UNIVERSITY OF CALIFORNIA
SANTA CRUZ

BAYESIAN MODELING AND SCALABLE INFERENCE FOR COUNT TIME SERIES IN INFECTIOUS DISEASE SURVEILLANCE

A dissertation submitted in partial satisfaction
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

STATISTICAL SCIENCE

by

Meini Tang

August 2026

The Dissertation of Meini Tang
is approved:

Raquel Prado, Chair

Athanasios Kottas

Zehang Li

Donald Smith
Interim Vice Provost and Dean of Graduate Studies

Contents

Figures and Tables viii

Abstract xxiii

Dedication xxv

Acknowledgments xxvi

1. Introduction 1
- 1.1. Motivation . . . 1
- 1.2. Statistical models for count time series in disease surveillance . . . 5
  - 1.2.1. State-space models for count time series . . . 5
  - 1.2.2. Self-exciting models and the renewal equation . . . 6
  - 1.2.3. From single-region to spatially connected populations . . . 8
  - 1.2.4. Posterior inference for non-Gaussian state-space models . . . 9
- 1.3. Contributions . . . 10
- 1.4. Outline of the thesis . . . 13

2. DGTF models for univariate count time series 14
- 2.1. Introduction . . . 14
- 2.2. Univariate state-space models for disease dynamics . . . 16
  - 2.2.1. Model components . . . 19
  - 2.2.2. Poisson AR/GARCH models . . . 20
  - 2.2.3. Discrete-time Hawkes-type models . . . 23
  - 2.2.4. Distributed lag models . . . 27
  - 2.2.5. Summary of DGTF model variants . . . 30
- 2.3. Posterior inference methods . . . 33
  - 2.3.1. Auxiliary linear-Gaussian representation . . . 33
  - 2.3.2. SMC for latent states . . . 34
  - 2.3.3. HVA . . . 44
  - 2.3.4. MCMC benchmark . . . 48
- 2.4. Simulation studies . . . 55
  - 2.4.1. Simulation design . . . 56

2.4.2. Model comparison setup . . . 57
2.4.3. Implementation and metrics . . . 57
2.4.4. Results: HVA versus MCMC . . . 60
2.4.5. Results: model comparison . . . 64
2.4.6. Discussion . . . 70
2.5. Application: COVID-19 in Santa Cruz County . . . 73
2.5.1. Benchmark models . . . 74
2.5.2. DGTF models . . . 75
2.5.3. Results . . . 77
2.5.4. Discussion . . . 82
2.6. Concluding remarks . . . 83

3. Network Hawkes models for spatio-temporal reproduction numbers 86
3.1. Introduction . . . 86
3.2. Methodology . . . 89
3.2.1. Model . . . 89
3.2.2. Reproduction number quantities . . . 93
3.2.3. Identifiability of the transmission matrix and reproduction numbers . . . 97
3.3. Bayesian inference . . . 98
3.3.1. Zero-inflation indicators and coefficients . . . 100
3.3.2. Column-wise HMC for spatial network parameters . . . 100
3.3.3. Block update for reproduction number trajectories . . . 101
3.3.4. Baseline intensity . . . 102
3.4. Simulation studies . . . 103
3.4.1. Simulation design . . . 103
3.4.2. Study 1: parameter estimation with heterogeneous retention . . . 105
3.4.3. Study 2: model structure comparison . . . 108
3.4.4. Summary . . . 111
3.4.5. Sensitivity analyses . . . 112
3.5. Application: COVID-19 across California counties . . . 114
3.5.1. Data and model specification . . . 114
3.5.2. Transmission network, decomposition and forecasting results . . . 116
3.6. Concluding remarks . . . 129

4. Particle Laplace variational approximation for network Hawkes models 134
4.1. Introduction . . . 134
4.2. Network Hawkes-type model and notation . . . 139
4.3. Particle Laplace variational approximation . . . 141
4.3.1. EIVA proposal for the latent reproduction trajectory . . . 146

4.3.2. Newton–Laplace mixture for the spatial parameters 150
4.3.3. Approximate moment updates for the horseshoe scales and auxiliaries 156
4.3.4. Forward filtering for the zero-inflation indicators 158
4.3.5. Stochastic gradient updates for global parameters 160
4.3.6. Block-coordinate algorithm and computational complexity . . . 165
4.4. Simulation studies 169
4.4.1. Simulation design 169
4.4.2. Convergence diagnostics 172
4.4.3. Parameter estimation 173
4.4.4. Predictive performance 177
4.4.5. Replicated comparison against Markov chain Monte Carlo (MCMC) 178
4.4.6. Computational efficiency 180
4.4.7. Sensitivity to algorithm settings 182
4.4.8. Trade-off summary and recommendation 185
4.5. Application to COVID-19 across California counties 187
4.5.1. Setup and convergence 187
4.5.2. Parameter estimation 189
4.5.3. Posterior predictive fit and forecasting 193
4.5.4. Summary 195
4.6. Concluding remarks 195

5. The `dgtf` R package **201**
5.1. Introduction 201
5.2. Model framework 203
5.2.1. General formulation 203
5.2.2. Canonical models 204
5.2.3. Seasonality and exogenous regressors 206
5.3. The dgtf package 207
5.3.1. Model construction 208
5.3.2. Priors 209
5.3.3. Model fitting 210
5.3.4. Forecasting 211
5.3.5. Inspecting a fitted model 211
5.3.6. Model comparison 212
5.4. Inference methods 214
5.4.1. HVA 214
5.4.2. MCMC 214
5.4.3. Convergence diagnostics 215

5.4.4. Choosing between methods . . . 216
5.5. Simulation study . . . 216
5.5.1. Model specification and data generation . . . 216
5.5.2. Fitting with HVA and MCMC . . . 218
5.5.3. Parameter recovery and comparison . . . 220
5.5.4. Posterior predictive checks and forecasting . . . 221
5.6. Application: COVID-19 in Santa Cruz County . . . 223
5.6.1. Data . . . 224
5.6.2. Model configurations . . . 224
5.6.3. Posterior inference . . . 226
5.6.4. In-sample comparison . . . 228
5.6.5. Out-of-sample forecasting . . . 229
5.7. Concluding remarks . . . 231

6. Discussion and future directions **233**

Appendix **240**

A. Supplement to Chapter 2 **240**
A.1. Simulation from discrete-time Hawkes-type model . . . 240
A.1.1. Model specification . . . 240
A.1.2. Gradient of the log joint probability for HVA and HMC . . . 241
A.1.3. Implementation . . . 247
A.1.4. Convergence diagnostics . . . 248
A.1.5. Particle sensitivity analysis for HVA . . . 251
A.1.6. Prior sensitivity analysis for HVA and MCMC . . . 253
A.1.7. Model selection . . . 257
A.2. Simulation from distributed lag model . . . 260
A.2.1. Model specification . . . 260
A.2.2. Hyperparameter selection . . . 261
A.2.3. Model comparison . . . 263
A.3. Application to COVID-19 case counts in Santa Cruz County . . . 268
A.3.1. Implementation . . . 268
A.3.2. Convergence of HVA algorithms . . . 270
A.3.3. Model selection . . . 270
A.3.4. Weekly seasonality on COVID count time series . . . 271

B. Supplement to Chapter 3 **275**
B.1. Posterior sampling details 275
B.1.1. Zero-inflation indicators and coefficients 275
B.1.2. Column-wise HMC for horseshoe parameters 276
B.1.3. HMC for baseline intensity 280
B.1.4. Block update for reproduction number disturbances 280
B.2. Simulation study 1: additional results 281
B.3. Simulation study 2: robustness results 283
B.4. Simulation study for zero-inflation component 284
B.5. Sensitivity analyses on simulated data 286
B.5.1. Sensitivity to the lag distribution 287
B.5.2. Sensitivity to the innovation variance 288
B.5.3. Prior sensitivity 289
B.5.4. Sensitivity to the gain function 290
B.6. Sensitivity analyses on the application data 292
B.6.1. Sensitivity to the innovation variance 292
B.6.2. Subperiod robustness 294
B.6.3. Residual seasonality 295
B.6.4. Zero inflation 297

C. Supplement to Chapter 5 **299**
C.1. Application: HVA vs MCMC comparison for the discrete-time Hawkes-type model 299

Bibliography **303**

Figures and Tables

List of Figures

2.1. Graphical model for the DGTF framework with three levels. The *system* level updates the latent state $\boldsymbol{\theta}_t$ from $(\boldsymbol{\theta}_{t-1}, \mathbf{y}_{t-1})$ via $\mathbf{g}_t(\cdot)$; the *link* level forms $\eta_t = \mathbf{x}_t'\boldsymbol{\varphi} + f_t(\mathbf{y}_{t-1}, \boldsymbol{\theta}_t)$; the *observation* level generates y_t from an exponential-family model $\mathcal{F}(\chi_t, \phi)$. Static parameters $\boldsymbol{\gamma}$ (including $\boldsymbol{\varphi}$, ϕ, and any parameters inside $f_t, \mathbf{g}_t$) affect both levels. 21

2.2. Simulated data from a discrete-time Hawkes-type model along with the corresponding R_t. 56

2.3. Posterior distributions of $\rho, \varphi, \mu, \sigma^2$ and W estimated by hybrid variational approximation (HVA) and Markov chain Monte Carlo (MCMC). True values are marked by black dashed lines. 61

2.4. Posterior predictive distribution of y_t and posterior distribution of R_t from fitting the true model with unknown $\boldsymbol{\gamma} = (\rho, \boldsymbol{\varphi}', \mu, \sigma^2, W)'$ to simulated data. Black lines show true values, while blue and red lines represent results from hybrid variational approximation (HVA) and Markov chain Monte Carlo (MCMC), respectively. Shaded regions indicate 95% credible intervals (CIs) for each method. 65

2.5. Multi-steps-ahead forecasting distributions using true model $\mathcal{M}_0$, whose parameters are estimated by hybrid variational approximation (HVA) and Markov chain Monte Carlo (MCMC). Solid lines represent the posterior medians, and the colored ribbons indicate the 95% credible intervals (CIs). 66

2.6. Estimated log-normal lag distribution from $\mathcal{M}_0$ and $\mathcal{M}_1$, along with the implied negative-binomial lag distribution from $\mathcal{M}_2$, where the lag parameters are set to their respective posterior medians. 69

2.7. Posterior distributions of R_t using three different models. Black lines represent true values. 69

2.8. Posterior predictive distributions estimated using hybrid variational approximation (HVA) by different models, compared against the observed count time series simulated from $\mathcal{M}_0$. The black dots indicate the true values, the solid lines represent posterior medians, and the ribbons represent 95% credible intervals (CIs). 72

2.9. Daily new cases of COVID-19 from July 1st, 2020 to Dec 1st, 2021 in Santa Cruz County. 73
2.10. Posterior distributions of the effective reproduction numbers $\{R_t\}$ and static parameters $(W, \rho, \boldsymbol{\varphi})$ for Santa Cruz County, estimated by different models. In (d), we illustrate the log-normal lag distribution in the discrete-time Hawkes-type model, using the posterior medians for μ and σ_2 and the implied negative-binomial leg distribution in the distributed lag model with κ set to its posterior median. 78
2.11. Posterior predictive distributions of daily new cases across two time periods for the dynamic generalized transfer function (DGTF) models: Oct 2020 to March 2021 (left) and Feb 2021 to Aug 2021 (right). The posterior medians along with the 95% credible intervals (CIs) are illustrated. Red dots represent the daily new cases. 80

3.1. Network reproduction number R_t^{net}: true values (black), posterior mean and 95% credible interval from the network Hawkes-type model, and reproduction numbers with 95% confidence intervals estimated by WT and EpiEstim applied to aggregated case counts across all five locations. 106
3.2. Reproduction number estimation for a simulated network Hawkes-type process with five locations and heterogeneous retention rates. Left panels compare the intrinsic source-specific reproduction number $R_{k,t}$ (posterior mean and 95% credible interval) with the true values and univariate benchmark estimates from WT and EpiEstim. Right panels decompose the observed effective reproduction number $R_{k,t}^{\text{eff}}$ into local and imported contributions. Locations with low retention (D, E) have higher importation than locations with high retention (A, B). 107
3.3. Estimation of the spatial weight matrix $\boldsymbol{\alpha} = (\alpha_{s,k})$ in Study 1. The model recovers both diagonal retention rates and off-diagonal transmission structure. 108
3.4. Posterior mean (a) and standard deviation (b) of the estimated spatial weight matrix $\hat{\alpha}_{s,k}$. Each element $\alpha_{s,k}$ gives the fraction of secondary cases generated by source county k that the fitted model allocates to destination county s. Diagonal entries correspond to retention rates ω_k, and larger off-diagonal values indicate a stronger estimated cross-county allocation. 116

3.5. Estimated transmission network (a) compared to pre-pandemic commuter mobility from the 2016 to 2020 American Community Survey (b). Panel (a) shows estimated transmission, while panel (b) shows the commuter baseline. Arrows indicate the direction of the estimated transmission allocation (a) or commuter travel (b), with thickness proportional to weight magnitude. Node shading reflects retention rate. Several estimated transmission pathways run counter to commuter direction, a pattern that is consistent with, but does not identify, transmission flowing from workplaces back to residences. 118
3.6. Reproduction number comparison and decomposition for five counties (alphabetical order, continued in Fig. 3.7). Left panels compare the intrinsic source-specific reproduction number $R_{k,t}$ against the observed effective reproduction number from the network Hawkes-type model and two benchmark methods (EpiEstim and Wallinga–Teunis). Right panels decompose the observed effective reproduction number into local transmission (orange) and importation (purple) components. Solid lines are posterior means with shaded 95% credible intervals. 121
3.7. Reproduction number comparison and decomposition for the remaining five counties. Left panels compare the intrinsic source-specific reproduction number $R_{k,t}$ against the observed effective reproduction number from the network Hawkes-type model and two benchmark methods (EpiEstim and Wallinga–Teunis). Right panels decompose the observed effective reproduction number into local transmission (orange) and importation (purple) components. Solid lines are posterior means with shaded 95% credible intervals. 122
3.8. Network reproduction number R_t^{net} over time, defined as the spectral radius of the transmission matrix. The network quantity is compared to reproduction numbers estimated by EpiEstim and Wallinga–Teunis applied to aggregated daily case counts across all ten counties. 123

4.1. particle Laplace variational approximation (PLVA) convergence diagnostics on the simulated data. Panel (a) shows the pseudo-likelihood objective $\widehat{\mathcal{L}}_{\text{pl}}$ of Eq. (4.3.46) for two runs, one at $M = 50$ and one at $M = 100$. Panels (b) and (c) come from the $M = 50$ run alone, and show the posterior mean retention rate for each column and the efficient importance variational approximation (EIVA) effective sample size. . . 173
4.2. Posterior median and 95% credible interval (CI) of reproduction number $R_{k,t}$ for each location. 174

4.3. Posterior densities for the retention rate w_k across the 5 destinations. particle Laplace variational approximation (PLVA) at $M = 50$ in blue and Markov chain Monte Carlo (MCMC) in red. The solid vertical line marks the truth. The variational posterior is narrower than the Markov chain Monte Carlo (MCMC) posterior across all five columns, and is pulled slightly further toward the prior mode on w_4. 176
4.4. Posterior summary of the transmission matrix $\boldsymbol{\alpha}$ at $M = 50$. The top row shows heatmaps of the true $\boldsymbol{\alpha}$, the particle Laplace variational approximation (PLVA) posterior mean, and the Markov chain Monte Carlo (MCMC) posterior mean. The bottom row shows the posterior standard deviations under particle Laplace variational approximation (PLVA) and Markov chain Monte Carlo (MCMC). 176
4.5. One to ten day ahead forecast of $y_{s,t}$ on the held-out window for each of the 5 destinations. particle Laplace variational approximation (PLVA) at $M = 50$ in purple and Markov chain Monte Carlo (MCMC) in cyan. Solid lines show the posterior median forecast, and shaded bands show 95% prediction intervals. The black line represents the observed trajectory. 177
4.6. PLVA convergence diagnostics on the California COVID-19 data over the 300 optimization iterations completed. Left: pseudo-likelihood objective trace $\widehat{\mathcal{L}}_{\text{pl}}$ of Eq. (4.3.46). Right: w_k trace by county, where the slowest visually converging county (San Mateo, highlighted in red) stabilized by approximately iteration 182. 188
4.7. Posterior summaries of the spatial transmission weight matrix $\boldsymbol{\alpha}$ on the California COVID-19 data. Top row: posterior mean under particle Laplace variational approximation (PLVA) (left) and Markov chain Monte Carlo (MCMC) (right). Bottom row: posterior standard deviation under the two methods. The dominant pathways agree between the methods. particle Laplace variational approximation (PLVA) produces a more concentrated posterior, compared to Markov chain Monte Carlo (MCMC). 190
4.8. Network reproduction number $R_t^{\text{net}} = \rho(\mathbf{B}_t)$ over the study period. Posterior medians and 95% credible bands under particle Laplace variational approximation (PLVA) and Markov chain Monte Carlo (MCMC) are overlaid for direct comparison. 193

4.9. Ten step ahead forecasts on the held-out window for each of the ten counties. Observed counts are shown as black lines, and particle Laplace variational approximation (PLVA) and Markov chain Monte Carlo (MCMC) posterior medians are shown with their 95% prediction bands. The forecast distributions of the two methods agreed closely, and both achieved calibrated 95% prediction interval coverage near the nominal level. 194

5.1. Workflow of the **dgtf** package. A model object and a prior specification feed into `dgtf()`, which produces a fit. The fit supports forecasting, posterior inspection including posterior predictive checks, and cross-model comparison through a `dgtf_compare()`that accepts either fit objects or forecast objects. 207
5.2. hybrid variational approximation (HVA) versus Markov chain Monte Carlo (MCMC) comparison. The left panel shows the posterior median R_t band from each method with the simulated truth overlaid. The right panel shows overlaid posterior histograms of the innovation variance W with the truth marked by a dashed vertical line. 221
5.3. Combined posterior predictive and forecast bands for the three model variants on the COVID-19 data. The in-sample posterior predictive band over the 250-day training window and the out-of-sample forecast band over the 14-day test window are drawn in the same color for each model, and a vertical dotted line marks the train/test boundary. The black dashed line represents observed counts. 230

A.1. Convergence diagnostics for the dispersion parameter ρ showing iterations from 2,001 to 5,000 using MCMC. 249
A.2. Convergence diagnostics for the baseline intensity φ showing iterations from 2,001 to 5,000 using MCMC. 250
A.3. Convergence diagnostics for the log-normal mean parameter μ showing iterations from 2,001 to 5,000 using MCMC. 250
A.4. Convergence diagnostics for log-normal variance parameter σ^2 showing iterations from 2,001 to 5,000 using MCMC. 251
A.5. Convergence diagnostics for the system variance W showing iterations from 2,001 to 5,000 using MCMC. 251
A.6. Evolution of the conditional marginal likelihood $p(y_{1:T} \mid \boldsymbol{\gamma})$ during the first 1,000 iterations of HVA, using simulated data from the discrete-time Hawkes-type model $\mathcal{M}_0$. The plot shows stabilization of the objective function by iteration 1,000. 252

A.7. MCMC posteriors of the static parameters $\boldsymbol{\gamma} = (\rho, \varphi, \mu, \sigma^2, W)$ in the discrete-time Hawkes-type model, with data simulated from the same model. The default prior settings are: $\rho, W, \sigma^2 \sim \text{IG}(1, 1)$, $\mu \sim N(0, 1)$, $\varphi \sim N(0, 10)$ truncated to $(0, \infty)$, and $\kappa \sim \text{Beta}(1, 1)$. Each panel shows the effect of altering the prior distribution for a single parameter while keeping the priors for all other parameters fixed at their default values. For W (panel e), the comparison includes priors that place positive density at zero: half-normal ($\text{N}^+(s = 1)$), half-t ($t^+(s = 0.1, \nu = 3)$), and half-Cauchy ($\text{C}^+(s = 0.1)$ and $\text{C}^+(s = 0.05)$). 256

A.8. HVA posterior distributions of the static parameters $\boldsymbol{\gamma} = (\rho, \varphi, \mu, \sigma^2, W)$ in the discrete-time Hawkes-type model, with data simulated from the same model. The default prior settings are: $\rho, W, \sigma^2 \sim \text{IG}(1, 1)$, $\mu \sim N(0, 1)$, $\varphi \sim N(0, 10)$ truncated to $(0, \infty)$, and $\kappa \sim \text{Beta}(1, 1)$. Each panel shows the effect of altering the prior distribution for a single parameter while keeping the priors for all other parameters fixed at their default values. 257

A.9. Posterior distributions of the reproduction number R_t estimated by three DGTF models for data simulated from the distributed lag model $\mathcal{M}_2$. Solid lines show posterior medians, shaded regions indicate 95% CIs, and black lines mark true R_t values. The discrete-time Hawkes-type models ($\mathcal{M}_0$, $\mathcal{M}_1$) and true distributed lag model ($\mathcal{M}_2$) produce similar estimates despite different lag parameterizations. 265

A.10. Posterior predictive distributions $p(\mathbf{y}_t^{\text{rep}} \mid \mathbf{y}_t)$ for observed counts from data generated by the distributed lag model $\mathcal{M}_2$ (true model), compared against estimates from four competing models. Black dots show observed values, solid lines represent posterior medians, and shaded ribbons show 95% prediction intervals. 266

A.11. Posterior distributions of lag distribution parameters μ, σ^2, and κ estimated by competing models fitted to data generated from the distributed lag model $\mathcal{M}_2$. Black vertical line marks the true value $\kappa = 0.395$. Point estimates use posterior medians with 95% CIs shown. 267

A.12. Comparison of lag distributions and parameter estimates across models fitted to data generated from distributed lag model $\mathcal{M}_2$. (a) Implied lag distributions using posterior median parameter estimates: log-normal distributions from discrete-time Hawkes-type models $\mathcal{M}_0$ and $\mathcal{M}_1$, and negative-binomial distribution from the true distributed lag model $\mathcal{M}_2$; (b-c) Posterior distributions of the dispersion ρ and the baseline intensity φ. 267

A.13. Evolution of the conditional marginal likelihood $p(y_{1:T} \mid \boldsymbol{\gamma})$ in the first 5,000 iterations of HVA. We fitted the COVID-19 case counts in Santa Cruz County to three different models: (a) a third-order negative-binomial autoregression; (b) a discrete-time Hawkes-type model; (c) a second-order distributed lag model. 271
A.14. (a) The posterior distributions of R_t, from July 1, 2020 to Nov 29, 2021, for Santa Cruz County are estimated by WT, EpiEstim, and our models both with and without weekly seasonality; (b) The posterior predictive distributions $p(y_t^{\text{rep}} \mid y_t)$ for models with and without seasonality from Feb 4, 2021 to Nov, 2021. 273

B.1. Simulation setup for Study 1. Panel (a) shows the simulated case counts for five locations over $T = 200$ time points with retention rates $w_k \in \{0.9, 0.7, 0.5, 0.3, 0.3\}$. Panel (b) shows the spatial weight matrix, where diagonal entries represent retention rates and off-diagonal entries represent cross-location transmission probabilities. 282

C.1. Posterior comparison of the discrete-time Hawkes-type model fitted by hybrid variational approximation (HVA) and Markov chain Monte Carlo (MCMC). The two methods produce similar R_t trajectories across the training window, and the seasonality posteriors overlap on every day of the week. 302

List of Tables

2.1. Comparisons of DGTF model variants in terms of model structure and epidemiological interpretation. 31
2.2. Performance comparison of HVA and MCMC across iterations. HVA converges in 1,000 iterations while MCMC requires 2,000, resulting in 65% reduction in total runtime. Computational time reported as mean±SD in seconds across five independent runs. These results are for model $\mathcal{M}_0$ in the simulation study. 64
2.3. Performance metrics for five candidate models fitted to data from $\mathcal{M}_0$. The table shows χ^2 discrepancy, CRPS, and 95% posterior predictive interval coverage for observed counts y_t, along with MAE and 95% credible interval coverage for reproduction numbers R_t. Models $\mathcal{M}_3$ and $\mathcal{M}_4$ do not estimate R_t directly. 68

2.4. Comprehensive performance comparison between dynamic generalized transfer function (DGTF) models for COVID-19 data in Santa Cruz County. Metrics include in-sample fit and posterior predictive interval calibration for observed counts y_t, calculated over the full time series (July 2020 - December 2021). 79
2.5. Comparison of HVA and MCMC inference for the discrete-time Hawkes-type model on COVID-19 data. Parameter estimates show posterior medians with 95% credible intervals. Performance metrics include χ^2 discrepancy, continuous ranked probability score (CRPS), and 95% posterior predictive interval coverage for in-sample fit, and χ^2 discrepancy, CRPS, and 95% posterior predictive interval width for one-step-ahead forecasting (both achieved 100% coverage for one-step forecasts). 81

3.1. Network with mobility effect in the data generating process ($S = 5$, $T = 200$). In these simulations, network models improve reproduction number estimation, while mobility mainly improves spatial weight recovery. 110
3.2. Network with mobility effect ($S = 8$, $T = 200$). Mobility covariates become more useful as network size grows, especially for spatial weight recovery. 111
3.3. Reproduction number decomposition across ten California counties, ordered by descending intrinsic source-specific reproduction number. All quantities are temporal averages over the study period. Retrospective quantities ($\bar{R}_k^{\text{eff}}$ and local percentage) describe the origins of cases appearing in each county, whereas prospective quantities ($\bar{R}_k$ and retention) describe the destinations of transmission originating in each county. Net flow: difference between total outgoing and incoming transmission weights. 120
3.4. Out-of-sample forecasting performance (mean CRPS and 95% prediction interval coverage across counties and forecast origins). Lower CRPS indicates better calibrated probabilistic forecasts. 126
3.5. Projected reduction in cumulative cases under four hypothetical scenarios evaluated under the fitted model. Values are posterior mean percentage reductions with 95% credible intervals. Network-wide effects are summed across all ten counties. Santa Clara and San Joaquin are shown as representative counties. 128

4.1. Absolute levels on the illustrative fit to the single simulated dataset of § 4.4.1, which also produces the figures of this section. The $R_{k,t}$ columns report continuous ranked probability score (CRPS) on the posterior distribution of $R_{k,t}$, mean absolute deviation between the posterior median and the truth, and empirical coverage of the 95% credible intervals, each averaged across the 5 destinations, the training window, and the runs. The remaining columns report in-sample posterior predictive continuous ranked probability score (CRPS) on $y_{s,t}$ averaged across the training window, and forecast continuous ranked probability score (CRPS) at horizons of 1, 5, and 10 days averaged across the 5 destinations. Tables 4.2 to 4.4 report the comparison against Markov chain Monte Carlo (MCMC) across replicated datasets. 175

4.2. Replicated comparison on the spatial parameters across ten independently simulated datasets. The mean absolute error (MAE) on the retention rates w_k and the Frobenius norm of the error in the transmission weights $\boldsymbol{\alpha}$ are reported as the percentage change of the particle Laplace variational approximation (PLVA) value against the Markov chain Monte Carlo (MCMC) value on the same dataset, averaged across datasets, with the standard deviation across datasets in parentheses. Positive values indicate the larger particle Laplace variational approximation (PLVA) error. Coverage is the empirical coverage of the 95% credible intervals and width is their average width, both reported as levels. 181

4.3. Replicated comparison on the reproduction trajectories and the in-sample predictive fit across ten independently simulated datasets. The mean absolute error (MAE) and continuous ranked probability score (CRPS) changes are percentage changes of the particle Laplace variational approximation (PLVA) value against the Markov chain Monte Carlo (MCMC) value on the same dataset, averaged across datasets, with the standard deviation across datasets in parentheses. Positive values indicate the larger particle Laplace variational approximation (PLVA) error. The coverage columns report empirical coverage of the 95% credible intervals for $R_{k,t}$ and of the 95% posterior predictive intervals for $y_{s,t}$. 181

4.4. Replicated forecast comparison on the held-out window across ten independently simulated datasets. The continuous ranked probability score (CRPS) change is the percentage change of the particle Laplace variational approximation (PLVA) forecast continuous ranked probability score (CRPS) against the Markov chain Monte Carlo (MCMC) value on the same dataset, averaged across datasets, with the standard deviation across datasets in parentheses. Positive values indicate the larger particle Laplace variational approximation (PLVA) error. Coverage is the empirical coverage of the 95% posterior predictive intervals. 182
4.5. Wall clock time, iterations run, speedup factor over Markov chain Monte Carlo (MCMC), and the converged efficient importance variational approximation (EIVA) effective sample size per column for each particle Laplace variational approximation (PLVA) configuration. Wall clock times are averaged over five particle Laplace variational approximation (PLVA) runs at each M and three Markov chain Monte Carlo (MCMC) runs, each run on a separate dataset simulated from the same data generating process. 183
4.6. particle Laplace variational approximation (PLVA) accuracy at small efficient importance variational approximation (EIVA) particle counts. Each metric is normalized within a dataset and then averaged over five independently simulated datasets. The relative $R_{k,t}$ continuous ranked probability score (CRPS) is the geometric mean of the $R_{k,t}$ forecast continuous ranked probability score (CRPS) against each dataset's best M (lower is better, 1.00 optimal), and the $R_{k,t}$ coverage is the mean empirical 95% interval coverage (target 0.95). The relative continuous ranked probability score (CRPS) is worst at $M = 1$ and flat from $M = 2$, while the coverage rises through $M = 10$. 184
4.7. Sensitivity of posterior summaries and predictive performance to the initial state at $M = 50$. CS: cold start, using a generalized linear model fit for the spatial parameters and zeros for the disturbance trajectories. MS: Markov chain Monte Carlo (MCMC) short start, using the posterior means of a short Markov chain Monte Carlo (MCMC) chain. TS: true start at the data generating values. The simulated truth for w_k is shown in the first row for reference. Coverage is the empirical coverage of the 95% credible intervals for $R_{k,t}$. 185
4.8. Posterior mean and standard deviation of the diagonal retention rates w_k and selected dominant off-diagonal transmission weights $\alpha_{s,k}$. The two methods agree on which entries dominate, and the particle Laplace variational approximation (PLVA) estimates are more concentrated. . . . 191

4.9. Per-county reproduction number decomposition under particle Laplace variational approximation (PLVA) algorithm and Markov chain Monte Carlo (MCMC) on the California COVID-19 data, ordered by descending Markov chain Monte Carlo (MCMC) $\bar{R}_k$. The Markov chain Monte Carlo (MCMC) values are reproduced from Table 3.3. The two methods agree on Net Flow signs for nine of the ten counties, with Sacramento sitting near zero under both, and agree on the most transmissible and least transmissible counties as well as on the negative correlation between $\bar{R}_k$ and case rate. 192
4.10. In-sample posterior predictive check and ten step ahead forecasting performance on the California COVID-19 data. Values are means across the ten counties. The coverage columns report the empirical 95% prediction interval coverage, the fraction of counts within the central 95% posterior predictive interval. 194

5.1. Component families in the **dgtf** package and the types used in this chapter. String shortcuts are accepted in place of the explicit constructor calls when default parameter values suffice. The observation row lists the count models used in the examples here, and the package help documents the full set of components. 208
5.2. Shortcut constructors in the **dgtf** package. Each pre-fills the component arguments of `dgtf_model()`to produce the named canonical model. Koyck is the $r = 1$ special case of the negative binomial distributed lag family. 209
5.3. Static parameter recovery for the hybrid variational approximation (HVA) and Markov chain Monte Carlo (MCMC) fits, computed using function `dgtf_compare_params()`on the training data. Columns report the posterior mean, median, standard deviation, central 95% credible interval, mean absolute error, continuous ranked probability score (CRPS), and an indicator for whether the credible interval covers the truth. . . . 222
5.4. Model comparison on the COVID-19 data from `dgtf_compare()`. The in-sample block reports the number of inferred static parameters, the hybrid variational approximation (HVA) proxy for the log marginal likelihood, the continuous ranked probability score (CRPS), the chi-square discrepancy, the empirical 95% coverage and mean width of the posterior predictive interval, the Winkler interval score, and the total runtime. The forecast block reports coverage, the chi-square discrepancy, and the continuous ranked probability score (CRPS) averaged across the 14-day held-out horizon. 228

A.1. Model-specific settings and hyperparameter choices for the five DGTF models compared in the simulation study. The table shows the model structure, key hyperparameters, prior specifications for lag distribution parameters, and variational rank k used in HVA inference. 246
A.2. Common inference settings applied consistently across all DGTF models in simulation. Prior specifications are shown for parameters shared across models, along with algorithm-specific settings for HVA (two-filter smoother and gradient ascent optimization) and MCMC sampling. 246
A.3. Convergence diagnostics and accuracy metrics for different numbers of iterations in MCMC. The metrics include the Gelman-Rubin statistics for convergence assessment, the χ^2 discrepancy for the posterior predictive distribution of y_t, and the MAE for the posterior distribution of R_t. Results show that MCMC requires approximately 2,000 iterations for full convergence. 249
A.4. Convergence diagnostics and accuracy metrics for different numbers of iterations in HVA. The metrics include the conditional marginal log likelihood for convergence assessment, the χ^2 discrepancy and 95% CI coverage for the posterior predictive distribution of y_t, and the MAE and 95% CI coverage for the posterior distribution of R_t. Results demonstrate that HVA achieves convergence by 1,000 iterations with accuracy comparable to MCMC at 2,000 iterations. 252
A.5. Performance metrics for different particle counts in the HVA algorithm applied to simulated data from the discretized Hawkes model $\mathcal{M}_0$. Metrics are averaged across 200 time points and five independent runs. Runtime measured in seconds on an Apple M1 Pro processor (10 cores, 8 threads). The table shows diminishing returns in accuracy beyond 500 particles, while computational cost continues to increase linearly. 254
A.6. Selection of the discount factor δ of $\mathbf{W}_t$ in $\mathcal{M}_1$, the order of distributed lags r in $\mathcal{M}_2$, and the autoregressive orders p in $\mathcal{M}_3$ and $\mathcal{M}_4$, based on the χ^2 discrepancy of y_t and MAE of R_t. The data is simulated from a discretized Hawkes model with constant system variance. 259
A.7. Hyperparameter selection through grid search for models fitted to data simulated from the distributed lag model $\mathcal{M}_2$ ($r = 6$, $\kappa = 0.395$). Optimal values minimize in-sample estimation errors measured by χ^2 discrepancy and CRPS for y_t and the MAE of R_t if applicable. All models use HVA inference with 500 particles and 1,000 gradient ascent iterations. 262

A.8. Performance comparison of five DGTF models fitted to data generated from distributed lag model $\mathcal{M}_2$. Models are evaluated on predictive accuracy (χ^2 discrepancy and CRPS), uncertainty calibration (95% CI coverage and width for observed counts yt), and reproduction number estimation accuracy (MAE, 95% CI coverage, and width for R_t). Models $\mathcal{M}_3$ and $\mathcal{M}_4$ do not estimate R_t directly. 263
A.9. Model-specific settings for the three DGTF variants applied to COVID-19 data from Santa Cruz County. Hyperparameters were selected through grid search to minimize χ^2 discrepancy. The variational rank k reflects the number of static parameters including seasonal components. 268
A.10. Common inference settings maintained across all models in real data analyses. The table shows prior specifications for shared parameters and algorithmic settings. 269
A.11. Optimal lag order r for the distributed lags model and optimal autoregressive order p for negative-binomial autoregression, measured by χ^2 discrepancy, for the COVID count time series of Santa Cruz county from July 1, 2020 to Nov 29, 2021. 272

B.1. Posterior summaries for the retention rates w_k in Study 1. All true values lie within the 95% credible intervals. 283
B.2. Robustness to misspecification ($S = 5$, $T = 200$). The data generating process has horseshoe-distributed deviations and no mobility covariate. Including mobility causes little harm, while a Gaussian prior mainly worsens forecast calibration at low retention. 284
B.3. Simulation with zero-inflated data ($S = 5$, $T = 200$, 70% retention). Including the zero-inflation component improves both reproduction number estimation and forecast calibration. 285
B.4. Simulation without zero-inflated data ($S = 5$, $T = 200$, 70% retention). Including the zero-inflation component when it is absent from the data generating process causes minimal harm. 286
B.5. Sensitivity to the lag distribution, relative to the kernel with mean 4.7. Columns report the root mean square change in the off-diagonal transmission weights, the largest absolute change in the retention weights, and the mean absolute change in the network reproduction number. . . . 288
B.6. Sensitivity to the innovation variance W, with the data generating value $W = 0.01$ as the baseline. Columns report the mean absolute error, the empirical coverage of the 95% credible interval, and the mean interval width for $R_{k,t}$, and the mean absolute change in the local and imported reproduction numbers relative to the baseline. 289

B.7. Prior sensitivity, relative to the default prior configuration. Columns report the root mean square change in the off-diagonal transmission weights, the largest absolute change in the retention weights, the ratio of the posterior median of the global shrinkage scale to its default, the absolute change in the posterior mean of the baseline intensity μ, the mean absolute change in the network reproduction number, and the ratio of the one-step forecast CRPS to the default. 290
B.8. Sensitivity to the gain function. Recovery of each gain function against its own data generating truth when data are generated and fit under the same gain function, showing the innovation variance W used to generate each process, the Frobenius error of the transmission matrix against the truth, the Spearman correlation of the ordering by net flow against the truth, the mean absolute error and empirical 95% coverage of $R_{k,t}$, and the mean network reproduction number. 291
B.9. Sensitivity of forecasting performance to the random walk innovation variance W. Values are mean CRPS across seven rolling forecast origins and ten counties. The choice $W = 0.005$ achieves the best overall performance across forecast horizons. 293
B.10. Subperiod robustness of the fitted network. Each subperiod is compared to the fit on the full window through the root mean square change in the off-diagonal transmission weights and the Spearman correlation of the ordering of the ten counties by net flow. The mean network reproduction number is reported for each window. 295
B.11. Residual seasonality on the application data. For each county, the dominant period of the standardized posterior predictive residuals from the periodogram, and the residual autocorrelation at lags of one week, two weeks, and one year. 296
B.12. Zero-inflated against plain Poisson observation model on the ten-county application. The transmission matrix and the ordering of counties by net flow are compared to the zero-inflated fit through the root mean square change in the matrix weights and the Spearman correlation of the ordering; the mean network reproduction number and the forecast CRPS at horizons of one, five, and ten days are reported for each. 298

C.1. Comparison of hybrid variational approximation (HVA) and Markov chain Monte Carlo (MCMC) for the discrete-time Hawkes-type model on the COVID-19 training data, combining in-sample posterior predictive scores from `dgtf_compare()` with 14-day forecast scores from `dgtf_forecast_fit()`. Reported metrics for the in-sample window are the continuous ranked probability score (CRPS), the chi-square discrepancy, the empirical coverage and mean width of the 95% credible interval, the Winkler interval score, and the total runtime in seconds. For the forecast horizon, we report empirical coverage, the chi-square discrepancy, and the continuous ranked probability score (CRPS) averaged across the horizon. 302

Abstract

Bayesian Modeling and Scalable Inference for Count Time Series in Infectious Disease Surveillance

by

Meini Tang

Real-time monitoring of infectious disease outbreaks calls for statistical models that recover interpretable quantities such as the time-varying reproduction number from noisy count data, track posterior uncertainty, and run on time scales compatible with daily updates. Existing methods address these aims through separate model classes. Discrete-time Hawkes-type models, Poisson autoregressions, and distributed lag models each capture self-exciting transmission through alternative parameterizations of the same conditional mean structure, but they have been developed across separate software packages with model-specific inference routines, which makes structural model comparison cumbersome in practice. This dissertation develops a unified Bayesian framework for count time series in disease surveillance, organized around three threads. First, a class of dynamic generalized transfer function models places the three modeling families inside a common modular state-space class built from six independent components. A hybrid variational algorithm combines sequential Monte Carlo on the latent trajectory with stochastic gradient ascent on the static parameters. Second, a multivariate extension to spatially connected regions, a Bayesian network Hawkes model, jointly estimates time-varying intrinsic source-specific reproduction numbers and a sparse transmission network learned from data through a regularized horseshoe prior.

The observed reproduction number at each location is decomposed into a local component and an imported component. Posterior inference proceeds through a block Markov chain Monte Carlo sampler, with a particle Laplace variational counterpart developed for routine updates at larger spatial scales. Third, an R package implements the unified univariate framework through a compositional specification interface aligned with the six modular components, with the two inference engines available behind a single entry point. The methods are illustrated through simulation studies and applications to daily COVID-19 case counts from Santa Cruz County and from ten California counties.

For whatever

Acknowledgements

First and foremost, I would like to thank my advisor, Raquel Prado, for her sharp statistical insight, her generosity with her time, and her steadfast guidance, which has never wavered even through difficult times. I cannot thank her enough.

Many other faculty at UCSC have left clear marks on this work. I am grateful to Athanasios Kottas and Zehang Li for serving on my defense committee, and for being teachers I continue to learn from. Thanasis taught with a clarity and care that I continue to aspire to. Zehang's instruction shaped how I think about data, and I am especially grateful that he welcomed me into his group meetings during a period when I most needed that community. Juhee Lee's teaching set a standard for rigor that I still hold myself to. Bruno Sansó introduced me to ways of thinking about spatial problems that I find myself returning to whenever I sit down with a new dataset. Robert Lund taught the stochastic processes material that became the methodological core of my research.

I have also been lucky in my friends and companions. Jizhou Kang gave me invaluable advice during my internship. Qi Wang and Qianyu Dong have made the past years much more enjoyable through their company in both research and life. My family and friends elsewhere have offered steady support that has reached me reliably across time zones. None of this would have been possible without them.

The text of Chapter 2 is adapted from the following previously published article:

Tang, M. and Prado, R. (2026), "Fast approximate posterior inference for modeling disease dynamics via state-space models," *Computational Statistics & Data Analysis*, 221, 108368.

Chapter 1

Introduction

§ 1.1. Motivation

Infectious disease surveillance provides the empirical foundation for public health decisions during an outbreak. Health agencies track case counts, hospitalizations, and deaths in close to real time, and these data streams are translated into quantitative summaries that support intervention, resource allocation, and communication with the public. The COVID-19 pandemic increased the demands on this translation process. Daily reporting became routine, models were expected to update overnight, and the time-varying reproduction number R_t became a widely used summary statistic for tracking the epidemic (Cori et al., 2013; Wallinga and Teunis, 2004). The pandemic also exposed several limitations of the existing statistical methods at this scale. Models that worked well for retrospective analysis often took hours to re-estimate, and models from different research traditions were difficult to compare on the same data because each came with its own software package and inference routine.

The reproduction number R_t is the expected number of secondary infections generated

by a typical infected individual at time t. It has a transparent epidemiological interpretation, where values above one indicate that transmission is increasing and values below one indicate that it is declining. This interpretability is part of the reason R_t became central during COVID-19, and it remains a useful target for any modeling framework that aims to be operationally relevant. Producing a credible R_t trajectory from observed case counts requires three ingredients. The first is a probabilistic model that links the observed counts to a latent transmission process. The second is an inference procedure that recovers the latent process from the data while quantifying uncertainty. The third is a computational implementation that runs in time scales compatible with daily updates.

Several modeling traditions have grown around this task, each capturing a different aspect of disease transmission. Compartmental models based on ordinary differential equations, such as the susceptible-infected-recovered family, encode mechanistic assumptions about how individuals move between disease states (Bertozzi et al., 2020). Compartmental models offer interpretability and are well suited to scenario analysis, but they assume deterministic dynamics and continuous-time evolution that fit awkwardly with the discrete and stochastic nature of daily case counts. Self-exciting point processes, in particular the Hawkes process, instead model new cases as triggered probabilistically by past cases through a kernel that admits a serial interval interpretation (Hawkes, 1971; Hawkes and Oakes, 1974; Koyama et al., 2021; Browning et al., 2021; Chiang et al., 2022). Poisson autoregressions relate the conditional intensity to recent counts through autoregressive coefficients (Agosto and Giudici, 2020; Fokianos et al., 2022). Distributed lag models represent cumulative effects of past observations through a lagged transfer function (Koyck, 1954; Alves et al., 2010; Welty et al., 2009). The renewal equation approach (Cori et al., 2013; Wallinga and Teunis, 2004) provides a

sliding-window estimator of R_t that has become a de facto standard for early warning monitoring.

These approaches share more structure than is usually acknowledged. A discrete-time Hawkes-type model and a Poisson autoregression both relate the current intensity to past counts, differing mainly in how the lag weights are parameterized. A distributed lag model is a state-space representation of the same dependency in recursive form. The renewal equation has the same discrete-time conditional mean form when the lag weights are interpreted as a serial interval distribution and the coefficient is interpreted as R_t. The connections are not obvious from the way each family is typically introduced, and the corresponding software packages have been developed with different interfaces and model-specific inference routines. An applied analyst who wants to compare a Hawkes-type model and a Poisson autoregression on the same surveillance series has to learn two packages, reformat the data twice, and reconcile two sets of output before any meaningful comparison takes place. The cost is high enough that the comparison is rarely done.

Spatial structure raises a second set of challenges. Standard reproduction number estimators assume that each region is closed, attributing every new case at a given location to prior cases at the same location. The assumption is plausible when mobility between regions is small relative to local transmission, and untenable when it is not (Tsang et al., 2021; Russell et al., 2021). A region whose case counts are sustained mainly by infections imported from a neighbor may appear to have controlled local transmission when in fact its intrinsic R_t is well below one and the apparent activity is driven entirely by importation. Conversely, a region that exports most of its secondary infections may appear controlled when it is in fact driving regional spread. Distinguish-

ing the two cases requires a model that explicitly represents cross-region transmission pathways and learns their strengths from the data rather than imposing them from a fixed adjacency matrix (Chiang et al., 2022; Laub et al., 2025; Santitissadeekorn et al., 2025; Lasalle and Pascal, 2025).

Computation forms the third bottleneck. Posterior inference for the model classes considered above is hard for two reasons. First, the observation model is typically a count distribution and the latent transmission process is typically continuous, so the joint posterior is non-Gaussian and Markov chain Monte Carlo (MCMC) samplers must mix across the latent trajectory while updating static parameters. Second, in spatial settings the transmission weights and the intrinsic source-specific reproduction numbers enter the intensity multiplicatively, which introduces ridges in the posterior that mean-field and joint Gaussian variational families collapse onto a single point. Standard variational approaches lose calibrated uncertainty in exactly the high-dimensional regime where they would otherwise be most useful, and MCMC samplers become expensive to update on a daily basis as the number of regions grows. A method that captures the dependence structure of the posterior while running in time compatible with daily updates is needed.

The central contribution of this thesis is to use a Bayesian state-space formulation to put several self-exciting count models on common ground. This common formulation is useful because the same surveillance problem usually involves three decisions at once: how past incidence should enter the current risk, how transmission should be shared across locations, and how the resulting posterior can be approximated quickly enough for repeated use. Treating these decisions in one setting makes it easier to compare model classes, separate local from imported transmission, and use fast approximate inference.

§ 1.2. Statistical models for count time series in disease surveillance

This section describes the model classes that recur throughout the thesis at a level sufficient for the contributions in § 1.3 to be stated precisely. The full mathematical development for each class is given in the chapter where it is used.

1.2.1. State-space models for count time series

A natural framework for the modeling tasks just described is the discrete-time state-space model. The observed counts y_t are treated as noisy realizations of an unobserved latent process x_t that evolves over time, with the joint distribution specified through an observation density $p(y_t \mid x_t)$ and a system density $p(x_t \mid x_{t-1})$. Static parameters $\boldsymbol{\theta}$ enter both densities and have their own prior. The structure separates two distinct sources of variation in surveillance data, namely the noise that arises in reporting and the dynamics of the underlying transmission process, and it makes the latent state available for posterior inference alongside the static parameters (West and Harrison, 1997; Durbin and Koopman, 2012; Prado et al., 2021).

The classical dynamic linear model (DLM) of West and Harrison (1997) is the special case in which the observation and system equations are linear and Gaussian. Conditional on the static parameters, the Kalman filter recursions deliver the filtering, smoothing, and forecast distributions in closed form. For surveillance data, however, the observations are nonnegative integers, often with substantial overdispersion and excess zeros, so a Gaussian observation density is not appropriate. The thesis works with models that retain the Markovian structure of the system equation while replacing

the observation density with a Poisson, negative binomial, or zero-inflated Poisson (ZIP) distribution whose mean is linked to the latent state through a transfer function. Closed-form recursions are no longer available, so posterior inference proceeds either through MCMC, through sequential Monte Carlo (SMC) on the latent trajectory, or through structured variational approximations. The trade-offs among these approaches are discussed in § 1.2.4 and developed in detail in Chapters 2 and 4.

1.2.2. Self-exciting models and the renewal equation

The most important non-Gaussian state-space class for the thesis is the family of self-exciting count models that includes discrete-time Hawkes-type models (Hawkes, 1971; Hawkes and Oakes, 1974; Laub et al., 2025), Poisson autoregressions (Agosto and Giudici, 2020; Fokianos et al., 2022), and distributed lag models (Koyck, 1954; Alves et al., 2010; Welty et al., 2009). The shared feature of this family is that the conditional mean of y_t is a weighted sum of past observations,

$$\mathrm{E}[y_t \mid \mathcal{D}_{t-1}] = \mu + R_t \sum_{k=1}^{t-1} \phi_k \, y_{t-k}, \tag{1.2.1}$$

where $\mathcal{D}_{t-1}$ is the history through time $t-1$, μ is a background importation rate, R_t is a transmission magnitude, and $\{\phi_k\}$ is a lag distribution. The form is a directly specified discrete-time self-exciting conditional mean, motivated by the ideas of Hawkes and renewal processes rather than derived from a continuous-time Hawkes likelihood. The three model families above correspond to specific parameterizations of R_t and $\{\phi_k\}$. Discrete-time Hawkes-type models leave $\{\phi_k\}$ as a probability mass function and interpret R_t as the reproduction number. Poisson autoregressions absorb R_t into the lag weights and leave a finite collection of autoregressive coefficients. Distributed lag

models replace the sum with a recursive update that requires only a low-dimensional state. Each parameterization has different statistical and computational properties, but the shared algebraic structure motivates the unified framework developed in Chapter 2.

Because several distinct objects share the Hawkes name, it is worth being precise at the outset about how the discrete-time models studied here relate to the continuous-time Hawkes process. We distinguish four cases: (a) continuous-time Hawkes processes observed as event-time data; (b) *binned* Hawkes processes, in which an underlying continuous-time Hawkes process is observed only through counts aggregated over time bins, and whose likelihood is derived from that latent continuous-time process (Shlomovich et al., 2020; Zhou and Papadogeorgou, 2025); (c) integer-valued autoregressive (INAR) processes, which admit a formal limiting connection to the Hawkes process (Kirchner, 2016, 2017); and (d) discrete-time self-exciting or renewal-type count models specified directly through a conditional mean equation such as Eq. (1.2.1). The models developed in this thesis belong to category (d). They are defined directly in discrete time from the ideas of Hawkes and renewal processes, and they neither claim a formally justified limit of a continuous-time Hawkes process nor approximate the binned likelihood of one. We refer to these models as *discrete-time Hawkes-type models* (equivalently, self-exciting discrete-time count models), following the discrete-time constructions of Browning et al. (2021); Koyama et al. (2021). The distinction from the binned Hawkes setting is methodologically important. In that setting the likelihood follows from a continuous-time point process observed through aggregation, whereas here the model is specified directly by its discrete-time conditional mean.

The right-hand side of Eq. (1.2.1) also coincides, after setting $\mu = 0$, with the

right-hand side of the renewal equation

$$\mathrm{E}[I_t \mid \mathcal{D}_{t-1}] = R_t \sum_{k=1}^{t-1} \phi_k \, I_{t-k}, \tag{1.2.2}$$

in which I_t is the number of new infections at time t, R_t is the time-varying reproduction number, and $\{\phi_k\}$ is the serial interval distribution (Fraser, 2007; Cori et al., 2013). Cori et al. (2013) fit Eq. (1.2.2) under a Poisson likelihood and a sliding-window assumption on R_t, leading to the closed-form posterior that drives the **EpiEstim** R package and serves as the de facto baseline for R_t estimation. The discrete-time Hawkes-type and dynamic generalized transfer function (DGTF) formulations adopted in this thesis instead treat R_t as a stochastic latent trajectory, which permits information to be shared across time in a way that respects the smoothness of the underlying transmission process. The connection between Eqs. (1.2.1) and (1.2.2) makes **EpiEstim** a natural baseline against which to evaluate the proposed methods.

1.2.3. From single-region to spatially connected populations

The closed-population assumption underlying Eq. (1.2.2) fails when surveillance covers several geographically connected regions. The relevant generalization is a multivariate version of Eq. (1.2.1) in which the count at location s and time t has conditional mean

$$\mathrm{E}[y_{s,t} \mid \mathcal{D}_{t-1}] = \mu_s + \sum_{k=1}^{S} \alpha_{s,k} \, R_{k,t} \sum_{l=1}^{t-1} \phi_{t-l} \, y_{k,l}, \tag{1.2.3}$$

where S is the number of locations, $R_{k,t}$ is the intrinsic source-specific reproduction number at location k, and $\alpha_{s,k}$ governs how an infection generated by a case at location k is allocated to destination location s. When the transmission matrix $\boldsymbol{\alpha} = [\alpha_{s,k}]$ is column-stochastic, the closed-population R_t at each destination decomposes additively into a

local component, given by $\alpha_{s,s}R_{s,t}$, and an imported component, given by $\sum_{k\neq s}\alpha_{s,k}R_{k,t}$.

The single-region epidemic threshold $R_t = 1$ no longer governs growth or decay of the multi-region system. The relevant quantity instead is the spectral radius of the offspring matrix (Diekmann et al., 1990; Allen and van den Driessche, 2013; Green et al., 2022). Writing $\mathbf{R}_t = \mathrm{diag}(R_{1,t}, \ldots, R_{S,t})$ for the diagonal matrix of regional reproduction numbers, the offspring matrix at time t is $\boldsymbol{\alpha}\mathbf{R}_t$, and the system is subcritical when its spectral radius is below one. The condition can hold even when some regional $R_{s,t}$ exceed one, provided those regions export most of their secondary infections rather than retaining them. Chapter 3 develops a Bayesian network Hawkes model that infers $\boldsymbol{\alpha}$ from incidence data under a sparsity-promoting regularized horseshoe prior (Carvalho et al., 2010a; Piironen and Vehtari, 2017; Makalic and Schmidt, 2016), and Chapter 4 provides a structured variational counterpart suitable for routine updates during an outbreak.

1.2.4. Posterior inference for non-Gaussian state-space models

Three families of inference algorithms appear repeatedly in the chapters that follow. MCMC provides asymptotically exact posterior summaries by simulating a Markov chain whose stationary distribution is the target posterior. The cost of MCMC grows with the dimension of the joint parameter space, and for state-space models with a long latent trajectory the cost of each iteration is large. SMC approximates the filtering distribution of the latent state by a weighted particle ensemble that is propagated forward in time (Doucet and Johansen, 2011; Kitagawa, 1996), and provides an unbiased estimator of the marginal likelihood as a by-product. Variational inference (VI) approximates the joint posterior by a tractable density chosen to minimize the Kullback–Leibler (KL)

divergence to the target (Hoffman et al., 2013; Wand, 2017), trading exactness for substantial gains in speed. Hybrid methods combine SMC on the latent trajectory with VI on the static parameters, balancing the flexibility of SMC with the efficiency of VI (Loaiza-Maya et al., 2022; Loaiza-Maya and Nibbering, 2025). Chapter 2 adopts the hybrid path for the univariate framework, and Chapter 4 develops a related variational scheme for the network setting where the multiplicative coupling between transmission weights and reproduction numbers requires a more elaborate variational family than the mean-field or joint Gaussian default.

§ 1.3. Contributions

The contributions build from this common formulation. The first contribution is a modular univariate state-space class for self-exciting count models, so that the observation model, lag distribution, link function, and latent dynamics can be changed without starting the inference problem from scratch. The second contribution is a spatial extension in which the observed reproduction number is decomposed into local and imported components through a learned transmission matrix. The third contribution is computational, comprising hybrid variational inference tools and a related R software package developed so that the models can be fitted and compared in repeated surveillance analyses.

Chapter 2 introduces the univariate part of the framework. The DGTF class is a state-space class that contains discrete-time Hawkes-type models, Poisson autoregressions, and distributed lag models as special cases of Eq. (1.2.1). The model is built from six independent components, namely an observation distribution, a link function, a

gain function, a lag distribution, a system equation, and an error distribution. Each component can be modified without re-deriving the inference procedure. A hybrid variational approximation (HVA) algorithm that combines SMC with variational Bayes provides fast approximate posterior inference, achieving roughly a three-fold speedup over MCMC in the studies considered, with similar point estimates but some loss of interval calibration typical of Gaussian variational approximations. The framework is illustrated through simulation studies and an application to COVID-19 case counts from Santa Cruz County, California. The chapter is based on Tang and Prado (2026).

Chapter 3 extends the framework to spatially connected regions through a Bayesian network Hawkes-type model of the form Eq. (1.2.3). Observed case counts at multiple locations are modeled as ZIP realizations of an intensity that combines a background rate with weighted contributions from prior cases at all locations. The transmission weights form a column-stochastic matrix whose off-diagonal entries are shrunk toward a mobility-based baseline by a regularized horseshoe prior, so the network structure is learned from the data with posterior uncertainty quantification. The chapter's main modeling contribution is the decomposition of the closed-population reproduction number into local and imported components, together with Bayesian regularization of the transmission matrix. The system-wide epidemic threshold is connected to the spectral radius of the inferred branching matrix. Posterior computation proceeds through a block MCMC scheme with non-centered parameterizations and structured Gaussian proposals for latent states.

Chapter 4 addresses the computational framework needed for the spatial model. The transmission weights and the source-specific reproduction trajectories enter the intensity multiplicatively, which produces flat directions in the likelihood and ridges in the poste-

rior whenever destination trajectories are highly correlated across source locations. Joint Gaussian and mean-field variational families collapse these ridges onto a single point. The proposed particle Laplace variational approximation (PLVA) method combines an efficient importance variational approximation (EIVA) for the latent reproduction trajectory with a mixture of Laplace approximations for the spatial parameters, whose components are placed at the conditional mode of each particle and whose weights are inherited from the importance sampler. The PLVA is a block-coordinate variational approximation built column by column. A finite-particle realization of the law of total variance shows that the mixture captures both the average local curvature of the spatial posterior and the spread of conditional modes along the ridge, while a joint Gaussian family captures only the first component. Simulation studies and an application to daily COVID-19 case counts across ten California counties show speedups of 2 to 15 times over MCMC with closely aligned predictive performance in the studied settings, while a replicated comparison across ten simulated datasets identifies residual under-coverage of the credible intervals for the spatial parameters.

Chapter 5 describes the software implementation of the univariate framework. The **dgtf** package for R implements the model class of Chapter 2 through a compositional specification interface aligned with the six components introduced there. The package supports posterior inference under either the HVA or a MCMC sampler through a common interface, so a user can trade computational speed against calibration without rewriting the model. Built-in functions provide posterior predictive checks, multi-step forecasting, parameter recovery diagnostics, and side-by-side model comparison. The software chapter supports dissemination and reproducibility. The interface is demonstrated through a simulation study comparing five model formulations against a

known ground truth and an application to COVID-19 surveillance data from Santa Cruz County, with the **EpiEstim** renewal equation estimator (Cori et al., 2013) included as a baseline.

§ 1.4. Outline of the thesis

The chapters build on one another. Chapter 2 develops the univariate DGTF framework and the hybrid variational algorithm used for fast inference. Chapter 3 extends the framework to a multivariate spatio-temporal setting and introduces the local/imported decomposition of transmission. Chapter 4 develops the structured variational approximation used to scale inference for the network model. Chapter 5 presents the **dgtf** software implementation, which makes the univariate framework reproducible and usable for applied surveillance analysis. Chapter 6 synthesizes the methodological contributions, discusses limitations, and outlines directions for further work. Appendices provide supplementary derivations, additional simulation results, and extended posterior summaries for Chapters 2 and 3.

Chapter 2

Dynamic Generalized Transfer Function models for univariate count time series

§ 2.1. Introduction

The self-exciting form $\mathrm{E}[y_t \mid \mathcal{D}_{t-1}] = \mu + R_t \sum_{k=1}^{t-1} \phi_k \, y_{t-k}$ introduced in § 1.2.2 contains discrete-time Hawkes-type models, Poisson autoregressions, and distributed lag models as alternative parameterizations of the same conditional mean structure. Each parameterization has been developed within its own research community and ships with its own software package and inference routine. An applied analyst who wants to compare a Hawkes-type model and a Poisson autoregression on the same surveillance series often has to move between different software tools, reformat the data, and reconcile distinct model outputs before any meaningful comparison can take place. The practical cost is high enough that such comparisons are rarely done. This chapter develops a unified Bayesian framework that places all three families inside a single non-linear non-Gaussian state-space model, the dynamic generalized transfer function (DGTF) class, with a single inference routine that applies across the family.

The DGTF model is constructed from six independent components: an observation distribution, a link function, a gain function, a lag distribution, a system equation, and an error distribution. Changing any one component while holding the others fixed produces a new model within the same class. Treating $\{\phi_k\}$ as a finite collection of free coefficients with R_t absorbed into them yields a Poisson autoregression. Treating $\{\phi_k\}$ as a probability mass function and R_t as a stochastic latent trajectory yields a discrete-time Hawkes-type model. Replacing the explicit weighted sum with a recursive update on a low-dimensional state yields a distributed lag model. The same framework accommodates negative binomial observations for overdispersion, alternative link and gain functions, weekly seasonality, and exogenous covariates, without re-deriving the inference procedure. § 2.2 develops these connections in detail and shows how each existing family appears as a special case.

The proposed approach is novel in that it provides a common Bayesian state-space representation and develops posterior inference tools that are shared across model classes that are usually treated separately. This common representation allows users to compare observation models, lag structures, gain functions, seasonal terms, and latent transmission dynamics within one inferential setting.

Posterior inference is challenging because the observation density is non-Gaussian and the latent state evolves non-linearly under most lag and gain specifications. MCMC provides asymptotically exact posterior summaries but is slow enough that routine daily updates become impractical even at the single-region scale. This chapter develops a hybrid variational approximation (HVA) algorithm that combines a SMC pass on the latent trajectory with stochastic gradient ascent (SGA) on the variational evidence lower bound (ELBO) for the static parameters. The proposed method reduces the dimension of

the variational problem to that of the static parameters and recovers the latent trajectory through the marginal likelihood estimate produced by the particle filter as a by-product (Loaiza-Maya et al., 2022; Frazier et al., 2023; Wang et al., 2022; Xuan et al., 2024; Cabral et al., 2024). Across the model variants in the simulation study, HVA produced posterior medians and predictive scores close to those from MCMC while running roughly three times faster, but its credible intervals for the latent reproduction number were narrower and showed some undercoverage.

The remainder of the chapter is organized as follows. § 2.2 introduces the DGTF class, describes the six components in detail, and shows how discrete-time Hawkes-type models, Poisson autoregressions, and distributed lag models emerge as special cases. § 2.3 develops the auxiliary linear Gaussian representation that supports the SMC pass and presents the HVA algorithm alongside the MCMC benchmark. § 2.4 reports simulation studies that compare the two inference methods and that compare DGTF variants under known data generating processes. § 2.5 applies the framework to daily COVID-19 case counts from Santa Cruz County, California, with comparisons against the **EpiEstim** renewal equation estimator (Cori et al., 2013) and the discrete-time Hawkes-type baseline of Koyama et al. (2021). § 2.6 summarizes the chapter and connects its contributions to those developed in the rest of the thesis.

§ 2.2. Univariate state-space models for disease dynamics

Disease modeling often relies on time series of count data, such as daily cases, hospitalizations, or deaths. The central challenge lies in capturing how infections propagate

through populations over time. Many statistical approaches, including Poisson autoregressions (ARs), discrete-time Hawkes-type models and distributed lag models, share a core epidemiological principle: current case counts depend on recent infections. This "self-exciting" behavior, where past infections trigger future ones, typically shows an effect that decays over time as infected individuals recover or isolation measures take effect. These models capture similar phenomena, yet they emerged independently and traditionally require distinct inference algorithms (see, for example, Davis et al., 2021 for a comprehensive review on models and methods for count time series data), which has obscured their underlying connections.

This section introduces the DGTF models as a unifying state-space framework that encompasses seemingly disparate approaches. State-space models naturally separate the noisy observation process (e.g., reported case counts) from the unobserved latent process driving the dynamics (e.g., the true transmission rate), providing a modular and interpretable representation of disease dynamics. The simplest state-space models, such as the dynamic normal linear model (West and Harrison, 1997; Prado et al., 2021) consider two levels (observational and system levels) and are typically linear and Gaussian. More general state-space models include models that are non-Gaussian, such as generalized DLMs (DLMs; Gamerman, 1998), non-linear dynamic models (e.g., models in Gordon et al., 1993 and Kitagawa, 1996), as well as hierarchical models that include more than two levels such as dynamic hierarchical models (Gamerman and Migon, 1993; Gamerman, 1998).

The proposed DGTF framework applies to any count time series where the underlying rate evolves stochastically over time and that may depend on past observations, covariates, and/or unobserved state parameters. It is essentially a generalized dynamic

model with a link function that allows us to consider potentially non-linear relationships with past values of the observations and the state parameters via transfer functions and potential covariates, and a system equation that is also potentially non-linear. More specifically, a DGTF model has the following components:

1. Observation model: An exponential family distribution at the observational level (e.g., Poisson or negative binomial).

2. Link structure: A function that links observations to a predictor that decomposes into a linear relationship with exogenous regressors and a linear or non-linear relationship with endogenous effects and the state parameters captured by a differentiable transfer function. This is similar to the setting in generalized dynamic models. The main difference is that we allow potentially non-linear relationships at this level between the link function and endogenous effects and the state parameters.

3. State dynamics: Markovian latent states with potentially non-linear evolution equations. The non-linear relationships are made through differentiable evolution functions.

The framework is most suitable for discrete-time count data with time-varying parameters and self-exciting dynamics. For fundamentally continuous-time processes, continuous-time alternatives may be more appropriate. Note that dynamic generalized linear models form a well-studied subclass of these models. Key references include (West et al., 1985; West and Harrison, 1997) and references therein, while (Gamerman et al., 2016) provides further review with particular focus on Bayesian approaches for posterior inference.

The remainder of this section presents the general DGTF specification and then shows how existing models such as Poisson ARs, discrete-time Hawkes-type models and distributed lag models emerge as special cases through specific choices of transfer functions and lag structures.

2.2.1. Model components

Let $y_t, t = 0, 1, 2, \ldots,$ denote observed counts at time t. A DGTF model comprises the following elements, where $\boldsymbol{\theta}_t$ represents the latent state vector, and $\boldsymbol{\gamma}$ contains time-invariant parameters:

1. **Observation equation:** $y_t|\chi_t, \phi \sim \mathcal{F}(\chi_t, \phi)$, where $\mathcal{F}(\chi_t, \phi)$ belongs to the exponential family with natural parameter χ_t and scale parameter $\phi > 0$. We further assume that $y_1, \ldots, y_T$ are conditionally independent given $\{\chi_t,\ t = 1 : T\}$ and ϕ. In other words,

$$p(y_t|\chi_t, \phi) = \mathsf{b}(y_t, \phi) \exp\{\phi[y_t\chi_t - \mathsf{a}(\chi_t)]\}, \tag{2.2.1}$$

   with $\mathrm{E}(y_t|\chi_t, \phi) = \mu_t(\chi_t) = \mathsf{a}'(\chi_t)$ and $\mathrm{Var}(y_t|\chi_t, \phi) = \mathsf{a}''(\chi_t)/\phi$.

2. **Link function:** The mean $\mu_t(\chi_t) = \mathrm{E}(y_t|\chi_t, \phi)$ connects to the state parameters $\boldsymbol{\theta}_t$ via the link function $\zeta(\cdot)$:

$$\eta_t := \zeta(\mu_t(\chi_t)) = \mathbf{x}_t'\boldsymbol{\varphi} + f_t(\mathbf{y}_{t-1}, \boldsymbol{\theta}_t). \tag{2.2.2}$$

   Here $\mathbf{x}_t'\boldsymbol{\varphi}$ is a regression component that captures exogenous effects through $\mathbf{x}_t$ and coefficients $\boldsymbol{\varphi}$, (including intercepts, potential seasonal patterns, or external covariates). $f_t(\mathbf{y}_{t-1}, \boldsymbol{\theta}_t)$ is a transfer function component that captures dependen-

cies on past observations $\mathbf{y}_{t-1} = (y_{t-1}, \ldots, y_{t-L})'$ and/or the state parameters $\boldsymbol{\theta}_t$. In epidemiological settings, this function allows us to model how past case counts influence current transmission intensity.

3. **System equation:** State evolution follows

$$\boldsymbol{\theta}_t = \mathbf{g}_t(\mathbf{y}_{t-1}, \boldsymbol{\theta}_{t-1}) + \mathbf{w}_t, \quad \mathbf{w}_t \sim N(\mathbf{0}, \mathbf{W}_t). \tag{2.2.3}$$

The evolution function $\mathbf{g}_t$ depends on past observations $\mathbf{y}_{t-1}$ and/or past state $\boldsymbol{\theta}_{t-1}$. For linear evolution: $\mathbf{g}_t(\mathbf{y}_{t-1}, \boldsymbol{\theta}_{t-1}) = \mathbf{G}_t\boldsymbol{\theta}_{t-1}$, with $\mathbf{G}_t$ a matrix possibly dependent on $\mathbf{y}_{t-1}$ but not on $\boldsymbol{\theta}_{t-1}$. In epidemiological settings, this function describes how the underlying disease transmission process evolves over time, capturing phenomena like gradual changes in population behavior, virus characteristics, or effects of public health interventions.

In addition to the components 1-3 above, the DGTF model has a set of **static parameters**, denoted as $\boldsymbol{\gamma}$, that includes the scale parameter ϕ, the regression coefficients $\boldsymbol{\varphi}$, and any additional parameters in f_t and $\mathbf{g}_t$. Gathering these parameters in $\boldsymbol{\gamma}$ is important for inference purposes as described in § 2.3.

Figure 2.1 summarizes the structure of the model. The next subsections illustrate how widely used models such as Poisson autoregressions, discrete-time Hawkes-type models and distributed lag models, are special cases of DGTF models.

2.2.2. Poisson AR/GARCH models

Poisson ARs and their generalizations represent the simplest self-exciting count models. (Agosto and Giudici, 2020) use a log-linear Poisson AR models to analyze daily new

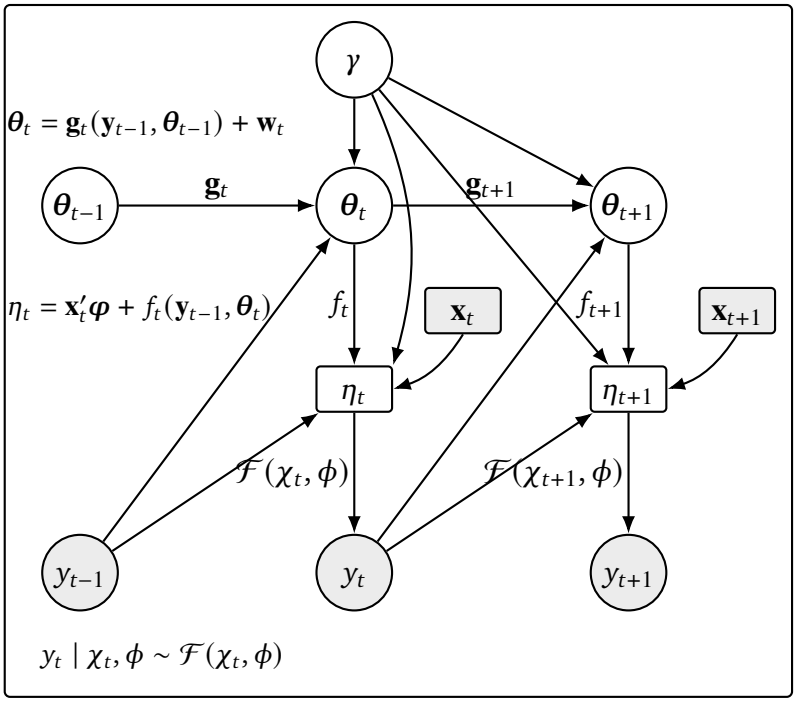


Figure 2.1.: Graphical model for the DGTF framework with three levels. The *system* level updates the latent state $\boldsymbol{\theta}_t$ from $(\boldsymbol{\theta}_{t-1}, \mathbf{y}_{t-1})$ via $\mathbf{g}_t(\cdot)$; the *link* level forms $\eta_t = \mathbf{x}_t'\boldsymbol{\varphi} + f_t(\mathbf{y}_{t-1}, \boldsymbol{\theta}_t)$; the *observation* level generates y_t from an exponential-family model $\mathcal{F}(\chi_t, \phi)$. Static parameters γ (including $\boldsymbol{\varphi}$, ϕ, and any parameters inside $f_t, \mathbf{g}_t$) affect both levels.

COVID-19 in several countries. (Giudici et al., 2023) extend these models to allow for time-varying parameters that can capture COVID-19 dynamics. Recent multivariate extensions are also available (Fokianos et al., 2022). Here we illustrate how the univariate Poisson AR can be written as the DGTF model.

A Poisson AR(p) model or integer ARCH(p) takes the form:

$$p(y_t|\lambda_t) = \frac{\lambda_t^{y_t}}{y_t!}\exp\{-\lambda_t\}, \quad \lambda_t = a + \sum_{i=1}^{p} b_i y_{t-i}, \tag{2.2.4}$$

with $a > 0$ and $b_i \geq 0$. For stationarity: $\sum_{i=1}^{p} b_i < 1$. Its DGTF representation has the following elements:

1. *Observation equation*: $y_t|\lambda_t \sim \text{Poisson}(\lambda_t)$, $\lambda_t > 0$. Therefore, in the form of Eq. (2.2.1) we have that $\phi = 1$, $\chi_t = \log(\lambda_t)$, $\mathsf{a}(\chi_t) = \exp(\chi_t)$, and $\mathsf{b}(y_t, \phi) = 1/y_t!$.

2. *Link function*: The identity link and the canonical link are two commonly used choices. The identity link is given by $\lambda_t = e^{\chi_t} = x_t\varphi + f_t(\mathbf{y}_{t-1}, \boldsymbol{\theta}_t)$, with $\mathbf{y}_{t-1} = (y_{t-1}, \ldots, y_{t-p})'$ (and so $L = p$). The identity link is used in (Rydberg and Shephard, 2000). Alternatively, the canonical link is given by $\log(\lambda_t) = \chi_t = x_t\varphi + f_t(\mathbf{y}_{t-1}, \boldsymbol{\theta}_t)$. The canonical link is used in (Held et al., 2005) and (Giudici et al., 2023). In both cases, $\boldsymbol{\theta}_t = (b_1, \ldots, b_p)$ is a p-dimensional state vector and the transfer function $f_t(\mathbf{y}_{t-1}, \boldsymbol{\theta}_t) = \mathbf{F}_t'\boldsymbol{\theta}_t$ is linear, where $\mathbf{F}_t = (y_{t-1}, \ldots, y_{t-p})'$. For the exogenous effects, we have $\varphi = a$ and $x_t = 1$, for all t.

3. *System equation*: $\boldsymbol{\theta}_t = \mathbf{g}_t(\mathbf{y}_{t-1}, \boldsymbol{\theta}_{t-1}) + \mathbf{w}_t = \mathbf{G}_t\boldsymbol{\theta}_{t-1}$, where $\mathbf{G}_t = \mathbf{I}$, $\mathbf{w}_t = \mathbf{0}$, and $\mathbf{W}_t = \mathbf{0}$ for all t, since the AR(p) parameters are not time-varying.

The static parameter is $\gamma = a$. Note that the AR coefficients are treated as latent states that remain constant over time.

Log-linear Poisson models have also been used in the literature. These models use the same observation equation and state vector, but the link function is the canonical log link, $\log(\lambda_t) = a + \sum_{i=1}^{p} b_i \log(y_{t-i}+1)$, and thus $\mathbf{F}_t = (\log(y_{t-1}+1), \ldots, \log(y_{t-p}+1))'$. Note that for this model the non-negativity of the baseline and AR coefficients is not required. Although Poisson AR models with static parameters are more parsimonious and computationally appealing, models that consider time-varying parameters offer more flexibility. This can be achieved by using a non-zero positive-definite variance-covariance matrix $\mathbf{W}_t$ at the system level. Therefore, Poisson autoregressive models

with time-varying parameters are also in the DGTF class.

The relationship to other self-exciting models becomes clear when viewing the AR coefficients $\{b_i\}$ as unnormalized lag weights analogous to the triggering kernel in the Hawkes process, a link that we made explicit in § 2.2.3. Unlike the discrete-time Hawkes-type model described next, these weights neither sum to one nor separate overall transmission intensity from temporal patterns.

2.2.3. Discrete-time Hawkes-type models

As discussed in § 1.2.2, the term *discrete-time Hawkes-type model* refers to a discrete-time self-exciting count model specified directly through its conditional mean, in the sense of Browning et al. (2021); Koyama et al. (2021). It is distinct from a binned Hawkes process (Shlomovich et al., 2020; Zhou and Papadogeorgou, 2025), whose likelihood derives from an underlying continuous-time point process observed through aggregated counts; we do not attempt to approximate that binned likelihood, nor do we claim a formal limiting connection of the kind established for INAR processes (Kirchner, 2016, 2017).

Branching processes play a central role in epidemiology, as they give us insights into the dynamics of epidemic spread. In the case of a univariate branching process, we define N_t as the number of events (e.g., number of cases of a disease) up until time t. Moving to the discrete-time domain, we focus on the number of incidents that occur during a unit-time interval $(t-1, t]$, denoted as $y_t = N_t - N_{t-1}$. In a Poisson process setting, it is assumed that the expected value of y_t is a function of the past values of y_t. For example, (Cori et al., 2013) propose the so-called *EpiEstim* model, which uses a

Poisson process and a renewal equation for conditional intensity λ_t, i.e.,

$$\lambda_t = R_t \sum_{l=1}^{t-1} y_{t-l} g_l. \tag{2.2.5}$$

Here, R_t has a piece-wise structure and g_l denotes the probability mass function of the discretized shifted Gamma distribution. This lag distribution g_l serves as a proxy for the serial interval, which measures the time between symptom onset in an infector and symptom onset of their infectee.

In the context of Covid-19 disease dynamics, (Browning et al., 2021) and (Koyama et al., 2021) consider a discrete-time Hawkes-type self-exciting count model with intensity:

$$\lambda_t = a + \sum_{i:\, t_i<t} R_{t_i} y_{t_i} k(t - t_i),\ a > 0,\ R_t > 0.$$

The first term, a, represents the baseline mean (imported cases), while the self-exciting component captures secondary transmission through a triggering kernel $k(t - t_i)$. We denote ϕ_l as the discrete-time equivalent of $k(t - l)$ and refer to $\{\phi_l, l \in \mathbb{N}^+\}$ as the lag distribution.

The discrete-time Hawkes-type model can be viewed as a normalized version of the Poisson AR model where the coefficients $\{b_i\}$ are replaced by $\{R_t\phi_l\}$, with ϕ_l forming a proper probability distribution. Furthermore, this normalization separates the overall transmission intensity (R_t) from the temporal pattern of transmission (ϕ_l), and ensures that the lag weights sum to one, facilitating interpretation and prior specification.

(Browning et al., 2021) assume that R_t has a piecewise structure and use the probability mass function of the geometric distribution as the triggering kernel. It also assumes that y_t follows a Poisson distribution. (Koyama et al., 2021) extend this using a state-space

approach to model where R_t evolves via a latent Markov random walk with independent Cauchy errors $\{w_t\}$, i.e.,

$$\lambda_t = a + \sum_{l=1}^{L} h(\psi_{t+1-l})\, y_{t-l}\, \phi_l, \;\; \psi_t = \psi_{t-1} + w_t, \tag{2.2.6}$$

where $R_t = h(\psi_t)$ with gain function $h(\cdot) \geq 0$. While (Koyama et al., 2021) use $h(\psi_t) = \max(\psi_t, 0)$, differentiable alternatives like $h(\psi_t) = \log(1+\exp(\psi_t))$ facilitate inference. The mean serial interval of COVID-19 is estimated to range from 4.2 to 7.5 days (Alene et al., 2021) and so, (Koyama et al., 2021) specify $\{\phi_l\}$ using the probability mass function (PMF) of a discretized log-normal distribution with parameters $\mu \approx 1.3863$ and $\sigma^2 \approx 0.3226$. This distribution has a mean of 4.7 and a standard deviation of 2.9.

Note that the finite sum in Eq. (2.2.6) truncates at L days. L is chosen such that $1 - \sum_{l=1}^{L} \phi_l < \epsilon$ (typically $\epsilon = 0.01$). (Koyama et al., 2021) used $L = 30$, reflecting the epidemiological assumption that infectiousness effectively ceases after 30 days. This truncation reflects the epidemiological reality that infectiousness decays to zero over time, a pattern also captured by the distributed lag models described below.

At the observation level, (Koyama et al., 2021) assumes that y_t follows a negative-binomial distribution to account for overdispersion. This model has a DGTF representation with an L-dimensional state space as follows:

1. *Observation equation*: A Poisson distribution or a negative-binomial distribution can be assumed at the observational level. In the latter case, we have that conditional on $\rho > 0$, $y_t|\lambda_t \sim \text{NB}(\lambda_t, \rho)$, $\lambda_t > 0$, i.e.:

$$p(y_t|\lambda_t, \rho) = \frac{\Gamma(y_t + \rho)}{\Gamma(\rho) y_t!} \left(\frac{\rho}{\lambda_t + \rho}\right)^{\rho} \left(\frac{\lambda_t}{\lambda_t + \rho}\right)^{y_t}. \tag{2.2.7}$$

   Again, conditioning on $\rho > 0$, this implies that $\phi = 1$, $\chi_t = \log\left(\frac{\lambda_t}{\lambda_t+\rho}\right)$, and so

$\lambda_t = \frac{\rho e^{\chi_t}}{(1-e^{\chi_t})}$. In addition $\mathsf{a}(\chi_t) = -\rho \log(1-e^{\chi_t})$, leading to $E(y_t|\lambda_t, \rho) = \mathsf{a}'(\chi_t) = \lambda_t$, and $V(y_t|\lambda_t, \rho) = \lambda_t(1 + \lambda_t/\rho)$. Note that if $\rho \to \infty$ we obtain the Poisson model with parameter λ_t.

2. *Link function*: f_t is a non-linear function of latent states $\boldsymbol{\theta}_t = (\psi_t, \ldots, \psi_{t+1-L})'$, as in Eq. (2.2.6). For the exogenous effect, we set $\varphi = a$ and $x_t = 1$ for all t. In this case, we have $\lambda_t = a + f_t(\mathbf{y}_{t-1}, \boldsymbol{\theta}_t) = a + \sum_{l=1}^{L} h(\psi_{t+1-l}) y_{t-l}\phi_l$, which corresponds to a linear discrete-time Hawkes-type conditional mean. While $h(\psi_t)$ can be any non-negative function, we prefer a $h(\psi_t)$ that is at least first-order differentiable, since differentiability supports the linearization step used in the approximate inference algorithm.

   A non-linear link can be considered. In particular, a logistic link $\rho_t = 1/(1 + e^{-\lambda_t})$ can be used, where ρ_t is the dispersion at the observation level, and $y_t \sim \text{NB2}(n, \rho_t)$, where the likelihood function is given by:

$$p(y_t \mid \rho_t, n) = \frac{\Gamma(y_t + n)}{\Gamma(n) y_t!} \rho_t^{y_t} (1-\rho_t)^n. \tag{2.2.8}$$

   This is an alternative parameterization of the negative-binomial distribution, where n represents the number of successes and ρ_t represents the probability of failures. The logistic link allows us to consider a non-linear self-exciting discrete-time count model where $R_t \in \mathbb{R}$ can describe self-exciting and self-inhibiting patterns.

3. *System equation*: $\boldsymbol{\theta}_t = \mathbf{g}_t(\mathbf{y}_{t-1}, \boldsymbol{\theta}_{t-1}) + \mathbf{w}_t = \mathbf{G}\boldsymbol{\theta}_{t-1} + \mathbf{w}_t$, with $\mathbf{w}_t = (w_t, 0, \ldots, 0)'$,

$\mathbf{W}_t = \text{diag}(W_t, 0, \ldots, 0)$, and

$$\mathbf{G}_t = \mathbf{G} = \begin{pmatrix} \boldsymbol{e}_1 & 0 \\ \mathbf{I} & \mathbf{0} \end{pmatrix}, \quad \boldsymbol{e}_1 = (1, 0, \ldots, 0).$$

For the system error distribution, (Koyama et al., 2021) assumes a Cauchy distribution. In contrast, we assume that system errors follow a Gaussian distribution with $W_t = W$ for all t. Alternatively, we can consider a time-varying W_t specified using a discount factor $\delta \in (0, 1]$. Discount factors typically range between 0.8 and 1, with smaller values indicating larger system uncertainty. A discount factor of 1 represents the special case where no system-level error exists. Details and examples related to modeling and inference with discount factors are explained and illustrated in (West and Harrison, 1997).

The static parameters $\boldsymbol{\gamma}$ in this model include ρ and a. Additionally, it may include the parameters of the lag distributions from which the PMF $\{\phi_l\}$ is derived.

2.2.4. Distributed lag models

Distributed lag models (Ravines et al., 2006) capture the same epidemiological principle as the discrete-time Hawkes-type model, namely that infectiousness decays to zero over time, but do so via recursive transfer functions that accumulate the effects of past cases. In disease modeling, this represents how individuals infected at time t continue to generate secondary infections at times $t + 1$, $t + 2$, and so on, with diminishing intensity. Such behavior can be modeled using transfer functions, denoted by β_t, in a state-space context by writing the intensity function of an observed Poisson process, denoted by λ_t,

(this is also the expected count at time t), as

$$\lambda_t = a + \beta_t.$$

The transfer function β_t captures the cumulative effect of all past observations on the current intensity.

If we consider y_t to be the number of cases of a disease at time t, we can use y_{t-1}, the number of cases of the disease at time $t-1$ as the covariate x_t. In such a case, it is reasonable to consider that the effect of $x_t = y_{t-1}$ may persist while the infected individuals are still infectious, but eventually decrease to zero. This can be modeled through the transfer function as

$$\beta_t = \kappa\beta_{t-1} + (1-\kappa)h(\psi_t)y_{t-1}.$$

Here $\kappa \in (0,1)$ controls past effects, while $(1-\kappa)h(\psi_t)y_{t-1}$ represents the immediate contribution of recent cases scaled by the gain function $h(\psi_t)$. This model, known as Koyck's model (Koyck, 1954) in the economics literature, represents κ as the memory of the effect of past observations and $(1-\kappa)h(\psi_t)$ as the immediate effect of y_{t-1}. It can be formulated as an infinite sum of the past observations:

$$\beta_t = \sum_{l=0}^{\infty} \kappa^l (1-\kappa)h(\psi_{t-l})y_{t-1-l}.$$

This form directly parallels the discrete-time Hawkes-type model in Eq. (2.2.6), with two key differences: (1) the sum extends to infinity rather than being truncated, and (2) the lag distribution ϕ_l follows a geometric distribution with $\phi_l = \kappa^l(1-\kappa)$. This geometric decay provides computational advantages through recursive representation while maintaining the epidemiological interpretation of decaying infectiousness.

(Solow, 1960) further generalized this approach by showing that we can replace the geometric distribution with the more flexible negative binomial distribution, specifically NB2(r, κ), as defined in Eq. (2.2.8), by setting ϕ_l as

$$\phi_l = \frac{\Gamma(r+l)}{\Gamma(r)\ l!}\kappa^l(1-\kappa)^r.$$

Here, r is a positive integer representing the number of successes and κ is the probability of failure. This generalization gives us the following transfer function:

$$\beta_t = \sum_{l=1}^{\infty} \phi_l h(\psi_{t+1-l}) y_{t-l}. \tag{2.2.9}$$

The geometric distribution can be considered as a special case of this model when $r = 1$. This negative-binomial lag structure provides additional flexibility to model delayed effects of past incidences, such as disease incubation periods. This infinite moving average on the right-hand side of Eq. (2.2.9) can be reformulated into a finite polynomial. Define B as the lag operation such that $B^p x_t = x_{t-p}$, then we have

$$(1-\kappa B)^r \beta_t = (1-\kappa)^r h(\psi_t) y_{t-1}. \tag{2.2.10}$$

Taking $r = 2$ as an example, the transfer function component is given by

$$\beta_t = 2\kappa\beta_{t-1} - \kappa^2\beta_{t-2} + (1-\kappa)^2 h(\psi_t) y_{t-1}.$$

Assuming a Poisson observation distribution with intensity λ_t, this model has a DGTF representation with an $(r+1)$-dimensional state space, i.e.,

1. *Observation equation*: $y_t \mid \lambda_t \sim \text{Pois}(\lambda_t)$, $\lambda_t > 0$.

2. *Link function*: We have $\boldsymbol{\theta}_t = (\psi_{t+1}, \beta_t, \beta_{t-1})'$, $\varphi = a$, $x_t = 1$ for all t and $L = 1$, implying that $\mathbf{y}_{t-1} = y_{t-1}$. The transfer function f_t is linear and can be expressed as $f_t(y_{t-1}, \boldsymbol{\theta}_t) = \mathbf{F}_t'\boldsymbol{\theta}_t$, where $\mathbf{F}_t = (0, 1, 0)'$ for all t, and thus

$$\lambda_t = x_t\varphi + f_t(y_{t-1}, \boldsymbol{\theta}_t) = x_t\varphi + \mathbf{F}_t'\boldsymbol{\theta}_t = a + \beta_t.$$

3. *System equation*: $\boldsymbol{\theta}_t = \mathbf{g}_t(y_{t-1}, \boldsymbol{\theta}_{t-1}) + \mathbf{w}_t$. Here $\mathbf{g}_t(y_{t-1}, \boldsymbol{\theta}_{t-1})$ is a non-linear function of $\boldsymbol{\theta}_{t-1}$ given by

$$\mathbf{g}_t(y_{t-1}, \boldsymbol{\theta}_{t-1}) = \begin{pmatrix} \theta_{t-1,1} \\ (1-\kappa)^2 h(\theta_{t-1,1}) y_{t-1} + 2\kappa\theta_{t-1,2} - \kappa^2\theta_{t-1,3} \\ \theta_{t-1,2} \end{pmatrix}, \tag{2.2.11}$$

   and the system error is $\mathbf{w}_t = (\varepsilon_t, 0, 0)'$ with variance-covariance matrix $\mathbf{W}_t = \text{diag}(W_t, 0, 0)$.

Finally, the vector of static parameter is $\boldsymbol{\gamma} = (a, \kappa)'$.

2.2.5. Summary of DGTF model variants

We showed above how Poisson ARs, discrete-time Hawkes-type models and distributed lag models are special model classes within the class of DGTF models. We also mentioned how the 3 types of models are related. Here we discuss the core structural differences between the 3 types of models. More specifically, Table 2.1 highlights the differences between the models within the DGTF framework. While all models incorporate past observations through self-exciting dynamics, they differ fundamentally in how they parameterize lag structures and time-varying transmission rates. Each model offers distinct computational and interpretational advantages for epidemiological

applications.

First we compare the transfer function structure. For the 3 models we can write $\alpha_t = a + \beta_t$, with β_t a transfer function. We see that in the case of the Poisson AR $\beta_t = \sum_{i=1}^{p} b_i y_{t-i}$, and so the relationship between β_t and past values of the observations y_t is linear. In the case of the discrete-time Hawkes-type model we have $\beta_t = \sum_{l=1}^{L} \phi_l h(\psi_{t+1-1}) y_{t-l}$, and so the relationship between β_t and past values of y_t is also linear, but we only consider a finite number of past values and there is a non-linear function $h(\cdot)$ that established the weights for each of the past observations as a non-linear function of the state parameters. In the case of the distributed lag models, β_t is a linear function of past values of y_t, but we have an infinite number of lags, the weights are determined through a non-linear gain function of the state parameters and there is also a dependency with past values of β_t.

Table 2.1.: Comparisons of DGTF model variants in terms of model structure and epidemiological interpretation.

	Poisson AR	Discrete-time Hawkes-type model	Distributed Lag
Transfer function β_t	Linear states and y_t $\sum_{i=1}^{p} b_i y_{t-l}$	Nonlinear states and linear y_t $\sum_{l=1}^{L} \phi_l h(\psi_{t+1-l}) y_{t-l}$	Nonlinear states and linear y_t $\sum_{l=1}^{\infty} \phi_l h(\psi_{t+1-l}) y_{t-l}$
Lag structure	Unnormalized $\{b_i\}$ or $\{b_{i,t}\}$	Normalized lag PMF $\{\phi_l\}$	NB2(r, κ) lag PMF $\{\phi_l\}$
Reproduction number	–	Explicit: $R_t = h(\psi_t)$	Explicit: $R_t = h(\psi_t)$
Infectivity duration	Finite: up to p days.	Finite: up to L days	Infinite but exponentially decaying
State dimension	p (number of lags)	L (truncation point)	$r + 1$ (polynomial order)

Similarly, we can compare the models in terms of the coefficients that weigh the lag structure. Poisson AR models offer computational simplicity for short-term dependencies. They directly model lag effects through coefficients b_i, but these unnormalized weights lack clear epidemiological interpretation. The model can incorporate time-varying parameters by allowing $b_{i,t}$ to evolve stochastically, although this increases computational complexity. Alternatively, Discrete-time Hawkes-type models provide a

clearer epidemiological interpretation by explicitly separating the reproduction number $R_t = h(\psi_t)$ from the temporal pattern of transmission captured by the lag distribution ϕ_l. This separation facilitates prior specification and interpretation of the results: the lag distribution can be chosen to match the estimated serial intervals, while R_t directly quantifies the transmission intensity. The main limitation is the need to truncate at lag L, which requires careful selection to balance computational cost against approximation error. Distributed lag models are well suited when computational efficiency matters or when dealing with long-memory processes. The recursive formulation of β_t as a function of past values of the observations, but also past values of β_t avoids storing all L past observations, requiring only r lagged values of the transfer function β_t. While the lag distribution is restricted to geometric ($r = 1$) or negative binomial ($r > 1$) families, these often provide adequate flexibility for epidemiological applications. The recursive structure also naturally accommodates infinite lags with exponential decay.

Therefore, the modular structure of the DGTF framework enables straightforward model customization. Practitioners can interchange observation distributions (e.g., Poisson vs. negative binomial), link functions, lag distributions, and gain functions without having to reimplement algorithms for posterior inference.

This unified perspective not only clarifies theoretical relationships between seemingly disparate models, but also enables practical benefits: principled model comparison using consistent inference procedures, efficient implementation through shared computational infrastructure, and principled model selection based on data characteristics and inferential goals.

§ 2.3. Posterior inference methods

The non-linear nature of DGTF models requires approximation strategies for tractable inference. We first construct an auxiliary linear Gaussian model through first-order Taylor expansion of the observation and evolution functions. This auxiliary model serves as the foundation for our inference algorithms, providing tractable proposal distributions while maintaining computational efficiency. We then present three inference approaches below: a SMC scheme for inference of the latent states $\boldsymbol{\theta}_t$, conditional on the static parameters $\boldsymbol{\gamma}$, a HVA for joint approximate parameter and state inference, and a MCMC approach for joint parameter and state inference. The HVA method is an iterative procedure with two-stages, one that uses SMC to obtain estimates of $\boldsymbol{\theta}_{1:T}$ conditional on $\boldsymbol{\gamma}$, and another stage that uses variational optimization to obtain estimates of $\boldsymbol{\gamma}$ conditional on $\boldsymbol{\theta}_{1:T}$. We use MCMC as a benchmark method and show the advantages of the proposed HVA method. We defer detailed comparisons of computational trade-offs to the empirical sections.

2.3.1. Auxiliary linear-Gaussian representation

For convenience, we omit the input notation for the dependence of f_t and $\mathbf{g}_t$ on $\mathbf{y}_{t-1}$ for the remainder of this section. Additionally, assume we have m static parameters $\boldsymbol{\gamma} = \{\gamma_1, \ldots, \gamma_m\}$.

For the evolution function $\mathbf{g}_t$, we approximate

$$\mathbf{g}_t(\boldsymbol{\theta}_{t-1}) \approx \mathbf{g}_t(\boldsymbol{\vartheta}_{t-1}) + \mathbf{G}_t(\boldsymbol{\theta}_{t-1} - \boldsymbol{\vartheta}_{t-1}),$$

where $\boldsymbol{\vartheta}_{t-1}$ is the point of approximation, and $\mathbf{G}_t$ denotes the gradient of $\mathbf{g}_t$ evaluated at

$\boldsymbol{\vartheta}_{t-1}$. The resulting linearized evolution equation is

$$\boldsymbol{\theta}_t \approx \mathbf{h}_t + \mathbf{G}_t\boldsymbol{\theta}_{t-1} + \mathbf{w}_t, \text{ where } \mathbf{h}_t = \mathbf{g}_t(\boldsymbol{\vartheta}_{t-1}) - \mathbf{G}_t\boldsymbol{\vartheta}_{t-1} \text{ and } \mathbf{w}_t \sim N(\mathbf{0}, \mathbf{W}_t). \tag{2.3.1}$$

Similarly, for the observation function f_t we have:

$$f_t(\boldsymbol{\theta}_t) \approx d_t + \mathbf{F}_t'\boldsymbol{\theta}_t, \text{ where } d_t = f_t(\mathbf{u}_t) - \mathbf{F}_t'\mathbf{u}_t. \tag{2.3.2}$$

Here, $\mathbf{u}_t$ is the approximation point and $\mathbf{F}_t$ is the gradient of f_t at $\mathbf{u}_t$. Setting $\tilde{y}_t = \zeta(y_t)$, we apply the delta method to obtain an approximate normal likelihood:

$$\tilde{y}_t \approx N(\tilde{\eta}_t, \tilde{V}_t), \quad \tilde{\eta}_t = \mathbf{x}_t'\boldsymbol{\varphi} + d_t + \mathbf{F}_t'\boldsymbol{\theta}_t, \quad \tilde{V}_t = \mathrm{Var}(y_t)[\zeta'(y_t)]^2, \tag{2.3.3}$$

where $\zeta(\cdot)$ represents the link function as in Eq. (2.2.2). These approximations enable efficient computation across all our inference methods.

We use Eqs. (2.3.1) to (2.3.3) to obtain filtering and smoothing of the state parameters in the SMC method, and equations similar to Eq. (2.3.3) to build proposal distributions for the Metropolis–Hastings (MH) steps in the MCMC algorithm.

2.3.2. SMC for latent states

SMC approximates the filtering and smoothing distributions of state parameters using a set of weighted particles. For DGTF models, SMC tracks the evolution of latent states over time conditional on the static parameters $\boldsymbol{\gamma}$, while handling the non-linear dynamics and non-Gaussian observations. This also allows us to track the evolution of functions of the state parameters such as the reproduction number in models for disease dynamics.

We adopt the modified two-filter smoothing algorithm of (Fearnhead et al., 2010). The two-filter smoother works by running two complementary filters: a forward filter

that propagates uncertainty from past to future, and a backward filter that incorporates future observations to refine estimates of past states.

The algorithm achieves computational complexity of $O(N)$ and generates new samples during the smoothing process, which helps address particle degeneracy issues common in SMC methods.

Forward filtering

The forward filter updates our belief about latent states sequentially as new data arrives. At each time step t, we maintain a collection of particles with associated weights representing the distribution of the state parameters $\boldsymbol{\theta}_t$.

More specifically, we seek to approximate the posterior density $p(\boldsymbol{\theta}_t \mid y_{1:t}, \boldsymbol{\gamma})$, which can be decomposed as follows:

$$p(\boldsymbol{\theta}_t \mid y_{1:t}, \boldsymbol{\gamma}) \propto p(y_t \mid \boldsymbol{\theta}_t, \boldsymbol{\gamma}) \int p(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{t-1}, \boldsymbol{\gamma}) p(\boldsymbol{\theta}_{t-1} \mid y_{1:t-1}, \boldsymbol{\gamma}) \mathrm{d}\boldsymbol{\theta}_{t-1}.$$

SMC methods approximate $p(\boldsymbol{\theta}_t \mid y_{1:t}, \boldsymbol{\gamma})$ by a set of particles $\{\boldsymbol{\theta}_t^{(i)}\}_{i=1}^N$ and a set of weights $\{\omega_t^{(i)}\}_{i=1}^N$. Assuming we have the same type of particle approximation at time $t-1$ for $p(\boldsymbol{\theta}_{t-1} \mid y_{1:t-1}, \boldsymbol{\gamma})$ we obtain

$$p(\boldsymbol{\theta}_t \mid y_{1:t}, \boldsymbol{\gamma}) \approx c p(y_t \mid \boldsymbol{\theta}_t, \boldsymbol{\gamma}) \sum_{i=1}^{N} \omega_{t-1}^{(i)} p(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{t-1}^{(i)}, \boldsymbol{\gamma}),$$

where c is a normalizing constant.

There are different SMC methods for obtaining a set of weighted particles that approximate the filtering distribution. The auxiliary particle filter of (Pitt and Shephard,

1999) offers a general method by approximating

$$cp(y_t \mid \boldsymbol{\theta}_t, \boldsymbol{\gamma})p(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{t-1}^{(i)}, \boldsymbol{\gamma})\omega_{t-1}^{(i)}$$

with $q(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{t-1}^{(i)}, y_t, \boldsymbol{\gamma})v_t^{(i)}$, such that $q(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{t-1}^{(i)}, y_t, \boldsymbol{\gamma})$ is the distribution from which we sample the new particles $\{\boldsymbol{\theta}_t^{(i)}\}$, and $\{v_t^{(i)}\}$ is a collection of normalized weights that sum to one. The simplest case is the algorithm of (Gordon et al., 1993) that sets

$$q(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{t-1}, y_t, \boldsymbol{\gamma}) = p(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}, \boldsymbol{\gamma}), \quad v_t^{(i)} = \omega_{t-1}^{(i)}.$$

The most efficient algorithm is obtained by taking

$$q(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{t-1}^{(i)}, y_t, \boldsymbol{\gamma}) = p(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{t-1}^{(i)}, y_t, \boldsymbol{\gamma}), \quad v_t^{(i)} \propto p(y_t \mid \boldsymbol{\theta}_{t-1}^{(i)}, \boldsymbol{\gamma})\omega_{t-1}^{(i)},$$

leading to optimal weights of $\omega_t^{(i)} = 1/N$.

Note that $p(y_t \mid \boldsymbol{\theta}_{t-1}, \boldsymbol{\gamma})$ and $p(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{t-1}, y_t, \boldsymbol{\gamma})$ have analytical forms in the DGTF model. To overcome this, we build importance densities using an approximated normal DLM based on our earlier linearizations (Eqs. (2.3.2) and (2.3.3)):

$$\tilde{y}_t \sim N(\tilde{\eta}_t, \tilde{V}_t) \text{ and } \boldsymbol{\theta}_t \sim N(\mathbf{g}_t(\boldsymbol{\theta}_{t-1}), \mathbf{W}_t).$$

The calculation of $\tilde{\eta}_t$ in Eq. (2.3.3) involves $\mathbf{F}_t$, which is the derivative of $f_t(\boldsymbol{\theta}_t)$ evaluated at $\mathbf{g}_t(\boldsymbol{\theta}_{t-1})$, ensuring our linearization is centered at the predicted state value. The auxiliary variable $\tilde{y}_t$ represents a Gaussian approximation to our count data. We transform discrete counts to a continuous scale through the link function and apply the delta method for variance approximation, enabling efficient Gaussian filtering techniques that noticeably reduce computational cost.

Then, we use $q(\tilde{y}_t \mid \boldsymbol{\theta}_t, \boldsymbol{\gamma})$ to denote the conditional likelihood of $\tilde{y}_t$ in our ap-

proximated model. By marginalizing over $\boldsymbol{\theta}_t$, we obtain the one-step-ahead predictive distribution:

$$q(\tilde{y}_t \mid \boldsymbol{\theta}_{t-1}, \boldsymbol{\gamma}) = \int q(\tilde{y}_t \mid \boldsymbol{\theta}_t, \boldsymbol{\gamma})\; p(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{t-1}, \boldsymbol{\gamma})\; \mathrm{d}\boldsymbol{\theta}_t,$$
$$= (2\pi\tilde{V}_t|\mathbf{W}_t|)^{-\frac{1}{2}}|\boldsymbol{\Sigma}_t|^{\frac{1}{2}} \exp\left\{-\frac{\frac{(\tilde{y}_t - d_t)^2}{\tilde{V}_t} + \mathbf{g}_t'(\boldsymbol{\theta}_{t-1})\mathbf{W}_t{}^{-1}\mathbf{g}_t(\boldsymbol{\theta}_{t-1}) - \boldsymbol{\mu}_t'\boldsymbol{\Sigma}_t^{-1}\boldsymbol{\mu}_t}{2}\right\}.$$

We propagate the latent states $\boldsymbol{\theta}_t$ using $q(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{t-1}, \tilde{y}_t, \boldsymbol{\gamma})$, which follows a normal distribution $N(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$ with:

$$\boldsymbol{\mu}_t = \boldsymbol{\Sigma}_t\left\{\mathbf{F}_t\tilde{V}_t^{-1}\left[\tilde{y}_t - \tilde{\eta}_t(\mathbf{g}_t(\boldsymbol{\theta}_{t-1}))\right] + \mathbf{W}_t{}^{-1}\mathbf{g}_t(\boldsymbol{\theta}_{t-1})\right\}, \quad \boldsymbol{\Sigma}_t = [\mathbf{F}_t\tilde{V}_t^{-1}\mathbf{F}_t' + \mathbf{W}_t{}^{-1}]^{-1}.$$

This Gaussian approximation performs well in practice because the linearization captures the local curvature of the likelihood around the predicted states, and the auxiliary particle filter framework corrects for approximation errors through importance weighting as explained below.

In an auxiliary particle filter, we first resample the particles $\{\boldsymbol{\theta}_{t-1}^{(i)}\}$ using weights that account for how well each particle predicts the next , and so we use $q(\tilde{y}_t \mid \boldsymbol{\theta}_{t-1}^{(i)}, \boldsymbol{\gamma})\omega_{t-1}^{(i)}$. Then, for each particle $\boldsymbol{\theta}_{t-1}^{(i)}$, we draw a new sample $\boldsymbol{\theta}_t^{(i)}$ from $q(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{t-1}^{(i)}, y_t, \boldsymbol{\gamma})$. Finally, we calculate the new importance weights for $\{\boldsymbol{\theta}_t^{(i)}\}$ via

$$\omega_t^{(i)} \propto \frac{p(y_t \mid \boldsymbol{\theta}_t, \boldsymbol{\gamma})p(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{t-1}^{(i)}, \boldsymbol{\gamma})}{q(\tilde{y}_t \mid \boldsymbol{\theta}_{t-1}^{(i)}, \boldsymbol{\gamma})q(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{t-1}^{(i)}, \tilde{y}_t, \boldsymbol{\gamma})}.$$

To minimize particle degeneracy, we resample based on the effective sample size

(ESS) computed as

$$\text{ESS} = \frac{\left(\sum_i \omega_t^{(i)}\right)^2}{\sum_i \left(\omega_t^{(i)}\right)^2}.$$

We resample when ESS $< mN$, where N is the total number of particles and m is a constant in $[0, 1]$, typically 0.5 (Doucet and Johansen, 2011). Then, we resample when roughly half our particles have negligible weight, preventing the filter from collapsing to just a few particles. If resampling is performed, the importance weights are reset to 1. Algorithm 1 summarizes the forward filtering steps.

Algorithm 1: Forward Filtering

Input: $y_{1:T}$, $\boldsymbol{\gamma}$, N particles, resample ratio m
Output: Particles $\{(\boldsymbol{\theta}_t^{(i)}, \omega_t^{(i)})\}_{i=1}^N$, $t = 1, \ldots, T$
Initialize: $\boldsymbol{\theta}_0^{(i)} \sim p(\boldsymbol{\theta}_0)$, $\omega_0^{(i)} = 1/N$ for $i = 1, \ldots, N$
for $t = 1$ **to** T **do**
  `// Resample`
  Resample indices $\{i_k\}$ based on predictive weights $v_t^{(i)} \propto q(\tilde{y}_t|\boldsymbol{\theta}_{t-1}^{(i)}, \boldsymbol{\gamma})\omega_{t-1}^{(i)}$;
  `// Propagate`
  for $k = 1$ **to** N **do**
    Sample $\boldsymbol{\theta}_t^{(k)} \sim N(\boldsymbol{\mu}_t^{(i_k)}, \boldsymbol{\Sigma}_t^{(i_k)})$;
  `// Reweight`
  Calculate weights $\{\omega_t^{(k)}\}$ and the ESS; **if** *ESS* $< mN$ **then**
    Resample and reset $\omega_t^{(k)} = 1/N$.

Backward filtering

The forward filter provides $p(\boldsymbol{\theta}_t \mid y_{1:t}, \boldsymbol{\gamma})$, our belief about states given data up to time t. To incorporate all the available data and obtain the complete smoothing distribution $p(\boldsymbol{\theta}_t \mid y_{1:T}, \boldsymbol{\gamma})$, we need a backward filter that computes the likelihood of each state

given the observations from t to T.

Following (Fearnhead et al., 2010), we construct backward filtering weighted particles $\{(\tilde{\boldsymbol{\theta}}_t^{(j)}, \tilde{\omega}_t^{(j)})\}_{j=1}^N$, that approximate a backward filter density

$$\tilde{p}(\boldsymbol{\theta}_t \mid y_{t:T}, \boldsymbol{\gamma}) \propto \pi_t(\boldsymbol{\theta}_t|\boldsymbol{\gamma}) p(y_{t:T} \mid \boldsymbol{\theta}_t, \boldsymbol{\gamma}).$$

Here, $\pi_t(\boldsymbol{\theta}_t|\boldsymbol{\gamma})$ is an artificial density that ensures the backward recursion remains well-defined, typically chosen as the predictive distribution from a simple forward model. We explain how to obtain the particle approximation of $\tilde{p}(\boldsymbol{\theta}_t \mid y_{t:T}, \boldsymbol{\gamma})$ before showing how to combine it with the forward particles.

First note that

$$p(y_{t:T} \mid \boldsymbol{\theta}_t, \boldsymbol{\gamma}) = p(y_t \mid \boldsymbol{\theta}_t, \boldsymbol{\gamma}) \int p(\boldsymbol{\theta}_{t+1} \mid \boldsymbol{\theta}_t, \boldsymbol{\gamma}) p(y_{t+1:T} \mid \boldsymbol{\theta}_{t+1}, \boldsymbol{\gamma}) \mathrm{d}\boldsymbol{\theta}_{t+1},$$

and so, we obtain

$$\begin{aligned}
\tilde{p}(\boldsymbol{\theta}_t \mid y_{t:T}, \boldsymbol{\gamma}) \propto & \pi_t(\boldsymbol{\theta}_t|\boldsymbol{\gamma}) p(y_t \mid \boldsymbol{\theta}_t, \boldsymbol{\gamma}) \int \frac{p(\boldsymbol{\theta}_{t+1} \mid \boldsymbol{\theta}_t, \boldsymbol{\gamma}) \tilde{p}(\boldsymbol{\theta}_{t+1} \mid y_{t+1:T}, \boldsymbol{\gamma})}{\pi_{t+1}(\boldsymbol{\theta}_{t+1} \mid \boldsymbol{\gamma})} d\boldsymbol{\theta}_{t+1}, \\
\approx & \pi_t(\boldsymbol{\theta}_t \mid \boldsymbol{\gamma}) p(y_t \mid \boldsymbol{\theta}_t, \boldsymbol{\gamma}) \sum_{j=1}^{J} \frac{p(\tilde{\boldsymbol{\theta}}_{t+1}^{(j)} \mid \boldsymbol{\theta}_t, \boldsymbol{\gamma})}{\pi_{t+1}(\tilde{\boldsymbol{\theta}}_{t+1}^{(j)} \mid \boldsymbol{\gamma})} \tilde{\omega}_{t+1}^{(j)},
\end{aligned}$$

where $\{(\tilde{\boldsymbol{\theta}}_{t+1}^{(j)}, \tilde{\omega}_{t+1}^{(j)})\}_{j=1}^N$ is a weighted particle approximation to $\tilde{p}(\boldsymbol{\theta}_{t+1} \mid y_{t+1:T}, \boldsymbol{\gamma})$ and $\pi_t(\boldsymbol{\theta}_t|\boldsymbol{\gamma})$ is an artificial marginal density (Briers et al., 2010; Fearnhead et al., 2010) such that $\pi_t(\boldsymbol{\theta}_t|\boldsymbol{\gamma}) > 0$ if $p(y_{t:T}|\boldsymbol{\theta}_t, \boldsymbol{\gamma}) > 0$.

Similar to the forward case, we need efficient proposals for the backward recursion. Following (Fearnhead et al., 2010), we use an auxiliary backward particle filter to obtain

a sampling distribution $\tilde{q}(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t+1}, \tilde{y}_t, \boldsymbol{\gamma})$, such that

$$\tilde{q}(\boldsymbol{\theta}_t \mid \tilde{\boldsymbol{\theta}}_{t+1}^{(j)}, \tilde{y}_t, \boldsymbol{\gamma})\tilde{v}_t^{(j)} \approx \pi_t(\boldsymbol{\theta}_t \mid \boldsymbol{\gamma})p(y_t \mid \boldsymbol{\theta}_t, \boldsymbol{\gamma})p(\tilde{\boldsymbol{\theta}}_{t+1}^{(j)}|\boldsymbol{\theta}_t, \boldsymbol{\gamma})\frac{\tilde{\omega}_{t+1}^{(j)}}{\pi_{t+1}(\tilde{\boldsymbol{\theta}}_{t+1}^{(j)} \mid \boldsymbol{\gamma})}.$$

We take $\pi_{t-1}(\boldsymbol{\theta}_{t-1}|\boldsymbol{\gamma})$ to be a normal distribution $N(\boldsymbol{\theta}_{t-1} \mid \boldsymbol{\mu}_{t-1}^*, \boldsymbol{\Sigma}_{t-1}^*)$. Then, using the linear approximation in Eq. (2.2.3), we obtain $\pi_t(\boldsymbol{\theta}_t \mid \boldsymbol{\gamma})$ to be $N(\boldsymbol{\theta}_t \mid \boldsymbol{\mu}_t^*, \boldsymbol{\Sigma}_t^*)$, with

$$\boldsymbol{\mu}_t^* = \mathbf{g}_t(\boldsymbol{\mu}_{t-1}^*), \quad \boldsymbol{\Sigma}_t^* = \mathbf{W}_t + \mathbf{G}_t\boldsymbol{\Sigma}_{t-1}^*\mathbf{G}_t'.$$

Here $\mathbf{G}_t$ is the Jacobian of $\mathbf{g}_t(\boldsymbol{\theta}_{t-1})$ evaluated at $\boldsymbol{\theta}_{t-1} = \boldsymbol{\mu}_{t-1}^*$. This choice ensures that the artificial density evolves according to the same dynamics as our state equation, making the backward recursion consistent with the model structure. We initialize this with $\boldsymbol{\theta}_0 \sim N(\boldsymbol{\mu}_0^*, \boldsymbol{\Sigma}_0^*)$, for example, $\boldsymbol{\mu}_0^* = \mathbf{0}$, and $\boldsymbol{\Sigma}_0^* = \mathbf{I}$.

The backward transition density is given by

$$q(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{t+1}, \boldsymbol{\gamma}) \propto \pi_t(\boldsymbol{\theta}_t \mid \boldsymbol{\gamma})q(\boldsymbol{\theta}_{t+1} \mid \boldsymbol{\theta}_t, \boldsymbol{\gamma}),$$

using $(\boldsymbol{\theta}_t \mid \boldsymbol{\gamma}) \sim N(\boldsymbol{\mu}_t^*, \boldsymbol{\Sigma}_t^*)$, and $(\boldsymbol{\theta}_{t+1} \mid \boldsymbol{\theta}_t, \boldsymbol{\gamma}) \sim N(\mathbf{h}_{t+1} + \mathbf{G}_{t+1}\boldsymbol{\theta}_t, \mathbf{W}_t)$, with $\mathbf{G}_{t+1}$ the first order derivative of $\mathbf{g}_{t+1}$ evaluated at $\boldsymbol{\mu}_t^*$. Then, $q(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{t+1}, \boldsymbol{\gamma})$ is a normal distribution with mean and variance given by

$$\mathbf{m}_t^* = \mathbf{C}_t^*[(\boldsymbol{\Sigma}_t^*)^{-1}\boldsymbol{\mu}_t^* + \mathbf{G}_{t+1}'\mathbf{W}_{t+1}^{-1}(\boldsymbol{\theta}_{t+1} - \mathbf{h}_{t+1})], \quad \mathbf{C}_t^* = [(\boldsymbol{\Sigma}_t^*)^{-1} + \mathbf{G}_{t+1}'\mathbf{W}_{t+1}^{-1}\mathbf{G}_{t+1}]^{-1}.$$

Using the normal approximation $\tilde{y}_t \sim N(\tilde{\eta}_t, \tilde{V}_t)$, we get

$$\begin{aligned}
\tilde{q}(\tilde{y}_t \mid \boldsymbol{\theta}_{t+1}, \boldsymbol{\gamma}) &= \int q(\tilde{y}_t \mid \boldsymbol{\theta}_t, \boldsymbol{\gamma})q(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{t+1}, \boldsymbol{\gamma})d\boldsymbol{\theta}_t \\
&= (2\pi\tilde{V}_t|\mathbf{C}_t^*|)^{-1/2}|\tilde{\boldsymbol{\Sigma}}_t^*|^{1/2} \times \\
&\quad \times \exp\left\{-\frac{1}{2}\left[\tilde{V}_t^{-1}\left(\tilde{y}_t - \boldsymbol{x}_t'\boldsymbol{\varphi} - d_t\right)^2 + \mathbf{m}_t^{*'}(\mathbf{C}_t^*)^{-1}\mathbf{m}_t^* + (\tilde{\boldsymbol{\mu}}_t^*)'\tilde{\boldsymbol{\Sigma}}_t^{*-1}\tilde{\boldsymbol{\mu}}_t^*\right]\right\}
\end{aligned}$$

and $q(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{t+1}, \tilde{y}_t, \boldsymbol{\gamma})$ a normal distribution with mean and variance given by

$$\tilde{\boldsymbol{\mu}}_t^* = \tilde{\boldsymbol{\Sigma}}_t^* \left[\mathbf{F}_t \tilde{V}_t^{-1} (\tilde{y}_t - \boldsymbol{x}_t' \boldsymbol{\varphi} - d_t) + \mathbf{C}_t^{*-1} \mathbf{m}_t^* \right], \quad \tilde{\boldsymbol{\Sigma}}_t^* = \left[\tilde{V}_t^{-1} \mathbf{F}_t \mathbf{F}_t' + (\mathbf{C}_t^*)^{-1} \right]^{-1}.$$

Finally, we can derive the importance weights as

$$\tilde{\omega}_t^{(j)} \propto \frac{\pi_t(\boldsymbol{\theta}_t \mid \boldsymbol{\gamma}) p(y_t \mid \boldsymbol{\theta}_t, \boldsymbol{\gamma}) p(\tilde{\boldsymbol{\theta}}_{t+1}^{(j)} \mid \boldsymbol{\theta}_t, \boldsymbol{\gamma})}{\pi_{t+1}(\tilde{\boldsymbol{\theta}}_{t+1}^{(j)} \mid \boldsymbol{\gamma}) \tilde{q}(\tilde{y}_t \mid \tilde{\boldsymbol{\theta}}_{t+1}^{(j)}, \boldsymbol{\gamma}) \tilde{q}(\boldsymbol{\theta}_t \mid \tilde{\boldsymbol{\theta}}_{t+1}^{(j)}, y_t, \boldsymbol{\gamma})}. \tag{2.3.4}$$

In this auxiliary backward filter, we first resample the particles $\{\boldsymbol{\theta}_{t+1}^{(j)}\}$ using weights proportional to $\tilde{q}(\tilde{y}_t \mid \tilde{\boldsymbol{\theta}}_{t+1}^{(j)}, \boldsymbol{\gamma}) \tilde{\omega}_{t+1}^{(j)}$. Then, we generate new particles $\{\tilde{\boldsymbol{\theta}}_t^{(j)}\}$ by sampling from $\tilde{q}(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{t+1}, y_t, \boldsymbol{\gamma})$. Finally, we calculate the new importance weights and perform resampling if needed. Algorithm 2 summarizes the steps of the backward sampling scheme.

Algorithm 2: Backward Filtering

Input: Forward particles at T: $\{(\boldsymbol{\theta}_T^{(j)}, \omega_T^{(j)})\}_{j=1}^N$, $y_{1:T}$, $\boldsymbol{\gamma}$, resampling ratio m

Output: Backward particles $\{(\tilde{\boldsymbol{\theta}}_t^{(j)}, \tilde{\omega}_t^{(j)})\}_{j=1}^N$, $t = T-1, \ldots, 1$

Initialize: $\tilde{\boldsymbol{\theta}}_T^{(j)} = \boldsymbol{\theta}_T^{(j)}$, $\tilde{\omega}_T^{(j)} = \omega_T^{(j)}$ for $j = 1, \ldots, N$

for $t = T-1$ **to** 1 **do**

`// Resample`

Resample indices $\{j_k\}$ with weights $\left\{\tilde{q}(\tilde{y}_t | \tilde{\boldsymbol{\theta}}_{t+1}^{(j)}, \boldsymbol{\gamma}) \tilde{\omega}_{t+1}^{(j)}\right\}$;

`// Propagate`

for $k = 1$ **to** N **do**

Sample $\tilde{\boldsymbol{\theta}}_t^{(k)} \sim N((\tilde{\boldsymbol{\mu}}_t^*)^{(j_k)}, (\tilde{\boldsymbol{\Sigma}}_t^*)^{(j_k)})$;

`// Reweight`

Calculate weights $\{\tilde{\omega}_t^{(k)}\}$ using Eq. (2.3.4); **if** $ESS < mN$ **then**

Resample and reset $\tilde{\omega}_t^{(k)} = 1/N$.

Two-filter smoothing

The key insight of the two-filter smoothing is combining information from both temporal directions. The forward filter provides particles representing $p(\boldsymbol{\theta}_t \mid y_{1:t})$ while the backward filter provides particles approximating $p(\boldsymbol{\theta}_t \mid y_{t:T})$. Combining these yields the full smoothing distribution $p(\boldsymbol{\theta}_t \mid y_{1:T})$, which incorporates all available information about the latent states.

Assume we have a set of weighted particles from the forward and backward filters, denoted as $\{(\boldsymbol{\theta}_t^{(i)}, \omega_t^{(i)})\}$ and $\{(\tilde{\boldsymbol{\theta}}_t^{(j)}, \tilde{\omega}_t^{(j)})\}$, respectively. Following the two-filter smoothing approach of (Fearnhead et al., 2010) we combine the forward weighted particles at time $t-1$ and the backward weighted particles at time $t+1$ to obtain the following approximation to the smoothing distribution $p(\boldsymbol{\theta}_t \mid y_{1:T}, \boldsymbol{\gamma})$:

$$p(\boldsymbol{\theta}_t|y_{1:T}, \boldsymbol{\gamma}) \approx c \sum_{i=1}^{N} \sum_{j=1}^{N} p(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}^{(i)}, \boldsymbol{\gamma}) \omega_{t-1}^{(i)} p(y_t|\boldsymbol{\theta}_t, \boldsymbol{\gamma}) \frac{p(\tilde{\boldsymbol{\theta}}_{t+1}^{(j)}|\boldsymbol{\theta}_t, \boldsymbol{\gamma})}{\pi_{t+1}(\tilde{\boldsymbol{\theta}}_{t+1}^{(j)} \mid \boldsymbol{\gamma})} \tilde{\omega}_{t+1}^{(j)}$$

with c a normalizing constant.

To sample from this complex distribution, we use an auxiliary particle filtering approach that finds a sampling distribution $\bar{q}(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}^{(i)}, \boldsymbol{\theta}_{t+1}^{(j)}, y_t, \boldsymbol{\gamma})$ and weights $\bar{v}_t^{(i,j)}$ such that

$$\bar{q}(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}^{(i)}, \boldsymbol{\theta}_{t+1}^{(j)}, y_t, \boldsymbol{\gamma}) \bar{v}_t^{(i,j)} \approx p(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}^{(i)}, \boldsymbol{\gamma}) p(y_t|\boldsymbol{\theta}_t, \boldsymbol{\gamma}) p(\tilde{\boldsymbol{\theta}}_{t+1}^{(j)}|\boldsymbol{\theta}_t, \boldsymbol{\gamma}) \frac{\omega_{t-1}^{(i)} \tilde{\omega}_{t+1}^{(j)}}{\pi_{t+1}(\tilde{\boldsymbol{\theta}}_{t+1}^{(j)} \mid \boldsymbol{\gamma})}.$$

Our linearization yields $\bar{q}(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}_{t+1}, y_t, \boldsymbol{\gamma})$ as a normal distribution with mean $\boldsymbol{\mu}_{t|T}$

and covariance matrix $\boldsymbol{\Sigma}_{t|T}$:

$$\boldsymbol{\mu}_{t|T} = \boldsymbol{\Sigma}_{t|T}\left[\tilde{V}_t^{-1}\mathbf{F}_t(\tilde{y}_t - \boldsymbol{x}_t'\boldsymbol{\varphi} - d_t) + \mathbf{W}_t^{-1}\mathbf{g}_t(\boldsymbol{\theta}_{t-1}) + \mathbf{G}_{t+1}'\mathbf{W}_{t+1}^{-1}(\boldsymbol{\theta}_{t+1} - \mathbf{h}_{t+1})\right], \tag{2.3.5}$$

$$\boldsymbol{\Sigma}_{t|T} = \left[\mathbf{W}_t^{-1} + \mathbf{G}_{t+1}'\mathbf{W}_{t+1}^{-1}\mathbf{G}_{t+1} + \tilde{V}_t^{-1}\mathbf{F}_t\mathbf{F}_t'\right]^{-1}. \tag{2.3.6}$$

Following the two-filter smoother of (Fearnhead et al., 2010), we obtain a particle approximation to $p(\boldsymbol{\theta}_t|y_{1:T}, \boldsymbol{\gamma})$ through four steps. First, we resample the forward filtering particles $\{\boldsymbol{\theta}_{t-1}^{(i)}\}$ with weights $\{q(\tilde{y}_t|\boldsymbol{\theta}_{t-1}^{(i)}, \boldsymbol{\gamma})\omega_{t-1}^{(i)}\}$, leading to a new set of indices $\{i_k\}$. Second, we resample the backward filtering particles $\{\tilde{\boldsymbol{\theta}}_{t+1}^{(j)}\}$ with weights $\{\tilde{q}(\tilde{y}_t \mid \tilde{\boldsymbol{\theta}}_{t+1}^{(j)}, \boldsymbol{\gamma})\tilde{\omega}_{t+1}^{(j)}\}$, leading to a new set of indices $\{j_k\}$. Third, we sample $\{\boldsymbol{\theta}_{t|T}^{(k)}\}$ from $N(\boldsymbol{\mu}_{t|T}^{(i_k,j_k)}, \boldsymbol{\Sigma}_{t|T}^{(i_k,j_k)})$, where the moments are computed as in Eqs. (2.3.5) and (2.3.6) using $\{\boldsymbol{\theta}_{t-1}^{(i_k)}\}$ and $\{\tilde{\boldsymbol{\theta}}_{t+1}^{(j_k)}\}$. Fourth, we compute weights for $\{\boldsymbol{\theta}_{t|T}^{(k)}\}$ as:

$$\omega_{t|T}^{(k)} \propto \frac{p\left(\boldsymbol{\theta}_{t|T}^{(k)} \mid \boldsymbol{\theta}_{t-1}^{(i_k)}, \boldsymbol{\gamma}\right) p\left(y_t \mid \boldsymbol{\theta}_{t|T}^{(k)}, \boldsymbol{\gamma}\right) p\left(\tilde{\boldsymbol{\theta}}_{t+1}^{(j_k)} \mid \boldsymbol{\theta}_{t|T}^{(k)}, \boldsymbol{\gamma}\right)}{\pi_{t+1}\left(\tilde{\boldsymbol{\theta}}_{t+1}^{(j_k)} \mid \boldsymbol{\gamma}\right) q(\tilde{y}_t \mid \boldsymbol{\theta}_{t-1}^{(i_k)}, \boldsymbol{\gamma})\tilde{q}(\tilde{y}_t \mid \tilde{\boldsymbol{\theta}}_{t+1}^{(j_k)}, \boldsymbol{\gamma})\bar{q}\left(\boldsymbol{\theta}_{t|T}^{(k)} \mid \boldsymbol{\theta}_{t-1}^{(i_k)}, \tilde{\boldsymbol{\theta}}_{t+1}^{(j_k)}, y_t, \boldsymbol{\gamma}\right)}.$$

Algorithm 3 summarizes the steps.

Algorithm 3: Two-filter Smoothing

Input: Forward particles $\{(\boldsymbol{\theta}_t^{(i)}, \omega_t^{(i)})\}_{i=1}^N$ and backward particles $\{(\tilde{\boldsymbol{\theta}}_t^{(j)}, \tilde{\omega}_t^{(j)})\}_{j=1}^N$, $t = 1, \ldots, T$

Output: Smoothed particles $\{(\boldsymbol{\theta}_{t|T}^{(k)}, \omega_{t|T}^{(k)})\}_{k=1}^N$, $t = 1, \ldots, T$

for $t = 1$ **to** T **do**

- Resample forward indices i_k using weights $q(\tilde{y}_t|\boldsymbol{\theta}_{t-1}^{(i)}, \boldsymbol{\gamma})\omega_{t-1}^{(i)}$;
- Resample backward indices j_k using weights $\tilde{q}(\tilde{y}_t|\tilde{\boldsymbol{\theta}}_{t+1}^{(j)}, \boldsymbol{\gamma})\tilde{\omega}_{t+1}^{(j)}$;
- **for** $k = 1$ **to** N **do**
  - Sample $\boldsymbol{\theta}_{t|T}^{(k)} \sim N(\boldsymbol{\mu}_{t|T}^{(k)}, \boldsymbol{\Sigma}_{t|T}^{(k)})$ using Eqs. (2.3.5) and (2.3.6);
- Calculate weights $\left\{\omega_{t|T}^{(k)}\right\}$ and resample $\{\boldsymbol{\theta}_{t|T}^{(k)}\}$.

The two-filter smoother provides an efficient $O(N)$ algorithm for computing smoothed estimates of the latent states. By running forward and backward passes independently, it enables parallelization while avoiding the particle degeneracy issues that plague standard forward-backward smoothers. For epidemiological applications, this means we can efficiently update our estimates of past reproduction numbers as new case data arrives.

We use SMC to obtain posterior inference for the state parameters conditional on the static parameters; however, in practice, we usually want to obtain joint inference for both static and state parameters. To achieve this, we present two additional approaches: HVA that combines SMC with variational optimization, and MCMC.

2.3.3. HVA

The HVA algorithm is an alternating scheme with two stages that combines the strengths of SMC for latent state inference with variational optimization for static parameter learning.

Static parameters in $\boldsymbol{\gamma}$ might be confined to a restricted parameter space. For example, variance parameters must be positive. For the HVA algorithm described below, we work with $\check{\boldsymbol{\gamma}}$, which represents the parameters mapped to the whole real line. We define the variational proposal distribution as $q_{\boldsymbol{\varsigma}}(\boldsymbol{\theta}_{1:T}, \check{\boldsymbol{\gamma}})$, where $\boldsymbol{\varsigma}$ denotes the parameters of the variational proposal distribution. This distribution serves as an approximation to the target posterior distribution, $p(\boldsymbol{\theta}_{1:T}, \check{\boldsymbol{\gamma}} \mid y_{1:T})$.

Optimization-based variational inference offers a scalable alternative to simulation-based methods. The goal is to minimize the KL divergence between the proposal and true posterior distributions. This is equivalent to maximizing the ELBO, which is given

by

$$\mathrm{E}\left\{\log p(\boldsymbol{\theta}_{1:T}, \check{\boldsymbol{\gamma}}, y_{1:T}) - \log q_{\boldsymbol{\varsigma}}(\boldsymbol{\theta}_{1:T}, \check{\boldsymbol{\gamma}})\right\}. \tag{2.3.7}$$

HVA proposal structure

While mean-field approximations assume independence among parameters, this assumption often fails for DGTF models where static parameters and latent states show strong dependencies. For example, the system variance parameter directly influences the smoothness of latent state trajectories, creating dependencies that mean-field approximations cannot capture. The hybrid approach of (Loaiza-Maya et al., 2022) addresses this limitation by preserving the exact conditional structure between parameters:

$$q_{\boldsymbol{\varsigma}}(\boldsymbol{\theta}_{1:T}, \check{\boldsymbol{\gamma}}) = p(\boldsymbol{\theta}_{1:T} \mid \check{\boldsymbol{\gamma}}, y_{1:T})\; q^0_{\boldsymbol{\varsigma}}(\check{\boldsymbol{\gamma}}). \tag{2.3.8}$$

This factorization respects the natural dependencies in the model. Rather than approximating the full joint distribution, we only approximate the marginal distribution of static parameters $q^0_{\boldsymbol{\varsigma}}(\check{\boldsymbol{\gamma}})$, while maintaining the exact conditional distribution of latent states given the static parameters.

Latent state marginalization

The key insight is that while $p(\boldsymbol{\theta}_{1:T} \mid \check{\boldsymbol{\gamma}}, y_{1:T})$ lacks a closed form, we can sample from it efficiently using the SMC two-filter smoother described above. This approach combines the computational efficiency of variational optimization for static parameters with the accuracy of Monte Carlo sampling for latent states, providing both full parallelization and linear computational complexity.

The choice of the particle number in the SMC requires careful consideration. Too few particles result in high variation in state trajectories and inflated variance parameter estimates, while too many particles increase computational cost unnecessarily. The optimal number also depends on the time series length. Our experiments indicate that 500 particles provides an effective balance between accuracy and speed for series of 200-500 observations.

Using Bayes theorem, this factorization enables us to marginalize out latent states $\boldsymbol{\theta}_{1:T}$ in the ELBO:

$$\mathcal{L}(\boldsymbol{\varsigma}) = \mathrm{E}\left\{\log p(\boldsymbol{\theta}_{1:T}, \check{\boldsymbol{\gamma}}, y_{1:T}) - \log q_{\boldsymbol{\varsigma}}(\boldsymbol{\theta}_{1:T}, \check{\boldsymbol{\gamma}})\right\} = \mathrm{E}\left\{\log p(\check{\boldsymbol{\gamma}}, y_{1:T}) - \log q^0_{\boldsymbol{\varsigma}}(\check{\boldsymbol{\gamma}})\right\}.$$

Thus, we only need to approximate the marginal posterior of static parameters rather than the full joint posterior, considerably simplifying the problem.

Static parameter optimization

Following the approach outlined in (Loaiza-Maya et al., 2022), we apply an element-wise transformation to $\check{\boldsymbol{\gamma}}$ to ensure that the distribution of the transformed variable $\widetilde{\boldsymbol{\gamma}}$ approximates normality. The proposal distribution is specified as $\widetilde{\boldsymbol{\gamma}} \sim N(\boldsymbol{\mu}_\varsigma, \mathbf{B}_\varsigma \mathbf{B}'_\varsigma + \mathbf{D}_\varsigma)$, where the variance-covariance matrix follows a latent factor structure. Here, $\boldsymbol{\mu}_\varsigma$, $\mathbf{B}_\varsigma$, and $\mathbf{D}_\varsigma$ are part of the unknown variational parameters $\boldsymbol{\varsigma}$ to be estimated. Specifically, $\mathbf{B}_\varsigma$ is an $m \times k$ matrix, where m is the number of static parameters and $k \leq m$ is the number of factors, while $\mathbf{D}_\varsigma$ is a diagonal matrix. This factor structure captures parameter correlations efficiently without requiring estimation of a full covariance matrix. The resulting distribution $q^0_{\boldsymbol{\varsigma}}(\check{\boldsymbol{\gamma}})$ follows a Gaussian copula, which flexibly accommodates different marginal distributions for each element of $\check{\boldsymbol{\gamma}}$.

(Loaiza-Maya et al., 2022) shows that $\mathcal{L}(\boldsymbol{\varsigma})$ can be maximized via gradient ascent:

$$\boldsymbol{\varsigma}^{(i+1)} = \boldsymbol{\varsigma}^{(i)} + \varrho^{(i)} \cdot \nabla_{\boldsymbol{\varsigma}} \widehat{\mathcal{L}} \left(\boldsymbol{\varsigma}^{(i)} \right), \tag{2.3.9}$$

$$\nabla_{\boldsymbol{\varsigma}} \widehat{\mathcal{L}} \left(\boldsymbol{\varsigma}^{(i)} \right) = \frac{\partial \check{\boldsymbol{\gamma}}'}{\partial \boldsymbol{\varsigma}} \left[\nabla_{\check{\boldsymbol{\gamma}}} \log p(\boldsymbol{\theta}_{1:T}, \check{\boldsymbol{\gamma}}, y_{1:T}) - \nabla_{\check{\boldsymbol{\gamma}}} \log q^0_{\boldsymbol{\varsigma}}(\check{\boldsymbol{\gamma}}) \right]. \tag{2.3.10}$$

The step size $\varrho^{(i)}$ is updated recursively using the ADADELTA method controlled by a user-specified learning rate. The terms $\partial \check{\boldsymbol{\gamma}}' / \partial \boldsymbol{\varsigma}$ and $\nabla_{\check{\boldsymbol{\gamma}}} \log q^0_{\boldsymbol{\varsigma}}(\check{\boldsymbol{\gamma}})$ have generic closed form expressions given by (Loaiza-Maya et al., 2022). For a specific model, the remaining task is to calculate the gradient of the log joint probability density, $\nabla_{\check{\boldsymbol{\gamma}}} \log p(\boldsymbol{\theta}_{1:T}, \check{\boldsymbol{\gamma}}, y_{1:T})$, which varies depending on the model structure. We provide examples of this calculation in the simulation section, with detailed derivations of the gradients for the discrete-time Hawkes-type model in § A.1.2.

Algorithm summary

Algorithm 4 summarizes the HVA procedure. Each iteration consists of two steps: first, we draw samples from our current static parameter approximation and use SMC to sample corresponding latent states, yielding an unbiased gradient estimate; second, we improve our approximation via gradient ascent. This approach uses Monte Carlo integration to marginalize out latent states at each iteration, distinguishing it from two-stage optimization schemes.

The algorithm typically runs for a predetermined number of iterations or until convergence criteria are met, such as when the ELBO stabilizes. However, the intractability of $p(\boldsymbol{\theta}_{1:T} \mid \check{\boldsymbol{\gamma}}, y_{1:T})$ in our proposal distribution (Eq. (2.3.8)) prevents direct ELBO computation. Instead, we monitor alternative metrics. (Loaiza-Maya et al., 2022) tracked the accuracy of the one-step-ahead posterior predictive. In our simulation examples, we

Algorithm 4: Hybrid Variational Approximation (HVA)

Input: $y_{1:T}$, initial $\boldsymbol{\varsigma}^{(0)}$, initial step size $\varrho^{(0)}$, learning rate ϵ, N iterations, P particles for SMC, M posterior samples

Output: Optimized $\boldsymbol{\varsigma}^*$, posterior samples $\{\widetilde{\boldsymbol{\gamma}}^{(j)}, \boldsymbol{\theta}_{1:T}^{(j)}\}_{j=1}^{M}$

Initialize: $\boldsymbol{\varsigma}^{(1)} = (\boldsymbol{\mu}_\varsigma, \mathbf{B}_\varsigma, \mathbf{D}_\varsigma)$, $\varrho^{(0)}$

`// Gradient ascent optimization`

for $i = 1$ **to** N **do**

- Sample $\widetilde{\boldsymbol{\gamma}}^{(i)} \sim q^0_{\boldsymbol{\varsigma}^{(i)}}$ (Gaussian copula);
- Run two-filter smoother (Algorithm 3) with P particles given $\widetilde{\boldsymbol{\gamma}}^{(i)}$;
- Compute $\boldsymbol{\theta}_{1:T}^{(i)} = \frac{1}{P}\sum_{p=1}^{P} \boldsymbol{\theta}_{1:T}^{(i,p)}$ (average over particles);
- Calculate the gradients $\nabla_{\boldsymbol{\gamma}} \log p(\boldsymbol{\theta}_{1:T}^{(i)}, \widetilde{\boldsymbol{\gamma}}^{(i)}, y_{1:T})$ and $\nabla_{\check{\boldsymbol{\gamma}}} \log q^0_{\boldsymbol{\varsigma}}(\check{\boldsymbol{\gamma}})$;
- Compute $\nabla_{\boldsymbol{\varsigma}} \widehat{\mathcal{L}}\left(\boldsymbol{\varsigma}^{(i)}\right)$ and update $\boldsymbol{\varsigma}^{(i+1)}$ via gradient ascent as in Eqs. (2.3.9) and (2.3.10);
- Update step size $\varrho^{(i)}$ via ADADELTA method;

`// Generate posterior samples`

for $j = 1$ **to** M **do**

- Sample $\widetilde{\boldsymbol{\gamma}}^{(j)} \sim q^0_{\boldsymbol{\varsigma}^*}$ using converged parameters $\boldsymbol{\varsigma}^* = \boldsymbol{\varsigma}^{(N)}$;
- Run two-filter smoother with P particles given $\widetilde{\boldsymbol{\gamma}}^{(j)}$;
- Compute $\boldsymbol{\theta}_{1:T}^{(j)} = \frac{1}{P}\sum_{p=1}^{P} \boldsymbol{\theta}_{1:T}^{(j,p)}$ (average over particles);

monitor changes in the conditional marginal likelihood $p(y_{1:T} \mid \check{\boldsymbol{\gamma}})$ as our convergence criterion, which typically stabilizes within 1,000 iterations.

2.3.4. MCMC benchmark

We now describe the MCMC algorithm for DGTF models, which serves as the gold standard for uncertainty quantification. Our implementation targets models with univariate latent states, specifically discrete-time Hawkes-type and distributed lag models, which suffice for our empirical comparisons between HVA and MCMC. While models with multivariate latent states (such as time-varying ARs) require custom state samplers,

they share the same generic static parameter updates through gradient-based methods.

Nevertheless, we include MCMC here for three reasons: it offers the most accurate posterior uncertainty estimates when computational resources permit; it enables validation of faster approximate methods like HVA, and practitioners may prefer it for final analyses where precise uncertainty quantification is needed.

General MCMC Framework for DGTF Models

The MCMC approach for DGTF models follows a general template that alternates between updating latent states and static parameters. The strategy combines model-specific and generic components: an update of the latent states $\{\boldsymbol{\theta}_t\}$ given current static parameters $\boldsymbol{\gamma}$ (model-specific), and an update of the static parameters $\boldsymbol{\gamma}$ given current latent states $\boldsymbol{\theta}_t$ (generic).

The key insight that makes MCMC applicable for DGTF models is that we can exploit the Markovian structure of the latent states. Rather than updating $\boldsymbol{\theta}_t$ directly, which would be computationally challenging due to their high correlation, we reparameterize in terms of the driving noise process w_t.

Model-Specific Latent State Updates

The latent state update requires model-specific derivation due to the particular structure of each DGTF variant. Below we illustrate a unified approach for discrete-time Hawkes-type and distributed lag models, but practitioners need to derive appropriate samplers for other specific model choices.

Discrete-time Hawkes-type and distributed lag models share a common intensity structure that enables a unified treatment. In both models, the intensity function can be

expressed as:

$$\eta_t = \zeta(\lambda_t) = \sum_{l=1}^{L} \phi_l h(\psi_{t+1-l}) y_{t-l}, \quad \psi_t = \psi_{t-1} + w_t.$$

The key difference is that $L = t$ for the infinite distributed lag model, while it fixed at a given L for the discrete-time Hawkes-type model.

For both models, $h(\psi_t)$ is required to be a non-negative smooth function that is at least first-order differentiable. When h is non-linear, we use a linearization strategy that enables efficient sampling:

$$h(\psi_t) \approx h(\psi_t^*) + h'(\psi_t^*)(\psi_t - \psi_t^*) = h_0(\psi_t^*) + h'(\psi_t^*)\psi_t, \tag{2.3.11}$$

where $h_0(\psi_t^*) = h(\psi_t^*) - h'(\psi_t^*)\psi_t^*$ and ψ_t^* is a reference value whose precise form during the MH sweep is specified below.

The central challenge in MCMC for state-space models is the high correlation among latent states ψ_t. Direct MH updates would suffer from poor mixing. Following (Gamerman, 1998), we overcome this by reparameterizing the problem in terms of the independent noise terms.

For simplicity, we assume that there is no regression term $\mathbf{x}_t'\boldsymbol{\varphi}$ in η_t and that $w_t \sim N(0, W)$, although the approach can be extended to more general cases where a regression term exists, w_t has a time-varying variance or follows a non-Gaussian distribution. The key idea is to represent the Markov random walk as a summation of independent white noise terms, $\psi_t = \sum_{j=1}^{t} w_j$, and sample $\{w_t\}$ directly using MH.

This reparameterization transforms a highly correlated sampling problem into one with independent components, dramatically improving mixing. To construct tractable proposal distributions for each w_t, we linearize $\eta_t = \zeta(\lambda_t)$ as a function of $w_1, \ldots, w_t$

using the approximation in Eq. (2.3.11), leading to

$$\eta_t \approx \eta_t^* = \sum_{l=1}^{L} \phi_l y_{t-l} h_0(\psi^*_{t+1-l}) + \sum_{j=1}^{t} w_j \sum_{l=1}^{t+1-j} \phi_l y_{t-l} h'(\psi^*_{t+1-l}).$$

This linearization yields a lower-triangular structure in the dependence of η_t^* on $\{w_j : j \leq t\}$. For each time point $t = 1, \ldots, T$:

$$\eta_1^* = \phi_1 y_0 h_0(\psi_1^*) + w_1 \phi_1 y_0 h'(\psi_1^*),$$
$$\eta_2^* = \sum_{l=1}^{2} \phi_l y_{2-l} h_0(\psi^*_{3-l}) + w_1 \sum_{l=1}^{2} \phi_l y_{2-l} h'(\psi^*_{3-l}) + w_2 \phi_1 y_1 h'(\psi_2^*),$$

$\ldots$

$$\eta_T^* = \sum_{l=1}^{\min(L,T)} \phi_l y_{T-l} h_0(T+1-l) + w_1 \sum_{l=1}^{\min(L,T)} \phi_l y_{T-l} h'(\psi_{T+1-l}) + \cdots + w_T \phi_1 y_{T-1} h'(\psi_T^*).$$

To express this system compactly, we define $\boldsymbol{\eta}^* = (\eta_1^*, \ldots, \eta_T^*)'$, $\mathbf{w} = (w_1, \ldots, w_T)$, and

$$\boldsymbol{\eta}_0^* = \left(\phi_1 y_0 h_0(\psi_1^*), \ldots, \sum_{l=1}^{\min(L,T)} \phi_l y_{T-l} h_0(\psi^*_{T+1-l}) \right)',$$

$$\mathbf{F}^* = \begin{pmatrix} \sum_{l=1}^{1} \phi_l y_{1-l} h'(\psi^*_{2-l}) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ \sum_{l=1}^{\min(L,T)} \phi_l y_{T-l} h'(\psi^*_{T+1-l}) & \cdots & \sum_{l=1}^{1} \phi_l y_{T-l} h'(\psi^*_{T+1-l}) \end{pmatrix}.$$

The system then has a compact matrix representation:

$$\boldsymbol{\eta}^* = \boldsymbol{\eta}_0^* + \mathbf{F}^* \mathbf{w}.$$

The linearized system enables us to construct an approximated normal likelihood that serves as the basis for our proposal distributions. Define $\boldsymbol{F}^*_{t,\cdot}$ as the row vector representing the tth row of matrix $\mathbf{F}^*$. Let $F^*_{t,j}$ denote an element in the tth row and jth column of

$\mathbf{F}$, and let $\eta^*_{0,t}$ represent the tth element of vector $\boldsymbol{\eta}^*_0$. Using this notation, we construct an approximated normal likelihood:

$$\tilde{y}_t \overset{a}{\sim} N(\eta^*_t, V^*_t),\ \eta^*_t = \eta^*_{0,t} + \boldsymbol{F}^*_{t,\cdot}\mathbf{w},\ \mathbf{w} \sim N(\mathbf{0}, W\mathbf{I}_T).$$

Here $V^*_t = \mathrm{Var}(y_t)[\zeta'(y_t)]^2$. The variance depends on the observation model: for Poisson $\mathrm{Var}(y_t) = \lambda^*_t$; for negative-binomial, $\mathrm{Var}(y_t) = \lambda^*_t(\lambda^*_t + \rho)/\rho$, where $\lambda^*_t = \zeta^{-1}(\eta^*_t)$.

The lower-triangular structure of $\mathbf{F}^*$ implies that η^*_t depends only on $w_1, \ldots, w_t$. Since $\eta^*_t = \eta^*_{0,t} + \sum_{k=1}^{t} F^*_{t,k} w_k$, any η^*_t with $t \geq j$ depends on w_j. This yields the conditional posterior will be used to generate the proposal:

$$p(w_j \mid \cdot) \propto p(w_j \mid W)\Pi_{t=j}^{T} p(\tilde{y}_t \mid \eta^*_t, V^*_t).$$

When constructing the proposal for a given w_j, we compute the linearization point ψ^* with w_j set to zero, so that $\psi^*_s = \sum_{k\neq j,\, k\leq s} w_k$ for $s \geq j$, using the most recent values of all other components. This ensures that $\mathbf{F}^*$, V^*_t, and the resulting proposal variance B_j depend only on $\{w_k : k \neq j\}$, which are held fixed during the update of w_j. If instead w_j were retained in the linearization, B_j would depend on the current value of w_j, making the random-walk proposal asymmetric and requiring a second linearization at the proposed state to correct the acceptance ratio, roughly doubling the computational cost per MH step. Because B_j depends on w_j only indirectly through $h'(\psi^*_s)$, excluding w_j has a negligible effect on proposal quality.

With this specification and the conjugacy introduced by the linearization, for $j =$

$2, \ldots, T$, the full conditional of w_j follows a normal distribution $N(b_j, B_j)$ with

$$B_j = \left[\frac{1}{W} + \sum_{t=j}^{T} \frac{(F^*_{t,j})^2}{V^*_t} \right]^{-1} \quad \text{and} \quad b_j = B_j \sum_{t=j}^{T} \frac{F^*_{t,j} \left(\tilde{y}_t - \eta^*_{0,t} - \sum_{k \neq j} F^*_{t,k} w_k \right)}{V^*_t}.$$

Following (Gamerman, 1998), we assume w_1 follows a normal prior $N(b_0, B_0)$. The conditional posterior for w_1 also follows a normal distribution $N(b_1, B_1)$ with

$$B_1 = \left[\frac{1}{B_0} + \sum_{t=1}^{T} \frac{(F^*_{t,1})^2}{V^*_t} \right]^{-1} \quad \text{and} \quad b_1 = B_1 \left[\frac{b_0}{B_0} + \sum_{t=1}^{T} \frac{F^*_{t,1} \left(\tilde{y}_t - \eta^*_{0,t} - \sum_{k \neq 1} F^*_{t,k} w_k \right)}{V^*_t} \right].$$

In (Gamerman, 1998), $N(b_j, B_j)$ is used directly as an independence proposal. We instead adopt a random-walk MH proposal that uses only the variance B_j to set the step size,

$$w_j^{(i)} \sim N(w_j^{(i-1)}, B_j), \tag{2.3.12}$$

where i indicates the ith MCMC sample. Because B_j depends only on $\{w_k : k \neq j\}$ (held fixed during the update of w_j), the proposal is symmetric, $q(w_j^{(i)} \mid w_j^{(i-1)}) = q(w_j^{(i-1)} \mid w_j^{(i)})$. The Hastings correction therefore cancels, and the acceptance ratio reduces to the posterior density ratio

$$r = \min \left\{ 1, \frac{p(w_j^{(i)} \mid W) \Pi_{t=j}^{T} p(y_t \mid \eta_t^{(i)}, \rho)}{p(w_j^{(i-1)} \mid W) \Pi_{t=j}^{T} p(y_t \mid \eta_t^{(i-1)}, \rho)} \right\}. \tag{2.3.13}$$

Generic Static Parameter Updates

The second component of our MCMC algorithm updates the static parameters $\boldsymbol{\gamma}$ given the current latent states. Unlike the model-specific latent state updates, we use a generic approach using Hamiltonian Monte Carlo (HMC) within MH steps that works for any

DGTF variant.

HMC brings concepts from physics into the sampling process to enhance efficiency. The key advantage for DGTF models is that HMC can efficiently explore high-dimensional parameter spaces and handle the complex posterior geometries that arise from non-linear transfer functions. We offer only a brief introduction here; (Neal, 2011) and (Betancourt, 2018) provide comprehensive resources for those seeking deeper insights.

We first transform $\boldsymbol{\gamma}$ to $\check{\boldsymbol{\gamma}}$ so that each component spans the entire real line. These transformed parameters become position variables in a physical system, while auxiliary momentum variables $\mathfrak{m}$ are drawn from a standard multivariate Gaussian distribution. The total energy of this system is captured by a Hamiltonian function:

$$H(\mathfrak{m}, \check{\boldsymbol{\gamma}}) = -\log p(\check{\boldsymbol{\gamma}}|y_{1:T}, w_{1:T}) + \frac{1}{2}\mathfrak{m}'\mathfrak{m}.$$

To generate proposals, we simulate the Hamiltonian dynamics for $\mathfrak{L}$ steps using the leapfrog integrator with step size e. These tuning parameters $\mathfrak{L}$ and e require careful calibration for good performance.

After completing the trajectory simulation, we negate the final momentum variables to ensure reversibility, resulting in a proposed state $(\mathfrak{m}^*, \check{\boldsymbol{\gamma}}^*)$. This proposal is accepted with probability:

$$\min\left\{1, \exp(-H(\mathfrak{m}^*, \check{\boldsymbol{\gamma}}^*) + H(\mathfrak{m}, \check{\boldsymbol{\gamma}}))\right\}.$$

A key component of HMC is calculating the gradient of the log posterior with respect to $\check{\boldsymbol{\gamma}}$. These gradients are identical to those required for HVA, enabling code reuse between inference methods and making this component generic across all DGTF variants.

Algorithm Summary

The complete MCMC sampling procedure is summarized in Algorithm 5. It combines model-specific latent state updates with generic HMC for static parameters. While latent state samplers need to be derived for each DGTF variant, the HMC component remains unchanged, requiring only gradient computations. Practitioners can either derive analytical expressions for their specific model or use automatic differentiation tools for convenience.

Algorithm 5: MCMC for DGTF Models

Input: $y_{1:T}$, priors, HMC parameters $(\mathfrak{L}, e)$
Output: Posterior samples $\{w_1^{(i)}, \ldots, w_T^{(i)}, \boldsymbol{\gamma}^{(i)}\}_{i=1}^N$
Initialize: $w_t^{(0)}$, $\boldsymbol{\gamma}^{(0)}$
for $i = 1$ **to** N **do**
  `// Update latent states`
  for $t = 1$ **to** T **do**
    Sample $w_t^{(i)} \sim N(w_t^{(i-1)}, B_t^{(i-1)})$ and accept with probability r given by Eq. (2.3.13);
  `// Update` $\boldsymbol{\gamma}$ `(HMC)`
  Run leapfrog for $\mathfrak{L}$ steps with step size e and accept with Hamiltonian criterion.

§ 2.4. Simulation studies

We designed a comprehensive simulation study with two primary objectives: (1) to compare the computational and statistical performance of our proposed inference methods (HVA versus MCMC) when the model is correctly specified, and (2) to evaluate model selection capabilities by assessing how different DGTF model structures perform when fitted to data generated from a known true model. This dual approach allows us to

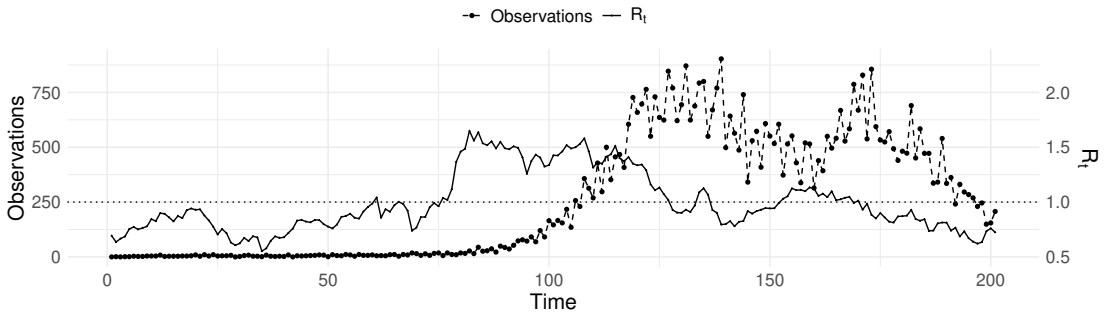


Figure 2.2.: Simulated data from a discrete-time Hawkes-type model along with the corresponding R_t.

separately evaluate inference methodology and model flexibility within our framework.

2.4.1. Simulation design

We generated data from a discrete-time Hawkes-type model ($\mathcal{M}_0$) designed to mimic realistic epidemic dynamics. The lag distribution followed a discretized log-normal with parameters $\mu = 1.39$ and $\sigma^2 = 0.32$, truncated at $L = 15$ lags so that $1 - \sum_l \phi_l < 0.01$. Observations were drawn from a negative-binomial distribution with dispersion parameter $\rho = 30$ to capture the overdispersion typical of epidemiological count data. We specified a constant baseline intensity with $\varphi = 1$ and $x_t = 1$, and used the softplus gain function $h(\psi_t) = \log(\exp(\psi_t) + 1)$ to ensure positivity. The latent process evolved as a random walk $\psi_t \sim N(\psi_{t-1}, W)$ with variance $W = 0.01$, allowing for gradual changes in transmission intensity. Figure 2.2 shows the resulting time series y_t and reproduction numbers $R_t = h(\psi_t)$. Additional details are provided in § A.1.1.

2.4.2. Model comparison setup

To evaluate the flexibility of our DGTF framework and understand the effects of model misspecification, we fitted multiple DGTF variants to data simulated from model $\mathcal{M}_0$. More specifically, we compared five model structures: $\mathcal{M}_0$ (true model): the true discrete-time Hawkes-type model with unknown parameters $(\rho, \varphi, \mu, \sigma^2, W)$; $\mathcal{M}_1$ (discounted Hawkes): a discrete-time Hawkes-type model with unknown time-varying covariance $\mathbf{W}_t$ specified using a discount factor $\delta \in (0, 1]$ (Prado et al., 2021); $\mathcal{M}_2$ (distributed lag): a distributed lag model with $\psi_t \sim N(\psi_{t-1}, W)$, with W unknown; $\mathcal{M}_3$ (NB-AR): a negative-binomial AR model with time-varying coefficients evolving according to a random walk evolution equation with covariance matrix $W\mathbf{I}$ and W unknown; $\mathcal{M}_4$ (Poisson TV-AR): a Poisson AR with time-varying coefficients evolving according to a random walk with covariance matrix $W\mathbf{I}$ and W unknown.

All models used weakly informative priors to ensure comparability: $\rho \sim IG(1, 1)$, $W \sim IG(1, 1)$, $\sigma^2 \sim IG(1, 1)$, $\mu \sim N(0, 1)$, $\phi \sim N(0, 10)\mathbf{1}(\phi \geq 0)$ and $\kappa \sim Beta(1, 1)$. Model-specific hyperparameters δ, r and p were tuned by grid search to minimize χ^2 discrepancies. Optimized values are provided in § 2.4.3 and Table 2.3. A comprehensive prior sensitivity analysis (detailed in § A.1.6) confirmed that different weakly informative priors had minimal impact on posterior inference, with all parameters concentrating around their true values across different prior specifications.

2.4.3. Implementation and metrics

HVA configuration. Unlike (Loaiza-Maya et al., 2022), who used MCMC to sample latent states with HVA, we opted for a SMC two-filter smoother as it can be easily

parallelized. We selected the number of particles for the SMC two-filter smoother through a comprehensive sensitivity analysis. For model $\mathcal{M}_0$, tests with 50 to 1,000 particles (detailed results in § A.1.5) showed that 500 particles achieved the best trade-off between computational cost and statistical accuracy: the χ^2 discrepancy stabilized at 1.19 for $N \geq 500$, with less than 1% improvement when doubling to 1,000 particles, while runtime increased linearly with N. We initialized the artificial marginal density with $\boldsymbol{\mu}_0^* = \mathbf{0}$ and $\boldsymbol{\Sigma}_0^* = \mathbf{I}$, and set the resampling ratio m to 0.5 (see Algorithm 1). Similar results were obtained for all other models, therefore, we used 500 particles throughout.

For static parameter estimation, HVA used gradient ascent to maximize the ELBO, with an initial step size of 10^{-5} and an ADADELTA learning rate of 0.02 for all models. To ensure a fair comparison across models, we selected model-specific hyperparameters through a grid search, choosing values that minimized the χ^2 discrepancy across specified parameter ranges. The rank parameter k of $\mathbf{B}_\varsigma$ for each model's variational proposal distribution was set to match the number of unknown parameters in the model, yielding to $k = 5$ for $\mathcal{M}_0$, $k = 4$ for $\mathcal{M}_1$ and $\mathcal{M}_2$, and $k = 3$ for $\mathcal{M}_3$ and $\mathcal{M}_4$.

Additionally, optimal values of hyperparameters for each model were set as follows. For model $\mathcal{M}_1$ the optimal discount factor was $\hat{\delta} = 0.9$ chosen from $\delta \in [0.87, 0.95]$. For model $\mathcal{M}_2$ the optimal lag order was $\hat{r} = 4$ chosen from $r \in \{2, \ldots, 7\}$. For $\mathcal{M}_3$ and $\mathcal{M}_4$ the optimal autoregressive orders were $\hat{p} = 3$ and $\hat{p} = 5$, respectively, chosen from $p \in \{1, \ldots, 6\}$.

The HVA algorithm ran for 1,000 iterations for all models, with convergence verified through the trace plots of $\log p(y_{1:T}|\boldsymbol{\gamma})$ (see Fig. A.6 for model $\mathcal{M}_0$; similar results were obtained for all models). After convergence, 5,000 posterior samples of $\boldsymbol{\gamma}$ and $\boldsymbol{\theta}_{1:T}$ were drawn.

MCMC configuration. To allow a direct comparison with HVA, we implemented MCMC for the true model $\mathcal{M}_0$. The implementation used HMC with a leapfrog integrator of 50 steps and step size of 0.005. The sampling run consisted of 12,000 iterations: the first 2,000 were discarded as burn-in, and the remaining 10,000 were thinned by a factor of 2, yielding 5,000 posterior samples.

Convergence diagnostics, including trace plots, posterior density plots, autocorrelation functions, and Gelman-Rubin statistics, confirmed satisfactory mixing (see § A.1.4).

Performance metrics. We assessed model performance through multiple metrics. The primary measure of posterior predictive fit was the averaged χ^2 discrepancy (Gelman et al., 1996), defined as:

$$X^2(y_{1:T}; \boldsymbol{\gamma}) = \frac{1}{NT} \sum_{t=1}^{T} \sum_{i=1}^{N} \frac{\left(y_t - \lambda_t^{(i)}\right)^2}{V_t^{(i)}}.$$

Here, $\lambda_t^{(i)}$ represents the conditional mean computed using parameter estimates from the ith posterior sample, drawn via HVA (post-convergence) or MCMC. The variance $V_t^{(i)}$ accounts for the observation model: $V_t^{(i)} = \lambda_t^{(i)}(1 + \lambda_t^{(i)}/\rho^{(i)})$ for negative-binomial models and $V_t^{(i)} = \lambda_t^{(i)}$ for Poisson models. This metric captures both bias and variance in the predictive distribution.

We also computed the continuous ranked probability score (CRPS), which provides a proper scoring rule for probabilistic forecasts. CRPS measures the integrated squared difference between the empirical distribution function of the predictive distribution and the observed value, offering a more nuanced assessment of forecast quality than point estimates alone.

For latent state estimation, we computed the mean absolute error (MAE) between

posterior median estimates of R_t and true values, providing a direct measure of estimation accuracy.

Finally, to evaluate uncertainty quantification, we computed empirical 95% coverage rates and average interval widths for both the latent reproduction numbers R_t and the observed counts y_t, using the interval appropriate to each. For a latent quantity such as R_t, coverage is measured against the posterior credible interval,

$$\frac{1}{T}\sum_{t=1}^{T}\mathbf{1}\left\{R_t^{\text{true}} \in \text{CI}_{0.95}(R_t \mid y_{1:T})\right\},$$

i.e., it corresponds to the fraction of time points at which the truth falls within the central 95% posterior interval of R_t. For the observed counts the relevant interval is the posterior predictive interval, not a credible interval, and coverage is given by

$$\frac{1}{T}\sum_{t=1}^{T}\mathbf{1}\left\{y_t^{\text{obs}} \in \text{PI}_{0.95}(y_t^{\text{rep}} \mid y_{1:T})\right\},$$

the fraction of time points at which the observed count falls within the central 95% interval of the posterior predictive distribution of a replicated count y_t^{rep}. The two quantities measure different things. The first assesses whether the posterior of a latent quantity is calibrated against the truth, whereas the second assesses the calibration of replicated observations. Because the study uses a single simulated series, both are averages over time within that series rather than frequentist coverages across independent replicate datasets.

2.4.4. Results: HVA versus MCMC

We first compare the performance of the HVA with the gold-standard MCMC under the correctly specified model $\mathcal{M}_0$. This benchmark comparison establishes the accuracy

and reliability of HVA prior to analyzing more complex model selection scenarios.

Parameter estimation. Figure 2.3 shows that HVA and MCMC produced similar posterior distributions for most parameters. Both methods accurately recovered ρ, φ, and μ. Minor differences emerged for the variance parameters: MCMC slightly overestimated σ^2 while HVA slightly overestimated W. This bias in the HVA's estimate of W reflects the limited particle count, inflating the inferred system variance. Overall, both methods yielded very similar posterior means and credible intervals.

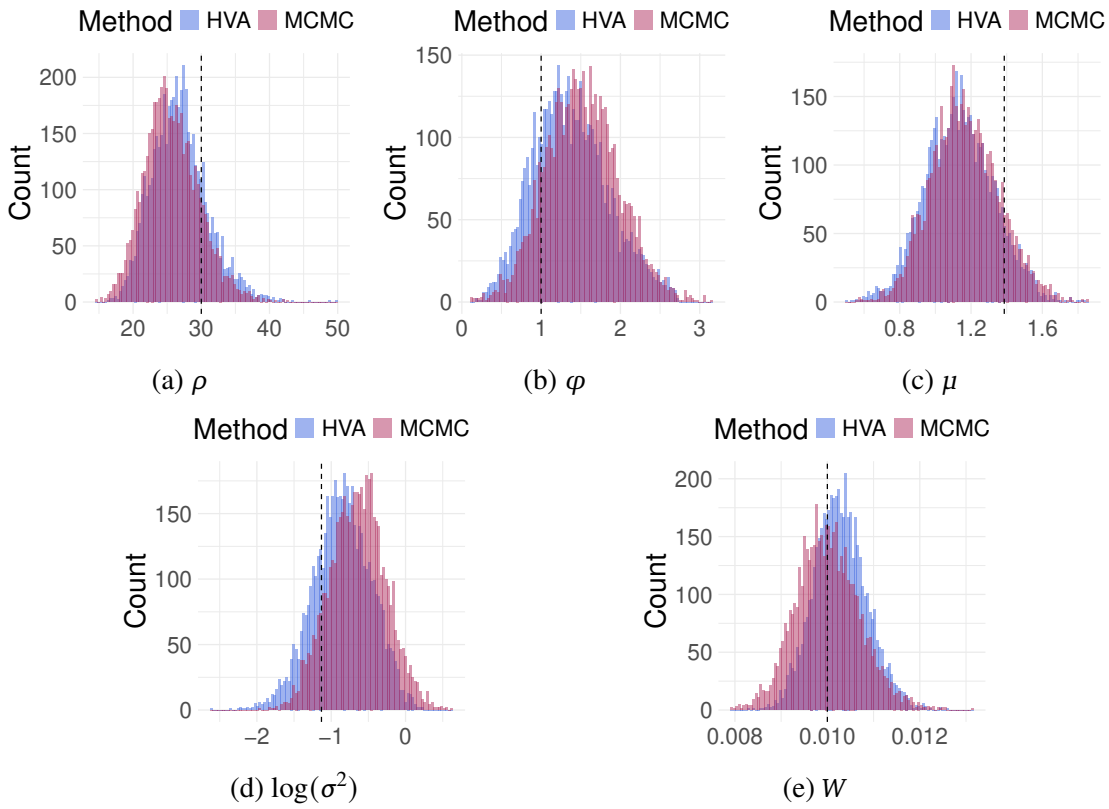


Figure 2.3.: Posterior distributions of ρ, φ, μ, σ^2 and W estimated by HVA and MCMC. True values are marked by black dashed lines.

Predictive performance. Table 2.2 presents key performance metrics. Both algorithms achieved nearly identical χ^2 discrepancies (1.19 for HVA and 1.18 for MCMC at convergence) and MAE values (0.12), confirming comparable accuracy in point estimation. The CRPS values further support this conclusion, with 27.16 for HVA and 27.1 for MCMC at their respective convergence points, confirming equivalent probabilistic forecast quality.

HVA produced narrower 95% credible intervals for R_t, leading to slightly lower coverage (≈ 0.87 vs. 0.97 for MCMC), a typical characteristic of variational approximations, which trade uncertainty accuracy for computational efficiency. This pattern persisted in out-of-sample forecasting. Figure 2.5 shows that while both methods produced similar posterior medians for one-step-ahead and five-steps-ahead forecasts ($t = 140, \ldots, 185$), HVA consistently showed less uncertainty in its predictive distributions. As expected, forecast uncertainty increased with longer prediction horizons for both methods, however, HVA's credible intervals remained consistently narrower.

The practical significance of this underestimation depends on the inferential goal. In our simulations, HVA's 95% credible intervals for R_t achieved approximately 87% empirical coverage compared with 97% for MCMC, roughly a 10 percentage gap. For the observed counts y_t, both methods achieved comparable coverage, suggesting that the variance underestimation is most consequential for latent state inference. Since both methods yield nearly identical χ^2 discrepancies and CRPS values, this coverage loss is unlikely to affect model selection or point estimation. The trade-off is therefore most relevant when calibrated interval estimates for R_t are needed for policy decisions. In that setting the HVA should not be read as a replacement for MCMC, which remains the reference for calibrated intervals; the HVA is best reserved for fast point estimation,

forecasting, and model comparison, where the two methods agree.

Several principled approaches exist for correcting variational posterior variances. (Giordano et al., 2018) developed a linear response method that recovers improved covariance estimates from mean-field variational posteriors by computing corrections based on the Hessian of the variational objective at the optimum. (Syring and Martin, 2019) proposed a general posterior calibration procedure that adjusts the posterior spread through a tempering parameter, tuned via simulation so that credible regions achieve nominal frequentist coverage. More recently, (Huggins et al., 2020) introduced post-hoc diagnostic bounds that can identify when variational credible intervals require correction. Although we do not pursue these corrections here, they offer promising directions for settings where both computational speed and calibrated uncertainty are needed. A simpler alternative is to use HVA as a warm-start mechanism to initialize a short MCMC chain, bypassing the costly burn-in phase while retaining the asymptotic exactness of MCMC for final calibration.

Computational efficiency and trade-offs. The primary computational advantage of HVA is computational efficiency. As shown in Table 2.2, HVA converged after 1,000 iterations versus 2,000 for MCMC, which is a 50% reduction in required iterations. This translated to a total runtime of 100 s for HVA versus 287 s for MCMC, approximately 65% faster overall. When examining runtime per iteration, HVA required an average of 0.1 s per iteration compared to MCMC with an average of 0.144 s per iteration, demonstrating that HVA is also faster per iteration, by approximately 31%.

Given the comparable predictive accuracy shown in Table 2.2 (differences $< 1\%$ across all point estimation metrics), this computational efficiency represents a favorable

Table 2.2.: Performance comparison of HVA and MCMC across iterations. HVA converges in 1,000 iterations while MCMC requires 2,000, resulting in 65% reduction in total runtime. Computational time reported as mean±SD in seconds across five independent runs. These results are for model $\mathcal{M}_0$ in the simulation study.

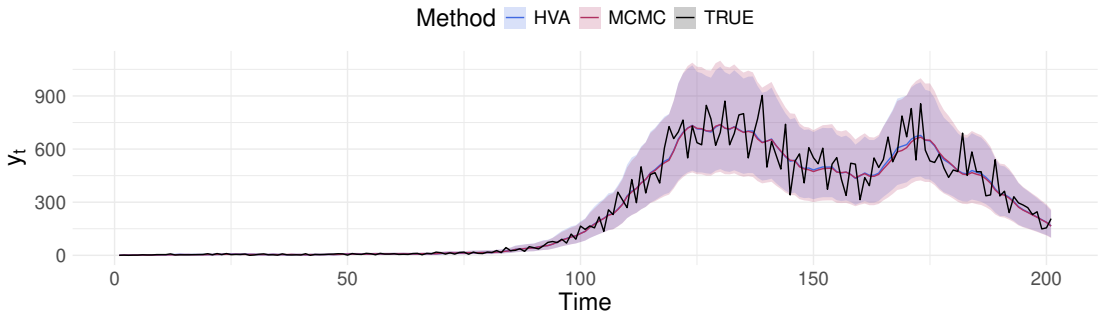
				y_t Metrics		R_t Metrics	
Method	Iter	Runtime (s)	χ^2	CRPS	Coverage	MAE	Coverage
HVA-500	100	12.25±0.31	1.94	29.62	0.97	0.19	1.00
	500	53.34±1.52	1.23	28.16	0.95	0.13	0.85
	1000	**99.95±1.05**	**1.19**	**27.16**	**0.96**	**0.12**	**0.87**
	2000	196.74±1.24	1.19	27.08	0.95	0.12	0.86
MCMC	500	73.35±2.13	1.31	28.32	0.98	0.14	0.91
	1000	141.31±2.39	1.22	27.58	0.97	0.14	0.94
	2000	**287.14±4.1**	**1.18**	**27.1**	**0.96**	**0.12**	**0.97**
	5000	730.64±9.91	1.19	27.1	0.96	0.12	0.98

trade-off for many applications. HVA allows more frequent model updates, which is useful for real-time epidemic surveillance and sensitivity analyses. However, MCMC is valuable for final analyses requiring full uncertainty quantification. HVA enables almost three times as many data analyses at the same computational cost. This gain increases when scaling to larger datasets or multiple locations.

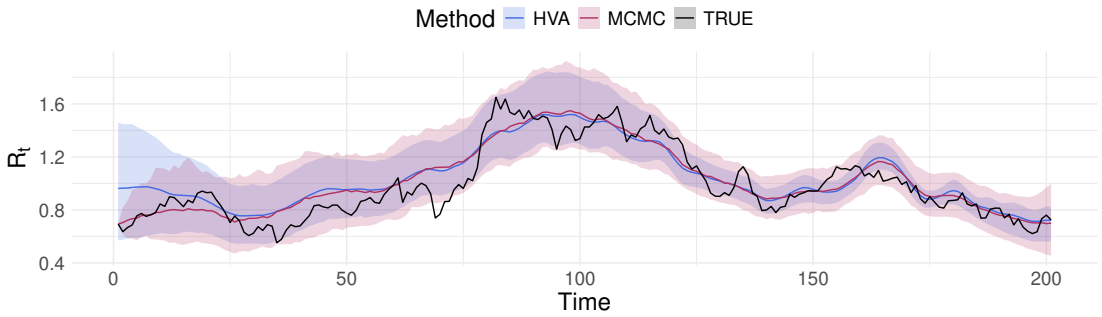
2.4.5. Results: model comparison

The model comparison presented here addresses a practical question: when the true data generating process is unknown, how do different DGTF model specifications perform? By fitting multiple models to data from a known source, we assess how closely each model approximates the data generating process, thereby illustrating how the unified DGTF framework can guide principled model selection.

For inference, we used HVA exclusively for all models. This choice ensures method-



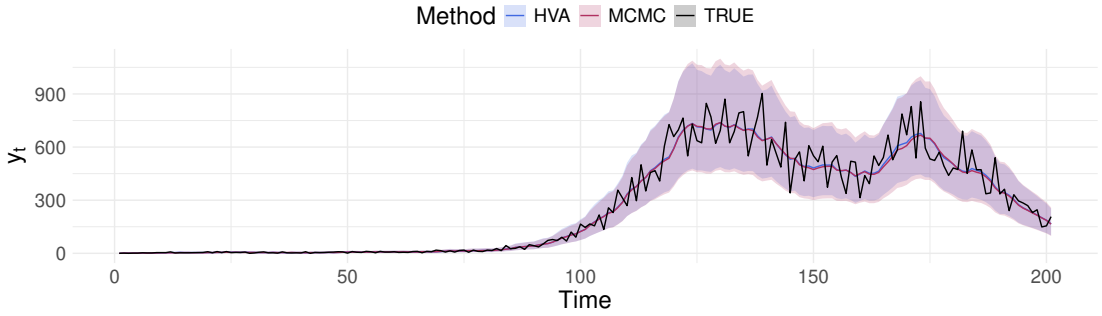
(a) Posterior predictive distributions of y_t.



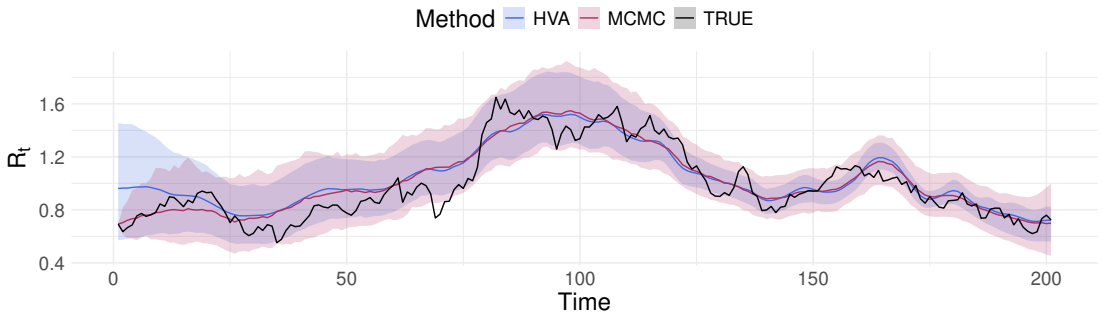
(b) Posterior distributions of R_t.

Figure 2.4.: Posterior predictive distribution of y_t and posterior distribution of R_t from fitting the true model with unknown $\boldsymbol{\gamma} = (\rho, \boldsymbol{\varphi}', \mu, \sigma^2, W)'$ to simulated data. Black lines show true values, while blue and red lines represent results from HVA and MCMC, respectively. Shaded regions indicate 95% credible intervals (CIs) for each method.

ological consistency while highlighting the flexibility of the unified inference scheme. Unlike MCMC, which requires model-specific samplers for each DGTF variant, HVA can be implemented with a single algorithmic structure, using the same two-filter smoother and gradient-based parameter updates. Furthermore, § 2.4.4 showed that HVA achieves comparable point estimates to MCMC with approximately 65% shorter runtime, albeit with some underestimation of posterior variance for latent states, making it well suited for comparative model evaluation where calibrated interval estimates are not the primary



(a) One-step-ahead forecasting distributions $p(y_{t+1} \mid y_{1:t})$ for $t = 140, \ldots, 185$.



(b) Five-steps-ahead forecasting distributions $p(y_{t+5} \mid y_{1:t})$ for $t = 140, \ldots, 185$.

Figure 2.5.: Multi-steps-ahead forecasting distributions using true model $\mathcal{M}_0$, whose parameters are estimated by HVA and MCMC. Solid lines represent the posterior medians, and the colored ribbons indicate the 95% credible intervals (CIs).

Lag distribution recovery. Before examining overall model performance, we first assessed how well each model recovered the true lag distribution, as this mechanistic component is important for capturing disease transmission dynamics. Figure 2.6 compares the true lag distribution with the estimated distributions from models $\mathcal{M}_0$, $\mathcal{M}_1$ and $\mathcal{M}_2$, using posterior medians of the parameters. Note that the distributed lag model can be rewritten as $\lambda_t = \mathbf{x}_t'\boldsymbol{\varphi} + \sum_{l=1}^{+\infty} \phi_l h(\psi_{t+1-l}) y_{t-l}$, where ϕ_l is the PMF of NB2$(4, \kappa)$, making it an implied lag distribution for this model.

Compared to the other two alternatives and as expected, $\mathcal{M}_0$ best recovered the key

characteristics of the true lag distribution: a small weight on $l = 1$, a peak at $l = 3$, and a relatively long tail. In contrast, the lag distribution estimated by $\mathcal{M}_1$ peaked at $l = 2$, and that for the negative-binomial lag distribution in $\mathcal{M}_2$ put more weight on $l = 1$ and had a relatively short tail. For the lag distribution parameters, $\mathcal{M}_0$ estimated $\mu = 1.19$ (95% credible interval (CI): 0.60-1.53) and $\sigma^2 = 0.44$ (0.21-0.97), closely matching true values. Model $\mathcal{M}_1$ underestimated μ at 0.95 (0.62-1.23) with $\sigma^2 = 0.38$ (0.21-0.76). The distributed lag model $\mathcal{M}_2$ estimated $\kappa = 0.5$ (0.38-0.59).

Model performance. Having established differences in lag structure recovery, we now turn to overall model performance across multiple metrics. Table 2.3 summarizes these results. The correctly specified discrete-time Hawkes-type model ($\mathcal{M}_0$) achieved the best performance across all measures in this simulation, with $\chi^2 = 1.19$ and CRPS = 27.16, closely followed by the distributed-lag model ($\mathcal{M}_2$) and the generalized Hawkes model ($\mathcal{M}_1$), both with $\chi^2 \approx 1.22$. By contrast, the autoregressive models $\mathcal{M}_3$ (negative-binomial AR) and $\mathcal{M}_4$ (Poisson AR) performed substantially worse ($\chi^2 = 1.42$ and 1.63, respectively), confirming that purely autoregressive formulations struggle to replicate the transmission dynamics.

Uncertainty quantification revealed distinct patterns across models. The first three models achieved near-nominal 95% CI coverage for y_t (0.97 for $\mathcal{M}_0$ and $\mathcal{M}_1$, 0.96 for $\mathcal{M}_2$), while $\mathcal{M}_3$ showed slight undercoverage at 0.93 and $\mathcal{M}_4$ severely underestimated uncertainty with only 0.73 coverage. For R_t estimation, all models showed undercoverage compared to nominal levels, with $\mathcal{M}_0$ achieving 0.87 coverage and MAE of 0.12, $\mathcal{M}_2$ performing similarly with 0.866 coverage and MAE of 0.13, while $\mathcal{M}_1$ showed poorer calibration at 0.736 coverage despite comparable MAE of 0.13.

Figure 2.7 shows that $\mathcal{M}_0$ and $\mathcal{M}_2$ maintain consistent coverage of the true R_t over time, while $\mathcal{M}_1$ progressively underestimate the uncertainty around R_t. This deterioration likely reflects oversmoothing from time-varying system variance, which can lead shrink state variability when the true state variance is constant.

Despite these differences in latent state estimation, the first three models produced broadly similar posterior predictive distributions for the observed counts y_t, as illustrated in Figs. 2.8a and 2.8b. This similarity in predictions, despite different latent dynamics, underscores that multiple model structures can provide adequate fit to count data, making careful evaluation of mechanistic assumptions important.

The AR models showed distinct limitations. Although $\mathcal{M}_3$'s CIs maintained reasonable coverage despite inflated uncertainty (Fig. 2.8c), the Poisson AR model $\mathcal{M}_4$ failed to capture the overdispersion of the data, resulting in poor predictive performance. This indicates that multiple DGTF formulations can yield adequate data level fits even when their internal mechanistic structures differ, reinforcing the importance of examining epidemiological interpretability, not just predictive accuracy.

Table 2.3.: Performance metrics for five candidate models fitted to data from $\mathcal{M}_0$. The table shows χ^2 discrepancy, CRPS, and 95% posterior predictive interval coverage for observed counts y_t, along with MAE and 95% credible interval coverage for reproduction numbers R_t. Models $\mathcal{M}_3$ and $\mathcal{M}_4$ do not estimate R_t directly.

		$\mathcal{M}_0$	$\mathcal{M}_1$ ($\hat{\delta} = 0.9$)	$\mathcal{M}_2$ ($\hat{r} = 4$)	$\mathcal{M}_3$ ($\hat{p} = 3$)	$\mathcal{M}_4$ ($\hat{p} = 5$)
y_t	χ^2 Discrepancy	1.19	1.22	1.22	1.42	1.63
	CRPS	27.16	27.92	27.18	30.25	31.02
	95% PI Coverage	0.96	0.97	0.96	0.93	0.73
R_t	MAE	0.12	0.13	0.13		
	95% credible interval coverage	0.87	0.736	0.866		

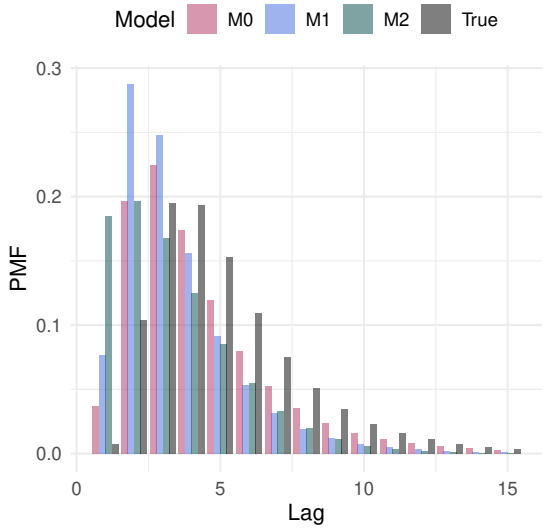


Figure 2.6.: Estimated log-normal lag distribution from $\mathcal{M}_0$ and $\mathcal{M}_1$, along with the implied negative-binomial lag distribution from $\mathcal{M}_2$, where the lag parameters are set to their respective posterior medians.

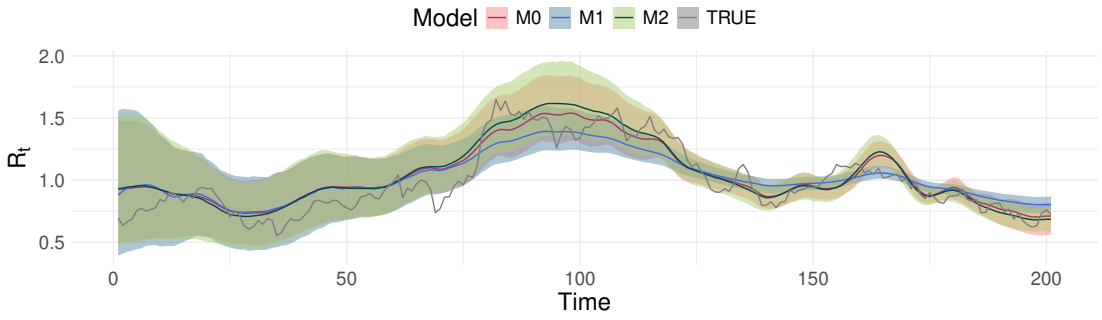


Figure 2.7.: Posterior distributions of R_t using three different models. Black lines represent true values.

2.4.6. Discussion

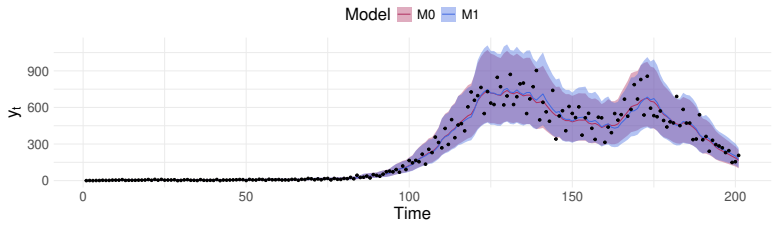
Our simulation study yields several key insights regarding the performance and practical use of DGTF models for epidemic time series. First, the comparison between HVA and MCMC highlights the trade-off between computational efficiency and calibration of posterior uncertainty. HVA provides nearly identical point estimates to MCMC while requiring roughly 65% less runtime, but underestimates posterior variance for latent states, with 95% credible interval coverage for R_t dropping from 0.97 to 0.87. This makes HVA ideal for routine surveillance and large-scale model screening, whereas MCMC remains preferable for final analyses demanding well-calibrated uncertainty quantification. As discussed in § 2.4.4, post-hoc covariance corrections (Giordano et al., 2018) or using HVA posteriors to warm-start short MCMC runs can help bridge this gap.

Second, the mechanistic structure of the model significantly impacts performance. When the true model is fitted to the data, it recovers the lag distribution and the reproduction numbers most accurately, highlighting the value of incorporating epidemiological knowledge into model specification. In the case in which the true model is the discrete-time Hawkes-type model, the distributed lag model provided a reasonable alternative. On the other hand, AR models show an important limitation: standard autoregressive structures cannot capture the renewal process dynamics central to disease transmission. Despite incorporating time-varying coefficients for flexibility, models $\mathcal{M}_3$ and $\mathcal{M}_4$ showed substantially worse performance, with the Poisson variant failing to accommodate overdispersion and the negative-binomial variant overcompensating with excessively wide uncertainty bands.

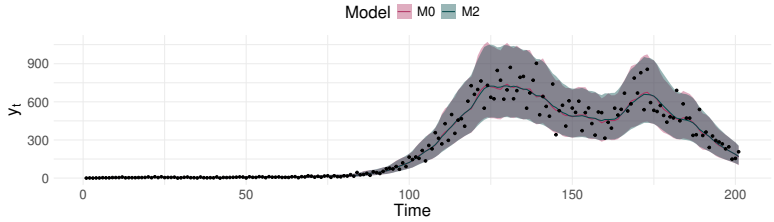
From a practical standpoint, these findings suggest an effective workflow for DGTF

modeling: use HVA for fast model comparison and parameter screening, identify the most appropriate lag and gain structures, and reserve MCMC for confirmatory analyses where precise uncertainty bounds are needed. This strategy combines interpretability, computational scalability, and robustness, principles that are useful for real-time epidemic surveillance.

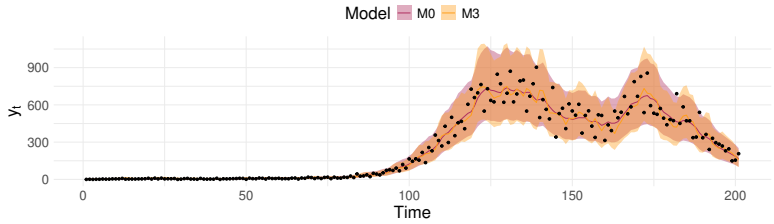
An additional simulation study using data generated from a distributed lag model (§ A.2) reinforced these findings, showing that both discrete-time Hawkes-type and distributed lag models can provide reasonable approximations when the underlying process has appropriate lag structure.



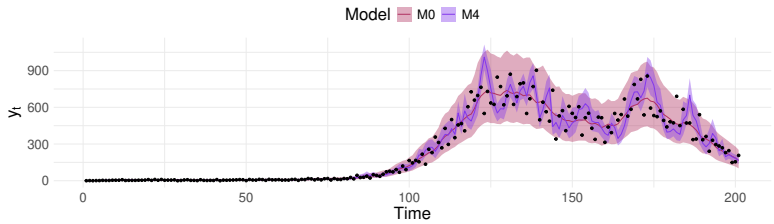
(a) Comparison of the true model $\mathcal{M}_0$ against the generalized discrete-time Hawkes-type model $\mathcal{M}_1$ with discount factor $\hat{\delta} = 0.89$.



(b) Comparison of the true model $\mathcal{M}_0$ against the distributed lag model $\mathcal{M}_2$ with lag order $\hat{r} = 6$.



(c) Comparison of the true model $\mathcal{M}_0$ against the negative-binomial AR model $\mathcal{M}_3$ with autoregressive lag $\hat{p} = 3$.



(d) Comparison of the true model $\mathcal{M}_0$ against the Poisson AR model $\mathcal{M}_4$ with autoregressive lag $\hat{p} = 6$.

Figure 2.8.: Posterior predictive distributions estimated using HVA by different models, compared against the observed count time series simulated from $\mathcal{M}_0$. The black dots indicate the true values, the solid lines represent posterior medians, and the ribbons represent 95% CIs.

§ 2.5. Application: COVID-19 in Santa Cruz County

We analyze daily confirmed COVID-19 cases from Santa Cruz County, California from the California Department of Public Health. The dataset spans $T = 518$ days from July 1, 2020 to December 1, 2021. This period spans two major pandemic waves and pronounced weekly seasonality due to testing and reporting practices (Fig. 2.9).

Our analysis addresses two key questions. First, we compare the DGTF framework against established epidemiological methods to demonstrate its practical advantages, particularly its ability to directly incorporate components such as seasonality while maintaining computational efficiency. Second, we compare different DGTF models. This within-framework comparison aims to guide standard practice in applied epidemiology, where analysts fit multiple models to identify the structure that best captures transmission dynamics.

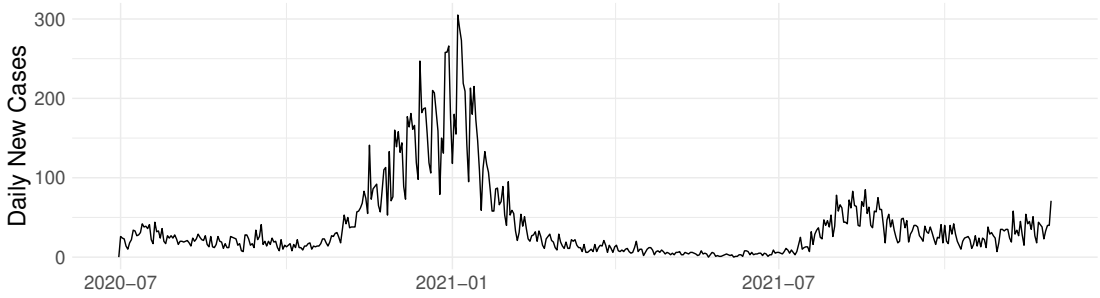


Figure 2.9.: Daily new cases of COVID-19 from July 1st, 2020 to Dec 1st, 2021 in Santa Cruz County.

2.5.1. Benchmark models

We evaluated our DGTF framework against three established methods for estimating effective reproduction numbers to demonstrate the advantages of our unified approach. The benchmark models represent some of the current standard approaches used in epidemiological surveillance. Our DGTF models offer enhanced flexibility through direct incorporation of seasonality and unified inference procedures. The benchmark models cannot directly incorporate seasonality, therefore, we pre-processed their input data using the day-of-week adjustment method from (Koyama et al., 2021).

(Koyama et al., 2021). A discrete-time Hawkes-type model with ramp gain function $h(\psi_t) = \max(\psi_t, 0)$ and a log-normal lag distribution ($\mu = 1.39$, $\sigma^2 = 0.32$ truncated at $L = 30$ days). These values yield a mean lag of 4.7 days and standard deviation of 2.9 days, consistent with COVID-19 serial interval estimates. The latent states follow a univariate random walk with Cauchy noise (scale = 0.001). Inference uses a fixed-lag smoother with 10,000 particles (Kitagawa, 1994).

EpiEstim (Cori et al., 2013). A discrete renewal model with Poisson observations model and Gamma-distributed lag weights with unknown μ and σ. We specified priors $\mu \sim N(4.7, 1.8^2)\mathbf{1}(1.1 < \mu < 8.3)$ and $\sigma \sim N(2.9, 1.4^2)\mathbf{1}(0.1 < \sigma < 5.7)$, centering the distributions around the same serial interval parameters while allowing flexibility. We implemented Bayesian inference via MCMC using the `EpiEstim` package. We sampled 1000 parameter pairs (μ, σ) from their posterior distribution, and generated 1000 R_ts per parameter pair using an adaptive sliding window that maintains the posterior coefficient of variation below a threshold. This analysis yielded posterior estimates of $\mu = 4.79$

(95% CI: 1.61-7.56) and $\sigma = 2.98$ (95% CI: 0.73-5.30).

The WT method (Wallinga and Teunis, 2004). This method probabilistically reconstructs transmission trees to count secondary cases per infected individual, estimating the likelihood of case i infecting case j through a Gamma distribution, which represents either the generation or serial interval. We calibrated the Gamma parameters to match our standard mean (4.7 days) and standard deviation (2.9 days), implementing the method through the `EpiEstim` R package.

2.5.2. DGTF models

We considered three variants from the DGTF family using the same negative binomial observation model, $y_t \sim \text{NB}(\lambda_t, \rho)$: (1) a discrete-time Hawkes-type model, (2) a distributed lag model, and (3) a AR with time-varying autoregressive coefficients. These variants represent different assumptions about transmission dynamics. We incorporated weekly seasonality through a seasonal parameter vector $\boldsymbol{\varphi} = (\varphi_1, \ldots, \varphi_7)'$ and a design matrix $\mathbf{X} = (\mathbf{x}_0, \ldots, \mathbf{x}_T)'$. Here, $\mathbf{x}_t = \mathbf{P}\mathbf{x}_{t-1}$ for $t = 1, \ldots, T$, starting with $\mathbf{x}_0 = (1, 0, \ldots, 0)'$. The matrix $\mathbf{P}$ was a 7×7 permutation matrix defined as:

$$\mathbf{P} = \begin{pmatrix} \mathbf{0} & \mathbf{I}_6 \\ 1 & \mathbf{0}' \end{pmatrix}.$$

All models used a softplus gain function $h(\psi_t) = \log(\exp(\psi_t) + 1)$ to ensure non-negativity while maintaining differentiability.

For the discrete-time Hawkes-type model, we used a log-normal lag distribution (truncated at $L = 30$) with unknown mean μ and variance σ^2. The latent states follow

univariate random walk processes $\psi_t \sim N(\psi_{t-1}, W)$. This model required the estimation of static parameters $\boldsymbol{\gamma} = (\rho, W, \mu, \sigma^2, \boldsymbol{\varphi})$. Similarly, the distributed lag model, denoted by DL(r), also assumed univariate random walk processes $\psi_t \sim N(\psi_{t-1}, W)$ at the system level. We chose the lag order r from $\{2, \ldots, 9\}$ by minimizing the χ^2 discrepancy and also estimated the unknown static parameters $\boldsymbol{\gamma} = (\boldsymbol{\varphi}', \rho, W, \kappa)'$.

The autoregressive NB-AR(p) model accommodates nonstationarity through time-varying coefficients:

$$\lambda_t = \boldsymbol{x}_t' \boldsymbol{\varphi} + \sum_{i=1}^{p} b_{i,t} y_{t-i}.$$

Non-negative coefficients are ensured through $b_{i,t} = h(\psi_{i,t})$, with latent states following $\boldsymbol{\theta}_t \sim N(\boldsymbol{\theta}_{t-1}, W\mathbf{I})$. We selected $p \in \{2, \ldots, 6\}$ by minimizing χ^2 discrepancy.

For all three models, we assumed priors $\rho, W \sim \text{IG}(1, 1)$ and $\varphi \sim N(0, 10)\mathbf{1}(\varphi_i > 0)$, where φ_i represented the ith seasonal component in $\boldsymbol{\varphi} = (\varphi_i, i = 1, \ldots)$. The discrete-time Hawkes-type model additionally used priors $\sigma^2 \sim \text{IG}(1, 1)$ and $\mu \sim N(0, 1)$, while the distributed lag model assumed $\kappa \sim \text{Beta}(1, 1)$.

We performed inference using the HVA algorithm with 5,000 iterations, monitoring convergence through trace plots of $\log p(y_{1:T} \mid \boldsymbol{\gamma})$, (see Fig. A.13). We generated 5,000 posterior samples of $\boldsymbol{\gamma}$ and $\boldsymbol{\theta}_{1:T}$ after convergence using the optimized parameters. Our HVA implementation used a two-filter smoother with 500 particles and a resampling ratio of $m = 0.5$. The algorithm used a step size of 10^{-5} and ADADELTA learning rate of 0.02, with variational proposal dimension k set to the number of static parameters minus one. This resulted in $k = 10$ for the discrete-time Hawkes-type model, $k = 9$ for the distributed lag model, and $k = 8$ for the autoregressive model.

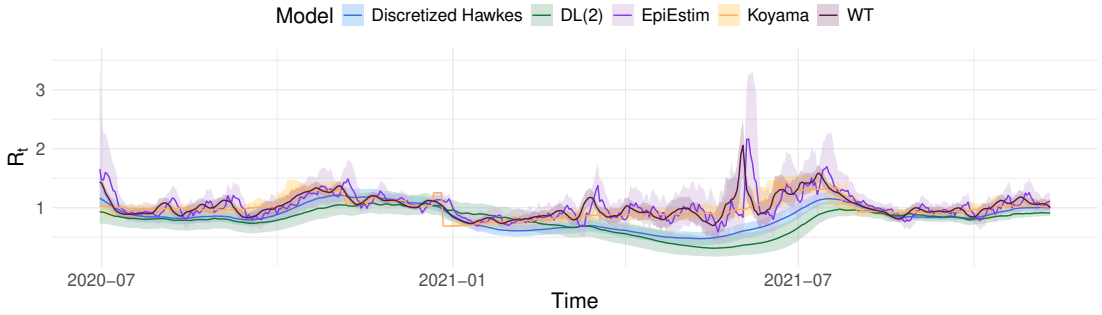
2.5.3. Results

Hyperparameter selection and model fit. We determined the optimal parameters for each model by minimizing the χ^2 discrepancy, with detailed results presented in Table A.11. For the distributed lag model, we identified an optimal lag order of $\hat{r} = 2$, while the NB-AR performed best with $\hat{p} = 3$. Using these optimal parameters, we compared the discrete-time Hawkes-type model, DL(2), and NB-AR(3) models against our benchmarks.

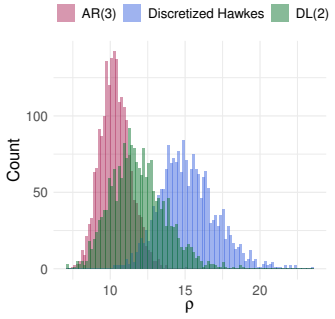
Next, we compare our DGTF framework against established benchmark methods to demonstrate the advantages of our unified approach. A key distinction between our approach and the benchmark models lies in how we handle seasonality. While benchmark models require pre-adjusted data, DGTF models directly incorporate weekly patterns as a covariate effect into their structure. This difference proves important during low-incidence periods. Figure 2.10a reveals that all models produce broadly similar R_t estimates during high-incidence periods, with overlapping 95% credible intervals. However, between March and July 2021, when case counts were low, DGTF models with seasonality yield consistently lower R_t estimates. This divergence shows that failing to account for reporting patterns can inflate transmission estimates when cases are sparse.

Within-DGTF model selection. Having established the advantages of our framework over benchmark methods, we explore model selection within our framework. This comparison serves two purposes: demonstrating how practitioners can use the unified framework to test different epidemiological assumptions, and identifying which structures best capture COVID-19 transmission dynamics.

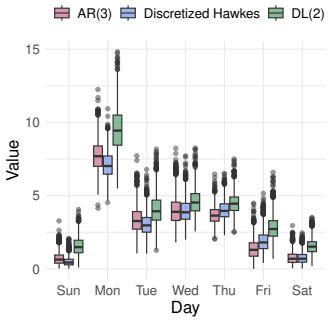
Table 2.4 provides the quantitative basis for model comparison. The discrete-time



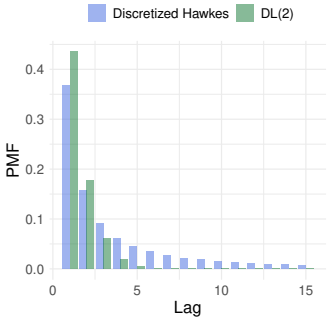
(a) Estimated effective reproduction numbers (R_t). NB-AR(3) is excluded (does not estimate R_t). Benchmark models used seasonally adjusted data, while DGTF variants incorporated seasonality directly and used the original case counts.



(b) Dispersion ρ.



(c) Weekly seasonality $\boldsymbol{\varphi}$.



(d) Lag distributions.

Figure 2.10.: Posterior distributions of the effective reproduction numbers $\{R_t\}$ and static parameters $(W, \rho, \boldsymbol{\varphi})$ for Santa Cruz County, estimated by different models. In (d), we illustrate the log-normal lag distribution in the discrete-time Hawkes-type model, using the posterior medians for μ and σ_2 and the implied negative-binomial leg distribution in the distributed lag model with κ set to its posterior median.

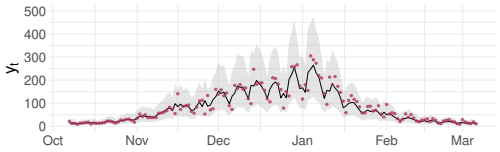
Hawkes-type model achieved the best predictive performance with $\chi^2 = 2.338$ and CRPS = 7.14, followed closely by DL(2) with $\chi^2 = 2.639$ and CRPS = 7.19. The autoregressive model performed substantially worse ($\chi^2 = 3.064$, CRPS = 9.52), suggesting that explicit modeling of transmission lags improves fit. Importantly, coverage rates near 95% for the first two models indicate well-calibrated uncertainty, while interval widths reflect different variance assumptions.

Table 2.4.: Comprehensive performance comparison between DGTF models for COVID-19 data in Santa Cruz County. Metrics include in-sample fit and posterior predictive interval calibration for observed counts y_t, calculated over the full time series (July 2020 - December 2021).

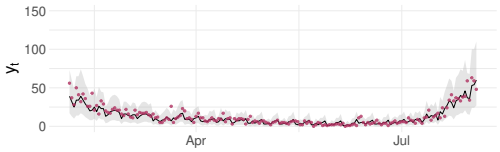
	χ^2 Discrepancy	CRPS	95% PI Coverage
Discrete-time Hawkes-type model	2.338	7.14	0.93
DL(2)	2.639	7.19	0.95
NB-AR(3)	3.064	9.52	0.92

Epidemiological insights from parameter estimates. The estimated seasonal parameters reveal clear weekly reporting patterns (Fig. 2.10c). All models identify approximately 80% fewer reported cases on weekends ($\hat{\varphi}_{\text{Sat}} \approx 1.3$, $\hat{\varphi}_{\text{Sun}} \approx 1.0$), compensated by a surge on Mondays ($\hat{\varphi}_{\text{Mon}} \approx 8.2$). This pattern, consistent across all model specifications, reflects administrative delays in testing and reporting rather than actual transmission dynamics. These artificial fluctuations need to be accounted for to avoid biased R_t estimates.

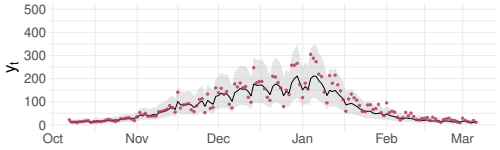
The estimated lag distributions provide insights into the process from transmission to reporting (Fig. 2.10d). The discrete-time Hawkes-type model estimates a mean lag of 8.2 days, longer than the COVID-19 serial interval alone (4.2-7.5 days; Alene et al., 2021). This extended lag captures the cascade from infection through symptom onset, testing,



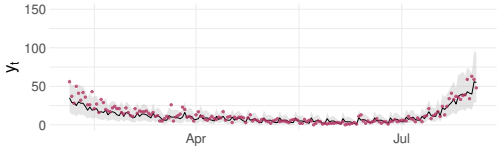
(a) NB-AR(3): Oct 2020 to March 2021



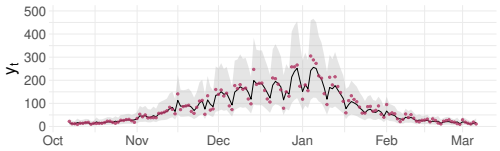
(b) NB-AR(3): Feb 2021 to Aug 2021



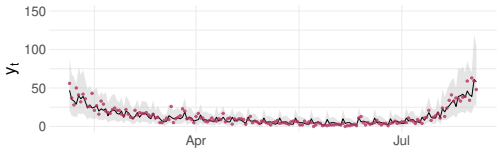
(c) Discrete-time Hawkes-type model: Oct 2020 to March 2021



(d) Discrete-time Hawkes-type model: Feb 2021 to Aug 2021



(e) DL(2): Oct 2020 to March 2021



(f) DL(2): Feb 2021 to Aug 2021

Figure 2.11.: Posterior predictive distributions of daily new cases across two time periods for the DGTF models: Oct 2020 to March 2021 (left) and Feb 2021 to Aug 2021 (right). The posterior medians along with the 95% CIs are illustrated. Red dots represent the daily new cases.

and administrative reporting. The DL(2) model's geometric lag structure may truncate this tail prematurely, partially explaining its slightly worse predictive performance.

Dispersion parameter estimates reveal important differences in how models partition variability (Fig. 2.10b). The discrete-time Hawkes-type model estimates $\hat{\rho} \approx 15$, indicating moderate overdispersion consistent with superspreading events. In contrast, NB-AR(3)'s lower estimate ($\hat{\rho} \approx 10$) suggests it attributes more variation to the observation process rather than true transmission dynamics. This difference in variance decomposition affects both uncertainty quantification and outbreak forecasting.

Computational performance. We compared HVA against MCMC for the best-performing discrete-time Hawkes-type model. To evaluate one-step-ahead forecasting, we used a rolling forecast procedure: starting from December 1, 2021, we fit each model using all data from July 2, 2020 to the current date, generated a one-step-ahead forecast, then rolled forward day-by-day through December 15, 2021. The reported metrics are averages across these 15 days.

Table 2.5.: Comparison of HVA and MCMC inference for the discrete-time Hawkes-type model on COVID-19 data. Parameter estimates show posterior medians with 95% credible intervals. Performance metrics include χ^2 discrepancy, CRPS, and 95% posterior predictive interval coverage for in-sample fit, and χ^2 discrepancy, CRPS, and 95% posterior predictive interval width for one-step-ahead forecasting (both achieved 100% coverage for one-step forecasts).

	Metric	HVA	MCMC
Runtime (s)		3,896	13,483
In-sample fit	χ^2 discrepancy	2.338	2.260
	CRPS	7.14	7.05
	95% PI Coverage	0.93	0.95
One-step forecasting	χ^2 discrepancy	0.133	0.132
	CRPS	8.382	8.127
	95% PI Width	48.43	53.29
Parameter estimates	ρ	15.0 (11.9, 19.3)	13.6 (11.2, 16.6)
	μ	0.59 (0.13, 1.00)	0.46 (0.09, 0.88)
	σ^2	3.02 (1.84, 5.10)	2.54 (1.50, 4.86)
	W	0.0047 (0.0037, 0.0059)	0.0039 (0.0035, 0.0052)

Table 2.5 reveals the computational and statistical trade-offs between the two inference approaches. HVA completed 5,000 gradient ascent iterations and generated 5,000 posterior samples in 3,896 s, while MCMC required 13,483 s for 5,000 burn-in iterations plus thinning (keeping every second sample) to obtain 5,000 effective samples. We note that both algorithms may have achieved convergence before completing all iterations

(see Fig. A.13); these runtimes reflect our conservative choice to ensure thorough exploration of the parameter space rather than the minimum necessary computation.

For in-sample fit, both methods produce very similar results, with MCMC showing marginally better performance (3% lower χ^2 discrepancy and 1% lower CRPS). For one-step-ahead forecasting, both methods achieve nearly identical χ^2 discrepancy (0.133 vs 0.132), while MCMC produces slightly better probabilistic forecasts with a 3% lower CRPS. Notably, HVA generates slightly narrower prediction intervals than MCMC (48.43 vs 53.29), yet both methods achieve 100% coverage for one-step-ahead predictions. Parameter estimates remained consistent between methods with overlapping credible intervals. These results confirm that HVA provides reliable inference at substantially reduced computational cost. The narrower prediction intervals are consistent with the variance underestimation discussed in § 2.4.4, though in this application both methods achieve full coverage for one-step-ahead predictions.

2.5.4. Discussion

Our analysis shows that model structure has considerable impacts on epidemiological inference. The relatively strong performance of the discrete-time Hawkes-type model in this application is consistent with its separation of transmission intensity (R_t) from the temporal pattern of transmission, together with the flexibility of the log-normal lag structure.

Direct performance comparison with benchmark models was limited because they require seasonally adjusted data. Our supplementary analysis (§ A.3.4) compared all models on original count data without seasonal adjustments. This revealed that DGTF and benchmark models produce similar R_t estimates when neither accounts

for seasonality. However, incorporating seasonal terms in DGTF improved model fit, reducing χ^2 discrepancy from 2.518 to 2.338.

These findings highlight key advantages of the DGTF framework. First, incorporation of weekly reporting patterns significantly improves model fit and prevents biased R_t estimates during low-incidence periods. The 80% weekend drop in reported cases reflects administrative delays, not transmission dynamics, emphasizing the importance of modeling these patterns explicitly. Second, the unified inference framework enables efficient comparison of different epidemiological assumptions. Using the same algorithm across models allows practitioners to explore structural uncertainty, particularly valuable for emerging diseases with unknown transmission mechanisms.

The AR models, while computationally efficient, lack direct R_t interpretation. Among models that estimate R_t, the discrete-time Hawkes-type model with log-normal lag distribution captures the prolonged transmission-to-reporting process better than the geometric structure in the distributed lags model DL(2). However, we caution against directly interpreting estimated lags as epidemiological parameters such as serial intervals or generation intervals, since observed data conflates multiple processes including infection-to-onset delays, testing delays, and reporting delays.

§ 2.6. Concluding remarks

This chapter introduced the DGTF class as a unified state-space framework for count time series in disease surveillance. This modeling framework places discrete-time Hawkes-type models, Poisson autoregressions, and distributed lag models inside a common modular structure, so that an analyst can move between model families by

changing one component while holding the inference routine fixed. The HVA algorithm combines a SMC pass on the latent trajectory with SGA on the variational ELBO for the static parameters, and produced point estimates that agreed with the MCMC target while running roughly three times faster across the model variants considered.

Two findings from the empirical work are worth highlighting. First, the gap between HVA and MCMC on credible interval coverage of R_t was modest but real, with coverage of nominal 95% intervals falling from 0.97 to 0.87. This is the variance underestimation that typically accompanies Gaussian-family variational methods, and it places HVA naturally in the model exploration and routine surveillance role rather than in the final inference role, where the calibrated intervals of MCMC remain preferable. Post-hoc covariance corrections (Giordano et al., 2018; Syring and Martin, 2019) and short MCMC runs warm-started from the variational posterior can close part of the gap. Second, the modular framework made it inexpensive to compare DGTF variants on the Santa Cruz data, and the comparison was informative. A discrete-time Hawkes-type model with a log-normal lag distribution better out-of-sample forecast scores than the geometric distributed-lag and autoregressive variants in the Santa Cruz analysis, and explicit weekly seasonality in the regression term reduced bias in the R_t estimates during low-incidence periods.

Three threads developed in the rest of the thesis pick up directly from this chapter. Chapter 3 extends the DGTF class to spatially connected populations through the multivariate Hawkes generalization sketched in Eq. (1.2.3), with a sparsity-promoting prior on the transmission matrix and a decomposition of the observed reproduction number into local and imported components. Chapter 4 develops a structured variational algorithm for the spatial model motivated by limitations of the joint Gaussian variational

family in the multiplicative geometry that arises there, and stands in the same relation to the network Hawkes model that HVA stands in to DGTF. Chapter 5 describes the **dgtf** package for R, which implements the model class and inference routines developed in this chapter through an interface aligned with the six modular components.

Several methodological directions remain open. Loss-based variational objectives (Frazier et al., 2025) would allow the inference procedure to be tailored to specific epidemiological loss functions, such as those that penalize false negatives during outbreak detection. Recursive updating (Tomasetti et al., 2022; Mastrototaro and Olsson, 2024) and nested SMC (Naesseth et al., 2019) would enable fully online inference as new observations arrive. Natural gradient updates (Zhang et al., 2024) could improve convergence of the variational optimization in regimes where the posterior is highly anisotropic. Continuous-time formulations of the DGTF class would also be of interest in settings where surveillance data are reported on irregular schedules.

Chapter 3

Network Hawkes models for spatio-temporal reproduction numbers

§ 3.1. Introduction

The multivariate generalization of the self-exciting form introduced in § 1.2.3 replaces the single-region reproduction number R_t of Chapter 2 with source-specific trajectories $R_{k,t}$, where k indexes the source location, and a transmission matrix $\boldsymbol{\alpha}$ whose entry $\alpha_{s,k}$ allocates secondary infections generated in source location k to destination location s. When $\boldsymbol{\alpha}$ is column-stochastic, the closed-population reproduction number at each destination decomposes additively into a local component, given by $\alpha_{s,s}R_{s,t}$, and an imported component, given by $\sum_{k\neq s}\alpha_{s,k}R_{k,t}$, and the relevant epidemic threshold for the multi-region system is the spectral radius of the offspring matrix $\boldsymbol{\alpha}\mathbf{R}_t$ rather than a single diagonal entry. The structural form is well understood. The substantive gap, and the focus of this chapter, is the lack of a Bayesian inference procedure that learns $\boldsymbol{\alpha}$ from incidence data with full posterior uncertainty quantification, makes the local and imported decomposition explicit as a posterior summary, and remains tractable at the

spatial scales typical of regional surveillance.

Several existing frameworks address spatial structure in disease surveillance, but each leaves part of this gap unfilled. The endemic-epidemic framework of Held et al. (2005) and Meyer et al. (2017) decomposes regional incidence into endemic and autoregressive components, but typically requires spatial weights to be pre-specified from adjacency or population flow rather than learned from data (Paul et al., 2008; Bracher and Held, 2022). Multivariate Hawkes processes provide a natural framework for spatial transmission (Rizoiu et al., 2018; Chiang et al., 2022; Laub et al., 2025), but applications to date have either fixed the reproduction numbers (Kim et al., 2019) or treated regions independently without cross-regional excitation (Unwin et al., 2021). More recently, Santitissadeekorn et al. (2025) and Lasalle and Pascal (2025) have proposed data-driven procedures that learn network structure from incidence directly, but rely on frequentist penalization without full posterior uncertainty quantification, and none of the existing approaches formally decomposes observed reproduction numbers into local and imported components. Holbrook et al. (2020b) develop scalable Bayesian inference for continuous-time spatio-temporal Hawkes models using GPU-accelerated likelihood calculations, with an application to large gunfire data. Related work addresses spatial coarsening in Hawkes models for gunfire, wildfire, and viral contagion (Holbrook et al., 2020a), and recent COVID-19 applications fit spatio-temporal Hawkes models to county-level case records by treating cases as individual events with latent locations within counties (Ko et al., 2024). The present chapter is complementary to this literature: it keeps the data in aggregated county-by-day form and focuses on Bayesian inference for intrinsic source-specific reproduction numbers, spatial allocation weights, and the decomposition of observed reproduction numbers into local and imported components.

More specifically, this chapter develops a Bayesian network Hawkes-type model that learns the spatial allocation matrix from incidence data, quantifies posterior uncertainty, provides the local/imported decomposition as a posterior summary, and remains tractable at the spatial scales typical of regional surveillance. The transmission matrix $\boldsymbol{\alpha}$ is given a regularized horseshoe prior (Carvalho et al., 2010b; Piironen and Vehtari, 2017) on its off-diagonal entries, with the diagonal retention rate parameterized separately to preserve the column-stochastic constraint, so the network structure is learned from incidence data with posterior uncertainty rather than imposed in advance. The intrinsic source-specific reproduction numbers $R_{k,t}$ are treated as latent random-walk trajectories, sharing information across time without imposing a sliding-window assumption. The decomposition of the closed-population reproduction number into local and imported components is provided as a posterior summary, and the connection to the system-level epidemic threshold is made through the spectral radius of the inferred offspring matrix. A Markov-switching zero-inflation component (Lambert, 1992; Douwes-Schultz and Schmidt, 2022; Douwes-Schultz et al., 2025) absorbs runs of structural zeros without conflating disease absence with low-intensity transmission, which matters in small-area surveillance where consecutive zeros are common.

Posterior inference is harder than in the univariate setting of Chapter 2 for two reasons. First, the latent reproduction trajectories at the S locations are strongly correlated across time, and the horseshoe prior induces complex dependence among the spatial parameters within each column of $\boldsymbol{\alpha}$. Second, the transmission weights and the reproduction numbers enter the intensity multiplicatively, which produces ridges in the joint posterior wherever destination trajectories are highly correlated across source locations. This chapter develops a block MCMC sampler that uses column-wise con-

ditional independence to update each column of $\boldsymbol{\alpha}$ independently, applies non-centered parameterizations (Betancourt and Girolami, 2013) to both the horseshoe block and the random-walk innovations, and uses a Gaussian approximation to the Poisson likelihood to construct structured multivariate normal proposals for the latent disturbances. Each iteration costs $O(ST^2 + S^2T)$, which is acceptable for the medium-term policy planning horizon and motivates the variational counterpart developed in Chapter 4 for routine updates over larger networks.

The remainder of this chapter is organized as follows. § 3.2 presents the network Hawkes-type model, including the spatial weight parameterization, the random-walk dynamics on the reproduction numbers, and the regularized horseshoe prior, together with the local and imported decomposition and the spectral radius threshold. § 3.3 describes the block MCMC algorithm. § 3.4 reports simulation studies on parameter recovery and predictive performance across transmission regimes. § 3.5 applies the model to COVID-19 surveillance data from ten California counties, with the local and imported decomposition, forecasting performance relative to standard methods, and a model-based scenario analysis of targeted interventions. § 3.6 summarizes the chapter and connects its contributions to the variational counterpart developed in Chapter 4.

§ 3.2. Methodology

3.2.1. Model

Let $y_{s,t}$ denote the observed case count at location $s \in \{1, \ldots, S\}$ and time $t \in \{1, \ldots, T\}$. To accommodate structural zeros commonly observed in small-area surveillance data

we adopt a zero-inflated Poisson formulation as follows

$$
\begin{aligned}
y_{s,t} &\sim \begin{cases} \text{Pois}(\lambda_{s,t}), & z_{s,t} = 1, \\ \delta_0, & z_{s,t} = 0, \end{cases} \\
\lambda_{s,t} &= \mu + \sum_{k=1}^{S} \sum_{l=1}^{t-1} \alpha_{s,k} R_{k,l} \phi_{t-l} y_{k,l}, \\
z_{s,t} &\sim \text{Bern}(p_{s,t}),\ \text{logit}(p_{s,t}) = \beta_0 + \beta_1 z_{s,t-1}.
\end{aligned}
$$

Here, δ_0 denotes a point mass at zero, and $z_{s,t}$ is a latent activation indicator. When $z_{s,t} = 0$, the model enforces a structural zero; when $z_{s,t} = 1$, counts follow the Poisson transmission mechanism. This separates structural zeros or reporting absences from low-intensity transmission and reduces the tendency to explain persistent zeros by shrinking reproduction numbers. Conditional on activation, the intensity $\lambda_{s,t}$ combines a baseline rate $\mu > 0$ for sporadic importations not explained by prior cases with transmission from earlier cases. Those transmission contributions are weighted by the spatial allocation parameter $\alpha_{s,k}$, the intrinsic source-specific reproduction number $R_{k,l}$, and the lag distribution ϕ_{t-l}, which is treated as fixed from external epidemiological studies (Alene et al., 2021). Fixing the serial interval rather than estimating it avoids a further source of weak identifiability. Because the lag kernel, the reproduction numbers, the spatial weights, and the baseline all shape the same conditional mean, a change in ϕ can be partly absorbed by a change in the smoothness of $R_{k,t}$, in the spatial weights, or in the baseline rate, so estimating ϕ jointly from incidence alone would be poorly determined. A sensitivity analysis over shorter and longer serial intervals is reported in § B.5.1. We place a normal prior on the log baseline rate, $\log(\mu) \sim N(a_\mu, b_\mu^2)$, to ensure positivity. The activation probability $p_{s,t}$ follows an autoregressive logistic model. The

coefficients β_0 and β_1 control the baseline activation probability and persistence of the active state. We assign priors $\beta_0 \sim N(a_{\beta_0}, b^2_{\beta_0})$ and $\beta_1 \sim N(a_{\beta_1}, b^2_{\beta_1})$.

Cases at location s arise from local transmission from prior cases at s and importation from other locations $k \neq s$. The spatial weight matrix $\boldsymbol{\alpha} = [\alpha_{s,k}]$ encodes this structure: diagonal elements $\alpha_{s,s}$ represent local retention, whereas off-diagonal elements represent cross-location transmission. The reproduction number $R_{k,t}$ captures time-varying transmissibility at location k. We model $R_{k,t}$ as follows:

$$R_{k,t} = h(r_{k,t}),\ r_{k,t} = r_{k,t-1} + \sqrt{W}\eta_{k,t},\ \eta_{k,t} \sim N(0,1).$$

Here $h(\cdot)$ is a monotone link function, such as softplus or the exponential, ensuring positivity; $r_{k,t}$ follows a random walk; and W controls trajectory smoothness. In the simulation studies, W is fixed at its true data generating value; in the application, it is selected by out-of-sample forecasting performance as described in § B.6.1. We assign the initial state the prior $r_{k,0} \sim N(0, W)$.

To preserve this interpretation and avoid scale non-identifiability between $\{R_{k,t}\}$ and $\boldsymbol{\alpha}$, we impose a column-stochastic constraint on the spatial weight matrix: $\sum_{s=1}^{S} \alpha_{s,k} = 1$ for each source location k. Under this constraint, $R_{k,t}$ is the total expected offspring generated by a case at (k,t), and $\alpha_{s,k}$ allocates those offspring across destination locations. We decompose $\boldsymbol{\alpha}$ into diagonal and off-diagonal components:

$$\alpha_{s,k} = \begin{cases} w_k, & s = k, \\ (1 - w_k)\dfrac{u_{s,k}}{U_k}, & s \neq k. \end{cases}$$

Here $w_k \in (0,1)$ is the retention rate at location k, and $u_{s,k}/U_k$ with $U_k = \sum_{j \neq k} u_{j,k}$ determines how exported cases are distributed across destinations. Larger w_k values

indicate more local transmission. We set $w_k \sim \text{Beta}(a_w, b_w)$. To model off-diagonal redistribution flexibly, we use a log-linear specification with mobility covariates and location-specific deviations:

$$u_{s,k} = \exp(v_{s,k}), \quad v_{s,k} = m_{s,k} + \tilde{\theta}_{s,k}, \quad s \neq k.$$

The exponential transformation ensures positivity of $u_{s,k}$, and the log-linear form yields additive effects on the log-scale. Here $m_{s,k}$ is a mobility or distance covariate capturing the baseline propensity for transmission from k to s, and $\tilde{\theta}_{s,k} = \theta_{s,k} - \bar{\theta}_k$ is a centered deviation term, with $\bar{\theta}_k = (S-1)^{-1} \sum_{j \neq k} \theta_{j,k}$. Centering removes column-wise mean shifts in log-weights, so deviations represent relative over- or under-transmission to specific destinations rather than global inflation or deflation of exported mass. Under the column-stochastic constraint, this preserves the interpretation of $R_{k,t}$ as total expected offspring.

Most location pairs are expected to follow baseline mobility patterns, with only a small subset showing substantial deviations. To encode this sparsity, we assign a regularized horseshoe prior to the terms $\theta_{s,k}$ (Carvalho et al., 2010b; Piironen and Vehtari, 2017):

$$\theta_{s,k} \sim N(0, \tau_k^2 \tilde{\gamma}_{s,k}^2), \quad s \neq k, \quad \tilde{\gamma}_{s,k}^2 = \frac{c^2 \gamma_{s,k}^2}{c^2 + \tau_k^2 \gamma_{s,k}^2},$$

Here τ_k controls column-level sparsity, $\gamma_{s,k}$ allows pathway-specific escape from shrinkage, c defines the slab scale, and $\tilde{\gamma}_{s,k}^2$ is the regularized local variance. This prior shrinks most deviations $\theta_{s,k}$ toward zero, corresponding to transmission proportional to mobility, while allowing a small number of edges to escape shrinkage when supported by the data. Because shrinkage is applied only to deviations from baseline mobility, not to w_k or reproduction numbers $R_{k,t}$, the prior regularizes network structure without directly

constraining intrinsic transmissibility. The priors on these variance components are

$$\gamma_{s,k} \sim C^{+}(0,1), \quad \tau_k \sim C^{+}(0,\tau_0),$$

where $C^{+}(0,s)$ denotes a half-Cauchy distribution with scale s. Following Piironen and Vehtari (2017), we set the global shrinkage hyperparameter as

$$\tau_0 = \frac{p_0 \sigma_y}{(S-1-p_0)\sqrt{T}},$$

where p_0 is the expected number of non-zero deviations per source location, σ_y is the scale of the observed counts, $S-1$ is the number of potential destinations, and T is the number of time points. This calibration ties global shrinkage to the expected sparsity and data scale, strongly shrinking small deviations toward zero while allowing anomalous pathways to be detected.

3.2.2. Reproduction number quantities

The model above implies several distinct reproduction number quantities that should not be interpreted interchangeably. The quantities introduced below are indexed either by source location k or by destination location s, reflecting two complementary inferential viewpoints. The intrinsic source-specific reproduction number $R_{k,t}$ is source-indexed and forward-looking, whereas the effective and local reproduction numbers are destination-indexed and backward-looking. This distinction is intentional and reflects the directional nature of transmission. In the absence of spatial coupling ($\alpha_{s,k} = 0$ for $s \neq k$), the source and destination indices coincide and $R^{\text{eff}}_{s,t} = R_{s,t}$.

Intrinsic intrinsic source-specific reproduction number. For a case arising in region k at time t, the intrinsic source-specific reproduction number $R_{k,t}$ represents the expected

total number of secondary infections generated across all regions and future times. Under the column-stochastic constraint $\sum_{s=1}^{S} \alpha_{s,k} = 1$, the expected number of offspring appearing in region s equals

$$B_t(s,k) = \alpha_{s,k} R_{k,t}.$$

Summing across locations yields $\sum_{s=1}^{S} B_t(s,k) = R_{k,t}$. Thus $R_{k,t}$ is a forward-looking measure of the intrinsic transmissibility of newly generated cases in region k. The quantity $w_k R_{k,t}$, with $w_k = \alpha_{k,k}$, corresponds to the expected number of offspring retained locally.

Network reproduction number. Let $\boldsymbol{B}_t = [B_t(s,k)]$ denote the offspring matrix at time t. Since $\sum_{h>0} \phi_h \approx 1$, the total expected number of offspring from a primary case at (k,t) equals $R_{k,t}$, and the spatial allocation is determined by $\boldsymbol{\alpha}$.

The matrix $\boldsymbol{B}_t$ plays the role of a next-generation matrix in multivariate branching process theory (Diekmann et al., 1990; Allen and van den Driessche, 2013). We define the network reproduction number as

$$R_t^{\text{net}} = \rho(\boldsymbol{B}_t),$$

where $\rho(\boldsymbol{B}_t)$ denotes the spectral radius (largest eigenvalue in modulus) of $\boldsymbol{B}_t$. From branching theory, the system is supercritical if $\rho(\boldsymbol{B}_t) > 1$, critical if $\rho(\boldsymbol{B}_t) = 1$, and subcritical if $\rho(\boldsymbol{B}_t) < 1$. Thus epidemic growth is determined by the spectral radius rather than by any single $R_{k,t}$.

Connection to classical univariate reproduction numbers. The definition of R_t^{net} generalizes the reproduction number from standard closed-population renewal models. When $S = 1$, the offspring matrix reduces to the scalar $B_t = (R_{1,t})$, and therefore $\rho(B_t) = R_{1,t}$. In this case, the intrinsic source-specific reproduction number coincides

exactly with the reproduction number estimated under univariate renewal models.

More generally, when working with multiple locations jointly, if it is assumed that there is no cross-regional transmission ($\alpha_{s,k} = 0$ for $s \neq k$), then $\boldsymbol{B}_t$ is diagonal and $\rho(\boldsymbol{B}_t) = \max_k R_{k,t}$. Hence, the network reproduction number reduces to the largest regional intrinsic source-specific reproduction number, and regions evolve independently. With spatial coupling, however, $\rho(\boldsymbol{B}_t)$ determines the epidemic threshold, extending the classical renewal framework to a multivariate setting.

Observed effective reproduction number. Closed-population renewal estimators (Wallinga and Teunis, 2004; Cori et al., 2013) are typically interpreted through

$$E(y_{s,t} \mid \mathcal{F}_{t-1}) = R^{\text{eff}}_{s,t} \sum_{l<t} \phi_{t-l} y_{s,l},$$

where $\mathcal{F}_{t-1}$ denotes the incidence history up to time $t-1$ across all locations. Under this model,

$$\lambda_{s,t} = \mu + \lambda^{\text{endo}}_{s,t}, \qquad \lambda^{\text{endo}}_{s,t} = \sum_{k=1}^{S} \sum_{l<t} \alpha_{s,k} R_{k,l} \phi_{t-l} y_{k,l}.$$

Because renewal-based estimators attribute incidence to past observed cases, the observed effective reproduction number corresponds to the endogenous component of transmission. Defining $R^{\text{eff}}_{s,t}$ via

$$\lambda^{\text{endo}}_{s,t} = R^{\text{eff}}_{s,t} \sum_{l<t} \phi_{t-l} y_{s,l},$$

implies

$$R^{\text{eff}}_{s,t} = \frac{\sum_{k=1}^{S} \alpha_{s,k} \sum_{l<t} R_{k,l} \phi_{t-l} y_{k,l}}{\sum_{l<t} \phi_{t-l} y_{s,l}}.$$

When $\sum_{l<t} \phi_{t-l} y_{s,l} = 0$, we define $R^{\text{eff}}_{s,t} = 0$ by convention.

Separating the $k = s$ term from imported contributions yields an additive decompo-

sition of observed effective transmission into local and imported components:

$$R_{s,t}^{\text{eff}} = R_{s,t}^{\text{local}} + R_{s,t}^{\text{import}},$$

where

$$R_{s,t}^{\text{local}} = \frac{w_s \sum_{l<t} R_{s,l} \phi_{t-l} y_{s,l}}{\sum_{l<t} \phi_{t-l} y_{s,l}} \quad \text{and} \quad R_{s,t}^{\text{import}} = \frac{\sum_{k \neq s} \alpha_{s,k} \sum_{l<t} R_{k,l} \phi_{t-l} y_{k,l}}{\sum_{l<t} \phi_{t-l} y_{s,l}}.$$

When $\sum_{l<t} \phi_{t-l} y_{s,l} = 0$, we define $R_{s,t}^{\text{local}} = 0$ and $R_{s,t}^{\text{import}} = 0$ by convention.

Unlike the source-specific $R_{k,t}$, both components, the effective and local reproduction numbers, are backward-looking quantities that measure contributions to explaining current cases and can be viewed as a weighted average of past reproduction numbers. For real-time epidemic monitoring, the source-specific $R_{k,t}$ is more informative because it reflects current transmissibility, while the observed effective reproduction number lags this signal.

When importation is negligible (i.e., $w_s \approx 1$ for all locations), the observed effective reproduction number closely approximates the intrinsic source-specific reproduction number, and standard methods provide accurate estimates. However, when cross-location transmission is substantial, standard methods conflate local transmissibility with importation. A location may appear to have an observed effective reproduction number near one even when local transmissibility is well below one, if importation sustains the case count. Therefore, this decomposition has practical implications for intervention design. If standard methods suggest a reproduction number near one for a given location, one might conclude that local transmission is barely controlled. Our decomposition, however, reveals whether this reflects genuine local transmissibility or sustained importation, and the two scenarios call for different responses. For instance,

when the local component is high, local measures such as social distancing or masking can be considered. When the imported component dominates, mobility restrictions or cross-boundary coordination become more relevant.

3.2.3. Identifiability of the transmission matrix and reproduction numbers

The intensity depends on the spatial weights and the reproduction numbers only through their product $\alpha_{s,k}R_{k,t}$, so the two cannot be recovered from separate parts of the data and must be estimated jointly. The column-stochastic constraint $\sum_{s=1}^{S} \alpha_{s,k} = 1$ fixes the overall scale of each column. This is what allows $R_{k,t}$ to be read as the total expected number of secondary cases generated by a case at location k, and $\alpha_{s,k}$ as the share of those cases allocated to location s.

Fixing the scale does not remove all ambiguity in finite samples. Because the likelihood sees only the observed incidence, several distinct mechanisms can leave a similar imprint on the data. Genuine exportation from one location to another, synchronized epidemic waves driven by a common external shock, and mobility or reporting patterns that the model does not represent can all produce comparable lagged dependence between locations. Exportation and synchronized waves are the hardest to tell apart, since a driver shared across locations produces lagged co-movement much like exportation, with both leaving correlated increases in incidence separated by the assumed lag structure. A further ambiguity is internal to the parameterization, because the reproduction numbers $R_{k,t}$, the allocation weights $\alpha_{s,k}$, and the baseline rate μ all enter the same conditional mean and can partly substitute for one another over short

windows, so a change in one can be partly absorbed by the others.

For these reasons the estimated $\boldsymbol{\alpha}$ is best read as a model-based allocation of secondary incidence under the assumed lag distribution, not as a measurement of physical transmission between locations. A large weight $\alpha_{s,k}$ indicates that past incidence at location k helps explain later incidence at location s, which is consistent with a source–sink relationship but is not by itself evidence that infections moved from k to s. Distinguishing statistical association from physical transmission would require information beyond case counts, such as human mobility, contact tracing, pathogen genomic data, or interventions that alter transmission between specific locations.

§ 3.3. Bayesian inference

The posterior distribution involves time-varying latent states $\{r_{k,t}\}$ and $\{z_{s,t}\}$, a structured spatial weight matrix $\boldsymbol{\alpha}$ with horseshoe priors and the baseline intensity μ. A single-site Gibbs sampler would mix poorly in this setting because the latent reproduction number trajectories $\{r_{k,t}\}$ are strongly correlated across time and the horseshoe prior induces complex dependence among the spatial parameters within each column of $\boldsymbol{\alpha}$. We address these challenges with a block MCMC sampler that groups parameters according to the conditional independence structure of the model.

Two non-centered parameterizations are introduced. First, the reproduction number at location k follows a random walk $r_{k,t} = r_{k,t-1} + \sqrt{W}\,\eta_{k,t}$ with $\eta_{k,t} \sim N(0,1)$, so that $r_{k,t} = \sqrt{W}\sum_{j=0}^{t}\eta_{k,j}$. Rather than sampling the correlated trajectory $\{r_{k,t}\}$ directly, we sample the disturbances $\boldsymbol{\eta}_k = (\eta_{k,0},\ldots,\eta_{k,T-1})'$, which improves sampling efficiency. Second, the off-diagonal deviations in the spatial weight parameterization are written

as $\theta_{s,k} = \tau_k \tilde{\gamma}_{s,k} \varepsilon_{s,k}$ with $\varepsilon_{s,k} \sim N(0,1)$, where τ_k is the column-specific global shrinkage and $\tilde{\gamma}_{s,k}$ is the local shrinkage parameter from the horseshoe prior. Rather than sampling $\theta_{s,k}$ directly, we sample the $\varepsilon_{s,k}$s, which avoids the pathological geometry that arises when the centered parameterization concentrates near zero with occasional escapes to large values (Betancourt and Girolami, 2013).

The column-stochastic constraint $\sum_s \alpha_{s,k} = 1$ implies that, although the likelihood couples all columns of the transmission matrix through the intensity $\lambda_{s,t}$, the contribution of a given source location k enters only through the corresponding column $\alpha_{\cdot,k}$. Therefore, conditional on the reproduction numbers and the remaining columns, the spatial parameters associated with column k can be updated independently. This motivates a column-wise blocking strategy in the MCMC algorithm.

Algorithm 6 summarizes the sampling scheme. Each iteration consists of four blocks, described in turn below.

Algorithm 6: Blocked MCMC sampling for the network Hawkes-type model.

Sample zero-inflation indicators $\{z_{s,t}\}$ via forward-filtering backward-sampling;
Sample zero-inflation coefficients $\{\beta_0, \beta_1\}$ via HMC;
for $k = 1$ **to** S **do**
  Sample spatial network parameters $\{\varepsilon_{\cdot,k}, \gamma_{\cdot,k}, \tau_k, \omega_k\}$ via HMC, where $\varepsilon_{s,k}$ are the standardized deviations in the non-centered horseshoe parameterization $\theta_{s,k} = \tau_k \tilde{\gamma}_{s,k} \varepsilon_{s,k}$;
  Sample reproduction number disturbances $\{\eta_{k,t}\}_{t=0}^{T-1}$ via a MH step with a Gaussian proposal derived from a linearized likelihood, where $\eta_{k,t}$ are the standardized innovations in the random walk $r_{k,t} = r_{k,t-1} + \sqrt{W}\,\eta_{k,t}$;
Sample baseline intensity μ via HMC;

3.3.1. Zero-inflation indicators and coefficients

Sampling details for the zero-inflation indicators $\{z_{s,t}\}$ and coefficients $\{\beta_0, \beta_1\}$ are provided in § B.1.1. The latent indicators are updated independently by location using forward-filtering backward-sampling, which yields a draw of each $\{z_{s,t}\}_{t=1}^{T}$ trajectory. The coefficients (β_0, β_1) are then updated jointly via HMC, targeting the logistic regression posterior under Gaussian priors.

3.3.2. Column-wise HMC for spatial network parameters

For each source location k, we jointly update the retention rate w_k, the standardized deviation $\{\varepsilon_{s,k}\}_{s\neq k}$, and the associated horseshoe shrinkage parameters $\{\gamma_{s,k}\}_{s\neq k}$ and τ_k. This joint update is needed because changing one deviation $\varepsilon_{s,k}$ affects the normalized weights at all destinations through U_k.

HMC operates on the unconstrained vector

$$\boldsymbol{\xi}_k = (\{\varepsilon_{s,k}\}_{s\neq k}, \operatorname{logit}(w_k), \{\log \gamma_{s,k}\}_{s\neq k}, \log \tau_k)',$$

where the diagonal terms are omitted because $\alpha_{k,k} = w_k$ is parameterized directly. The log posterior target is

$$\begin{aligned}\log p(\boldsymbol{\xi}_k \mid \cdot) \propto & \sum_{s,t} z_{s,t} \log \operatorname{Pois}(y_{s,t} \mid \lambda_{s,t}) + a_w \log w_k + b_w \log(1 - w_k) \\ & + \sum_{s\neq k} \left[\log N(\varepsilon_{s,k} \mid 0, 1) + \log p(\gamma_{s,k}) + \log \gamma_{s,k}\right] \\ & + \log p(\tau_k) + \log \tau_k, \end{aligned} \tag{3.3.1}$$

where $a_w \log w_k + b_w \log(1 - w_k)$ combines the $\operatorname{Beta}(a_w, b_w)$ prior on w_k with the

logit Jacobian, and $\log \gamma_{s,k}$ and $\log \tau_k$ are Jacobian corrections for the log-transformed half-Cauchy priors from § 3.2. Because $\alpha_{s,k}$ depends only on column k, the gradients of Eq. (3.3.1) are available in closed form; their expressions are given in § B.1.2.

3.3.3. Block update for reproduction number trajectories

We update the entire disturbance vector $\boldsymbol{\eta}_k = (\eta_{k,0}, \eta_{k,1}, \ldots, \eta_{k,T-1})'$ jointly using a MH step with a Gaussian proposal constructed from a linearized Poisson likelihood. The reproduction number trajectory is then recovered from $r_{k,t} = \sqrt{W} \sum_{j=0}^{t} \eta_{k,j}$.

We linearize $h(r_{k,l})$ around the current iterate and use a moment-matched Gaussian approximation to the Poisson likelihood to construct the proposal:

$$h(r_{k,l}) \approx h(r_{k,l}^{(i)}) + h'(r_{k,l}^{(i)})(r_{k,l} - r_{k,l}^{(i)}).$$

Under these approximations, $\lambda_{s,t}$ becomes linear in $\boldsymbol{\eta}_k$.

Substituting the linearized link function into the intensity and isolating the contribution from source k yields

$$\lambda_{s,t}^{(i)} = \mu_{s,t}^{(i)} + \sqrt{W}\, \alpha_{s,k} \sum_{j=0}^{t-1} c_{k,j,t}^{(i)}\, \eta_{k,j}, \tag{3.3.2}$$

where $\mu_{s,t}^{(i)}$ collects terms not involving $\boldsymbol{\eta}_k$, and

$$c_{k,j,t}^{(i)} = \sum_{l=j}^{t-1} h'(r_{k,l}^{(i)}) \phi_{t-l}\, y_{k,l}.$$

Stacking over time gives $\boldsymbol{\lambda}_s^{(i)} = \boldsymbol{\mu}_s^{(i)} + \sqrt{W}\, \alpha_{s,k}\, \mathbf{C}_k^{(i)}\, \boldsymbol{\eta}_k$, where $\mathbf{C}_k^{(i)}$ is lower triangular.

Combining the $N(\mathbf{0}, \mathbf{I})$ prior with the linearized likelihood gives the Gaussian proposal

$$\boldsymbol{\eta}_k \mid \cdot \;\sim\; N\left(\boldsymbol{u}_k^{(i)},\; \left[\boldsymbol{\Omega}_k^{(i)}\right]^{-1}\right), \tag{3.3.3}$$

with precision matrix and mean

$$\boldsymbol{\Omega}_k^{(i)} = \mathbf{I} + W \sum_{s=1}^{S} \alpha_{s,k}^2 \, \mathbf{C}_k^{(i)\prime} \left[\boldsymbol{\Lambda}_s^{(i)}\right]^{-1} \mathbf{C}_k^{(i)},$$

$$\boldsymbol{u}_k^{(i)} = \sqrt{W} \left[\boldsymbol{\Omega}_k^{(i)}\right]^{-1} \sum_{s=1}^{S} \alpha_{s,k} \, \mathbf{C}_k^{(i)\prime} \left[\boldsymbol{\Lambda}_s^{(i)}\right]^{-1} \boldsymbol{d}_s^{(i)},$$

where $\boldsymbol{\Lambda}_s^{(i)} = \operatorname{diag}(\lambda_{s,2}^{(i)}, \ldots, \lambda_{s,T}^{(i)})$ and $\boldsymbol{d}_s^{(i)} = \boldsymbol{y}_s - \boldsymbol{\mu}_s^{(i)}$. We draw $\boldsymbol{\eta}_k^*$ from this Gaussian proposal and accept or reject it with a Metropolis–Hastings step targeting the exact posterior under the Poisson likelihood.

The precision matrix $\boldsymbol{\Omega}_k^{(i)}$ is the sum of the identity and a product of lower-triangular matrices, and its Cholesky factor can be computed in $O(T^2)$ operations per source location. Since each source column is updated independently, the total cost per MCMC iteration scales as $O(ST^2)$ for the reproduction number block.

3.3.4. Baseline intensity

The baseline intensity μ enters $\lambda_{s,t}$ additively and affects all locations and time points. HMC therefore operates on the unconstrained scalar $\log \mu$ targeting

$$\log p(\log \mu \mid \cdot) \propto \log N(\log \mu \mid a_\mu, b_\mu^2) + \sum_{s,t} z_{s,t} \log \operatorname{Pois}(y_{s,t} \mid \lambda_{s,t}),$$

with the Jacobian correction for the log transformation on μ.

§ 3.4. Simulation studies

We report two simulation studies targeting distinct inferential questions. Study 1 evaluates recovery of epidemiologically interpretable quantities under heterogeneous retention, including $R_{k,t}$ trajectories, network-level transmissibility through R_t^{net}, and decomposition into local and imported components. Study 2 evaluates robustness and predictive performance under structural variation, with emphasis on network specification, mobility covariates, and prior choice. To focus on spatial transmission effects, the data generating processes below do not include zero inflation; a dedicated zero-inflation experiment is provided in § B.4.

3.4.1. Simulation design

All simulations generate count data $\{y_{s,t}\}$ for S locations over $T = 200$ time points from a Poisson model with intensity

$$\lambda_{s,t} = \mu + \sum_{k=1}^{S} \sum_{l=1}^{t-1} \alpha_{s,k} R_{k,l} \phi_{t-l} y_{k,l},$$

where $\mu = 2$, $R_{k,t} = h(r_{k,t})$, $h(\cdot)$ is a softplus function, and $r_{k,t}$ follows a random walk with innovation variance $W = 0.01$. The lag distribution $\{\phi_l\}$ is a discretized log-normal with mean 4.7 days and standard deviation 2.9 days, truncated at $L = 30$ days and renormalized to sum to one. The spatial weights $\alpha_{s,k}$ are column-stochastic, and the diagonal element $\alpha_{k,k} = w_k$ gives the retention rate. When mobility information is used, $m_{s,k}$ encodes pairwise mobility between locations and $\tilde{\theta}_{s,k}$ captures deviations from the mobility-based pattern. When mobility information is omitted, off-diagonal weights depend only on $\tilde{\theta}_{s,k}$.

Prior specifications are shared across simulation studies as follows: $w_k \sim \text{Beta}(1, 1)$, the horseshoe slab scale is $c^2 = 4$, and $\log(\mu) \sim N(0, 9)$. W is fixed at its true value for all simulation studies (for the Covid-19 data analysis, W is selected using a grid-based analysis as reported in § B.6.1). HMC step sizes and diagonal mass matrices are adapted during burn-in via dual averaging (Hoffman and Gelman, 2014) with a target acceptance rate of 0.75 and a target integration time of $T = 2.0$. After burn-in, step sizes are finalized and leapfrog counts are determined by the target integration time divided by the adapted step size. MCMC inference uses 30,000 iterations with 5,000 burn-in and thinning to every 5th sample. Convergence is assessed via trace plots and Gelman-Rubin statistics, with parameters achieving $\hat{R} < 1.1$.

We compare four approaches in the simulation studies: the proposed network Hawkes-type model, a univariate Hawkes-type model fit separately to each location (Tang and Prado, 2026), EpiEstim (Cori et al., 2013), and Wallinga–Teunis (WT; Wallinga and Teunis, 2004). The network Hawkes-type model is fit with and without mobility covariates, and with either regularized horseshoe or Gaussian priors on the off-diagonal deviations $\tilde{\theta}_{s,k}$. EpiEstim and WT are used in Study 1 and are also applied to aggregated counts for comparison with the network reproduction number.

Performance is assessed using the mean absolute error (MAE) for reproduction number estimation, the continuous ranked probability score (CRPS) for probabilistic forecasting at horizons of 1, 5, and 10 steps, and the Frobenius norm between the posterior mean and true spatial weight matrix.

3.4.2. Study 1: parameter estimation with heterogeneous retention

Study 1 examines parameter recovery under heterogeneous retention in a five-location network with $w_k \in \{0.9, 0.7, 0.5, 0.3, 0.3\}$ for locations A–E and no mobility covariates, so off-diagonal structure is driven only by $\tilde{\theta}_{s,k}$. This yields a clear source–sink pattern: location A retains 90% of secondary cases locally, locations D and E retain 30%, and locations B and C are intermediate. We assess recovery of location-specific transmissibility, network-level transmissibility, and the local/imported decomposition implied by the fitted branching structure. Figure B.1 shows the case counts and spatial weight matrix for this study. The simulated trajectories show a major peak with a sustained tail, and mean incidence is highest in A and B (52 and 77), compared with C, D, and E (30, 33, and 27), consistent with accumulation driven by retention versus redistribution across the network.

Figure 3.1 compares the network reproduction number $R_t^{\text{net}} = \rho(\mathbf{B}_t)$ with reproduction numbers obtained by applying WT and EpiEstim to case counts aggregated over all five locations. The network Hawkes-type posterior mean tracks the true spectral radius closely and the 95% credible intervals maintain full coverage over time in this simulation. All methods detect the major peak near $t = 50$, but the aggregated WT and EpiEstim curves are consistently upward biased when $R_t^{\text{net}} < 1$, most clearly for $t \approx 100$–150. This discrepancy is expected: in the aggregated renewal formulations, the baseline immigration is absorbed into apparent onward transmission, whereas the network Hawkes-type model separates immigration from secondary spread.

Figure 3.2 compares location-specific reproduction number estimates. The left panels compare the intrinsic source-specific reproduction number $R_{k,t}$ with WT and EpiEstim estimates for each location, while the right panels decompose the effective reproduction

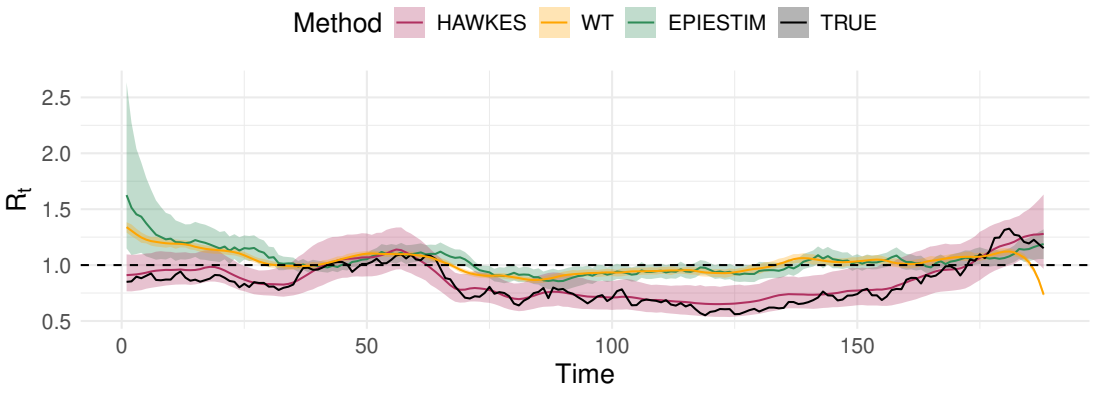
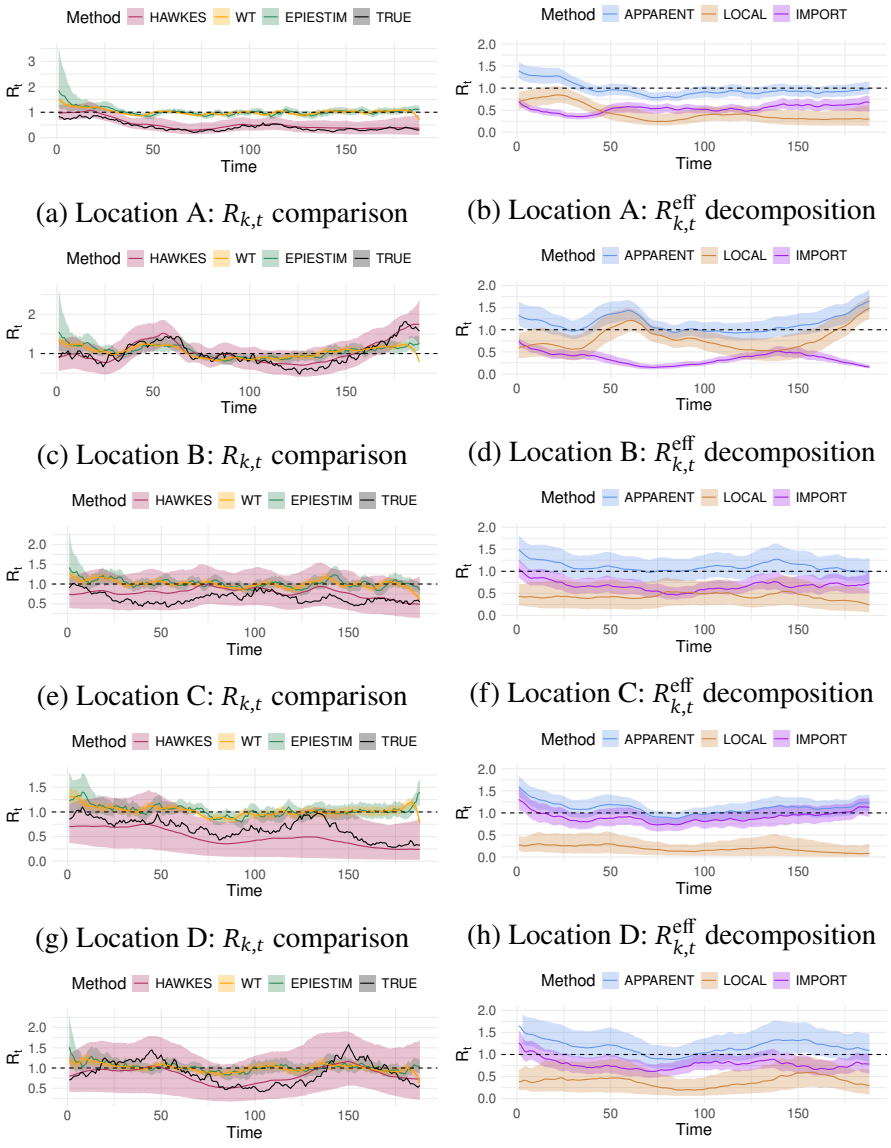


Figure 3.1.: Network reproduction number R_t^{net}: true values (black), posterior mean and 95% credible interval from the network Hawkes-type model, and reproduction numbers with 95% confidence intervals estimated by WT and EpiEstim applied to aggregated case counts across all five locations.

number $R_{k,t}^{\text{eff}}$ into local and imported components. Posterior means for $R_{k,t}$ track the true curves well across locations, with narrower credible intervals for locations A and B, which have higher incidence. Differences between $R_{k,t}$ and the univariate renewal estimates vary with retention. In locations with high retention, the estimands are closer because local transmission dominates; in locations with low retention, importation contributes more strongly to $R_{k,t}^{\text{eff}}$, so WT and EpiEstim lie above the intrinsic $R_{k,t}$. Thus, the univariate methods track an effective closed-population quantity, whereas the network Hawkes-type model separates intrinsic transmission from importation.

Figure 3.3 shows that the model recovers the spatial transmission structure, while retention rate summaries are reported in Table B.1. All true ω_k values fall within their 95% credible intervals. Off-diagonal weights are also well recovered: source–destination pairs with high and low transmission are separated in the posterior mean matrix. Uncertainty is larger for columns D and E because lower incidence provides less information about how exported offspring are distributed across destinations. Together,



(a) Location A: $R_{k,t}$ comparison

(b) Location A: $R^{\text{eff}}_{k,t}$ decomposition

(c) Location B: $R_{k,t}$ comparison

(d) Location B: $R^{\text{eff}}_{k,t}$ decomposition

(e) Location C: $R_{k,t}$ comparison

(f) Location C: $R^{\text{eff}}_{k,t}$ decomposition

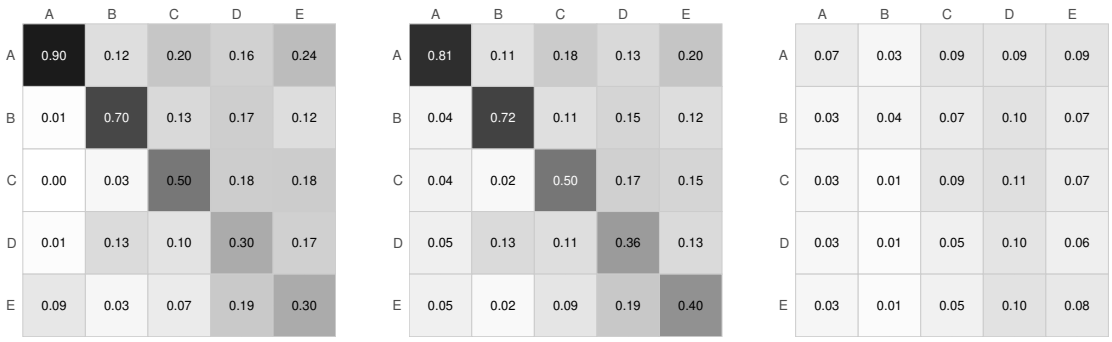
(g) Location D: $R_{k,t}$ comparison

(h) Location D: $R^{\text{eff}}_{k,t}$ decomposition

(i) Location E: $R_{k,t}$ comparison

(j) Location E: $R^{\text{eff}}_{k,t}$ decomposition

Figure 3.2.: Reproduction number estimation for a simulated network Hawkes-type process with five locations and heterogeneous retention rates. Left panels compare the intrinsic source-specific reproduction number $R_{k,t}$ (posterior mean and 95% credible interval) with the true values and univariate benchmark estimates from WT and EpiEstim. Right panels decompose the observed effective reproduction number $R^{\text{eff}}_{k,t}$ into local and imported contributions. Locations with low retention (D, E) have higher importation than locations with high retention (A, B).



(a) True (b) Posterior mean (c) Posterior SD

Figure 3.3.: Estimation of the spatial weight matrix $\boldsymbol{\alpha} = (\alpha_{s,k})$ in Study 1. The model recovers both diagonal retention rates and off-diagonal transmission structure.

these results recover source–sink patterns that univariate analyses cannot identify. The recovered source–sink structure illustrates how the fitted model can support hypothetical intervention scenarios. Locations D and E, with retention rates near 0.3–0.4, export most of their secondary cases to the network, so a hypothetical reduction in transmission there would affect multiple destinations. By contrast, location A, with retention near 0.8, is largely self-sustaining, so analogous perturbations primarily affect local case counts.

3.4.3. Study 2: model structure comparison

Study 2 examines when network structure, mobility covariates, and prior choice improve inference relative to simpler alternatives. Unlike Study 1, which used a single realization with heterogeneous retention, Study 2 varies a common retention rate across simulation settings to isolate how retention governs the value of each modeling component. Each setting is replicated three times with independent random seeds, and results are averaged across replications. These replications are intended to check that the qualitative patterns

are not driven by a single random seed, rather than to provide a full Monte Carlo study.

We compare four models: a univariate Hawkes model fit separately to each location, a horseshoe network model that estimates the full spatial weight matrix, a horseshoe + mobility variant that adds a mobility covariate to the off-diagonal weights, and a Gaussian network model that replaces the horseshoe prior with independent Gaussian shrinkage.

Study 2a: Network with distance effect

In this scenario, the data generating process includes a mobility covariate in the off-diagonal weights, $\upsilon_{s,k} = m_{s,k} + \tilde{\theta}_{s,k}$, where $m_{s,k}$ encodes pairwise mobility and $\tilde{\theta}_{s,k}$ captures deviations from that baseline. We use $S = 5$ locations and vary a common retention rate across 30%, 70%, 90%, and 95% to assess how the value of network structure changes as transmission becomes more local.

Table 3.1 shows two main patterns. First, across these simulations network models improve reproduction number estimation relative to the univariate benchmark, with the largest gains when cross-location transmission is strongest. Forecasting gains are more selective: they are largest at low retention and longer horizons, but become negligible when retention is 95% and transmission is mostly local. Second, adding mobility primarily improves spatial weight recovery rather than forecasting. The Frobenius norm decreases most at low retention, with smaller gains as retention rises. This pattern suggests that when off-diagonal transmission is weak, incidence data provide limited information about cross-location weights, so mobility contributes only modest additional value.

Table 3.1.: Network with mobility effect in the data generating process ($S = 5$, $T = 200$). In these simulations, network models improve reproduction number estimation, while mobility mainly improves spatial weight recovery.

Retention	Model	MAE($R_{k,t}$)	CRPS_1	CRPS_5	CRPS_{10}	Frobenius
30%	Univariate	0.379	2.79	4.15	4.24	—
	Horseshoe	0.129	2.95	**3.61**	**3.62**	0.543
	Horseshoe + Mob.	**0.128**	**2.91**	3.62	3.63	**0.499**
70%	Univariate	0.141	19.5	23.7	36.7	—
	Horseshoe	0.079	16.6	21.5	29.4	0.380
	Horseshoe + Mob.	**0.077**	**16.4**	**21.3**	**29.2**	**0.364**
90%	Univariate	0.449	**2.91**	3.05	3.54	—
	Horseshoe	0.067	3.18	2.98	3.20	**0.523**
	Horseshoe + Mob.	**0.065**	3.14	**2.97**	**3.19**	0.528
95%	Univariate	0.162	**3.70**	**3.44**	4.37	—
	Horseshoe	**0.056**	3.81	3.48	4.36	0.575
	Horseshoe + Mob.	**0.056**	3.83	3.52	**4.34**	**0.546**

Study 2b: Robustness to misspecification

We assess robustness to two forms of misspecification. For covariate misspecification, we generate data from a network model with no mobility effect and fit models both with and without the mobility covariate. For prior misspecification, we replace the horseshoe prior on $\tilde{\theta}_{s,k}$ with an independent Gaussian prior. Detailed numerical results are reported in Table S2. They show that including an unnecessary mobility term causes little harm, while replacing the horseshoe prior with Gaussian shrinkage mainly worsens forecast calibration when retention is low.

Study 2c: Scaling to more locations

The final scenario examines how the value of mobility covariates changes with network size. As the number of locations increases, the off-diagonal spatial weight parameter space grows quadratically, increasing the potential value of structural regularization. We repeat the Study 2a design with $S = 8$ locations and retention rates of 30%, 70%, and 95%.

Table 3.2.: Network with mobility effect ($S = 8$, $T = 200$). Mobility covariates become more useful as network size grows, especially for spatial weight recovery.

Retention	Model	MAE($R_{k,t}$)	CRPS_1	CRPS_5	CRPS_{10}	Frobenius
30%	Univariate	0.180	4.37	5.31	6.83	—
	Horseshoe	0.176	4.08	5.04	5.83	0.827
	Horseshoe + Mob.	**0.173**	**3.82**	**4.93**	**5.66**	**0.704**
70%	Univariate	0.115	8.32	10.4	14.2	—
	Horseshoe	0.108	7.33	9.53	11.9	0.533
	Horseshoe + Mob.	**0.094**	**7.22**	**9.42**	**11.8**	**0.516**
95%	Univariate	0.074	2.50	2.54	3.14	—
	Horseshoe	**0.073**	**2.26**	2.43	3.07	0.890
	Horseshoe + Mob.	0.078	2.28	**2.43**	**3.05**	**0.864**

Table 3.2 supports the same overall conclusion at larger scale: mobility covariates become more useful as spatial dimension grows. The largest gains appear in spatial weight recovery, with more modest improvements in forecasting and reproduction number estimation. This suggests that mobility covariates provide increasingly useful information as the network becomes larger and the spatial parameter space expands.

3.4.4. Summary

The simulation study suggests several practical conclusions. First, network structure substantially improves reproduction number estimation whenever cross-location trans-

mission is non-negligible. Second, mobility covariates mainly improve spatial weight recovery and interpretation rather than forecasting, and their value increases with network size. Third, the horseshoe prior contributes most through forecast calibration, especially at low to moderate retention. Finally, in these simulations, the model is reasonably robust to both covariate and prior misspecification, so including mobility structure and horseshoe regularization has limited downside when the true data generating process is uncertain.

Finally, the simulation in § B.4 shows that the zero-inflation component improves reproduction number estimation and forecast calibration when reporting artifacts are present (28% MAE improvement and 13% long-horizon CRPS improvement) and causes negligible harm when they are absent.

3.4.5. Sensitivity analyses

We report how the model's outputs change when the components that are fixed or tuned rather than estimated, namely the lag kernel, the innovation variance W of the random walk, the priors, and the gain function, are varied, using the data generating process of Study 1. The full results are in § B.5.1 to B.5.4 and are summarized here.

When the serial interval mean is varied from 2.5 to 7.5 days, and its dispersion is varied at fixed mean, the excitation structure changes little. The off-diagonal transmission weights change by about 0.03 in root mean square, the ordering of locations by net flow changes by at most a single adjacent transposition, at the shortest lag, and the network reproduction number changes by at most 0.08. The local and imported reproduction numbers are deterministic functions of the transmission matrix and the reproduction trajectories, so their variation over the grid is of the same order.

The innovation variance W controls the smoothness of the reproduction trajectories and the width of their credible intervals. Over the grid, the interval width increases monotonically with W, and coverage rises toward the nominal 95% level near the data generating value $W = 0.01$; the mean absolute error of $R_{k,t}$ is smallest at that value. The ordering of locations by net flow and the local and imported decomposition stay close to their values at $W = 0.01$ and move further as W departs from it. Forecasting performance as a function of W is examined on the application data in § B.6.1.

When the prior hyperparameters are varied one at a time, across the retention, horseshoe slab, global shrinkage, baseline, and activation priors, the ordering of locations by net flow is unchanged, the network reproduction number moves by at most 0.02, the one-step forecast score stays within a few percent of the default, and the posterior mean of the baseline intensity moves by at most 0.06 on a scale where $\mu = 2$. The two quantities that a prior parameterizes directly respond to it. The retention weights move under the Beta prior on ω_k, and the posterior median of the global shrinkage scale τ_k scales with the prior scale τ_0, roughly halving and doubling with it. A change to any one prior leaves the other targets close to the default.

The reproduction number enters through the gain function, $R_{k,t} = h(r_{k,t})$. When data are generated and fit under the same gain function, the softplus and exponential gain functions both recover their own data generating truth, with comparable error in the transmission matrix, ordering of locations by net flow, and coverage of $R_{k,t}$. Fitting softplus-generated data under the exponential gain function leaves the ordering unchanged and gives comparable reproduction number and transmission matrix errors, so the recovered network is largely insensitive to the choice of gain function at estimation. We use the softplus gain function for numerical stability, since the exponential gain

function can produce very large reproduction numbers.

§ 3.5. Application: COVID-19 across California counties

We apply the network Hawkes-type model to COVID-19 surveillance data from ten California counties observed from July 2020 through November 2021. The counties include five from the San Francisco Bay Area (Alameda, Contra Costa, San Francisco, San Mateo, and Santa Clara) and five from the Central Valley or nearby areas linked by commuting (Merced, Sacramento, San Joaquin, Santa Cruz, and Stanislaus). We assess whether the model yields substantively different surveillance conclusions from standard reproduction number analyses that treat each county independently, by separating local transmission from importation across a regional network.

3.5.1. Data and model specification

Daily confirmed COVID-19 case counts were obtained from the California Department of Public Health and converted to cases per 100,000 population using 2019 American Community Survey estimates. The study period comprises $T = 517$ days, with cumulative incidence ranging from approximately 11,000 (San Francisco) to 28,000 (Merced) per 100,000 population.

Pre-pandemic commuter flow data from the 2016 to 2020 American Community Survey provide the mobility covariate matrix $\mathbf{M} = (m_{s,k})$. Off-diagonal entries $m_{s,k}$ give the share of county k's out-commuters traveling to county s, normalized so that

$\sum_{s \neq k} m_{s,k} = 1$, while diagonal entries are set to $m_{k,k} = 1$ as a reference level for within-county transmission. Because these commuter flows are measured before the pandemic, they need not reflect movement during the study period, when mobility changed substantially. We therefore treat $\mathbf{M}$ as a structural baseline for spatial connectivity rather than as contemporaneous movement, and let departures during 2020–2021 be captured through the deviation parameters $\theta_{s,k}$.

The lag distribution $\{\phi_l\}$ is obtained by discretizing a log-normal density with mean 4.7 days and standard deviation 2.9 days, truncated at $L = 30$ days and normalized to sum to one. We fix this distribution to avoid confounding temporal transmission patterns with the spatial allocation matrix and the time-varying reproduction trajectories. The reproduction number random walk variance is fixed at $W = 0.005$; sensitivity results for this choice are reported in § B.6.1.

The horseshoe global shrinkage parameter is calibrated following Piironen and Vehtari (2017). Setting the expected number of non-zero deviations from the mobility baseline to $p_0 = 2.8$ per source column and using the data scale $\hat{\sigma}_y = 0.82$ gives $\tau_0 = p_0/(S-1-p_0) \cdot \hat{\sigma}_y/\sqrt{T} = 0.016$. The remaining priors are $w_k \sim \text{Beta}(1,1)$, $c^2 = 4$, $\log(\mu) \sim N(0,9)$, and independent $N(0,9)$ priors for the zero-inflation coefficients, with latent indicators initialized at $z_{s,0} = 0$.

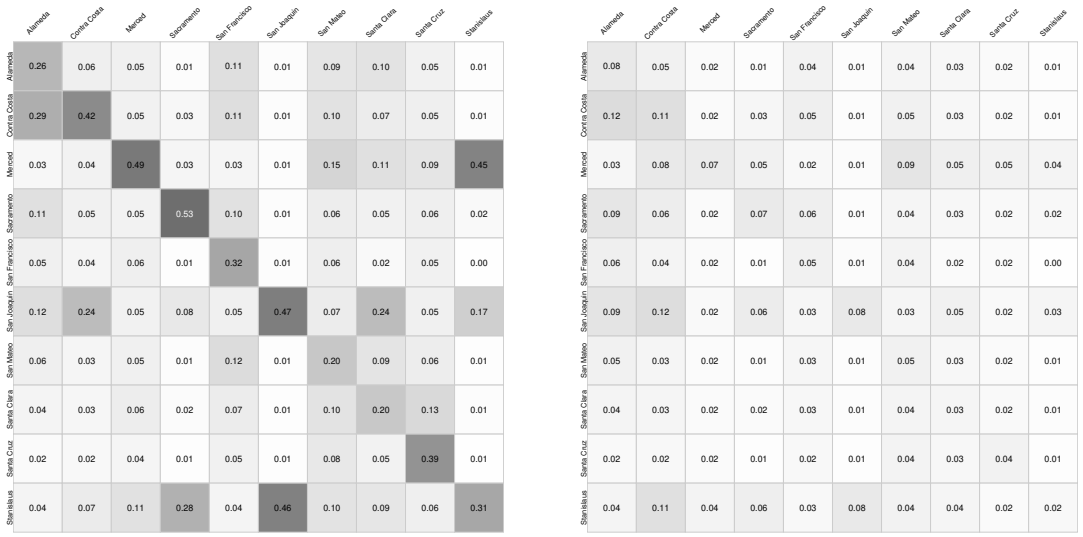
Posterior inference uses 35,000 MCMC iterations, with 10,000 discarded as burn-in and every fifth draw retained thereafter. HMC step sizes and diagonal mass matrices are adapted during burn-in using dual averaging (Hoffman and Gelman, 2014) with target acceptance rate 0.75 and target integration time 2. Convergence was assessed using trace plots and Gelman–Rubin diagnostics, with all reported parameters satisfying $\hat{R} < 1.1$. Full analysis required approximately 65 hours on a laptop with an Apple M1

Pro processor and 32 GB of memory.

3.5.2. Transmission network, decomposition and forecasting results

We first summarize the inferred spatial transmission network, then examine how the decomposition of reproduction numbers changes county-level interpretation, and finally assess forecasting and scenario-based planning implications.

Estimated spatial transmission network



(a) Posterior mean. (b) Posterior standard deviation.

Figure 3.4.: Posterior mean (a) and standard deviation (b) of the estimated spatial weight matrix $\hat{\alpha}_{s,k}$. Each element $\alpha_{s,k}$ gives the fraction of secondary cases generated by source county k that the fitted model allocates to destination county s. Diagonal entries correspond to retention rates w_k, and larger off-diagonal values indicate a stronger estimated cross-county allocation.

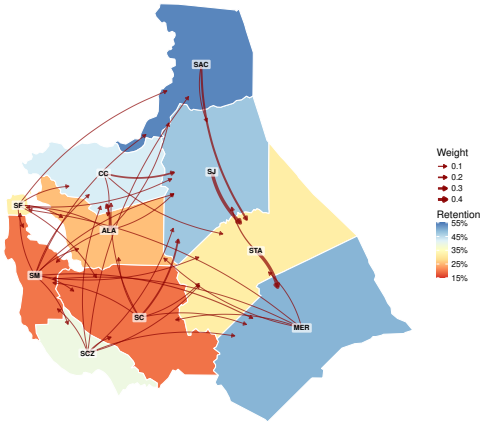
Figure 3.4 presents the estimated spatial weight matrix. Retention rates vary substantially across counties, ranging from 0.20 (San Mateo, Santa Clara) to 0.53 (Sacramento).

Counties with lower retention rates generate a larger fraction of secondary cases in other locations, while those with higher retention rates show more localized transmission. The Bay Area employment centers tend to have lower retention, whereas Central Valley residential counties and the geographically distant Sacramento have higher retention.

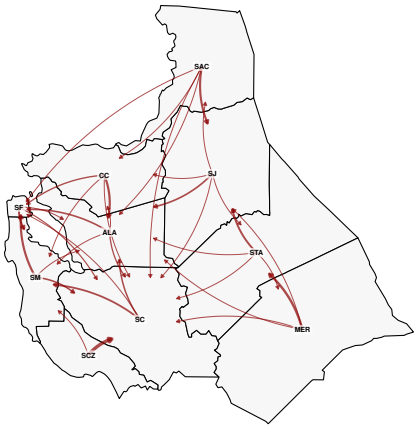
The off-diagonal weights reveal geographic clustering. Central Valley counties (Stanislaus, Merced, San Joaquin) form a strongly connected cluster, with transmission weights from Stanislaus to Merced ($\hat{\alpha}_{\text{Mer,Stan}} = 0.45$) and from San Joaquin to Stanislaus ($\hat{\alpha}_{\text{Stan,SJ}} = 0.46$) among the largest in the network. Bay Area counties similarly cluster, with strong connections from Alameda to Contra Costa ($\hat{\alpha}_{\text{CC,Ala}} = 0.29$) and from San Francisco to San Mateo.

Figure 3.5 compares the inferred transmission network with the pre-pandemic commuter mobility network, providing a visual benchmark for how the estimated transmission pathways align with or depart from baseline mobility patterns. For visual clarity, panel (a) displays only edges with $\alpha_{s,k} > 0.05$, so the map emphasizes dominant pathways while the complete estimated matrix is reported in Figure 3.4.

Several estimated transmission pathways run counter to commuter mobility patterns, as shown in Fig. 3.5. For example, pre-pandemic mobility data indicate more commuters traveling from Contra Costa to Alameda than the reverse, yet the estimated transmission weight from Alameda to Contra Costa ($\hat{\alpha}_{\text{CC,Ala}} = 0.29$) is much larger than in the opposite direction ($\hat{\alpha}_{\text{Ala,CC}} = 0.06$). Because $\alpha_{s,k}$ gives the fraction of secondary cases generated in county k that the model allocates to county s, this asymmetry means that, under the fitted network model, past incidence in Alameda receives a relatively large allocation weight for explaining later incidence in Contra Costa. This is consistent with a source–sink pattern, but it should not be read as direct causal evidence of



(a) Estimated transmission network (edges with $\alpha_{s,k} > 0.05$).



(b) Pre-pandemic commuter mobility from 2016–2020 (edges > 0.05).

Figure 3.5.: Estimated transmission network (a) compared to pre-pandemic commuter mobility from the 2016 to 2020 American Community Survey (b). Panel (a) shows estimated transmission, while panel (b) shows the commuter baseline. Arrows indicate the direction of the estimated transmission allocation (a) or commuter travel (b), with thickness proportional to weight magnitude. Node shading reflects retention rate. Several estimated transmission pathways run counter to commuter direction, a pattern that is consistent with, but does not identify, transmission flowing from workplaces back to residences.

physical transmission from Alameda to Contra Costa. The asymmetry is compatible with exposure occurring disproportionately in employment centers and subsequent infections being realized in residential counties; it is also compatible with synchronized epidemic waves, shared policy shocks, or time-varying mobility not captured by the model, which the incidence data cannot distinguish from one another. The posterior standard deviation map in Figure 3.4 indicates that these dominant directional links are estimated more precisely than weaker off-diagonal edges.

Reproduction number decomposition

The key contribution of the proposed approach in this setting is the decomposition of the observed effective reproduction number into local and imported components. The intrinsic source-specific reproduction number $R_{k,t}$ and retention rate w_k are prospective quantities: they describe how many secondary cases are generated by infections in county k and what fraction remain local. By contrast, the observed effective reproduction number $R^{\text{eff}}_{s,t}$ and its local contribution are retrospective quantities: they describe the origins of cases observed in county s.

We also summarize each county's network role through the net flow of transmission, defined as the difference between total outgoing and incoming transmission weights:

$$\text{Net Flow}_k = \sum_{s \neq k} \alpha_{s,k} - \sum_{j \neq k} \alpha_{k,j}.$$

Positive values indicate net exporters and negative values net importers. Because the spatial weight matrix is column-stochastic ($\sum_s \alpha_{s,k} = 1$), net flow sums to zero across counties. At a broad level, Bay Area counties tend to be net exporters, whereas Central Valley counties tend to be net importers, with San Joaquin and Stanislaus showing the

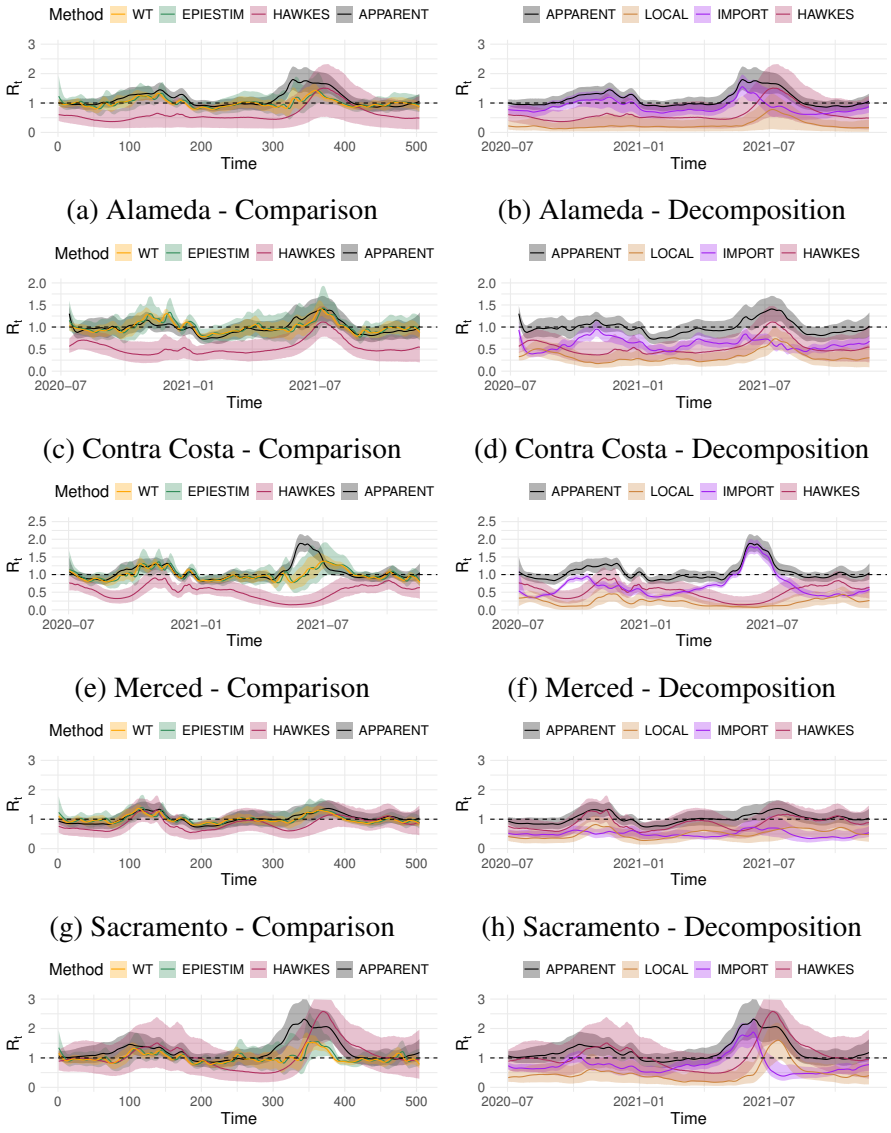
largest negative net flows. This source–sink pattern helps explain why counties with similar observed reproduction numbers can nevertheless play very different roles in regional transmission.

Table 3.3.: Reproduction number decomposition across ten California counties, ordered by descending intrinsic source-specific reproduction number. All quantities are temporal averages over the study period. Retrospective quantities ($\bar{R}_k^{\text{eff}}$ and local percentage) describe the origins of cases appearing in each county, whereas prospective quantities ($\bar{R}_k$ and retention) describe the destinations of transmission originating in each county. Net flow: difference between total outgoing and incoming transmission weights.

	Retrospective		Prospective		Net	Case	Household
County	$\bar{R}_k^{\text{eff}}$	Local%	$\bar{R}_k$	Retention%	Flow	Rate	Size
Santa Clara	1.25	24.40	1.14	19.80	+0.34	14.40	2.97
San Francisco	1.25	37.75	1.10	31.77	+0.39	10.99	2.36
Santa Cruz	1.29	31.60	1.07	39.06	+0.34	14.79	2.73
Stanislaus	1.04	43.27	1.01	31.44	−0.54	26.35	3.09
San Mateo	1.24	20.01	0.93	19.54	+0.35	12.94	2.89
Sacramento	1.02	49.77	0.85	53.43	−0.04	19.74	2.77
Alameda	1.13	21.89	0.79	25.62	+0.26	13.24	2.81
Contra Costa	0.99	37.40	0.68	42.23	−0.15	16.35	2.86
San Joaquin	1.10	21.44	0.55	47.23	−0.53	23.94	3.16
Merced	1.07	22.08	0.65	49.21	−0.44	27.85	3.31

Table 3.3 presents temporal averages of the estimated quantities for all ten counties. Although the observed effective reproduction number remains near 1.0 across counties, the underlying dynamics differ sharply. The intrinsic source-specific reproduction number ranges from 0.55 (San Joaquin) to 1.14 (Santa Clara), and the local contribution to the observed effective reproduction number ranges from 20% (San Mateo) to 50% (Sacramento). This heterogeneity is not visible under closed-population methods.

Counties with higher case rates tend to have lower intrinsic source-specific reproduction numbers. For example, Merced has the highest mean daily new case rate (27.9 per



(a) Alameda - Comparison

(b) Alameda - Decomposition

(c) Contra Costa - Comparison

(d) Contra Costa - Decomposition

(e) Merced - Comparison

(f) Merced - Decomposition

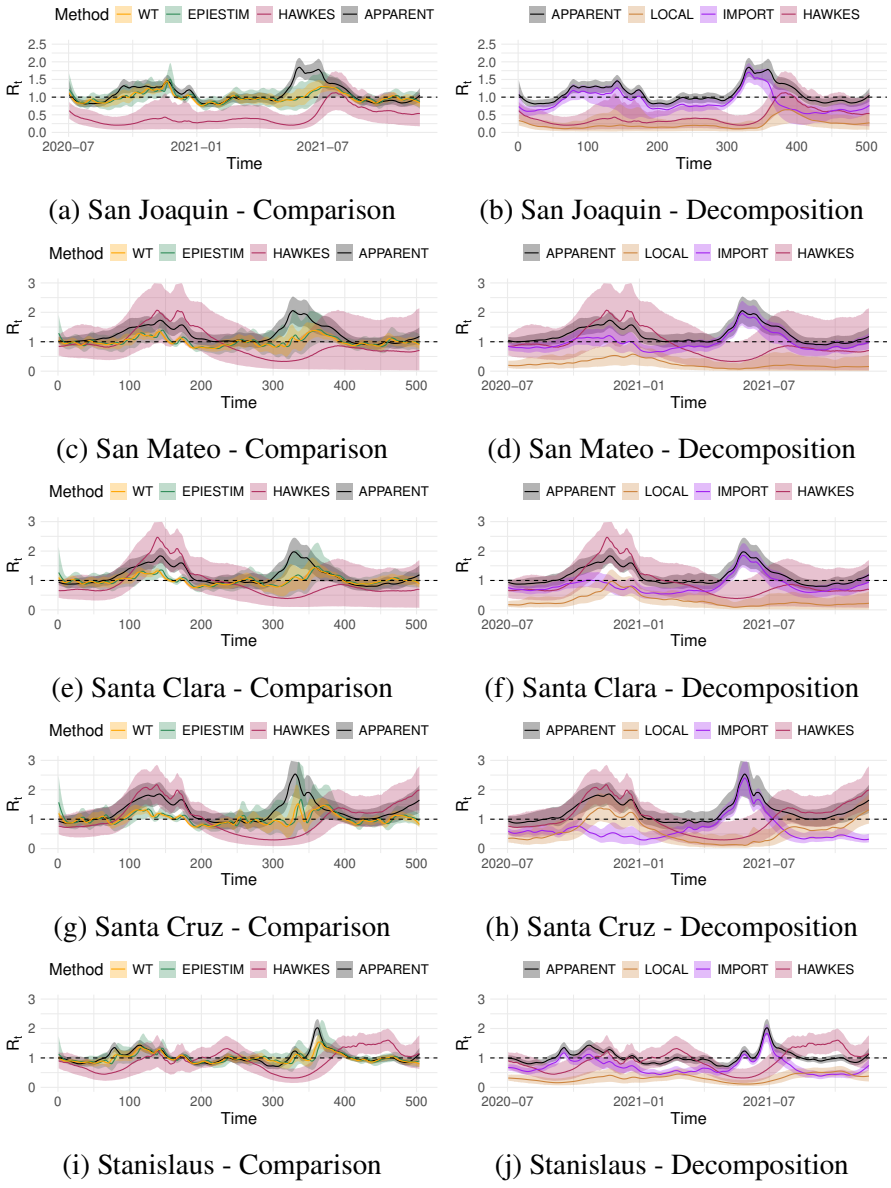
(g) Sacramento - Comparison

(h) Sacramento - Decomposition

(i) San Francisco - Comparison

(j) San Francisco - Decomposition

Figure 3.6.: Reproduction number comparison and decomposition for five counties (alphabetical order, continued in Fig. 3.7). Left panels compare the intrinsic source-specific reproduction number $R_{k,t}$ against the observed effective reproduction number from the network Hawkes-type model and two benchmark methods (EpiEstim and Wallinga–Teunis). Right panels decompose the observed effective reproduction number into local transmission (orange) and importation (purple) components. Solid lines are posterior means with shaded 95% credible intervals.



(a) San Joaquin - Comparison

(b) San Joaquin - Decomposition

(c) San Mateo - Comparison

(d) San Mateo - Decomposition

(e) Santa Clara - Comparison

(f) Santa Clara - Decomposition

(g) Santa Cruz - Comparison

(h) Santa Cruz - Decomposition

(i) Stanislaus - Comparison

(j) Stanislaus - Decomposition

Figure 3.7.: Reproduction number comparison and decomposition for the remaining five counties. Left panels compare the intrinsic source-specific reproduction number $R_{k,t}$ against the observed effective reproduction number from the network Hawkes-type model and two benchmark methods (EpiEstim and Wallinga–Teunis). Right panels decompose the observed effective reproduction number into local transmission (orange) and importation (purple) components. Solid lines are posterior means with shaded 95% credible intervals.

100,000) but a low intrinsic source-specific reproduction number (0.65), whereas San Francisco has the lowest case rate (11.0 per 100,000) but a comparatively high intrinsic source-specific reproduction number (1.10). Across counties, the correlation between $\bar{R}_k$ and mean daily new case rate is -0.52. This negative relationship is consistent with the network structure: counties with high intrinsic transmissibility but low retention generate secondary cases that appear elsewhere, whereas counties with lower intrinsic transmissibility can accumulate cases through importation. Case rates therefore measure where disease appears, while the intrinsic source-specific reproduction number summarizes where the fitted model attributes onward transmission.

The estimates also show a clear source–sink geographic structure. Bay Area counties tend to have higher intrinsic source-specific reproduction numbers, lower retention rates, and positive net flow, whereas Central Valley counties such as San Joaquin and Merced tend to have lower intrinsic source-specific reproduction numbers, higher retention rates, and negative net flow. Sacramento appears more self-contained, with the highest retention rate (53%) and highest local contribution (50%).

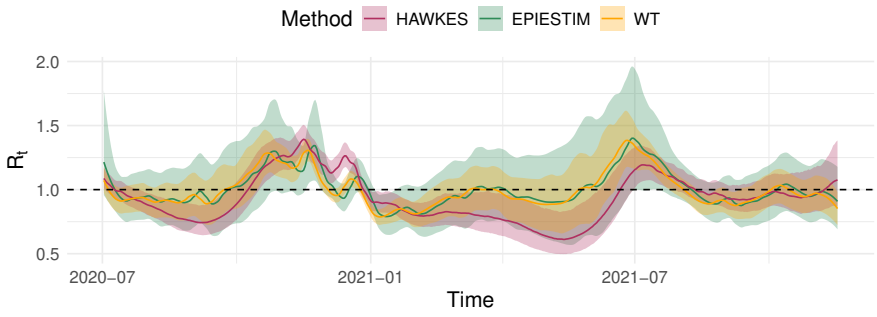


Figure 3.8.: Network reproduction number R_t^{net} over time, defined as the spectral radius of the transmission matrix. The network quantity is compared to reproduction numbers estimated by EpiEstim and Wallinga–Teunis applied to aggregated daily case counts across all ten counties.

While county reproduction numbers vary substantially, the overall epidemic trajectory is governed by the spectral radius of the network transmission matrix. Figure 3.8 shows that the estimated network reproduction number broadly aligns with EpiEstim and Wallinga–Teunis applied to aggregated county counts, although the aggregated estimates sit slightly above the network quantity when it falls below one. The network reproduction number nevertheless cannot replace the county-level decomposition in Table 3.3, which shows where the fitted model attributes onward transmission and where cases accumulate.

Taken together, these results show that decomposition changes epidemiological interpretation in ways that aggregate or closed-population methods cannot recover: counties with similar observed effective reproduction numbers can have very different balances of local transmission and importation. This distinction matters for interpreting hypothetical control scenarios, because perturbations to local transmissibility are most relevant in source counties, whereas counties dominated by importation are more affected by perturbations to cross-county transmission pathways. One possible interpretation of the source–sink pattern is that some infections are acquired in Bay Area employment centers but recorded later in residential counties, producing estimated transmission flows that run opposite to commuter direction. This interpretation is consistent with the combination of higher intrinsic source-specific reproduction numbers in Bay Area counties, larger case burdens in some residential counties, and external evidence that household transmission was an important amplifier of COVID-19 spread (Madewell et al., 2020; Thompson et al., 2021; Chang et al., 2021; Gatto et al., 2020). We emphasize, however, that the model does not identify this mechanism directly, and these comparisons should be viewed as contextual support rather than causal evidence.

Out-of-sample forecasting performance

We evaluate forecasting performance using a rolling-window design with fourteen forecast origins, one at each of the last fourteen days of the training data. At each origin t_0 we fit the model to observations through time t_0 and generate probabilistic forecasts at horizons $h \in \{1, 5, 10\}$ days. All four methods forecast the same daily incidence series on the cases per 100,000 scale, and each reported score is averaged over the fourteen origins and the ten counties, so the scores are comparable across methods. Performance is assessed using the continuous ranked probability score (CRPS) and the empirical coverage of 95% prediction intervals, that is, the fraction of held-out observations falling within the central 95% interval of the posterior predictive distribution. Of the network model's settings, only the innovation variance W is chosen with reference to forecasting performance, while the lag kernel is fixed in advance and the horseshoe scale is calibrated from the scale of the observed counts. The value $W = 0.005$ is selected on an earlier rolling-origin window that does not overlap these fourteen evaluation origins (§ B.6.1), so the reported comparison does not reuse the data that informed the choice, and the univariate and EpiEstim baselines are run at their standard settings without tuning to this objective.

We compare four methods. The network Hawkes-type model developed in this paper. The univariate Hawkes-type model from (Tang and Prado, 2026), which fits each county independently using a single-location version of the same framework without cross-county transmission. Two EpiEstim-based approaches, both of which first estimate county-level reproduction numbers and then generate forward simulations. The "vanilla" variant holds the most recent reproduction number estimate constant over the forecast horizon, following the approach in the `EpiEstim` and `projections` R pack-

age vignettes. The "AR" variant models the resulting $\log(R_{s,t})$ series with a univariate autoregressive process and simulates incidence forward using the renewal equation with the forecasted reproduction numbers.

Table 3.4.: Out-of-sample forecasting performance (mean CRPS and 95% prediction interval coverage across counties and forecast origins). Lower CRPS indicates better calibrated probabilistic forecasts.

	One-step-ahead		Five-step-ahead		Ten-step-ahead	
	CRPS	Coverage	CRPS	Coverage	CRPS	Coverage
Univariate Hawkes-type	**1.43**	1.00	4.23	0.85	6.15	0.84
EpiEstim/Vanilla	2.53	0.97	3.24	0.90	6.52	0.83
EpiEstim/AR	2.56	0.97	3.29	0.86	6.90	0.88
Network Hawkes-type	2.04	0.98	**2.40**	0.91	**3.39**	0.84

Table 3.4 shows a clear horizon-dependent pattern. At the one-step horizon, the univariate Hawkes-type model achieves the lowest CRPS, which is consistent with the limited role of cross-county importation in next-day prediction and with the advantage of estimating fewer parameters. At longer horizons, however, the network Hawkes-type model performs best, reducing CRPS by 43% relative to the univariate Hawkes-type model and by 48% relative to the EpiEstim variants at ten steps. This comparison is particularly informative because the univariate and network Hawkes-type models differ mainly in whether cross-county transmission is included. The univariate model deteriorates sharply across horizons (from 1.43 to 6.15), whereas the network model deteriorates more gradually (from 2.04 to 3.39). The EpiEstim variants show a pattern closer to the univariate Hawkes-type model. Overall, these results suggest that network structure contributes most to forecasting skill at medium and long horizons, when importation from neighboring counties has more time to accumulate. This is consistent with the simulation study, where the advantage of the network model was strongest at

longer horizons and lower retention rates, matching the application setting in which estimated retention rates range from 0.20 to 0.53. These results indicate that the network model improves medium- and longer-horizon forecasts where cross-location transmission is non-negligible, while the univariate model is more accurate at the one-step horizon. The prediction intervals are close to their nominal level at the one-step horizon and fall somewhat below it at the longer horizons for all four methods, and the network model's longer-horizon intervals are about as well calibrated as those of the alternatives.

Model-based scenario analysis

The fitted model can also be used to generate illustrative scenario projections that reflect the spatial heterogeneity revealed by the network decomposition. We consider four hypothetical parameter perturbations, each initiated at time t_0 and evaluated over the remaining observation period $(t_0, T]$.

1. Source targeting: Reduce $R_{k,t}$ by 20% for $t > t_0$ in the three counties with the highest transmissibility (Santa Clara, San Francisco, Santa Cruz), mimicking a hypothetical reduction in source-county transmissibility.

2. Sink targeting: Reduce $R_{k,t}$ by 20% for $t > t_0$ in the three counties with the lowest transmissibility and negative net flow (Merced, San Joaquin, Contra Costa), mimicking hypothetical local reductions in counties whose fitted counts are more strongly shaped by importation.

3. All-pathway reduction: Reduce all incoming cross-county transmission weights to San Joaquin by 50%, setting $\alpha_{\mathrm{SJ},k} \leftarrow 0.5 \cdot \alpha_{\mathrm{SJ},k}$ for all $k \neq \mathrm{SJ}$, while holding

reproduction numbers constant. This mimics a broad reduction in incoming cross-county pathways into San Joaquin.

4. Targeted pathway removal: Remove the single largest incoming transmission pathway from Santa Clara to San Joaquin by setting $\alpha_{SJ,SC} = 0$, while leaving all other weights unchanged. This mimics a narrow perturbation to the fitted pathway from Santa Clara to San Joaquin.

We set t_0 to Nov 1, 2020, before the winter surge. For each scenario, we simulate epidemic trajectories under the modified parameter values and compare cumulative cases with the corresponding fitted-model baseline. These comparisons are descriptive summaries of the fitted model under hypothetical perturbations and are not interpreted as identified causal effects. Posterior uncertainty is propagated across MCMC samples.

Table 3.5.: Projected reduction in cumulative cases under four hypothetical scenarios evaluated under the fitted model. Values are posterior mean percentage reductions with 95% credible intervals. Network-wide effects are summed across all ten counties. Santa Clara and San Joaquin are shown as representative counties.

Scenario	Network		Santa Clara		San Joaquin	
Source targeting	37.51	[30.52, 45.35]	43.47	[35.98, 51.88]	38.33	[31.38, 46.38]
Sink targeting	3.29	[1.52, 6.89]	2.02	[0.76, 5.12]	5.49	[1.88, 10.09]
Reduce all pathways	4.58	[3.49, 6.11]	0.31	[0.05, 1.07]	29.80	[24.18, 35.67]
Remove targeted pathway	4.25	[2.36, 6.68]	0.29	[0.05, 1.07]	27.92	[15.81, 41.46]

Table 3.5 shows how these hypothetical perturbations differ under the fitted model. In these simulations, reducing reproduction numbers in the three counties with the highest transmissibility produces larger network-wide reductions than the corresponding perturbation applied to the three counties with the lowest transmissibility and negative net flow. For San Joaquin, perturbations to incoming pathways produce larger projected

reductions than the county-level reduction applied in the scenario that targets sinks. The two pathway-based perturbations for San Joaquin also give similar projected reductions. We emphasize that these contrasts are useful for illustrating how the fitted model translates the estimated source–sink structure into medium-term projections, but they should not be interpreted as causal policy effects.

§ 3.6. Concluding remarks

This chapter developed a Bayesian network Hawkes-type model for spatio-temporal disease surveillance that estimates time-varying intrinsic source-specific reproduction numbers, learns a sparse transmission network from incidence data, and provides an explicit decomposition of the observed reproduction number into local and imported components. The four contributions of the model class, namely the decomposition itself, the spectral radius interpretation of the system-level epidemic threshold, the regularized horseshoe prior on the off-diagonal entries of $\boldsymbol{\alpha}$, and the Markov-switching zero-inflation component for structural absences, are coupled in the likelihood and recovered in a single Bayesian update through the block MCMC sampler of Algorithm 6.

The simulation studies showed the main features of the model class across a range of transmission regimes. Network structure improved reproduction number estimation by 56% to 86% in mean absolute error over single-location baselines whenever spatial transmission was non-negligible, with forecasting gains concentrated at longer horizons and lower retention rates. The horseshoe prior on the spatial weight deviations remained robust to covariate misspecification, and the value of the distance covariate increased with network size, reflecting the quadratic growth in the number of off-diagonal pa-

rameters. The Markov-switching zero-inflation component improved both reproduction number estimation and longer-horizon forecast calibration when runs of structural zeros were present, and remained close to the baseline model when they were not, which suggests that including the component carries little risk in surveillance settings with low incidence.

The California application revealed substantial heterogeneity in transmission dynamics that standard univariate or closed-population multivariate methods would not capture. All ten counties had observed effective reproduction numbers near one under closed-population estimation, but the network decomposition produced intrinsic source-specific reproduction numbers ranging from 0.55 in San Joaquin to 1.14 in Santa Clara, with local contributions to the observed effective reproduction number as low as 20%. The estimated transmission network showed a source and sink structure, with Bay Area employment counties appearing as net sources and Central Valley residential counties as net sinks. A scenario analysis illustrated how the same observed reproduction number can imply very different projected trajectories under hypothetical targeted perturbations of the fitted model, depending on whether transmission at the location is driven primarily by local spread or by importation.

Two limitations deserve emphasis. First, the spatial weights are treated as static parameters, so the model does not capture temporal changes in transmission pathways over the course of an epidemic. The estimated $\boldsymbol{\alpha}$ is therefore best read as an average or persistent connectivity structure over the study period rather than as a time-invariant mechanism, which matters over a span as long and as epidemiologically variable as the one considered here. A fit on three epidemic subperiods, reported in § B.6.2, shows that the excitation matrix does shift across regimes while the ordering of counties

by net flow is broadly preserved, consistent with this reading. Temporal variation is assigned to $R_{k,t}$ while $\boldsymbol{\alpha}$ is held fixed because letting both vary freely would sharpen the identifiability problems discussed in § 3.2.3 and add substantial computational cost. Allowing time-varying $\boldsymbol{\alpha}$ would be a substantial extension, and is outside the scope of this chapter. Second, validation of the estimated transmission pathways is necessarily indirect. The model identifies lagged statistical dependence in incidence rather than physical transmission, and synchronized regional epidemics make this distinction harder to draw (Reich et al., 2021). The scenario analysis should likewise be read as descriptive of the fitted model rather than as causal evidence, particularly under large perturbations that could themselves alter behavior or network structure.

The model also contains no explicit seasonal or day-of-week term. A residual analysis on the application data (§ B.6.3) finds a weekly cycle in the standardized posterior predictive residuals, with a dominant residual period of one week for all ten counties and positive residual autocorrelation at lags of one and two weeks, while showing no monthly or yearly seasonality. This weekly structure is consistent with day-of-week reporting patterns rather than with weekly variation in transmission, and the reproduction trajectories, smoothed by the small innovation variance W, do not absorb it. Its most direct effect is at the observation level, where the Poisson model understates the predictive dispersion at short horizons. Because the weekly cycle is faster than the transmission dynamics that the excitation matrix and the reproduction trajectories represent, we expect it to have limited influence on the source and sink ordering, the network reproduction number, and the local and imported decomposition, although we do not fit a model with a seasonal term to verify this directly. Incorporating a day-of-week or seasonal term in the intensity, or aggregating to weekly counts, is a natural

extension left to future work.

Zero inflation, finally, is not needed for the California application, where the counts are not sparse. Only 0.2% of the observed counts are zero, and those zeros are isolated rather than occurring in runs. When the full model is fit with the zero-inflation component, the activation process switches itself off, learning an activation probability near one and flagging no observation as a structural zero (§ B.6.4). Fitting without the component, as a plain Poisson observation model, leaves the network reproduction number, the ordering of counties by net flow, and the forecast scores essentially unchanged. We retain the component in the reported fit as a safeguard that reduces to Poisson when structural zeros are absent, and the simulation in § B.4 shows that it helps when runs of structural zeros are present, as in sparser small-area settings.

On the computational side, a cost of $O(ST^2 + S^2T)$ per iteration is compatible with the weekly cadence of medium-term policy planning at the ten-county scale of the application, but becomes prohibitive for routine daily updates across larger networks. This computational bottleneck is different from that in event-level spatio-temporal Hawkes models, where the main cost comes from the quadratic number of pairwise event-triggering calculations and has been addressed using GPU and distributed computing (Holbrook et al., 2020b; Ko et al., 2024). Here the goal is instead to exploit the aggregated count structure to obtain interpretable regional surveillance summaries without introducing latent within-county event locations. Chapter 4 develops a structured variational counterpart to the present sampler. The proposed particle Laplace variational approximation (PLVA) follows the same column-wise factorization but replaces the block MCMC updates with an importance-weighted particle approximation for the latent reproduction trajectory and a mixture of Laplace approximations for the spatial

parameters. It stands in the same relation to the network Hawkes-type model that the hybrid variational approximation (HVA) of Chapter 2 stands in to the dynamic generalized transfer function (DGTF) class.

Chapter 4

Particle Laplace variational approximation for network Hawkes models

§ 4.1. Introduction

The block MCMC sampler developed in Chapter 3 provides asymptotically exact posterior inference for the network Hawkes-type model, but at a cost of $O(ST^2 + S^2T)$ per iteration. This together with the several thousand iterations needed for adequate mixing make routine daily updates impractical when the number of locations S grows beyond the ten-county scale considered there. Variational inference is the natural computational alternative for this regime, and this chapter develops an algorithm in that spirit, matched to the geometry of the network Hawkes-type posterior.

The matching is motivated by features of the posterior that simple approximating families represent poorly. The transmission weights $\alpha_{\cdot,k}$ and the source-specific reproduction trajectory $r_k = (r_{k,0}, \ldots, r_{k,T})$ enter the intensity multiplicatively as $\lambda_{s,t} \propto \alpha_{s,k}\, h(r_{k,t})$, where h is the positive monotone gain function used in Chapter 3. The column-stochastic constraint removes the pure rescaling degeneracy between the two components, and the

regularized horseshoe prior shrinks most off-diagonal entries of $\boldsymbol{\alpha}$ toward a mobility-based baseline. Even so, the likelihood remains nearly flat along directions that trade off spatial allocation against reproduction number magnitude when destination trajectories are highly correlated across source locations. Such correlation arises routinely in epidemic surveillance, where synchronized waves of cases propagate across geographically connected counties, and the corresponding posterior shows pronounced ridges along the flat directions. The conditional posterior of either block given the other remains concentrated near its mode, allowing the block sampler of Chapter 3 to traverse the ridges by alternating block updates without collapsing them. These ridges need not be elliptical. They can be curved, skewed, heavy tailed, or locally multimodal, and it is that departure from elliptical geometry, rather than correlation itself, that a joint Gaussian variational approximation cannot follow. A correlated joint Gaussian does represent linear dependence between the two blocks, but it constrains the mean of either block given the other to be affine in that other block, and the corresponding covariance to be the same wherever the ridge is evaluated. A single Gaussian centered at one mode can therefore misrepresent the spread of a ridge whose location and curvature change along its length. A mean-field factorization fares worse, since it removes the cross-block dependence on which the conditional structure rests. The variational Bayes method of Sulem et al. (2025) for sparse continuous-time Hawkes processes adopts a mean-field family and addresses graph sparsity through thresholded penalties on triggering kernels, while the Cox-Hawkes method of Miscouridou et al. (2022) for continuous-space spatio-temporal Hawkes inherits the unimodal Gaussian variational geometry of automatic differentiation variational inference. Neither method addresses the ridge structure that the multiplicative parameterization produces in this setting.

This chapter develops the particle Laplace variational approximation (PLVA), an iterative conditional approximation that preserves the conditional update structure of the block sampler of Chapter 3. The method cycles through four blocks, each given an approximating family matched to the conditional geometry of the corresponding posterior. It does not optimize a single global evidence lower bound. What it returns instead is a stable collection of conditional approximations, whose accuracy is assessed empirically against the block sampler. The efficient importance variational approximation of Loaiza-Maya and Nibbering (2025), referred to hereafter as EIVA, handles the latent reproduction trajectory. The EIVA draws weighted particles from a Gaussian proposal calibrated against the observation density through a single Taylor linearization, and corrects the bias of the linearization through self-normalized importance weights against the exact observation density with the zero-inflation indicators marginalized. The spatial parameters then receive a mixture of Laplace approximations whose components are placed at the conditional posterior modes for each EIVA particle, with mixture weights inherited from the EIVA importance weights. The horseshoe scales and inverse-gamma auxiliaries are updated by approximate moment formulas based on the centered, unregularized Makalic–Schmidt augmentation (Makalic and Schmidt, 2016). The zero-inflation indicators are marginalized analytically by forward filtering of the binary Markov chain because the PLVA requires only the marginal observation density and its derivatives, not posterior samples of the indicators. This differs from the forward-filtering backward-sampling step used inside the MCMC sampler of Chapter 3, which explicitly samples the indicator trajectories.

The methodological argument turns on a variance decomposition for the mixture on the spatial block that mirrors the law of total variance applied to the column-conditional

variational joint distribution. The within-particle component of the mixture variance aggregates the local conditional curvatures across particles, and the between-particle component captures how the conditional mode of the spatial parameters varies across the importance-weighted reproduction trajectories. Both components have analogues under a correlated joint Gaussian, since the same decomposition holds there. What separates the mixture is that its components are free to move and to change shape from one trajectory to the next, whereas a joint Gaussian ties the conditional mean to an affine function of the trajectory and holds the conditional covariance fixed. The mixture is therefore expected to help when the conditional mode or curvature of the spatial block varies nonlinearly with the trajectory, and a correlated Gaussian may be adequate when that dependence is close to affine. Which case obtains is an empirical question, and the size of any gain depends on how sharply the conditional spatial mode responds to $\boldsymbol{\eta}$ in the regimes we tested. A mean-field family remains more restrictive than either, since it drops the dependence between the blocks altogether. The empirical contribution of the chapter is to document, through simulation studies and an application to daily COVID-19 case counts across ten California counties, that the PLVA matches the posterior medians, the reproduction number trajectories, and the predictive summaries of the block MCMC sampler of Chapter 3 at 2 to 15 times lower wall clock cost depending on the particle count. A replicated comparison against MCMC on ten independently simulated datasets also shows that the credible intervals on the spatial parameters stay too narrow at every particle count, which identifies a structural property of the Laplace mixture that the discussion section returns to with two concrete refinements.

The PLVA draws on several research threads. Within variational inference for state-space models, the closest precedent is the Gaussian variational approximation of Quiroz

et al. (2023) for high-dimensional state-space models, together with the broader structured variational methods of Hoffman et al. (2013) and the streamlined variational Bayes approach of Wand (2017). The use of importance-weighted Laplace components extends these approaches to settings in which the conditional distribution of the spatial block has a well-defined local mode and curvature but is non-conjugate, while the temporal block requires an importance correction in place of analytic conditional sampling. Within Bayesian inference for aggregated Hawkes processes, the closest precedent is Zhou and Papadogeorgou (2025) that develops a MCMC treatment for a *continuous-time* Hawkes process whose events are observed only as counts aggregated into time bins. The model presented in this chapter is instead a *discrete-time network* Hawkes-type model. The renewal mechanism is discretized directly through a fixed serial interval, the unit of analysis is the daily count rather than a binned realization of an underlying continuous-time process, and the process is multivariate, with a column-stochastic transmission matrix, a regularized horseshoe prior on its off-diagonal entries, and a Markov-switching zero-inflation component that the single-process aggregated setting does not contain. The contribution is also methodologically distinct. Zhou and Papadogeorgou (2025) address the continuous-time aggregation problem by sampling, whereas this chapter develops a structured *variational* algorithm for the discrete-time network problem, designed around the ridge geometry that the multiplicative parameterization produces. The two approaches are therefore complementary rather than competing. The PLVA also generalizes the modified hybrid variational approximation (HVA) of Loaiza-Maya et al. (2022) that was applied to the univariate dynamic generalized transfer function (DGTF) class in Chapter 2. The global block of § 4.3.5 retains that factorization, and the spatial and temporal blocks are augmented to handle the

multiplicative coupling that distinguishes the network setting from the univariate one.

The remainder of the chapter is organized as follows. § 4.2 fixes the notation and recalls the components of the network Hawkes-type model that the PLVA relies on. § 4.3 develops the proposed particle Laplace variational approximation (PLVA) algorithm, presenting each of the four blocks of this algorithm and the block-coordinate algorithm. § 4.4 reports simulation studies that compare the proposed method with the block MCMC sampler of Chapter 3 and baseline algorithms that use mean-field and Gaussian copula variational approaches. § 4.5 applies the proposed method to the daily COVID-19 case counts across ten California counties analyzed in the previous chapter. § 4.6 summarizes the chapter and discusses limitations.

§ 4.2. Network Hawkes-type model and notation

The variational approximation builds on the network Hawkes-type model developed in Chapter 3. This section fixes the notation used throughout the chapter and recalls the components that the variational method relies on.

Let $y_{s,t}$ denote the observed case count at destination $s \in \{1, \dots, S\}$ and time $t \in \{1, \dots, T\}$. The observation model is the zero-inflated Poisson of Chapter 3, governed by a latent activation indicator $z_{s,t}$ that follows a first-order Markov chain on $\{0, 1\}$ with $\text{logit}(p_{s,t}) = \beta_0 + \beta_1 z_{s,t-1}$. Conditional on $z_{s,t} = 1$, the count follows a $\text{Pois}(\lambda_{s,t})$ distribution with intensity

$$\lambda_{s,t} = \mu + \sum_{k=1}^{S} \sum_{l=1}^{t-1} \alpha_{s,k} R_{k,l} \phi_{t-l} y_{k,l}, \tag{4.2.1}$$

combining the baseline rate μ, the column-stochastic transmission weights $\alpha_{s,k}$, the

intrinsic source-specific reproduction numbers $R_{k,l}$, and the fixed serial interval distribution ϕ_{t-l}.

The reproduction number $R_{k,t} = h(r_{k,t})$ is a positive monotone transformation of a latent state $r_{k,t}$ that follows a Gaussian random walk $r_{k,t} = r_{k,t-1} + \sqrt{W}\,\eta_{k,t}$ with $\eta_{k,t} \sim N(0,1)$, $r_{k,0} = 0$, and gain function $h(r) = \log(1 + e^r)$. The non-centered representation $r_{k,t} = \sqrt{W}\sum_{j=0}^{t-1}\eta_{k,j}$ encodes the entire latent trajectory at source k by the disturbance vector $\boldsymbol{\eta}_k = (\eta_{k,0}, \ldots, \eta_{k,T-1})'$ of dimension T. The transmission matrix $\boldsymbol{\alpha} = [\alpha_{s,k}]$ uses the column-by-column parameterization of Chapter 3, with retention rate $\alpha_{k,k} = w_k \in (0,1)$ and off-diagonal entries $\alpha_{s,k} = (1 - w_k)\,u_{s,k}/U_k$ for $s \neq k$, where $u_{s,k} = \exp(m_{s,k} + \tilde{\theta}_{s,k})$, $m_{s,k}$ is a fixed mobility covariate, $\tilde{\theta}_{s,k} = \theta_{s,k} - \bar{\theta}_k$ is the column-centered residual deviation, and $U_k = \sum_{j\neq k} u_{j,k}$.

The deviation terms $\theta_{s,k}$ receive the regularized horseshoe prior of Chapter 3, written here in the non-centered form

$$\theta_{s,k} = \tau_k\,\tilde{\gamma}_{s,k}\,\varepsilon_{s,k}, \qquad \varepsilon_{s,k} \sim N(0,1), \qquad \tilde{\gamma}^2_{s,k} = \frac{c^2\gamma^2_{s,k}}{c^2 + \tau_k^2\gamma^2_{s,k}}, \tag{4.2.2}$$

with $\gamma_{s,k} \sim C^+(0,1)$ and $\tau_k \sim C^+(0, \tau_{0,k})$. The half-Cauchy scales are represented through the inverse-gamma augmentation of Makalic and Schmidt (2016) with auxiliaries $a_{s,k}$ and b_k, exactly as in Chapter 3. For the centered, unregularized horseshoe this augmentation yields conjugate inverse-gamma conditionals. § 4.3.3 describes how those moments are used for the regularized non-centered parameterization of the spatial block.

For the variational approximation, the unknowns are partitioned into four blocks. The *temporal block* consists of the disturbance vectors $\boldsymbol{\eta}_k$ for $k = 1, \ldots, S$, which encode the latent reproduction trajectories through the non-centered random walk. The *spatial*

block consists of $\zeta_k = (\text{logit}(\omega_k),\ \{\varepsilon_{s,k}\}_{s\neq k})' \in \mathbb{R}^S$ for $k = 1, \ldots, S$, which encodes the retention rate and the non-centered horseshoe deviations on a common unconstrained scale. The *horseshoe block* consists of the local and global shrinkage scales together with their Makalic–Schmidt auxiliaries, $\xi_k = (\{\gamma_{s,k}\}_{s\neq k},\ \tau_k,\ \{a_{s,k}\}_{s\neq k},\ b_k)$ for $k = 1, \ldots, S$. The *zero-inflation block* consists of the indicators $\boldsymbol{z} = \{z_{s,t}\}$. A small set of global parameters $\vartheta_{\text{glb}} = (\log \mu,\ \beta_0,\ \beta_1)$ enters all column updates additively through the offset intensity.

Three reproduction number quantities, developed in detail in Chapter 3, are recovered as deterministic transformations of the variational posterior. The intrinsic source-specific reproduction number $R_{k,t}$ gives the total expected offspring generated by a case at source k at time t. The matrix $\boldsymbol{B}_t = [\alpha_{s,k} R_{k,t}]$ plays the role of a next-generation matrix (Diekmann et al., 1990), and its spectral radius $R_t^{\text{net}} = \rho(\boldsymbol{B}_t)$ governs system-wide epidemic growth. The observed effective reproduction number $R_{s,t}^{\text{eff}}$ targeted by closed-population renewal estimators admits the additive decomposition $R_{s,t}^{\text{eff}} = R_{s,t}^{\text{local}} + R_{s,t}^{\text{import}}$ derived in Chapter 3.

§ 4.3. Particle Laplace variational approximation

This section develops the methodological core of the chapter. The PLVA is an iterative conditional approximation in which each block update uses an approximating family matched to the conditional geometry of the corresponding posterior, with blocks as defined in § 4.2.

Three properties of the construction hold throughout this section. First, the method is variational in the broad sense that it optimizes parametric families within blocks, but

the complete iteration is not coordinate ascent on a single global ELBO. Second, the objects the algorithm forms are approximations to conditional laws, each evaluated at the current values of the other blocks, so they are not exact conditionals of one normalized joint density, and a fixed point makes them mutually consistent without removing that conditioning. Third, the horseshoe scales are updated by conjugate moment formulas borrowed from a centered, unregularized version of the prior, which the fitted model does not satisfy exactly. § 4.4 and 4.5 assess the resulting approximation against the block MCMC sampler.

In plain terms, the PLVA approximates the posterior one source region at a time. The uncertain reproduction trajectory of a region is represented by a set of weighted *particles*, each representing one candidate trajectory. The spatial parameters attached to that region are then approximated, separately for each particle, by a *Laplace approximation* built at the mode of their conditional posterior. Averaging these Laplace approximations over the particle weights produces a mixture whose component centers and curvatures can change across candidate trajectories. This provides more flexibility than one local Gaussian approximation when the relevant dependence is non-elliptical, although it does not guarantee a better approximation in every setting. These two ingredients give the method its name, with *particles* representing the latent reproduction trajectories and *Laplace* approximations representing the conditional spatial block.

The object of inference is the joint posterior of the four parameter blocks defined in § 4.2 together with the global parameters,

$$p\big(\{\boldsymbol{\eta}_k\}_{k=1}^{S},\ \{\zeta_k\}_{k=1}^{S},\ \{\xi_k\}_{k=1}^{S},\ \boldsymbol{z},\ \vartheta_{\mathrm{glb}} \,\big|\, y_{1:T}\big), \tag{4.3.1}$$

formed from the temporal blocks $\boldsymbol{\eta}_k$, the spatial blocks ζ_k, the horseshoe blocks ξ_k, the

zero-inflation indicators $\boldsymbol{z}$, and the global parameters ϑ_{glb}. Rather than positing a single normalized joint density over all of these quantities, the PLVA maintains the following collection of blockwise approximations at outer iteration i:

$$\boldsymbol{Q}^{(i)} = \left\{ q_{\eta_k}^{(i)},\ q_{\zeta_k}^{(i)},\ q_{\xi_k}^{(i)} \right\}_{k=1}^{S} \ \cup\ \left\{ q_{\text{glb}}^{(i)} \right\}. \tag{4.3.2}$$

Equation (4.3.2) collects the approximations the algorithm maintains at iteration i. For column k, $q_{\eta_k}^{(i)}$ is the importance-weighted particle approximation to the temporal conditional evaluated at the preceding spatial summary, and $q_{\zeta_k}^{(i)}$ is the Laplace mixture obtained from those temporal particles, with one component per particle. The algorithm stores which component belongs to which particle, and that pairing is what encodes the dependence between the temporal and spatial blocks. The horseshoe block receives inverse-gamma approximations, and the global block receives a Gaussian factor-covariance approximation. The zero-inflation indicators $\boldsymbol{z}$ do not receive an approximation of their own, but are marginalized analytically inside every observation density by the forward filter of § 4.3.4.

More explicitly, Eq. (4.3.4) defines the temporal and spatial updates conditional on the current values $C_{-k}^{(i)}$ outside column k:

$$q_{\eta_k}^{(i)} \approx p\big(\boldsymbol{\eta}_k \mid y_{1:T},\ \hat{\boldsymbol{\zeta}}_k^{(i-1)},\ C_{-k}^{(i)}\big), \tag{4.3.3}$$

$$q_{\zeta_k}^{(i)} = \sum_{m=1}^{M} \bar{\varpi}_k^{(m,i)}\, N\big(\zeta_k^{*(m,i)},\ \Sigma_k^{(m,i)}\big). \tag{4.3.4}$$

The outer iteration recomputes the plug-in summaries and repeats these updates until the monitored quantities stabilize. Its output is a self-consistent collection of conditional approximations at an algorithmic fixed point.

Three related approximations appear in this thesis and are worth separating at the

outset. The HVA of Chapter 2, following Loaiza-Maya et al. (2022), is a hybrid variational scheme for the *univariate* DGTF class that marginalizes the latent state by SMC and optimizes a Gaussian copula over the static parameters. The EIVA of Loaiza-Maya and Nibbering (2025) is not used here as a standalone inference method, but as a *component* of this algorithm, where it is the importance sampling routine that produces the weighted particle approximation of a single temporal block $\boldsymbol{\eta}_k$. The PLVA is the full method of this chapter. It runs an EIVA step for each temporal block inside a block-coordinate loop, pairs it with the Newton–Laplace mixture for the spatial block, and retains the HVA treatment for the global parameters. The PLVA thus extends the HVA from the univariate to the network setting, where the multiplicative coupling between transmission weights and reproduction numbers forces the spatial block to be handled by a Laplace mixture indexed by particles in place of a single Gaussian copula.

At a high level, each iteration of the PLVA proceeds through the following steps. The full statement, with the gradient and curvature expressions, is given as Algorithm 7 in § 4.3.6.

1. *Temporal block.* For each source column k, update the latent reproduction trajectory by drawing M EIVA importance particles $\{\boldsymbol{\eta}_k^{(m)}\}_{m=1}^{M}$ with normalized weights $\{\bar{\varpi}_k^{(m)}\}_{m=1}^{M}$ (§ 4.3.1). The particle count M is the principal tuning constant of the method, controlling the accuracy of the importance estimate. Its effect is studied in § 4.4.7. The remaining settings, namely the Newton tolerance and maximum iteration count applied for each particle and the SGA learning rate, are technical and held at the defaults of Algorithm 7 throughout this chapter.

2. *Spatial block.* For each particle, find the conditional mode of the spatial parame-

ters ζ_k by Newton's method (§ 4.3.2).

3. *Mixture.* Assemble the Laplace approximations across particles into a weighted mixture, which serves as the variational distribution for the spatial block (§ 4.3.2).

4. *Remaining blocks.* Update the horseshoe scales and auxiliaries using the approximate inverse-gamma moment updates of § 4.3.3, and update the global parameters by stochastic gradient ascent (§ 4.3.5).

5. *Iterate.* Repeat for a fixed number of iterations N_{iter}, monitoring the convergence diagnostics of § 4.4.2.

Each block receives a family matched to its conditional geometry. For the temporal block, the EIVA proposal of Loaiza-Maya and Nibbering (2025) constructs a Gaussian proposal calibrated against a moment-matched Gaussian working approximation to the Poisson likelihood through a single Taylor linearization, draws M weighted particles from the proposal, and corrects the bias of the linearization through self-normalized importance weights that target the exact observation density with the zero-inflation indicators marginalized. For the spatial block, a mixture of Laplace approximations places one Gaussian component at the conditional mode of ζ_k given each EIVA particle and assigns the component the corresponding importance weight, so that the spatial block inherits the particle approximation from the temporal block without a separate variational optimization. For the horseshoe block, the algorithm updates the scales and auxiliaries by the inverse-gamma moment formulas of the centered, unregularized Makalic–Schmidt augmentation, as described in § 4.3.3. For the zero-inflation block, forward filtering of the binary Markov chain marginalizes $z_{s,t}$ analytically inside the

observation density. The global parameters are updated by SGA on a Gaussian factor covariance approximation, following the scheme of Loaiza-Maya et al. (2022).

Throughout this section, the hat notation, such as $\hat{\boldsymbol{\zeta}}_k$ or $\hat{r}_k$, denotes the current plug-in summary, recomputed once per outer iteration of Algorithm 7. At column k, the EIVA proposal approximates the conditional posterior of $\boldsymbol{\eta}_k$ given the data, the current spatial vector $\hat{\boldsymbol{\zeta}}_k$, and the current values of the parameters in all other blocks and columns. The Newton–Laplace mixture approximates the conditional posterior of $\boldsymbol{\zeta}_k$ given each $\boldsymbol{\eta}_k$ particle and those same current values. The within-particle and between-particle terms in § 4.3.2 summarize, respectively, average local conditional curvature and variation of the conditional modes across the temporal particles. When the iteration stabilizes, these quantities define the approximation returned for the column.

The remainder of this section develops each block in turn. § 4.3.1 presents the EIVA proposal for the temporal block. § 4.3.2 presents the Newton–Laplace mixture for the spatial block, including the law of total variance decomposition that motivates the mixture. § 4.3.3 presents the approximate moment updates for the horseshoe block. § 4.3.4 presents the forward filtering recursion for the zero-inflation indicators. § 4.3.5 presents the stochastic gradient updates for the global parameters. § 4.3.6 assembles the four blocks into the block-coordinate algorithm and reports its computational cost per iteration.

4.3.1. EIVA proposal for the latent reproduction trajectory

For each source column k, the EIVA step targets the conditional posterior of the disturbance vector $\boldsymbol{\eta}_k$ given the data and the current values of the parameters in the other blocks and other columns. The proposal is constructed by linearizing the gain function

h around the current trajectory $\hat{r}_{k,t}$, applying a moment-matched Gaussian working approximation to the Poisson likelihood, and isolating the contribution of column k to the intensity.

To isolate the column-k contribution, define the offset intensity that collects contributions from all other source columns:

$$\tilde{\mu}_{s,t}^{(-k)} = \hat{\mu} + \sum_{k' \neq k} \hat{\alpha}_{s,k'} \sum_{l=1}^{t-1} h(\hat{r}_{k',l})\, \phi_{t-l}\, y_{k',l}, \tag{4.3.5}$$

where $\hat{\mu}$, $\hat{\alpha}_{s,k'}$, and $\hat{r}_{k',l}$ are evaluated at the current values. The full intensity can then be written as

$$\lambda_{s,t} = \tilde{\mu}_{s,t}^{(-k)} + \alpha_{s,k} \sum_{l=1}^{t-1} h(r_{k,l})\, \phi_{t-l}\, y_{k,l}. \tag{4.3.6}$$

Linearizing the gain function around the current trajectory gives $h(r_{k,l}) \approx h(\hat{r}_{k,l}) + h'(\hat{r}_{k,l})(r_{k,l} - \hat{r}_{k,l})$. Using the non-centered representation $r_{k,l} = \sqrt{W} \sum_{j=0}^{l-1} \eta_{k,j}$, the linearized intensity can be written as a linear function of $\boldsymbol{\eta}_k$:

$$\lambda_{s,t} \approx \mu_{s,t}^{(0)} + \sqrt{W}\, \alpha_{s,k} \sum_{j=0}^{T-1} c_{k,j,t}\, \eta_{k,j}, \quad \text{with} \quad c_{k,j,t} = \sum_{l=j+1}^{t-1} h'(\hat{r}_{k,l})\, \phi_{t-l}\, y_{k,l}, \tag{4.3.7}$$

where

$$\mu_{s,t}^{(0)} = \tilde{\mu}_{s,t}^{(-k)} + \alpha_{s,k} \sum_{l=1}^{t-1} \left\{ h(\hat{r}_{k,l}) - h'(\hat{r}_{k,l}) \hat{r}_{k,l} \right\} \phi_{t-l} y_{k,l}, \quad c_{k,j,t} = \sum_{l=j+1}^{t-1} h'(\hat{r}_{k,l})\, \phi_{t-l}\, y_{k,l}. \tag{4.3.8}$$

Stacking over observation times $t = 2, \ldots, T$ gives

$$\boldsymbol{\lambda}_s \approx \boldsymbol{\mu}_s^{(0)} + \sqrt{W}\, \alpha_{s,k} \mathbf{C}_k \boldsymbol{\eta}_k, \tag{4.3.9}$$

where $\boldsymbol{\lambda}_s = (\lambda_{s,2}, \ldots, \lambda_{s,T})'$, $\boldsymbol{\mu}_s^{(0)} = (\mu_{s,2}^{(0)}, \ldots, \mu_{s,T}^{(0)})'$, and $\mathbf{C}_k$ is the $(T-1) \times T$ matrix

with entries $c_{k,j,t}$. Although the serial interval has effective support L_ϕ, $\mathbf{C}_k$ is generally dense below its triangular boundary. An early innovation $\eta_{k,j}$ enters every later random-walk state $r_{k,l}$ for $l > j$, so truncating ϕ_{t-l} does not bound the distance $t - j$ over which $c_{k,j,t}$ can be nonzero.

Combining the normal $\boldsymbol{\eta}_k \sim N(\mathbf{0}, \mathbf{I})$ with the Gaussian working observation equation $\boldsymbol{y}_s \approx \boldsymbol{\mu}_s^{(0)} + \sqrt{W}\,\alpha_{s,k}\mathbf{C}_k\boldsymbol{\eta}_k + \boldsymbol{\varepsilon}_s, \quad \boldsymbol{\varepsilon}_s \sim N(\mathbf{0}, \boldsymbol{\Lambda}_s)$, where $\boldsymbol{y}_s = (y_{s,2}, \ldots, y_{s,T})'$ and $\boldsymbol{\Lambda}_s = \mathrm{diag}(\hat{\lambda}_{s,2}, \ldots, \hat{\lambda}_{s,T})$, gives the Gaussian proposal

$$g_k(\boldsymbol{\eta}_k) = N\left(\boldsymbol{\eta}_k \mid \mathbf{u}_k, \boldsymbol{\Omega}_k^{-1}\right), \tag{4.3.10}$$

with precision and mean

$$\boldsymbol{\Omega}_k = \mathbf{I} + W\sum_{s=1}^{S}\alpha_{s,k}^2\mathbf{C}_k'\boldsymbol{\Lambda}_s^{-1}\mathbf{C}_k, \quad \mathbf{u}_k = \sqrt{W}\,\boldsymbol{\Omega}_k^{-1}\sum_{s=1}^{S}\alpha_{s,k}\mathbf{C}_k'\boldsymbol{\Lambda}_s^{-1}\mathbf{d}_s, \tag{4.3.11}$$

where $\mathbf{d}_s = \boldsymbol{y}_s - \boldsymbol{\mu}_s^{(0)}$. This proposal is the linearized Gaussian approximation used inside the MH step of the block sampler in Chapter 3, except that here it is used as an importance proposal rather than as an MH proposal. The proposal calibration uses the Poisson likelihood rather than the likelihood that marginalizes the zero-inflation indicators, because the moment-matched Gaussian working approximation has a simpler form under the Poisson model. The importance correction below restores the marginalized target.

To correct the bias due to the linearization and the Poisson working approximation, draw $\boldsymbol{\eta}_k^{(m)} \sim g_k(\boldsymbol{\eta}_k)$, for $m = 1, \ldots, M$, and reconstruct $r_{k,t}^{(m)} = \sqrt{W}\sum_{j=0}^{t-1}\eta_{k,j}^{(m)}$, for $t = 1, \ldots, T$. For each particle, compute the exact nonlinear intensity

$$\lambda_{s,t}^{(m)} = \tilde{\mu}_{s,t}^{(-k)} + \alpha_{s,k}\sum_{l=1}^{t-1}h(r_{k,l}^{(m)})\,\phi_{t-l}\,y_{k,l}. \tag{4.3.12}$$

The observation density with the zero-inflation indicators marginalized is then evaluated for each particle using the forward filtering recursion of § 4.3.4. Writing $p_{\mathrm{zi}}\left(y_{s,t} \mid y_{s,1:t-1}, \lambda_{s,1:t}^{(m)}\right)$ for the one-step-ahead predictive density returned by the filter, the unnormalized importance weight is

$$\tilde{\varpi}_k^{(m)} = \frac{\prod_{s=1}^{S} \prod_{t=1}^{T} p_{\mathrm{zi}}\left(y_{s,t} \mid y_{s,1:t-1}, \lambda_{s,1:t}^{(m)}\right) \prod_{j=0}^{T-1} N(\eta_{k,j}^{(m)} \mid 0, 1)}{N\left(\boldsymbol{\eta}_k^{(m)} \mid \mathbf{u}_k, \boldsymbol{\Omega}_k^{-1}\right)}. \tag{4.3.13}$$

The normalized weight is

$$\bar{\varpi}_k^{(m)} = \frac{\tilde{\varpi}_k^{(m)}}{\sum_{m'=1}^{M} \tilde{\varpi}_k^{(m')}}. \tag{4.3.14}$$

For any function $f(\boldsymbol{\eta}_k)$, the self-normalized importance sampling estimator of the corresponding conditional posterior expectation is

$$\widehat{\mathbb{E}}\{f(\boldsymbol{\eta}_k) \mid y_{1:T}, \text{current values}\} = \sum_{m=1}^{M} \bar{\varpi}_k^{(m)} f(\boldsymbol{\eta}_k^{(m)}). \tag{4.3.15}$$

The effective sample size of the importance sampler is

$$\mathrm{ESS}_k = \frac{1}{\sum_{m=1}^{M} (\bar{\varpi}_k^{(m)})^2}. \tag{4.3.16}$$

The ESS_k is the primary diagnostic for the quality of the EIVA proposal at column k. Values close to M indicate that the linearized Gaussian proposal is well matched to the column-conditional target, whereas small values indicate poor proposal calibration. This is an importance sampling effective sample size that measures how many equally weighted independent draws the M weighted particles are worth. It is not an MCMC effective sample size, since the particles are independent draws from g_k rather than autocorrelated states of a Markov chain. The MCMC effective sample sizes reported

for the block sampler of Chapter 3 in § 4.4 and 4.5 measure a different quantity, namely the autocorrelation of the Markov chain.

The EIVA particle set

$$\left\{\boldsymbol{\eta}_k^{(m)}, \bar{\varpi}_k^{(m)}\right\}_{m=1}^{M}$$

enters the spatial block of § 4.3.2 both as the conditioning trajectories for the Laplace components and as the source of the mixture weights. The same particle set also yields the importance sampling estimate of the column-integrated log likelihood,

$$\widehat{\log p}_k\left(y_{1:T} \mid \zeta_k, \xi_k, \vartheta_{\text{glb}}, \hat{\alpha}_{\cdot,-k}, \hat{r}_{-k}\right) = \log\left\{\frac{1}{M}\sum_{m=1}^{M} \tilde{\varpi}_k^{(m)}\right\}. \tag{4.3.17}$$

This quantity estimates the likelihood contribution obtained after integrating out the column trajectory $\boldsymbol{\eta}_k$ under its standard multivariate normal prior, conditional on the current offsets induced by all other columns. The notation $\hat{\alpha}_{\cdot,-k}$ and $\hat{r}_{-k}$ denotes that the offset depends on the current values of the spatial weights and trajectories in all columns other than k. The subscript k on $\widehat{\log p}_k$ emphasizes that this is a column-integrated likelihood estimate, rather than the full global marginal log likelihood.

4.3.2. Newton–Laplace mixture for the spatial parameters

For each source column k, the blockwise approximation for the spatial parameter vector $\zeta_k \in \mathbb{R}^S$ is a finite mixture of Gaussian components indexed by EIVA particles:

$$q_{\zeta_k}(\zeta_k) = \sum_{m=1}^{M} \bar{\varpi}_k^{(m)} N\left(\zeta_k \mid \zeta_k^{*(m)}, \Sigma_k^{(m)}\right), \tag{4.3.18}$$

where the mixture weights $\{\bar{\varpi}_k^{(m)}\}_{m=1}^{M}$ are the normalized importance weights from EIVA in Eq. (4.3.13) and each component is a Laplace approximation to the conditional

posterior of ζ_k given the m-th particle's reproduction trajectory $r_k^{(m)}$ and the current values of the parameters in the other blocks and other columns. The variational parameters for column k are

$$\varsigma_k = \left\{\zeta_k^{*(m)},\ \Sigma_k^{(m)},\ \bar{\varpi}_k^{(m)} :\ m = 1, \ldots, M,\ \ \textstyle\sum_m \bar{\varpi}_k^{(m)} = 1\right\}. \tag{4.3.19}$$

The component means $\zeta_k^{*(m)}$ are determined by Newton's method on the conditional log-posterior of each particle, the component covariances $\Sigma_k^{(m)}$ are determined by the local curvature at each mode, and the mixture weights are inherited from EIVA. No separate ELBO optimization is carried out for the variational parameters of the spatial block within an iteration of the block-coordinate algorithm. The component means and covariances are determined by the Newton–Laplace approximations at each particle.

Column-conditional approximation

The EIVA proposal in § 4.3.1 is built from the current spatial parameter vector $\hat{\zeta}_k$, which enters through the column-k weights $\alpha_{\cdot,k}$ in both the proposal moments in Eq. (4.3.11) and the exact intensity in Eq. (4.3.12) used in the importance weights. The importance-weighted particle set $\{\boldsymbol{\eta}_k^{(m)}, \bar{\varpi}_k^{(m)}\}_{m=1}^M$ therefore targets the column-conditional law

$$p(\boldsymbol{\eta}_k \mid y_{1:T},\ \hat{\zeta}_k,\ C_{-k}), \tag{4.3.20}$$

which conditions on $\hat{\zeta}_k$ rather than marginalizing over it. The mixture on the spatial block in Eq. (4.3.18) pairs each such particle with the Laplace expansion of $p(\zeta_k \mid y_{1:T}, \boldsymbol{\eta}_k^{(m)}, C_{-k})$ at its conditional mode. The resulting column-conditional variational

distribution takes the form

$$\begin{aligned}\tilde{q}_k(\zeta_k) &= \int p(\zeta_k \mid y_{1:T},\, \boldsymbol{\eta}_k,\, C_{-k})\, p(\boldsymbol{\eta}_k \mid y_{1:T},\, \hat{\zeta}_k,\, C_{-k})\, d\boldsymbol{\eta}_k \\ &\approx \sum_{m=1}^{M} \bar{\varpi}_k^{(m)}\, N\Big(\zeta_k \mid \zeta_k^{*(m)},\, \Sigma_k^{(m)}\Big) = q_{\boldsymbol{\varsigma}_k}(\zeta_k), \end{aligned} \tag{4.3.21}$$

where the outer measure is the $\hat{\zeta}_k$-conditional law that EIVA targets and C_{-k} collects the current values of all parameters in blocks other than the column-k temporal and spatial blocks. This distribution is not exactly the full block-conditional marginal $p(\boldsymbol{\zeta}_k \mid y_{1:T}, C_{-k})$, because the EIVA particles are drawn from the conditional distribution of $\boldsymbol{\eta}_k$ given the current value $\widehat{\boldsymbol{\zeta}}_k$, rather than from the marginal distribution of $\boldsymbol{\eta}_k$ with $\boldsymbol{\zeta}_k$ integrated out. This construction is not the exact block-conditional marginal $p(\boldsymbol{\zeta}_k \mid y_{1:T}, C_{-k})$, because its outer measure conditions on the current plug-in value $\widehat{\boldsymbol{\zeta}}_k$. Reaching a fixed point makes the update summaries self-consistent but does not remove this conditioning. The approximation is expected to be most accurate when, near the fixed point, the conditional distribution of $\boldsymbol{\eta}_k$ changes only weakly with $\boldsymbol{\zeta}_k$. § 4.4 and 4.5 assess that approximation empirically against MCMC.

When distinct trajectory particles lead to distinct conditional spatial modes, the mixture in Eq. (4.3.21) can represent skewness, curvature, or local multimodality through variation between components. A single joint Gaussian can represent linear cross-block dependence and a nonzero variance of conditional means, but it constrains those conditional means to be affine functions of the conditioning variables and its conditional covariance to be constant. The mixture indexed by particles relaxes those restrictions by allowing both the mode and local curvature to vary across trajectories. A mean-field family remains more restrictive because it removes the spatio-temporal dependence.

Per-particle Newton solver

For each EIVA particle m, the conditional mode is the minimizer of the negative log-posterior

$$f^{(m)}(\zeta_k) = -\sum_{s,t} \log p_{\mathrm{zi}}(y_{s,t} \mid \lambda_{s,t}(\zeta_k, \boldsymbol{\eta}_k^{(m)}), \hat{\pi}_{s,t}) - \log \pi(\zeta_k \mid \xi_k), \tag{4.3.22}$$

where $\pi(\zeta_k \mid \xi_k)$ is the prior on ζ_k implied by the Beta prior on w_k and the standard-normal prior on the non-centered horseshoe variables. The log-prior includes the Jacobian for the $\mathrm{logit}(w_k)$ transformation, contributing $\log[w_k(1 - w_k)]$ to the prior density. Newton's method on $f^{(m)}$ uses the gradient and Hessian

$$\mathbf{g}^{(m)}(\zeta_k) = \nabla f^{(m)}(\zeta_k), \qquad \mathbf{H}^{(m)}(\zeta_k) = \nabla^2 f^{(m)}(\zeta_k), \tag{4.3.23}$$

in which the chain rule propagates through both the intensity $\lambda_{s,t} = \lambda_{s,t}(\zeta_k, \boldsymbol{\eta}_k^{(m)})$ and the filtered activation probability $\hat{\pi}_{s,t} = \hat{\pi}_{s,t}(\zeta_k, \boldsymbol{\eta}_k^{(m)})$. Since $f^{(m)}$ is the negative log posterior, the Laplace covariance is based on the inverse Hessian of $f^{(m)}$ at its local minimum. The dependence of $\hat{\pi}_{s,t}$ on ζ_k propagates through the forward filtering recursion of § 4.3.4, and its contribution to the gradient is computed by backpropagating through that recursion in a single pass at $O(T)$ cost per location.

The Poisson contribution to $f^{(m)}$ is concave in $\lambda_{s,t}$ for entries with $y_{s,t} > 0$, and $\lambda_{s,t}$ is linear in $\alpha_{\cdot,k}$, so the corresponding terms contribute log-concavity in $\alpha_{\cdot,k}$. The zero-inflation marginalization, however, introduces a small convex contribution from entries with $y_{s,t} = 0$, of magnitude scaling with $\hat{\pi}_{s,t}(1 - \hat{\pi}_{s,t})$. When activation probabilities are bounded away from zero across most observations, the Hessian of the negative log posterior at the conditional mode remains positive definite under the priors on $\boldsymbol{\zeta}_k$ and the softmax-like transformation $\boldsymbol{\zeta}_k \mapsto \alpha_{\cdot,k}$ near the mode. In regimes where

activation probabilities are heterogeneous and substantial mass concentrates on zero, the raw Hessian may be indefinite at intermediate iterates and, in extreme cases, at the converged mode itself. To guarantee a descent direction at every step and a well-defined Laplace covariance, the Hessian is replaced by a Levenberg-Marquardt-damped surrogate

$$\mathbf{A}_{\delta}^{(m)}(\zeta_k) = \mathbf{H}^{(m)}(\zeta_k) + \delta\,\mathbf{I}, \tag{4.3.24}$$

where $\delta \geq 0$ is the smallest value along a geometric schedule $\{0, 10^{-8}, 10^{-7}, \ldots, \delta_{\max}\}$ such that $\mathbf{A}_{\delta}^{(m)} \succ 0$. The Newton step at iterate $\zeta^{(\ell)}$ takes the form

$$\zeta^{(\ell+1)} = \zeta^{(\ell)} - \kappa^{(\ell)} \left[\mathbf{A}_{\delta}^{(m)}\right]^{-1} \mathbf{g}^{(m)}(\zeta^{(\ell)}), \tag{4.3.25}$$

where $\kappa^{(\ell)} \in (0, 1]$ is chosen by backtracking line search, starting at $\kappa^{(\ell)} = 1$ and halving until $f^{(m)}(\zeta^{(\ell+1)}) < f^{(m)}(\zeta^{(\ell)})$. The iteration terminates when $\|\mathbf{g}^{(m)}(\zeta^{(\ell)})\|_{\infty} < \epsilon_{\text{tol}}$ or after N_{Newton} iterations. In typical activation regimes for daily disease surveillance data, convergence requires two to five iterations. At the converged mode, the Laplace covariance is

$$\Sigma_k^{(m)} = \left[\mathbf{A}_{\delta^*}^{(m)}(\zeta_k^{*(m)})\right]^{-1}, \tag{4.3.26}$$

with δ^* the smallest schedule value achieving positive definiteness at the mode. The algorithm reports the damping value used at convergence as a diagnostic, and a non-zero δ^* flags columns where the Laplace approximation may be unreliable.

Mixture moments and the conditional variance decomposition

The first two moments of the mixture in Eq. (4.3.18) are

$$\mathrm{E}_{q_{\varsigma_k}}[\zeta_k] = \sum_{m=1}^{M} \bar{\varpi}_k^{(m)}\, \zeta_k^{*(m)}, \tag{4.3.27}$$

$$\mathrm{Var}_{q_{\varsigma_k}}[\zeta_k] = \underbrace{\sum_{m=1}^{M} \bar{\varpi}_k^{(m)}\, \Sigma_k^{(m)}}_{\text{within-particle}} + \underbrace{\sum_{m=1}^{M} \bar{\varpi}_k^{(m)} \big(\zeta_k^{*(m)} - \bar{\zeta}_k\big)\big(\zeta_k^{*(m)} - \bar{\zeta}_k\big)'}_{\text{between-particle}}, \tag{4.3.28}$$

where $\bar{\zeta}_k = \mathrm{E}_{q_{\varsigma_k}}[\zeta_k]$. The decomposition in Eq. (4.3.28) is the finite-particle realization of the law of total variance applied to the column-conditional variational joint in Eq. (4.3.21):

$$\mathrm{Var}_{\tilde{q}_k}(\zeta_k) = \mathrm{E}_{\boldsymbol{\eta}_k}\big[\mathrm{Var}(\zeta_k \mid \boldsymbol{\eta}_k,\, y_{1:T},\, C_{-k})\big] + \mathrm{Var}_{\boldsymbol{\eta}_k}\big[\mathrm{E}(\zeta_k \mid \boldsymbol{\eta}_k,\, y_{1:T},\, C_{-k})\big], \tag{4.3.29}$$

where the outer expectation and variance are over the column-conditional outer measure $p(\boldsymbol{\eta}_k \mid y_{1:T},\, \hat{\zeta}_k,\, C_{-k})$. The first term aggregates the local conditional curvatures across particles, with each $\Sigma_k^{(m)}$ approximating $\mathrm{Var}(\zeta_k \mid \boldsymbol{\eta}_k^{(m)}, y_{1:T},\, C_{-k})$ through the local Laplace expansion. The second term captures the spread of conditional modes induced by the EIVA proposal's exploration of the trajectory space. This identity is not unique to the mixture. A correlated joint Gaussian also splits into an average conditional variance and a variance of the conditional means. The potential advantage of the mixture is instead its ability to represent nonlinear changes in conditional location and curvature across particles. In regimes where these changes are substantial, the between-particle term may contribute importantly to the marginal spread, and § 4.4 reports its magnitude. A single local Gaussian approximation represents the curvature around one mode. By contrast, the mixture in Eq. (4.3.18), with moments in Eqs. (4.3.27) and (4.3.28), retains

both the average within-particle curvature and the between-particle variation in the conditional modes. This added flexibility is relevant when those modes or curvatures vary nonlinearly with the temporal trajectory, while a sufficiently accurate correlated Gaussian may remain adequate when the dependence is approximately elliptical.

EIVA proposal centering and the block-coordinate cycle within column k

The EIVA proposal in § 4.3.1 is built from the current trajectory $\hat{r}_{k,t}$ and the current spatial parameter vector $\hat{\zeta}_k$. Within column k at iteration i of the block-coordinate algorithm, both are set from the column's mixture at iteration $i-1$. Specifically, $\hat{\zeta}_k$ is the previous mixture mean $\bar{\zeta}_k^{(i-1)}$, and $\hat{r}_{k,t}$ is the previous EIVA-weighted mean trajectory $\sum_m \bar{\varpi}_k^{(m,i-1)} r_{k,t}^{(m,i-1)}$. The column-$k$ update then proceeds as follows. The EIVA proposal in Eq. (4.3.10) is built from the current values. Particles are drawn and importance weights are computed via Eq. (4.3.13). For each particle, Newton's method on $f^{(m)}$ returns $\zeta_k^{*(m)}$ and $\Sigma_k^{(m)}$. The mixture is assembled and the current values are updated for the next column or the next iteration. Centering the EIVA proposal on the mixture mean ensures that, as $\bar{\zeta}_k^{(i)}$ converges, the proposal aligns with the block-conditional posterior at the conditional mean.

4.3.3. Approximate moment updates for the horseshoe scales and auxiliaries

The horseshoe block consists of the local scales $\gamma_{s,k}$, the global scale τ_k, and the Makalic–Schmidt auxiliaries $a_{s,k}$ and b_k. The approximation for the horseshoe block takes the product form $q(\xi_k) = q(\tau_k^2)q(b_k)\prod_{s\neq k} q(\gamma_{s,k}^2)q(a_{s,k})$, with each factor taken

to be inverse-gamma.

The fitted model departs from the setting in which these factors would be conjugate, in two ways. The regularized horseshoe couples the scales nonlinearly through $\tilde{\gamma}_{s,k}^2 = c^2\gamma_{s,k}^2/(c^2 + \tau_k^2\gamma_{s,k}^2)$, which breaks the conditional conjugacy that the Makalic–Schmidt augmentation supplies for the unregularized horseshoe. The spatial block is also written in non-centered form as $\theta_{s,k} = \tau_k\tilde{\gamma}_{s,k}\varepsilon_{s,k}$ with $\varepsilon_{s,k} \sim N(0, 1)$, so under a factorization in $(\varepsilon, \tau, \gamma)$ the likelihood depends on the scales through a deterministic nonlinear map.

The algorithm therefore updates the scales and auxiliaries by the conjugate moment formulas of the centered, unregularized prior $\theta_{s,k} \mid \tau_k^2, \gamma_{s,k}^2 \sim N(0, \tau_k^2\gamma_{s,k}^2)$, which transfer shrinkage information between the spatial and scale blocks at negligible cost. The slab cap c enters the spatial Newton problem through the effective precision $1/(\tilde{\gamma}_{s,k}^2\tau_k^2) = 1/(\gamma_{s,k}^2\tau_k^2) + 1/c^2$, with scales evaluated at their current point summaries. The approximation is most plausible when the slab regularization is inactive for most pathways, and § 4.4 reports how the resulting shrinkage compares with the MCMC reference.

Under that centered, unregularized prior, the inverse-gamma factors have shape parameters 1 for auxiliaries, 1 for the local scales, and $S/2$ for the global scale. The rate updates are

$$\text{rate}\big[q^*(a_{s,k})\big] = 1 + \mathrm{E}_q[1/\gamma_{s,k}^2], \tag{4.3.30}$$

$$\text{rate}\big[q^*(b_k)\big] = \tau_{0,k}^{-2} + \mathrm{E}_q[1/\tau_k^2], \tag{4.3.31}$$

$$\text{rate}\big[q^*(\gamma_{s,k}^2)\big] = \mathrm{E}_q[1/a_{s,k}] + \tfrac{1}{2}\,\mathrm{E}_q[\theta_{s,k}^2]\,\mathrm{E}_q[1/\tau_k^2], \tag{4.3.32}$$

$$\text{rate}\big[q^*(\tau_k^2)\big] = \mathrm{E}_q[1/b_k] + \tfrac{1}{2}\sum_{s\neq k}\mathrm{E}_q[\theta_{s,k}^2]\,\mathrm{E}_q[1/\gamma_{s,k}^2], \tag{4.3.33}$$

with the inverse-gamma expectations given by $\mathrm{E}_q[1/a_{s,k}] = 1/\text{rate}[q^*(a_{s,k})]$, $\mathrm{E}_q[1/b_k] =$

$1/\mathrm{rate}[q^*(b_k)]$, $\mathrm{E}_q[1/\gamma_{s,k}^2] = 1/\mathrm{rate}[q^*(\gamma_{s,k}^2)]$, and $\mathrm{E}_q[1/\tau_k^2] = (S/2)/\mathrm{rate}[q^*(\tau_k^2)]$.

The plug-in second moment $\mathrm{E}_q[\theta_{s,k}^2]$ that drives the scale updates is computed from the mixture on the spatial block as

$$\mathrm{E}_q[\theta_{s,k}^2] = \tilde{\gamma}_{s,k}^2\,\tau_k^2\,\mathrm{E}_q[\varepsilon_{s,k}^2], \qquad \mathrm{E}_q[\varepsilon_{s,k}^2] = \big(\bar{\zeta}_k^{[s]}\big)^2 + \big[\mathrm{Var}_{q_{\zeta_k}}(\zeta_k)\big]_{[s,s]}. \tag{4.3.34}$$

Here the regularized scale $\tilde{\gamma}_{s,k}^2$ and the global scale τ_k^2 enter as plug-in summaries from the preceding scale update, while $\bar{\zeta}_k^{[s]}$ and $\big[\mathrm{Var}_{q_{\zeta_k}}(\zeta_k)\big]_{[s,s]}$ are the mean and variance of the coordinate corresponding to $\varepsilon_{s,k}$ in the spatial mixture. The expression is a moment-mathching bridge between the non-centered spatial representation and the centered working updates. It should not be interpreted as a derivation of the centered conjugate conditional from the non-centered model.

After each update, the horseshoe scales are propagated back to the model state so that the next iteration's Newton–Laplace problem uses the updated shrinkage. The point summaries used in the spatial block are

$$\gamma_{s,k} \leftarrow \big[\mathrm{E}_q[1/\gamma_{s,k}^2]\big]^{-1/2}, \qquad \tau_k \leftarrow \big[\mathrm{E}_q[1/\tau_k^2]\big]^{-1/2}, \tag{4.3.35}$$

and the model's $\theta_{s,k}$ and $\alpha_{s,k}$ are recomputed from the updated scales and the unchanged $\varepsilon_{s,k}$ values through $\theta_{s,k} = \tau_k\,\tilde{\gamma}_{s,k}\,\varepsilon_{s,k}$.

4.3.4. Forward filtering for the zero-inflation indicators

Rather than introducing a variational factor over the binary indicators $z_{s,t}$, the PLVA marginalizes them analytically. This is sufficient for the PLVA because the EIVA weights and the Newton–Laplace updates require only the marginalized observation density and its derivatives, not posterior samples of the latent indicators. By contrast, the MCMC

sampler of Chapter 3 uses forward-filtering backward-sampling because it explicitly samples the indicator trajectories.

For each location s, define $\pi_{s,t|t-1} = p(z_{s,t} = 1 \mid y_{s,1:t-1}, \lambda_{s,1:t-1})$ as the predicted activation probability before observing $y_{s,t}$, and $\pi_{s,t|t} = p(z_{s,t} = 1 \mid y_{s,1:t}, \lambda_{s,1:t})$ as the filtered activation probability after observing $y_{s,t}$. The Markov transition model implies

$$\pi_{s,t|t-1} = p_{01}(1 - \pi_{s,t-1|t-1}) + p_{11}\pi_{s,t-1|t-1}, \tag{4.3.36}$$

where $p_{01} = p(z_{s,t} = 1 \mid z_{s,t-1} = 0) = \text{logit}^{-1}(\beta_0)$ and $p_{11} = p(z_{s,t} = 1 \mid z_{s,t-1} = 1) = \text{logit}^{-1}(\beta_0 + \beta_1)$. The corresponding one-step-ahead marginalized observation density is

$$p_{\text{zi}}(y_{s,t} \mid \lambda_{s,t}, \pi_{s,t|t-1}) = \pi_{s,t|t-1}\,\text{Pois}(y_{s,t} \mid \lambda_{s,t}) + (1 - \pi_{s,t|t-1})\mathbf{1}(y_{s,t} = 0). \tag{4.3.37}$$

The filtering update is then

$$\pi_{s,t|t} = \frac{\pi_{s,t|t-1}\,\text{Pois}(y_{s,t} \mid \lambda_{s,t})}{p_{\text{zi}}(y_{s,t} \mid \lambda_{s,t}, \pi_{s,t|t-1})}. \tag{4.3.38}$$

Equations (4.3.36) and (4.3.38) are applied forward in time for each location s, starting from an initial probability $\pi_{s,0|0}$.

The log marginal contribution of location s is therefore

$$\ell_s^{\text{zi}} = \sum_{t=1}^{T} \log p_{\text{zi}}(y_{s,t} \mid \lambda_{s,t}, \pi_{s,t|t-1}), \tag{4.3.39}$$

and the full marginalized log likelihood is $\sum_{s=1}^{S} \ell_s^{\text{zi}}$. This likelihood is the quantity used in the EIVA importance weights and in the Newton–Laplace updates for each particle.

The gradient of $\log p_{\text{zi}}(y_{s,t} \mid \lambda_{s,t}, \pi_{s,t|t-1})$ with respect to the spatial parameters $\boldsymbol{\zeta}_k$ has two contributions. The first comes from the direct dependence of $\lambda_{s,t}$ on $\boldsymbol{\zeta}_k$. The

second comes from the dependence of the predicted activation probability $\pi_{s,t|t-1}$ on earlier intensities $\lambda_{s,1:t-1}$ through the filtering recursion. We compute the full gradient by differentiating through the forward filter. A forward sweep gives the probabilities $\{\pi_{s,t|t-1}, \pi_{s,t|t}\}_{t=1}^{T}$, and a reverse sweep accumulates the derivatives with respect to the parameters. This costs $O(T)$ per location for a fixed intensity trajectory.

4.3.5. Stochastic gradient updates for global parameters

The global parameters $\vartheta_{\text{glb}} = (\log \mu, \beta_0, \beta_1)$ enter the intensity in Eq. (4.2.1) and the zero-inflation transition probabilities additively across all source columns and time points. The variational family for this block follows the hybrid variational approach of Loaiza-Maya et al. (2022) that was used for the univariate DGTF class in Chapter 2. In the present network setting, the exact conditional distribution of the latent trajectories is not available, so we use the current column-wise EIVA particle approximations to construct a stochastic update for the global parameters. We approximate the marginal posterior of $\boldsymbol{\vartheta}_{\text{glb}}$ by $q^0_{\boldsymbol{\varsigma}_{\text{glb}}}(\boldsymbol{\vartheta}_{\text{glb}})$, where $\boldsymbol{\varsigma}_{\text{glb}}$ denotes the corresponding variational parameters. All components of ϑ_{glb} are on unconstrained scales, so a multivariate normal approximation applies directly without an intermediate transformation.

If the marginal likelihood $p(y_{1:T} \mid \boldsymbol{\vartheta}_{\text{glb}})$ were available, the global variational parameters could be updated by maximizing the ideal marginal evidence lower bound

$$\mathcal{L}(\boldsymbol{\varsigma}_{\text{glb}}) = E_{q^0_{\boldsymbol{\varsigma}_{\text{glb}}}}\left[\log p(\boldsymbol{\vartheta}_{\text{glb}}, y_{1:T}) - \log q^0_{\boldsymbol{\varsigma}_{\text{glb}}}(\boldsymbol{\vartheta}_{\text{glb}})\right]. \tag{4.3.40}$$

In the network Hawkes-type model this marginal likelihood is not tractable because it requires integration over the coupled latent reproduction trajectories. We therefore use the column-wise EIVA particle sets from the current PLVA iteration to construct a

stochastic pseudo-likelihood surrogate for the global parameter update.

The variational distribution $q^0_{\boldsymbol{\varsigma}_{\text{glb}}}$ takes the form of a Gaussian with factor covariance structure,

$$q^0_{\boldsymbol{\varsigma}_{\text{glb}}}(\vartheta_{\text{glb}}) = N\Big(\vartheta_{\text{glb}} \mid \boldsymbol{\mu}_\varsigma,\ \boldsymbol{B}_\varsigma \boldsymbol{B}'_\varsigma + \boldsymbol{D}_\varsigma\Big), \tag{4.3.41}$$

where $\boldsymbol{\mu}_\varsigma$ is the mean vector, $\boldsymbol{B}_\varsigma$ is an $m \times k$ loading matrix with m the number of global parameters and $k \leq m$ the number of latent factors, and $\boldsymbol{D}_\varsigma$ is a diagonal matrix capturing residual variance. The factor structure has $O(mk+m)$ parameters and captures the dominant posterior dependencies between the global parameters at a cost much lower than a dense covariance (Loaiza-Maya et al., 2022). The variational parameters for the global block are $\boldsymbol{\varsigma}_{\text{glb}} = (\boldsymbol{\mu}_\varsigma,\ \boldsymbol{B}_\varsigma,\ \boldsymbol{D}_\varsigma)$.

The ideal bound in Eq. (4.3.40) cannot be computed in closed form because the marginal log-likelihood $\log p(y_{1:T} \mid \boldsymbol{\vartheta}_{\text{glb}})$ is intractable. For the ideal marginal objective, Fisher's identity gives

$$\nabla_{\vartheta_{\text{glb}}} \log p(y_{1:T} \mid \vartheta_{\text{glb}}) = \mathrm{E}_{p(\boldsymbol{\eta} \mid y_{1:T},\, \vartheta_{\text{glb}})}\big[\nabla_{\vartheta_{\text{glb}}} \log p(y_{1:T} \mid \boldsymbol{\eta},\ \vartheta_{\text{glb}})\big], \tag{4.3.42}$$

which uses that the prior on $\boldsymbol{\eta}$ does not depend on ϑ_{glb}.

Exact sampling from the joint conditional $p(\boldsymbol{\eta} \mid y_{1:T},\ \vartheta_{\text{glb}})$ is itself intractable, and the algorithm approximates it by the pseudo-likelihood in the sense of Besag (1975),

$$\tilde{p}(\boldsymbol{\eta} \mid y_{1:T},\ \vartheta_{\text{glb}}) = \prod_{k=1}^{S} p(\boldsymbol{\eta}_k \mid y_{1:T},\ \hat{\boldsymbol{\zeta}}_k,\ C_{-k}), \tag{4.3.43}$$

which is the same column-conditional law that the EIVA particle sets target in Eq. (4.3.20). At each iteration of SGA, the global parameters are drawn $\vartheta_{\text{glb}} \sim q^0_{\boldsymbol{\varsigma}_{\text{glb}}}$ and a joint latent trajectory is assembled from the particle sets produced by the column updates.

For each column $k = 1, \ldots, S$, draw $m_k^* \sim \mathrm{Cat}(\bar{\varpi}_k^{(1)}, \ldots, \bar{\varpi}_k^{(M)})$ and set $\boldsymbol{\eta}_k^* = \boldsymbol{\eta}_k^{(m_k^*)}$ with $r_{k,t}^* = \sqrt{W} \sum_{j=0}^{t-1} \eta_{k,j}^*$. The joint draw $(\boldsymbol{\eta}_1^*, \ldots, \boldsymbol{\eta}_S^*)$ is a single sample from the pseudo-likelihood in Eq. (4.3.43). It differs from a draw of $p(\boldsymbol{\eta} \mid y_{1:T}, \vartheta_{\mathrm{glb}})$ through the cross-column coupling that the pseudo-likelihood factorization omits. The resulting approximation error relative to the ideal gradient in is the price of the column-wise factorization that the rest of the algorithm relies on, and in regimes where the column-conditional posterior of $\boldsymbol{\eta}_k$ is approximately invariant to the spatial weights of other columns the bias is small.

The intensity at the joint draw is

$$\lambda_{s,t}^*(\vartheta_{\mathrm{glb}}) = \exp(\log \mu) + \sum_{k=1}^{S} \hat{\alpha}_{s,k} \sum_{l=1}^{t-1} h(r_{k,l}^*)\, \phi_{t-l}\, y_{k,l}, \tag{4.3.44}$$

where $\widehat{\alpha}_{s,k}$ is the current spatial weight from the column updates. For the same draw (β_0, β_1) of the global parameters, the forward filter of § 4.3.4 gives the predicted activation probabilities $\pi_{s,t|t-1}^*(\boldsymbol{\vartheta}_{\mathrm{glb}})$. of § 4.3.4 using λ^* together with the draw (β_0, β_1) of the global parameters. Combining the Fisher identity in Eq. (4.3.42) with the pseudo-likelihood draw motivates the pseudo-likelihood stochastic objective

$$\widehat{\mathcal{L}}_{\mathrm{pl}}(\boldsymbol{\varsigma}_{\mathrm{glb}}; \boldsymbol{\eta}^*) = \log p(\vartheta_{\mathrm{glb}}) + \sum_{s,t} \log p_{\mathrm{zi}}\left(y_{s,t} \mid \lambda_{s,t}^*(\boldsymbol{\vartheta}_{\mathrm{glb}}), \pi_{s,t|t-1}^*(\boldsymbol{\vartheta}_{\mathrm{glb}})\right) - \log q_{\boldsymbol{\varsigma}_{\mathrm{glb}}}^0(\vartheta_{\mathrm{glb}}), \tag{4.3.45}$$

whose gradient with respect to $\boldsymbol{\vartheta}_{\mathrm{glb}}$ at the joint draw is a single-sample score for the pseudo-likelihood surrogate in Eq. (4.3.43). The summed log-density in Eq. (4.3.45) is the complete-data log-likelihood at $\boldsymbol{\eta}^*$ rather than an estimator of $\log p(y_{1:T} \mid \vartheta_{\mathrm{glb}})$; the connection to the bound is through the gradient, not through the scalar value.

The variational parameters are updated by SGA on the pseudo-likelihood objec-

tive, $\boldsymbol{\varsigma}_{\text{glb}}^{(i+1)} = \boldsymbol{\varsigma}_{\text{glb}}^{(i)} + \varrho^{(i)} \cdot \nabla_{\boldsymbol{\varsigma}_{\text{glb}}} \widehat{\mathcal{L}}_{\text{pl}}$, with step size $\varrho^{(i)}$ updated recursively by the ADADELTA scheme following Loaiza-Maya et al. (2022). The reparameterization Jacobian $\partial \vartheta'_{\text{glb}} / \partial \boldsymbol{\varsigma}_{\text{glb}}$ and the score $\nabla_{\vartheta_{\text{glb}}} \log q^0_{\boldsymbol{\varsigma}_{\text{glb}}}$ admit generic closed-form expressions for the Gaussian with factor covariance structure. The gradient of the complete-data log-likelihood with respect to $\log \mu$ flows through the additive baseline in the intensity, contributing $\sum_{s,t} (\partial \log p_{\text{zi}} / \partial \lambda_{s,t}) \cdot \exp(\log \mu)$. The gradient with respect to β_0 and β_1 flows through the filtered activation probability $\hat{\pi}^*_{s,t}$ and requires backpropagation through the forward filter recursion of § 4.3.4.

Convergence of the global block is monitored through the trajectory of a pseudo-likelihood diagnostic across iterations, supplemented by the column-wise ESSs from § 4.3.1. We plot

$$\widehat{\mathcal{L}}_{\text{pl}}^{(i)} = \sum_{k=1}^{S} \widehat{\log p_k}^{(i)}, \qquad \widehat{\log p_k}^{(i)} = \log\Big(\tfrac{1}{M} \textstyle\sum_{m=1}^{M} \tilde{\varpi}_k^{(m,i)} \Big), \tag{4.3.46}$$

the self-normalized importance sampling estimator of the column-conditional log marginal likelihood summed across columns, evaluated at the current state at iteration i. The quantity in Eq. (4.3.46) is a Monte Carlo estimate of $\log \prod_k p(y_{1:T} \mid \hat{\boldsymbol{\zeta}}_k, C_{-k})$, the pseudo-likelihood evidence introduced in Eq. (4.3.43). It is *not* a lower bound on the model log marginal likelihood $\log p(y_{1:T})$, since the pseudo-likelihood factorization replaces the joint conditional distribution of $\boldsymbol{\eta}$ by a product of column-conditional laws. We therefore call $\widehat{\mathcal{L}}_{\text{pl}}^{(i)}$ the *pseudo-likelihood objective*. It is used only to detect when the block-coordinate iteration has stopped moving, so that the loop can be stopped once the traces have settled. Its value does not measure how well the converged variational distribution fits the data. The global block is updated once per iteration after the column updates, with the column updates providing the latent state particle sets that the joint

draw samples from.

The PLVA is best viewed as a blockwise approximation related to structured variational and message passing methods, including Hoffman et al. (2013), Wand (2017), and expectation propagation (Minka, 2001). Its approximating families are chosen to retain the spatial–temporal dependence induced by the multiplicative parameterization, but its complete update cycle is not derived as optimization of a single Kullback–Leibler objective or a lower bound on $\log p(y_{1:T})$. Methodological validity therefore rests on a transparent definition of the conditional approximations and on empirical validation against MCMC, rather than on a global variational optimality claim.

The quality of the fit is assessed by the column-wise EIVA effective sample sizes, the stability of the blockwise summaries, the agreement across initializations, and the held-out predictive scores and coverage reported in § 4.4 and 4.5 against the MCMC reference.

A bound on the evidence is within reach of this construction and is a natural next step. The importance weights the algorithm already stores are self-normalized, and the spatial approximation is a mixture indexed by those particles, so the joint proposal has no normalized density that can be evaluated at an arbitrary draw. Giving the EIVA proposal and the mixture a parametric form with an evaluable joint density would supply the importance ratio that the importance-weighted variational bound (IWAE) of Burda et al. (2016) requires, and would yield a lower bound on $\log p(y_{1:T})$ that tightens as M grows. § 4.6 returns to this as a direction for future work.

4.3.6. Block-coordinate algorithm and computational complexity

Algorithm 7 assembles the four blocks into a single block-coordinate optimization loop. Each iteration cycles through the S source columns, updating the temporal and spatial blocks at each column through the EIVA proposal, the Newton–Laplace mixture, and the current column summaries. After the column updates, the global block is updated by SGA on the pseudo-likelihood objective in Eq. (4.3.45), and the horseshoe scales and auxiliaries are updated by the moment formulas of § 4.3.3. The algorithm stops when the pseudo-likelihood diagnostic and the column-wise retention rate traces stabilize according to the rule described below. Once the variational parameters are fixed, the posterior samples are drawn using the procedure of Algorithm 8.

Algorithm 8 samples from the finite particle Laplace approximation actually constructed at the final iteration. Selecting one index m_k^* and retaining both its stored temporal particle and its associated Laplace component preserves the dependence encoded by the mixture. Running a fresh EIVA pass after sampling $\boldsymbol{\zeta}_k$ would instead define a different approximation and would break the stored particle–component pairing unless that new joint construction were specified separately. The paired rule also removes the additional EIVA cost during posterior sampling.

The cost of an iteration must therefore be evaluated with a dense $\mathbf{C}_k$. Within a column, the diagonal weights for each destination are first combined, and the precision contribution $\mathbf{C}_k'\mathbf{D}_k\mathbf{C}_k$ is then formed. Forming that matrix and factorizing it by Cholesky costs at most $O(T^3)$ per column. Drawing M particles, reconstructing trajectories, and evaluating exact intensities and forward-filtered observation densities cost $O(MSL_\phi T)$ per column. The spatial Newton solver costs $O\{N_{\text{Newton}}(SL_\phi T + S^3)\}$ for every particle, and assembling the mixture moments costs $O(MS^2)$ per column. Summing across the S

source columns gives

$$O\big(ST^3 + MS^2L_\phi T + MN_{\text{Newton}}(S^2L_\phi T + S^4) + MS^3\big). \tag{4.3.47}$$

The leading term in S is $O(MN_{\text{Newton}}S^4)$, from the Newton solves, and the dense temporal factorization contributes $O(ST^3)$, so this formulation is cubic rather than linear in the length of the series. A state-space implementation that exploits the Markov precision of the latent random walk would reduce the temporal term, at the cost of a separate derivation. The timings reported below hold for the values of S, T, and M that were run, and we do not extrapolate them beyond that range. For comparison, the block MCMC algorithm in Chapter 3 costs a similar amount per iteration but typically requires several thousand iterations for adequate mixing. The proposed algorithm instead uses block-coordinate optimization, so its runtime is mainly driven by the particle count M, the number of Newton steps in the spatial block, and the number of outer iterations required for stabilization. The empirical wall-clock comparison is reported in § 4.4.

This cost supports a limited claim about scale. The method is demonstrated for the moderate spatial networks and series lengths of the simulations and the ten-county application. Scaling to substantially longer series would require a state-space solver for the temporal block, and scaling to many more locations would require parallel column updates, sparsity or low-rank structure in the spatial Newton problems, or screening of candidate pathways before fitting.

The iteration bound N_{iter} in Algorithm 7 is fixed in advance. Each fit in this chapter uses $N_{\text{iter}} = 300$, and the monitored traces reported in § 4.4 stabilize well before that bound.

Algorithm 7: Particle Laplace variational approximation: optimization loop.

Input: Observations $\{y_{s,t}\}$; serial interval $\{\phi_l\}$; hyperparameters $(\tau_{0,k}, c, a_w, b_w, \ldots)$; EIVA particle count M; Newton settings $(N_{\text{Newton}}, \epsilon_{\text{tol}}, \delta_{\max})$; stochastic gradient settings.

Output: Spatial mixtures $\{q_k^*(\boldsymbol{\zeta}_k)\}_{k=1}^S$, horseshoe factors $\{q^*(\xi_k)\}_{k=1}^S$, and global variational parameters $\boldsymbol{\varsigma}_{\text{glb}}^*$.

Initialize $\bar{\zeta}_k^{(0)}$ from a generalized linear model fit; set $\xi_k^{(0)}$ at the prior modes; set $\hat{r}_{k,t}^{(0)} = 0$; set $\boldsymbol{\varsigma}_{\text{glb}}^{(0)}$ at prior means;

for $i = 1$ **to** N_{iter} **do**

  for $k = 1$ **to** S **do**

    Form the offset $\tilde{\mu}_{s,t}^{(-k)}$ via Eq. (4.3.5) from the current values;

    Construct the EIVA proposal g_k via Eqs. (4.3.10) and (4.3.11);

    Draw $\boldsymbol{\eta}_k^{(m)} \sim g_k$ for $m = 1, \ldots, M$, reconstruct $r_k^{(m)}$, and compute normalized importance weights $\bar{\varpi}_k^{(m)}$ via Eq. (4.3.13);

    for $m = 1$ **to** M **do**

      Set $\boldsymbol{\eta}_k = \boldsymbol{\eta}_k^{(m)}$ and $\zeta^{(0)} = \bar{\zeta}_k^{(i-1)}$;

      for $\ell = 0, 1, \ldots, N_{\text{Newton}}$ or until $\|\mathbf{g}^{(m)}(\zeta^{(\ell)})\|_\infty < \epsilon_{\text{tol}}$ **do**

        Compute $\mathbf{g}^{(m)}(\zeta^{(\ell)})$ and $\mathbf{H}^{(m)}(\zeta^{(\ell)})$;

        Form $\mathbf{A}_\delta^{(m)} = \mathbf{H}^{(m)} + \delta\mathbf{I}$ with the smallest $\delta \in \{0, 10^{-8}, \ldots, \delta_{\max}\}$ such that $\mathbf{A}_\delta^{(m)} \succ 0$;

        Take the Newton step with backtracking line search;

      Set $\zeta_k^{*(m)} = \zeta^{(\ell^*)}$ and $\Sigma_k^{(m)} = [\mathbf{A}_{\delta^*}^{(m)}]^{-1}$ at convergence;

    Assemble the mixture $q_k^{(i)}(\boldsymbol{\zeta}_k)$ from $\{(\boldsymbol{\zeta}_k^{*(m)}, \Sigma_k^{(m)}, \bar{\varpi}_k^{(m)})\}_{m=1}^M$; Compute $\bar{\zeta}_k^{(i)}$ via Eq. (4.3.27), update $\hat{r}_{k,t} = \sum_m \bar{\varpi}_k^{(m)} r_{k,t}^{(m)}$ and $\hat{\alpha}_{\cdot,k}$;

  `// Global block`

  Draw indices $m_k^* \sim \text{Cat}(\bar{\varpi}_k)$ for $k = 1, \ldots, S$ from the particle sets produced by the column updates to form the joint draw $\boldsymbol{\eta}^*$. Update the global variational parameters by SGA on the pseudo-likelihood objective in Eq. (4.3.45). Gradients with respect to (β_0, β_1) are computed by differentiating through the forward filter; `// Horseshoe block`

  Compute $\text{E}_q[\theta_{s,k}^2]$ via Eq. (4.3.34), update the inverse-gamma factors via Eqs. (4.3.30) to (4.3.33), and propagate the scales via Eq. (4.3.35);

  `// Convergence check`

  Track $\widehat{\mathcal{L}}_{\text{pl}}(\boldsymbol{\mu}_\varsigma^{(i)})$, the column-wise ESSs, and the mixture means $\bar{\zeta}_k$;

Algorithm 8: Particle Laplace variational approximation: posterior sampling.

Input: Variational parameters $\{\varsigma_k^*, \xi_k^*, \varsigma_{\text{glb}}^*\}_{k=1}^S$ from Algorithm 7; observations $\{y_{s,t}\}$; serial interval $\{\phi_l\}$; number of posterior draws N_{post}.

Output: Posterior samples $\{\zeta_k^{(j)}, \boldsymbol{\eta}_k^{(j)}, \vartheta_{\text{glb}}^{(j)}, \xi_k^{(j)}\}_{j=1}^{N_{\text{post}}}$.

for $j = 1$ **to** N_{post} **do**

  for $k = 1$ **to** S **do**

    Draw one stored component index $m_k^* \sim \text{Cat}(\bar{\varpi}_k^{(1)}, \ldots, \bar{\varpi}_k^{(M)})$;

    Set $\boldsymbol{\eta}_k^{(j)} = \boldsymbol{\eta}_k^{(m_k^*)}$ and $r_{k,t}^{(j)} = \sqrt{W} \sum_{j'=0}^{t-1} \eta_{k,j'}^{(j)}$;

    Draw $\zeta_k^{(j)} \sim N(\zeta_k^{*(m_k^*)}, \Sigma_k^{(m_k^*)})$ from the Laplace component paired with that same particle;

    Unpack $\zeta_k^{(j)}$ to update the column-k spatial weights $\alpha_{\cdot,k}^{(j)}$;

    Draw the horseshoe variables $\xi_k^{(j)}$ from the inverse-gamma factors of § 4.3.3;

  Draw $\vartheta_{\text{glb}}^{(j)} \sim q_{\varsigma_{\text{glb}}^*}^0$;

§ 4.4. Simulation studies

The simulation is designed to evaluate which posterior summaries the PLVA matches against the block MCMC sampler of Chapter 3, where it departs from that reference, and how much computation it saves. The data generating process contains no structural zeros, isolating the spatio-temporal coupling that the PLVA targets.

The evidence comes in two parts. A fit to one simulated dataset, reported in § 4.4.3 and 4.4.4, illustrates the posterior summaries and supplies the figures of this section. A replicated comparison against MCMC on ten independently simulated datasets, reported in § 4.4.5, provides the quantitative comparison between the two methods.

4.4.1. Simulation design

We compare the PLVA against the block MCMC sampler of Chapter 3 on the simulated dataset of Study 1 in § 3.4.2, which is drawn from the network Hawkes-type model and is illustrated in Fig. B.1. The simulator generates daily case counts $y_{s,t}$ at $S = 5$ destinations over $T = 210$ days. The data generating retention rates are $w = (0.9, 0.7, 0.5, 0.3, 0.3)$, the background intensity is $\mu = 2$, the gain function h is the softplus, and the latent reproduction trajectories follow Gaussian random walks with innovation variance $W = 0.01$. The serial interval $\{\phi_l\}$ is a discretized log-normal with mean 4.7 days and standard deviation 2.9 days, truncated at $L = 30$ days and renormalized to sum to one. The transmission weight matrix $\boldsymbol{\alpha}$ is column-stochastic with sparse and heterogeneous off-diagonal entries, generated without mobility covariates so that the off-diagonal structure is driven entirely by the deviation terms. The data are generated without structural zeros, so every activation indicator $z_{s,t}$ equals one. We hold out the last 10

days for forecast evaluation and fit both algorithms on the first $T_{\text{train}} = 200$ days.

Both algorithms fit the identical network Hawkes-type model of Chapter 3 under the identical prior specification, and differ only in the inference procedure. We use the prior $w_k \sim \text{Beta}(1, 1)$ for the retention rates and $\log \mu \sim N(0, 9)$ for the log baseline rate, and the regularized horseshoe on the off-diagonal deviations uses local scales $\gamma_{s,k} \sim C^+(0, 1)$, column-specific global scales $\tau_k \sim C^+(0, \tau_0)$, and slab scale $c^2 = 4$, exactly as in the simulation studies of Chapter 3. The zero-inflation coefficients β_0 and β_1 use the normal priors of Chapter 3. The zero-inflation block is not switched off. Although the data contain no structural zeros, both algorithms still estimate β_0 and β_1, and both recover a high activation probability, so the experiment also confirms that the component does no harm when structural zeros are absent. The innovation variance W is fixed at its data generating value, as in the simulation studies of Chapter 3.

The horseshoe global scale is fixed at $\tau_0 = 1$ before fitting. This is a deliberate, simulation-specific choice rather than the data-driven calibration $\tau_0 = p_0\sigma_y/\big((S - 1 - p_0)\sqrt{T}\big)$ of Piironen and Vehtari (2017) adopted in Chapter 3, whose inputs, an expected number of active pathways p_0 and a scale σ_y of the observed counts, are most natural for the application data. The reported quantities are not sensitive to the precise value of τ_0 as long as it is not set extremely small, because τ_0 enters only as the scale of the half-Cauchy prior on the column-specific global scale τ_k, which is heavy-tailed and lets the data override the prior. The same $\tau_0 = 1$ is used for both algorithms.

The PLVA is run at EIVA particle counts $M \in \{10, 20, 30, 50, 100\}$, where M is the number of EIVA particles introduced in § 4.3.1. Each configuration is run for 300 iterations, so that all configurations are compared at the same number of iterations. The spatial parameters are initialized by generalized linear model estimations and the

disturbance trajectories at zero, with the remaining Newton and stochastic gradient settings as in Algorithm 7. After the variational optimization completes, $N_{\text{post}} = 1000$ posterior samples are drawn via Algorithm 8 for posterior inference. The PLVA accuracy summaries reported in Table 4.1 are averaged across five independent runs at each M on this dataset, with identical settings but different random seeds. The posterior estimates shown in Figs. 4.2 to 4.5 are taken from a single one of the five $M = 50$ runs, rather than pooled across runs, and the posterior intervals shown in those figures should be read accordingly. In addition to the main comparison across M, we report a single PLVA run at $M = 50$ to illustrate the convergence diagnostics in Fig. 4.1.

For the replicated comparison of § 4.4.5, we simulated ten additional datasets. The retention rates w, the background intensity μ, the innovation variance W, and the serial interval $\{\phi_l\}$ are held at the values given above, while the off-diagonal deviations of $\boldsymbol{\alpha}$ and the simulated counts are drawn afresh for each dataset. We fit the PLVA at each particle count and the block MCMC sampler to all ten datasets under the settings described above, and compare every fit against the simulated truth of the dataset it was fitted to.

The MCMC reference runs the block sampler of Chapter 3 on the same data, model, and priors. The chain runs for 25,000 iterations after a 5,000 iteration burn-in, thinned to every fifth draw, matching the number of iterations used for the simulation studies of that chapter. Its convergence is monitored through the MCMC effective sample size on the retention rates w_k, reported in § 4.4.2. The MCMC accuracy summaries are averaged across three independent runs on this dataset with identical settings but different random seeds, and are computed from the retained posterior draws after convergence. All runs use a single CPU core. The wall clock times of Table 4.5 come from a separate set of

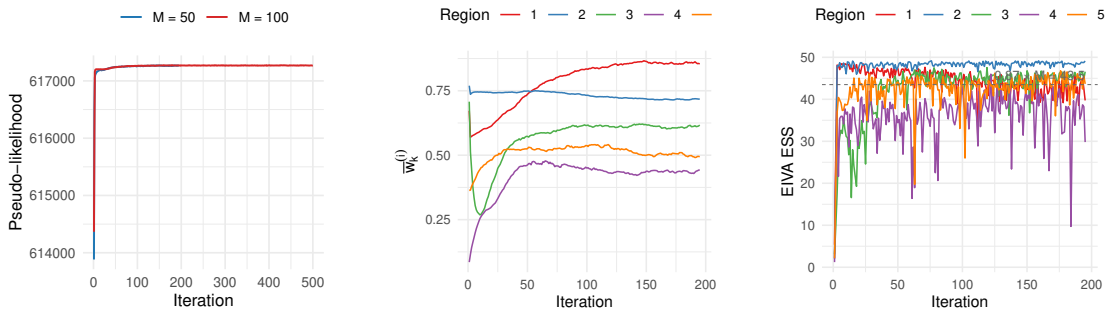
runs, five for the PLVA at each M and three for MCMC, each run on its own dataset simulated from the same data generating process.

4.4.2. Convergence diagnostics

Both algorithms reached stable estimates before the posterior analysis. The MCMC sampler produced an average MCMC effective sample size of 354 on the retention rates w_k, ranging from 168 to 446 across columns, in approximately 44 minutes of wall clock time. This is a chain autocorrelation diagnostic, distinct from the EIVA importance sampling ESS discussed below. The PLVA pseudo-likelihood objective trace $\widehat{\mathcal{L}}_{\text{pl}}$ of Eq. (4.3.46) settled within 30 iterations across all M settings, and the slowest converging w_k trace stabilized well before the 300th iteration.

The pseudo-likelihood objective signals when the iteration has stopped moving. Because the algorithm does not ascend a single global objective monotonically, a settled trace does not on its own establish that the stopped fit is a good approximation. Support for that comes from the coverage and predictive scores in § 4.4 and from the stability of the posterior summaries across the initialization schemes reported in § 4.4.7.

The EIVA importance sampler's effective sample size as a fraction of M ranged between 0.77 and 0.88 across configurations (Fig. 4.1). The mean column-wise EIVA ESS at convergence was 8.8 at $M = 10$, 17.5 at $M = 20$, 25.9 at $M = 30$, 38.6 at $M = 50$, and 85.9 at $M = 100$. These fractions indicate that the EIVA Gaussian proposal matches the column-conditional posterior reasonably well across particle counts, so M mainly controls the precision of the importance estimate rather than the stability of the proposal.



(a) Pseudo-likelihood. (b) Posterior mean $\bar{w}_k$. (c) EIVA ESS per column.

Figure 4.1.: PLVA convergence diagnostics on the simulated data. Panel (a) shows the pseudo-likelihood objective $\widehat{\mathcal{L}}_{\text{pl}}$ of Eq. (4.3.46) for two runs, one at $M = 50$ and one at $M = 100$. Panels (b) and (c) come from the $M = 50$ run alone, and show the posterior mean retention rate for each column and the EIVA effective sample size.

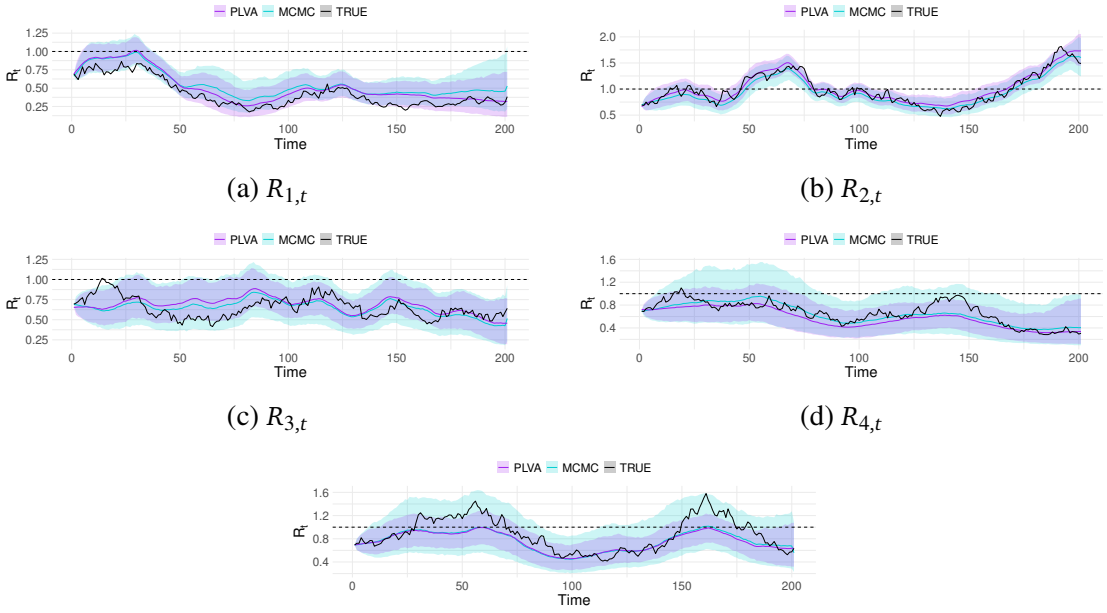
4.4.3. Parameter estimation

The estimates in this subsection and in § 4.4.4 come from the single simulated dataset of § 4.4.1 and are illustrative. § 4.4.5 reports the same comparison across ten independently simulated datasets.

We assess parameter estimation on three quantities of interest: the reproduction trajectory $R_{k,t}$, the retention rates w_k, and the spatial transmission matrix $\boldsymbol{\alpha}$. Across these targets, PLVA matched the MCMC reference on point estimates while producing narrower credible intervals.

Reproduction trajectories. Both methods tracked the true $R_{k,t}$ trajectories on the training window, as shown in Fig. 4.2. The CRPS on $R_{k,t}$ averaged 0.096 for MCMC and ranged from 0.09 to 0.10 across PLVA settings (Table 4.1). The MAE between the posterior median trajectory and the truth averaged 0.136 for MCMC and 0.12 to 0.14 for PLVA. The point estimates of the reproduction trajectories were of comparable accuracy

across methods. Empirical 95% credible interval coverage was 0.99 for MCMC, which was conservative relative to the nominal level, and 0.82 to 0.86 for PLVA. PLVA's intervals were narrower than MCMC's by roughly the same factor across all M values. § 4.4.5 finds the same shortfall across independently simulated datasets, which attributes it to the variational family and not to Monte Carlo noise from a small particle count.



(a) $R_{1,t}$ (b) $R_{2,t}$

(c) $R_{3,t}$ (d) $R_{4,t}$

(e) $R_{5,t}$

Figure 4.2.: Posterior median and 95% CI of reproduction number $R_{k,t}$ for each location.

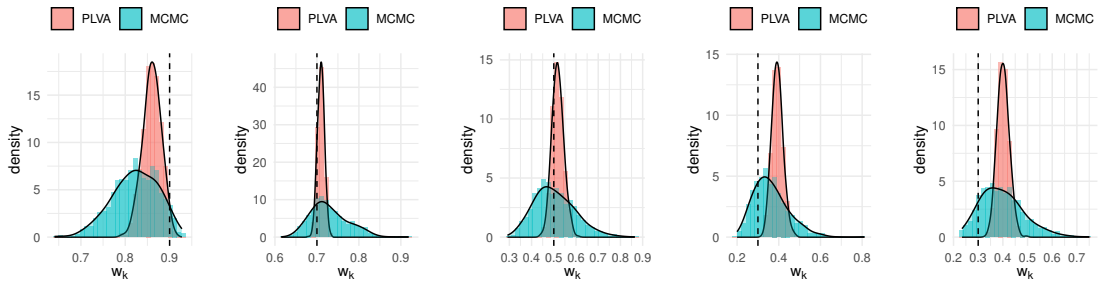
Retention rates. The MCMC posterior median for w_k recovered the truth on the well-identified columns, returning 0.81 and 0.72 against truths of 0.9 and 0.7. On the columns with low true values, the MCMC posterior median sat above the truth, returning $w_4 = 0.36$ and $w_5 = 0.40$ against truths of 0.30. PLVA matched MCMC on the three well-identified columns to within 0.04, with posterior medians of 0.86, 0.72, and 0.50. Its w_4 estimate of 0.41 sat higher again, while its w_5 estimate of 0.39 was comparable to

Table 4.1.: Absolute levels on the illustrative fit to the single simulated dataset of § 4.4.1, which also produces the figures of this section. The $R_{k,t}$ columns report CRPS on the posterior distribution of $R_{k,t}$, mean absolute deviation between the posterior median and the truth, and empirical coverage of the 95% credible intervals, each averaged across the 5 destinations, the training window, and the runs. The remaining columns report in-sample posterior predictive CRPS on $y_{s,t}$ averaged across the training window, and forecast CRPS at horizons of 1, 5, and 10 days averaged across the 5 destinations. Tables 4.2 to 4.4 report the comparison against MCMC across replicated datasets.

	$R_{k,t}$			CRPS on $y_{s,t}$			
Method	CRPS	MAE	Coverage	In-sample	$h = 1$	$h = 5$	$h = 10$
MCMC	0.096	0.136	0.99	3.39	5.34	7.23	12.51
PLVA, $M = 10$	0.09	0.13	0.85	3.37	5.13	7.58	13.47
PLVA, $M = 20$	0.09	0.12	0.86	3.38	4.96	7.33	12.89
PLVA, $M = 30$	0.10	0.14	0.82	3.37	5.05	7.50	13.19
PLVA, $M = 50$	0.09	0.13	0.85	3.38	4.96	7.42	13.11
PLVA, $M = 100$	0.09	0.13	0.85	3.38	4.98	7.39	13.06

MCMC's. Posterior 95% interval widths for w_k averaged 0.27 for MCMC and 0.08 for PLVA, so PLVA's intervals were roughly 2 to 5 times narrower across columns (Fig. 4.3). Both methods departed from the truth in the same direction on the low signal columns, so the displacement is not specific to the variational approximation. The PLVA estimate of w_4 stayed nearly constant across M values. § 4.4.5 shows that both the larger error and the narrow intervals on the retention rates persist across independently simulated datasets, which points to the remedies discussed in § 4.6.

Transmission weights. The recovered transmission matrix $\boldsymbol{\alpha}$ matched the qualitative network structure under both methods (Fig. 4.4). Both PLVA and MCMC produced diagonals close to their w_k estimates. The pointwise discrepancy between the PLVA and MCMC posterior mean matrices was at most 0.05 in absolute value across all entries.



(a) w_1 (true 0.9). (b) w_2 (true 0.7). (c) w_3 (true 0.5). (d) w_4 (true 0.3). (e) w_5 (true 0.3).

Figure 4.3.: Posterior densities for the retention rate w_k across the 5 destinations. PLVA at $M = 50$ in blue and MCMC in red. The solid vertical line marks the truth. The variational posterior is narrower than the MCMC posterior across all five columns, and is pulled slightly further toward the prior mode on w_4.

	1	2	3	4	5
1	0.90	0.12	0.20	0.16	0.24
2	0.01	0.70	0.13	0.17	0.12
3	0.00	0.03	0.50	0.18	0.18
4	0.01	0.13	0.10	0.30	0.17
5	0.09	0.03	0.07	0.19	0.30

(a) True $\boldsymbol{\alpha}$.

	1	2	3	4	5
1	0.86	0.12	0.18	0.15	0.20
2	0.03	0.72	0.11	0.14	0.14
3	0.03	0.02	0.50	0.14	0.14
4	0.04	0.13	0.11	0.41	0.14
5	0.04	0.02	0.10	0.15	0.39

(b) PLVA mean at $M = 50$.

	1	2	3	4	5
1	0.81	0.11	0.18	0.13	0.20
2	0.04	0.72	0.11	0.15	0.12
3	0.04	0.02	0.50	0.17	0.15
4	0.05	0.13	0.11	0.36	0.13
5	0.05	0.02	0.09	0.19	0.40

(c) MCMC mean.

	1	2	3	4	5
1	0.02	0.04	0.05	0.05	0.06
2	0.02	0.01	0.04	0.05	0.04
3	0.02	0.01	0.02	0.04	0.04
4	0.02	0.04	0.03	0.03	0.04
5	0.02	0.01	0.03	0.04	0.02

(d) PLVA SD.

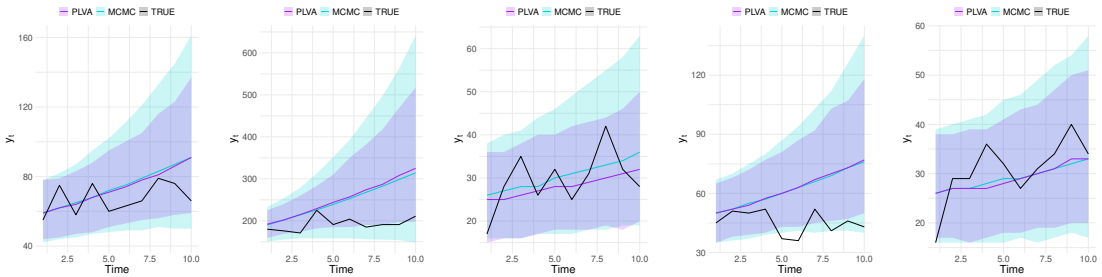
	1	2	3	4	5
1	0.07	0.03	0.09	0.09	0.09
2	0.03	0.04	0.07	0.10	0.07
3	0.03	0.01	0.09	0.11	0.07
4	0.03	0.01	0.05	0.10	0.06
5	0.03	0.01	0.05	0.10	0.08

(e) MCMC SD.

Figure 4.4.: Posterior summary of the transmission matrix $\boldsymbol{\alpha}$ at $M = 50$. The top row shows heatmaps of the true $\boldsymbol{\alpha}$, the PLVA posterior mean, and the MCMC posterior mean. The bottom row shows the posterior standard deviations under PLVA and MCMC.

4.4.4. Predictive performance

We evaluate in-sample posterior predictive fit on the training window and out-of-sample forecast accuracy on the held-out final 10 days. The in-sample CRPS on observed counts $y_{s,t}$ averaged 3.37 to 3.38 across PLVA settings and 3.39 for MCMC. Forecast CRPS on $y_{s,t}$ at horizon $h = 1$ day averaged 5.34 for MCMC and 4.96 to 5.13 for PLVA, so PLVA matched or improved on MCMC at every M. At $h = 5$ days the CRPS averaged 7.23 for MCMC and 7.33 to 7.58 for PLVA. At $h = 10$ days it averaged 12.51 for MCMC and 12.89 to 13.47 for PLVA (Table 4.1, with the forecast trajectories at each location shown in Fig. 4.5). Although $M = 20$ gave the best PLVA CRPS at the 5 and 10 day horizons, it led the next best particle count by under 1.5% at both, and the differences across M on the forecast metrics for this single dataset were small at every horizon. The ordering across M does not persist under replication, where $M = 50$ gave the lower forecast CRPS at both horizons (§ 4.4.5). On the ten replicated datasets of § 4.4.5, the forecast differences between the two methods are smaller than their variation across datasets at all three horizons.



(a) Location 1. (b) Location 2. (c) Location 3. (d) Location 4. (e) Location 5.

Figure 4.5.: One to ten day ahead forecast of $y_{s,t}$ on the held-out window for each of the 5 destinations. PLVA at $M = 50$ in purple and MCMC in cyan. Solid lines show the posterior median forecast, and shaded bands show 95% prediction intervals. The black line represents the observed trajectory.

4.4.5. Replicated comparison against MCMC

We repeated the comparison of § 4.4.3 and 4.4.4 on the ten independently simulated datasets described in § 4.4.1, fitting both methods to each dataset. The simulated series differ in scale from one dataset to the next, which makes absolute error metrics hard to compare across replications. For the error metrics we therefore report the percentage change of the PLVA value against the MCMC value on the same dataset, averaged over the ten datasets, with the standard deviation across datasets in parentheses. A positive change means the PLVA error was the larger of the two. Coverage and interval width are already comparable across datasets, so these are reported as levels.

Reproduction trajectories and in-sample fit. At $M = 50$, the MAE between the posterior median trajectory of $R_{k,t}$ and the truth changed by -10.3% (44.6%) against MCMC, a difference far smaller than its spread across datasets, so the two methods were of comparable accuracy on the point estimates. The CRPS on $R_{k,t}$ was 20.8% (29.8%) higher under the PLVA, and the empirical coverage of its 95% credible intervals was 0.79 (0.06) against the nominal 0.95 (Table 4.3). This coverage stayed between 0.79 and 0.81 at every particle count, matching the level reached in the small particle count study of Table 4.6. The shortfall on $R_{k,t}$ is therefore in the width of the credible intervals and not in the posterior median trajectory. On the observed counts, the in-sample posterior predictive CRPS changed by 0.5% (26.3%) and the empirical coverage of the 95% posterior predictive intervals was 0.96 (0.01), close to nominal at every particle count.

Spatial parameters. The two methods differed most on the spatial block. At $M = 50$ the MAE on the retention rates was 23.4% (19.5%) higher under the PLVA, its 95%

credible intervals covered the truth 0.52 of the time against 0.88 under MCMC, and their average width was 0.109 against 0.299, a factor of 2.8 narrower (Table 4.2). The transmission weights showed the same pattern less severely. The Frobenius norm of the error in $\boldsymbol{\alpha}$ was 5.3% (4.8%) higher under the PLVA, with coverage 0.72 against 0.97 and average width 0.156 against 0.270, a factor of 1.7 narrower. The MCMC intervals on $\boldsymbol{\alpha}$ were close to the nominal level across the ten datasets, so the reference is well calibrated on the quantity that determines the estimated network.

The under-coverage on the spatial block appeared in all ten datasets and at every particle count, so it is a property of the approximating family and not an artifact of one simulated realization. Beyond $M = 10$ the coverage on the retention rates moved between 0.48 and 0.64 without a trend, and the average interval width changed by less than 6%, so raising the particle count does not close the gap. Closing it calls for refinements of the variational family, a point that § 4.6 returns to with two concrete extensions.

Forecasting. The forecast comparison did not separate the two methods at any horizon. At $M = 50$ the forecast CRPS on the held-out counts changed by -5.4% (21.8%) at $h = 1$, 2.1% (18.3%) at $h = 5$, and 2.6% (14.1%) at $h = 10$, each small relative to its variation across datasets (Table 4.4). Empirical coverage of the 95% posterior predictive intervals was 1.00 at $h = 1$, 0.99 (0.02) at $h = 5$, and 0.97 (0.03) at $h = 10$, at or above the nominal level at every horizon and particle count.

The contrast between the calibrated predictive intervals and the narrow spatial intervals follows from how the two are constructed. The predictive distribution of a count combines parameter uncertainty with Poisson observation noise, and the observation

noise dominates at the counts simulated here, so a variational posterior that is too concentrated on the spatial block has little effect on the predictive scores. Credible intervals for the spatial parameters and for $R_{k,t}$ contain no observation noise term, which is why the under-coverage appears there.

Choice of the particle count. The replicated results also determine the particle count used in this chapter. The error metrics improve up to about $M = 50$ and change little beyond it. Between $M = 10$ and $M = 50$ the MAE on the retention rates falls from 26.3% to 23.4% above MCMC, the Frobenius error on $\boldsymbol{\alpha}$ from 9.7% to 5.3%, and the ten day forecast CRPS from 4.7% to 2.6%, whereas moving to $M = 100$ changes each of the three by about a percentage point or less. Coverage changes little with M on either spatial block or on $R_{k,t}$. The one quantity that keeps improving is the CRPS on $R_{k,t}$, which falls from 20.8% to 15.7% above MCMC between $M = 50$ and $M = 100$, at roughly twice the wall clock cost, since M enters the cost of an iteration linearly through Eq. (4.3.47). We therefore use $M = 50$ throughout the simulations of this chapter. Within this simulation setting, $M = 100$ is the better choice where the reproduction trajectories are the primary target, and $M = 20$ costs under half as much (§ 4.4.6), with forecast CRPS within about a percentage point of $M = 50$ at every horizon but a larger Frobenius error on $\boldsymbol{\alpha}$ and a larger CRPS on $R_{k,t}$.

4.4.6. Computational efficiency

Wall clock times scaled approximately linearly in M as the complexity analysis of § 4.3 predicts. The times below are averages over five PLVA runs at each particle count and three MCMC runs, each run on a separate dataset simulated from the same data

Table 4.2.: Replicated comparison on the spatial parameters across ten independently simulated datasets. The MAE on the retention rates w_k and the Frobenius norm of the error in the transmission weights $\boldsymbol{\alpha}$ are reported as the percentage change of the PLVA value against the MCMC value on the same dataset, averaged across datasets, with the standard deviation across datasets in parentheses. Positive values indicate the larger PLVA error. Coverage is the empirical coverage of the 95% credible intervals and width is their average width, both reported as levels.

	Retention rates w_k			Transmission weights $\boldsymbol{\alpha}$		
M	MAE change	Coverage	Width	Frobenius change	Coverage	Width
10	26.3 (22.6)	0.40	0.098	9.7 (6.6)	0.688	0.147
20	24.3 (19.7)	0.48	0.107	8.1 (5.5)	0.714	0.153
30	24.3 (20.3)	0.60	0.105	8.4 (4.9)	0.704	0.155
50	23.4 (19.5)	0.52	0.109	5.3 (4.8)	0.722	0.156
100	23.2 (22.4)	0.64	0.111	6.3 (5.1)	0.728	0.156
MCMC	—	0.88	0.299	—	0.966	0.270

Table 4.3.: Replicated comparison on the reproduction trajectories and the in-sample predictive fit across ten independently simulated datasets. The MAE and CRPS changes are percentage changes of the PLVA value against the MCMC value on the same dataset, averaged across datasets, with the standard deviation across datasets in parentheses. Positive values indicate the larger PLVA error. The coverage columns report empirical coverage of the 95% credible intervals for $R_{k,t}$ and of the 95% posterior predictive intervals for $y_{s,t}$.

	$R_{k,t}$			In-sample $y_{s,t}$	
M	MAE change	CRPS change	Coverage	CRPS change	Coverage
10	9.0 (39.2)	27.1 (30.6)	0.790 (0.056)	3.5 (22.3)	0.964 (0.009)
20	6.8 (41.7)	28.7 (26.3)	0.810 (0.052)	0.1 (21.5)	0.966 (0.006)
30	7.5 (31.0)	21.4 (28.0)	0.794 (0.046)	−4.3 (15.0)	0.964 (0.006)
50	−10.3 (44.6)	20.8 (29.8)	0.786 (0.061)	0.5 (26.3)	0.964 (0.006)
100	−3.4 (33.9)	15.7 (31.4)	0.790 (0.052)	−3.6 (14.6)	0.966 (0.006)

Table 4.4.: Replicated forecast comparison on the held-out window across ten independently simulated datasets. The CRPS change is the percentage change of the PLVA forecast CRPS against the MCMC value on the same dataset, averaged across datasets, with the standard deviation across datasets in parentheses. Positive values indicate the larger PLVA error. Coverage is the empirical coverage of the 95% posterior predictive intervals.

	$h = 1$ day		$h = 5$ days		$h = 10$ days	
M	CRPS change	Coverage	CRPS change	Coverage	CRPS change	Coverage
10	−4.8 (13.1)	1.000 (0.000)	2.2 (21.1)	0.968 (0.044)	4.7 (34.7)	0.964 (0.036)
20	−4.3 (23.0)	1.000 (0.000)	2.9 (18.3)	0.968 (0.072)	3.4 (15.5)	0.976 (0.026)
30	−4.1 (18.6)	1.000 (0.000)	2.2 (18.5)	0.968 (0.052)	4.0 (23.3)	0.976 (0.022)
50	−5.4 (21.8)	1.000 (0.000)	2.1 (18.3)	0.992 (0.018)	2.6 (14.1)	0.968 (0.034)
100	−5.6 (22.1)	1.000 (0.000)	2.0 (19.4)	0.968 (0.034)	2.6 (15.8)	0.972 (0.018)

generating process, all on a single CPU core. The 300 iteration PLVA runs took 174 seconds at $M = 10$, 313 seconds at $M = 20$, 451 seconds at $M = 30$, 805 seconds at $M = 50$, and 1480 seconds at $M = 100$ (Table 4.5). MCMC required 2641 seconds, approximately 44 minutes, to reach the effective sample size of 354 described above.

Translating into speedup over MCMC, PLVA was 15.2 times faster at $M = 10$, 8.4 times faster at $M = 20$, 5.9 times faster at $M = 30$, 3.3 times faster at $M = 50$, and 1.8 times faster at $M = 100$. At the $M = 50$ used throughout this chapter (§ 4.4.5), this is a 3.3 times speedup over MCMC at the accuracy reported in § 4.4.5, and $M = 20$ raises it to 8.4 times.

4.4.7. Sensitivity to algorithm settings

Very small particle counts. The replicated comparison of § 4.4.5, which fixes the particle count used in this chapter, covers $M \geq 10$. To see how the approximation behaves below that range, we ran the PLVA across $M \in \{1, 2, 5, 10, 20, 30, 50\}$ on five independently simulated datasets and summarized two metrics, each normalized within

Table 4.5.: Wall clock time, iterations run, speedup factor over MCMC, and the converged EIVA effective sample size per column for each PLVA configuration. Wall clock times are averaged over five PLVA runs at each M and three MCMC runs, each run on a separate dataset simulated from the same data generating process.

Method	Iterations	Wall clock (s)	Speedup	EIVA ESS
MCMC	—	2641	1.0	—
PLVA, $M = 10$	300	174	15.2	8.8
PLVA, $M = 20$	300	313	8.4	17.5
PLVA, $M = 30$	300	451	5.9	25.9
PLVA, $M = 50$	300	805	3.3	38.6
PLVA, $M = 100$	300	1480	1.8	85.9

a dataset and then averaged across datasets (Table 4.6). The first is the $R_{k,t}$ forecast CRPS relative to each dataset's best M, and the second is the empirical 95% coverage of $R_{k,t}$. The relative $R_{k,t}$ CRPS is 1.49 at $M = 1$, where the mixture reduces to a single Laplace component and the importance weights are inactive, falls to 1.22 at $M = 2$, and is flat from there. The $R_{k,t}$ coverage improves more gradually, from 0.62 at $M = 1$ to 0.76 at $M = 5$ and roughly 0.80 by $M = 10$, after which it plateaus. A single Laplace component is therefore inadequate on both metrics, and a handful of particles recovers most of what the mixture supplies.

Initialization. We assess sensitivity to the initial spatial and temporal state by comparing three schemes at $M = 50$. The cold start (CS) uses a generalized linear model initialization for the spatial parameters and zeros for the disturbance trajectories. The MCMC short start (MS) uses the posterior means of a short MCMC chain of 100 burn-in and 100 retained iterations. The true start (TS) uses the true data generating values as the initialization. All other algorithm settings are identical to the main runs of § 4.4.1.

The three initialization schemes converged to nearly the same posterior. Posterior

Table 4.6.: PLVA accuracy at small EIVA particle counts. Each metric is normalized within a dataset and then averaged over five independently simulated datasets. The relative $R_{k,t}$ CRPS is the geometric mean of the $R_{k,t}$ forecast CRPS against each dataset's best M (lower is better, 1.00 optimal), and the $R_{k,t}$ coverage is the mean empirical 95% interval coverage (target 0.95). The relative CRPS is worst at $M = 1$ and flat from $M = 2$, while the coverage rises through $M = 10$.

M	$R_{k,t}$ CRPS (rel.)	$R_{k,t}$ coverage
1	1.490	0.620
2	1.218	0.714
5	1.214	0.758
10	1.201	0.800
20	1.205	0.801
30	1.182	0.810
50	1.222	0.803

medians on w_k matched within 0.07 across schemes on every column (Table 4.7), with the largest gap appearing on w_3. The EIVA effective sample size at convergence stayed between 38.6 and 40.6 across the three runs. Predictive performance was also stable. In-sample CRPS on $y_{s,t}$ varied within 0.02 across schemes, 1 to 10 day forecast CRPS varied within 0.15, and posterior CRPS on $R_{k,t}$ stayed at 0.09 across the three schemes. Coverage of $R_{k,t}$ at the nominal 95% level ranged from 0.85 to 0.88. Although the true start began at the data generating parameters, it drifted to the same fixed point as the cold start, so the departure from the truth on the low signal retention rates does not reflect a failure of optimization. The under-coverage documented in § 4.4.5 is a property of the approximating family, and § 4.6 outlines two ways of widening it.

Table 4.7.: Sensitivity of posterior summaries and predictive performance to the initial state at $M = 50$. CS: cold start, using a generalized linear model fit for the spatial parameters and zeros for the disturbance trajectories. MS: MCMC short start, using the posterior means of a short MCMC chain. TS: true start at the data generating values. The simulated truth for w_k is shown in the first row for reference. Coverage is the empirical coverage of the 95% credible intervals for $R_{k,t}$.

Init	w_1	w_2	w_3	w_4	w_5	EIVA ESS
Truth	0.90	0.70	0.50	0.30	0.30	—
CS	0.86	0.72	0.50	0.41	0.39	38.6
MS	0.85	0.72	0.46	0.41	0.41	40.6
TS	0.85	0.71	0.53	0.42	0.38	39.5

Init	PPC CRPS	$\text{CRPS}_{R_{k,t}}$	$\text{Coverage}_{R_{k,t}}$	1 day CRPS	10 day CRPS
CS	3.38	0.09	0.85	4.96	13.11
MS	3.37	0.09	0.85	5.07	13.26
TS	3.36	0.09	0.88	5.10	13.14

4.4.8. Trade-off summary and recommendation

The simulation results support a scoped claim. The PLVA matches the MCMC posterior medians, the reproduction trajectories, and the predictive summaries at a fraction of the wall clock cost, while its credible intervals on the spatial parameters stay too narrow. Across the ten replicated datasets at $M = 50$, the forecast CRPS on the held-out counts differed from MCMC by -5.4% at $h = 1$, 2.1% at $h = 5$, and 2.6% at $h = 10$, each small relative to its variation across datasets, and the posterior predictive interval coverage was at or above the nominal level at all three horizons. The MAE of the posterior median reproduction trajectories against the truth differed by -10.3% with a standard deviation of 44.6% across datasets, so the two methods were of comparable point accuracy on $R_{k,t}$. The wall clock cost was 2 to 15 times lower than MCMC depending on M.

The cost of the variational approximation appeared in the credible interval widths.Across

the ten replicated datasets at $M = 50$, the empirical coverage of the PLVA 95% credible intervals was 0.79 for $R_{k,t}$, 0.72 for the transmission weights against MCMC's 0.97, and 0.52 for the retention rates against MCMC's 0.88. The retention rate intervals were 2.8 times narrower than MCMC's and the transmission weight intervals 1.7 times narrower. Beyond the smallest particle counts the under-coverage on ω_k was flat in M, which again suggests that closing the gap calls for modifications to the variational family and not for larger particle counts. § 4.6 names two such modifications, a joint importance proposal over $(\boldsymbol{\eta}_k, \zeta_k)$ and a heavier-tailed mixture family, and we defer the details to that section. Users with a primary need for fast posterior inference on $R_{k,t}$ and forecasting can rely on PLVA at $M = 50$, the count used throughout this chapter (§ 4.4.5), which gives a 3.3 times speedup over MCMC (Table 4.5). Where forecasting alone is the target, $M = 20$ gave forecast CRPS within about a percentage point of $M = 50$ at every horizon and raises the speedup to 8.4 times. Users requiring calibrated uncertainty quantification on the spatial parameters should pair PLVA with periodic MCMC validation or accept the broader intervals of MCMC.

The qualitative patterns we observed, namely predictive parity, narrower credible intervals on the retention rates, and approximately linear wall clock scaling in M, align with what the theoretical analysis of § 4.3 predicts. Scaling the study to a larger destination count and longer time series, and an extension to the joint importance sampling variant, are natural directions for future empirical work.

§ 4.5. Application to COVID-19 across California counties

We apply the proposed PLVA algorithm to the network Hawkes-type model of Chapter 3 on daily COVID-19 case counts from ten California counties between July 2020 and November 2021. Chapter 3 establishes the surveillance findings using the block MCMC sampler. Here we examine whether PLVA yields the same point estimates and predictive summaries at lower wall clock cost, and we report where the variational approximation departs from the MCMC reference. The model specification, the mobility covariate, and the data preprocessing follow Chapter 3 without modification, and this section focuses on the algorithmic comparison.

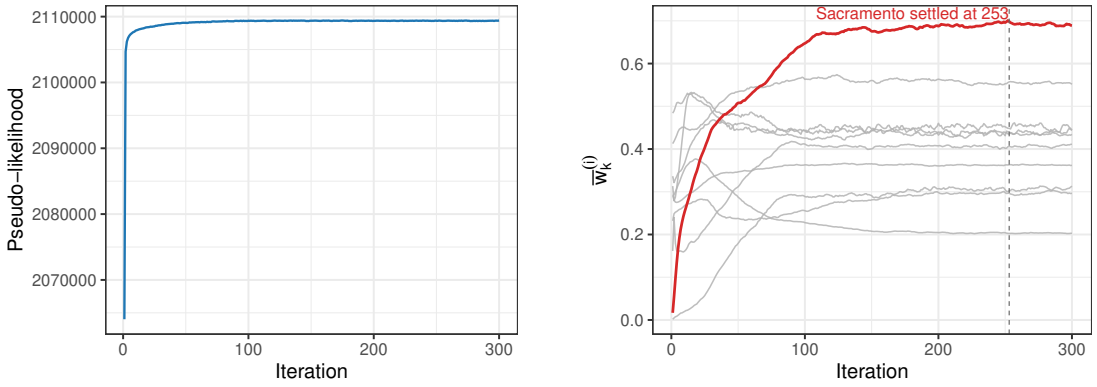
4.5.1. Setup and convergence

The model is fitted on $T = 517$ days of incidence data at $S = 10$ counties. Priors and the horseshoe global scale are set exactly as in Chapter 3. The MCMC reference runs the block MCMC sampler of Chapter 3 for 25,000 iterations after a 10,000 iteration burn-in, thinned to retain every fifth draw. PLVA runs at $M = 100$ for $N_{\text{iter}} = 300$ iterations. The particle count $M = 100$ is a deliberately conservative choice, above the $M = 50$ used in the simulations of § 4.4, so that the importance approximation is as accurate as possible for the reported application results. The wall clock comparison below is correspondingly conservative, as $M = 50$ would run roughly twice as fast. After the variational optimization completes, $N_{\text{post}} = 1000$ posterior samples are drawn via Algorithm 8.

The pseudo-likelihood objective trace $\widehat{\mathcal{L}}_{\text{pl}}$ of Eq. (4.3.46) settled within approximately

50 iterations, and the slowest visually converging column-wise mean retention rate, Sacramento, stabilized by approximately iteration 253, well within the 300 iterations completed. The EIVA importance sampler over the final 20 iterations produced a mean effective sample size of 39.1 out of $M = 100$ particles across the ten columns, ranging from 26.7 to 47.5 across columns. The MCMC reference produced a mean column-wise effective sample size of 362 for the retention rates out of the 5,000 retained samples.

The diagnostic traces are shown in Fig. 4.6. Wall clock cost differed by approximately 2.9 times. PLVA required 12.8 hours on a single CPU core, against 37.2 hours for the MCMC chain on the same hardware. Both diagnostic traces stabilized before the 300th iteration, suggesting that a shorter run may have produced similar posterior summaries at lower computational cost.



(a) Pseudo-likelihood objective trace.

(b) Posterior mean retention rate per county.

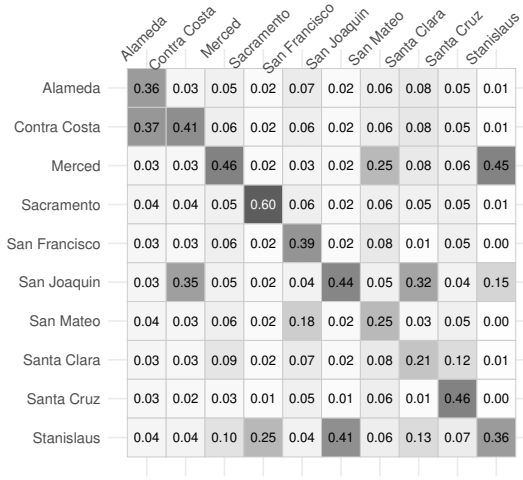
Figure 4.6.: PLVA convergence diagnostics on the California COVID-19 data over the 300 optimization iterations completed. Left: pseudo-likelihood objective trace $\widehat{\mathcal{L}}_{\text{pl}}$ of Eq. (4.3.46). Right: w_k trace by county, where the slowest visually converging county (San Mateo, highlighted in red) stabilized by approximately iteration 182.

4.5.2. Parameter estimation

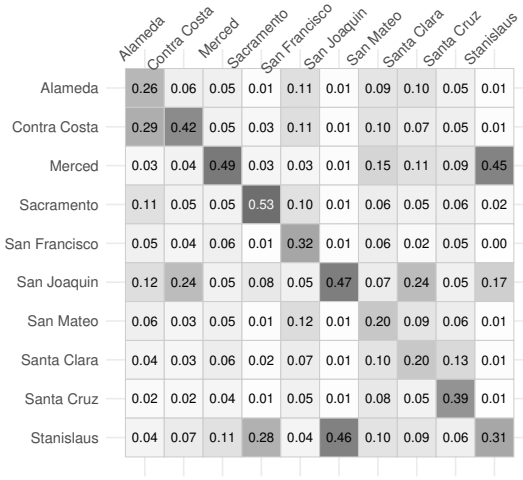
We first compare the posterior mean and standard deviation of the spatial weight matrix $\boldsymbol{\alpha}$ between the two algorithms. The qualitative network structure agreed between PLVA and MCMC. Both methods identified the same dominant directional pathways, including the Central Valley cluster (Stanislaus, Merced, San Joaquin) and the Bay Area cluster (Alameda, Contra Costa, San Francisco, San Mateo). The two largest off-diagonal entries in the PLVA posterior mean matched those in the MCMC posterior mean (Fig. 4.7), including the Stanislaus-to-Merced edge, $\hat{\alpha}_{\text{Mer,Stan}} = 0.45$ under both methods, and the San Joaquin-to-Stanislaus edge, $\hat{\alpha}_{\text{Stan,SJ}} \approx 0.41$ under PLVA and ≈ 0.46 under MCMC, that Chapter 3 highlights.

Two quantitative departures of PLVA from MCMC parallel the pattern observed in simulation. First, PLVA's posterior mean of the diagonal entries w_k averaged 0.39 across the ten counties, slightly larger than MCMC's mean of 0.36 (Table 4.8). The direction of the gap matches the over-shrinkage pattern documented in § 4.4.3. When the off-diagonal $\theta_{s,k}$ values are shrunk more strongly under the variational posterior, the column-stochastic constraint sends some of the unaccounted mass back into the diagonal. Second, the posterior standard deviation of w_k under PLVA was uniformly smaller than under MCMC, with an average standard deviation of 0.02 for PLVA and 0.06 for MCMC. Off-diagonal standard deviations averaged 0.03 for PLVA and 0.035 for MCMC, with the largest MCMC standard deviation of 0.22 corresponding to a PLVA standard deviation of 0.1 on the same entry. Overall, the variational approximation produced a more concentrated posterior on the network parameters than the MCMC reference, consistent with the simulation finding.

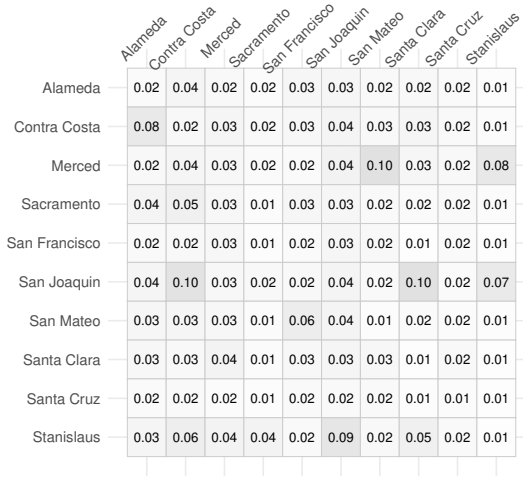
Beyond the spatial weight matrix, we compare the reproduction number decomposi-



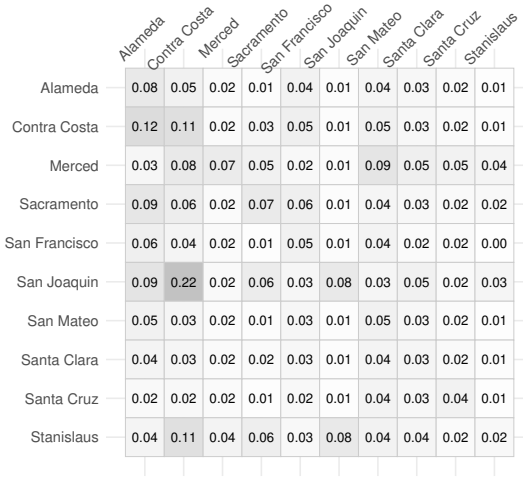
(a) Posterior mean under PLVA.



(b) Posterior mean under MCMC.



(c) Posterior SD under PLVA.



(d) Posterior SD under MCMC.

Figure 4.7.: Posterior summaries of the spatial transmission weight matrix $\boldsymbol{\alpha}$ on the California COVID-19 data. Top row: posterior mean under PLVA (left) and MCMC (right). Bottom row: posterior standard deviation under the two methods. The dominant pathways agree between the methods. PLVA produces a more concentrated posterior, compared to MCMC.

Table 4.8.: Posterior mean and standard deviation of the diagonal retention rates w_k and selected dominant off-diagonal transmission weights $\alpha_{s,k}$. The two methods agree on which entries dominate, and the PLVA estimates are more concentrated.

Entry	PLVA mean	PLVA SD	MCMC mean	MCMC SD
Avg diagonal $\bar{w}$	0.39	0.02	0.36	0.06
$\alpha_{\text{Mer,Stan}}$	0.45	0.08	0.45	0.04
$\alpha_{\text{Stan,SJ}}$	0.41	0.09	0.46	0.08
$\alpha_{\text{CC,Ala}}$	0.37	0.08	0.29	0.12
$\alpha_{\text{SJ,CC}}$	0.35	0.1	0.24	0.22

tion by county that Chapter 3 establishes as the chapter's central epidemiological output. Table 4.9 reports the PLVA posterior means of the intrinsic source-specific reproduction number $\bar{R}_k$ and the local contribution Local%, alongside the MCMC values reproduced from Table 3.3. The two methods agree on the sign of the net flow for nine of the ten counties. Sacramento lies close to zero under both methods but has opposite estimated signs (+0.02 under PLVA and −0.04 under MCMC). Bay Area counties tend to be net exporters under both methods, and Central Valley counties tend to be net importers, recovering the regional pattern of Chapter 3. The most transmissible and least transmissible counties also match, with Santa Clara at the top of $\bar{R}_k$ (1.32 under PLVA, 1.14 under MCMC) and San Joaquin at the bottom (0.60, 0.55). The Local% range is similar under the two methods, from roughly 20% in Merced and San Mateo to roughly 50% in Sacramento. The negative correlation between intrinsic transmissibility and case rate that Chapter 3 highlights is recovered, with the counties with the highest incidence having the lowest $\bar{R}_k$ and those with the lowest incidence the highest under both methods.

The county-level disagreements concentrate where the simulation predicted. The over-shrinkage of off-diagonal $\theta_{s,k}$ values noted earlier in this subsection inflates reten-

tion rates under PLVA by a few percentage points relative to MCMC, and the corresponding adjustments to the intrinsic transmissibility appear most clearly on the Bay Area sources, which have the highest transmissibility. San Francisco and Santa Cruz fall below 1.0 under PLVA but remain above 1.0 under MCMC. Apart from Sacramento's near-zero net flow, these differences do not change the source and sink classification or the qualitative correlation pattern. Calibrated statements about whether the $\bar{R}_k$ of a specific county sits above or below the critical threshold of 1.0 should still rely on MCMC, in line with the recommendation of § 4.6.

Table 4.9.: Per-county reproduction number decomposition under PLVA algorithm and MCMC on the California COVID-19 data, ordered by descending MCMC $\bar{R}_k$. The MCMC values are reproduced from Table 3.3. The two methods agree on Net Flow signs for nine of the ten counties, with Sacramento sitting near zero under both, and agree on the most transmissible and least transmissible counties as well as on the negative correlation between $\bar{R}_k$ and case rate.

	$\bar{R}_k$		Local%		Net Flow	
County	PLVA	MCMC	PLVA	MCMC	PLVA	MCMC
Santa Clara	1.32	1.14	27.8	24.4	+0.34	+0.34
San Francisco	0.81	1.10	30.5	37.8	+0.33	+0.39
Santa Cruz	0.87	1.07	28.9	31.6	+0.32	+0.34
Stanislaus	0.99	1.01	37.8	43.3	−0.50	−0.54
San Mateo	0.97	0.93	27.4	20.0	+0.32	+0.35
Sacramento	0.83	0.85	52.3	49.8	+0.02	−0.04
Alameda	0.70	0.79	23.4	21.9	+0.25	+0.26
Contra Costa	0.71	0.68	38.7	37.4	−0.14	−0.15
Merced	0.61	0.65	20.1	22.1	−0.43	−0.44
San Joaquin	0.60	0.55	24.4	21.4	−0.49	−0.53

We also compared PLVA and MCMC on the network reproduction number $R_t^{\text{net}} = \rho(\mathbf{B}_t)$, shown in Fig. 4.8. The posterior median trajectories under PLVA tracked the MCMC reference closely, and the variational posterior was modestly narrower than

MCMC's, as in the simulation.

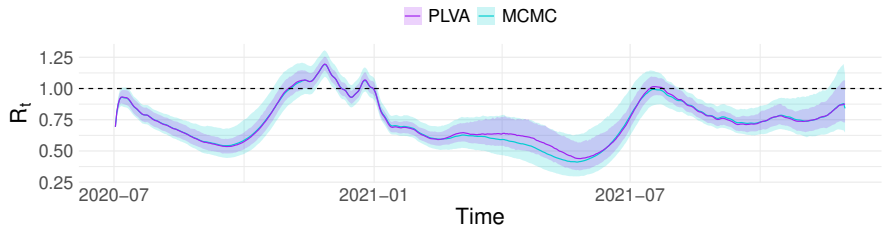


Figure 4.8.: Network reproduction number $R_t^{\text{net}} = \rho(\mathbf{B}_t)$ over the study period. Posterior medians and 95% credible bands under PLVA and MCMC are overlaid for direct comparison.

4.5.3. Posterior predictive fit and forecasting

We evaluated in-sample posterior predictive fit on the full training window and ten step ahead forecasting on a held-out window of 10 days. PLVA and MCMC produced closely aligned predictive distributions on the observed case counts.

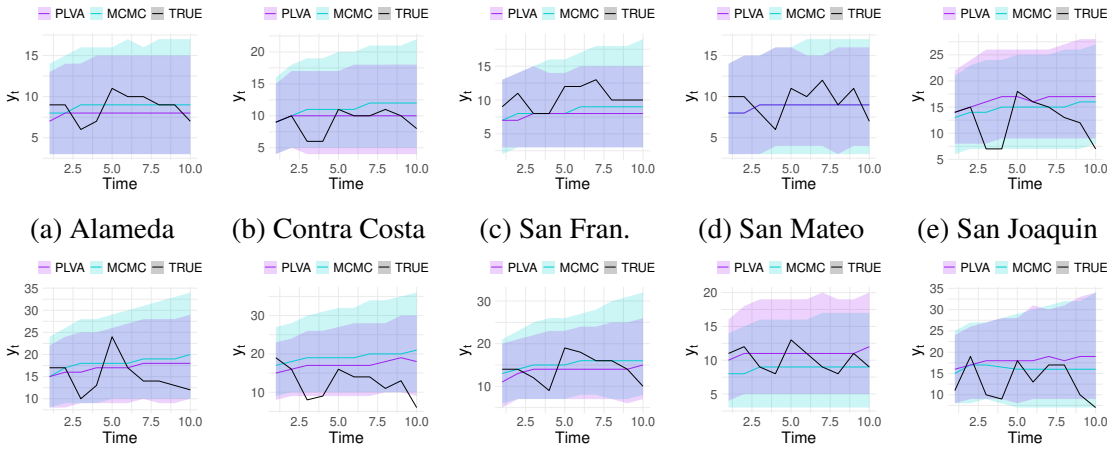
In-sample, the CRPS on $y_{s,t}$ averaged 3.708 for PLVA and 3.698 for MCMC across the ten counties, a relative difference smaller than 0.3% (Table 4.10). The empirical 95% prediction interval coverage averaged 0.88 for PLVA and 0.89 for MCMC, both modestly below the nominal 0.95. The two methods agreed on which counties were hardest to fit. Both produced the lowest coverage on San Francisco (county 3) at approximately 0.71, suggesting that the Poisson observation distribution understated the dispersion in that county's case counts.

On the held-out forecast window, the ten step ahead CRPS averaged 2.191 for PLVA and 2.146 for MCMC, a 2% difference well within the variation expected from a single forecast origin. The prediction interval coverage, the fraction of held-out counts within the central 95% posterior predictive interval, averaged 0.94 for PLVA and 0.96 for

MCMC. Both methods produced calibrated probabilistic forecasts and agreed closely on which counties had the largest forecast uncertainty (Fig. 4.9).

Table 4.10.: In-sample posterior predictive check and ten step ahead forecasting performance on the California COVID-19 data. Values are means across the ten counties. The coverage columns report the empirical 95% prediction interval coverage, the fraction of counts within the central 95% posterior predictive interval.

	Posterior predictive check		Ten step ahead forecast	
Method	CRPS	Coverage at 0.95	CRPS	Coverage at 0.95
PLVA	3.708	0.881	2.191	0.94
MCMC	3.698	0.888	2.146	0.96



(a) Alameda (b) Contra Costa (c) San Fran. (d) San Mateo (e) San Joaquin

(f) Stanislaus (g) Merced (h) Sacramento (i) Santa Clara (j) Santa Cruz

Figure 4.9.: Ten step ahead forecasts on the held-out window for each of the ten counties. Observed counts are shown as black lines, and PLVA and MCMC posterior medians are shown with their 95% prediction bands. The forecast distributions of the two methods agreed closely, and both achieved calibrated 95% prediction interval coverage near the nominal level.

4.5.4. Summary

PLVA matched the main point estimates and predictive summaries of Chapter 3 at 2.9 times lower wall clock cost. The parameter estimation agreed with the MCMC reference at posterior medians, and the predictive distribution on the observed case counts agreed within a relative difference of 0.3% on in-sample CRPS and 2% on ten step ahead forecast CRPS. The variational posterior on the spatial weight matrix was more concentrated than the MCMC posterior, mirroring the simulation finding of § 4.4, with little effect on the predictive metrics in this application.

The practical implication for surveillance work is that the network Hawkes-type model can be updated daily using PLVA, with periodic MCMC validation when calibrated uncertainty on the spatial parameters is required. The wall clock saving demonstrated on the ten county California data supports the routine use of PLVA for medium-term forecasting and scenario analysis at this scale.

§ 4.6. Concluding remarks

This chapter developed the PLVA for the network Hawkes-type model of Chapter 3. The method is an iterative conditional approximation that cycles through block updates, each using an approximating family matched to the conditional geometry of the corresponding posterior. The temporal block uses the EIVA proposal of Loaiza-Maya and Nibbering (2025) to obtain importance-weighted particles. The spatial block uses a mixture of Laplace approximations centered at the conditional mode of each particle, with weights inherited from the EIVA importance sampler. The horseshoe scales and auxiliaries are updated by the moment formulas of the centered, unregularized

Makalic–Schmidt augmentation. The zero-inflation indicators are marginalized analytically through a forward filter. The global parameters are updated by SGA on a Gaussian factor covariance approximation, mirroring the scheme used for the univariate DGTF class in Chapter 2. The complete procedure seeks a stable collection of conditional approximations, and its accuracy is established empirically, for the regimes studied, by comparison with the block MCMC sampler of Chapter 3.

The variance decomposition of Eq. (4.3.28) describes the additional flexibility supplied by the particle mixture. Its within-particle term summarizes average local curvature, and its between-particle term summarizes variation of the conditional spatial mode across temporal trajectories. While a correlated joint Gaussian can also accommodate both types of variance, the distinction arises when the conditional modes vary nonlinearly, the conditional covariance changes across trajectories, or the posterior is skewed, heavy-tailed, or multimodal. In such cases, one Gaussian may be inadequate, while the mixture can adapt its component locations and curvatures. A mean-field family is more restrictive because it removes cross-block dependence. The empirical contribution of the chapter, documented in § 4.4 and 4.5, is to assess this construction against the MCMC benchmark on simulated data and the California county application. In the regimes studied, the approximation matched the predictive distribution and the reproduction number trajectories to within the variation across replicated datasets while running at 2 to 15 times lower wall clock cost, and it underestimated posterior uncertainty on the spatial parameters at every particle count examined.

Two limitations of the current method stand out, both visible in the simulation study and both pointing to concrete extensions. First, across ten independently simulated datasets the credible intervals on the spatial block were narrower than the MCMC

reference at every particle count, by a factor of 2.8 on the retention rates and 1.7 on the transmission weights at $M = 50$, with empirical coverage of 0.52 and 0.72 against MCMC's 0.88 and 0.97. This pattern is consistent with the structural form of the Laplace mixture, which represents the spatial uncertainty as a finite mixture of Gaussians whose centers and curvatures are determined by Newton's method on the conditional log-posterior of each particle. Two refinements would broaden the coverage on the spatial block without abandoning the column-wise factorization that the rest of the algorithm relies on. The first is a joint importance proposal for column k that draws one $\zeta^{(m)} \sim N(\zeta^{*(m)}, \Sigma^{(m)})$ for each EIVA particle $\boldsymbol{\eta}^{(m)}$ and weights particles on the joint log-density, producing a weighted empirical posterior $\{(\zeta^{(m)}, \bar{w}^{(m)})\}$ on the joint $(\boldsymbol{\eta}_k, \zeta_k)$ rather than a Laplace mixture on ζ_k alone. The second is a heavier-tailed mixture family, for example a mixture of t-distribution components in place of the Gaussian Laplace expansions, that allows the variational posterior to explore wider regions of the parameter space. Either modification leaves the rest of the block-coordinate algorithm intact and can be evaluated against the same simulation and application baselines used here.

A related limitation concerns the horseshoe scales. What the experiments validate against MCMC are the transmission weights and predictive quantities that the scales induce, so τ_k and $\gamma_{s,k}$ are best read as internal regularization quantities, not as reported inferential targets. Where the shrinkage scales are themselves of interest, that block can be optimized numerically under the regularized non-centered model or sampled by MCMC.

Second, the Laplace components rest on a local unimodality assumption that the algorithm cannot verify internally, and the temporal approximation conditions on the

current spatial vector $\hat{\boldsymbol{\zeta}}_k$ rather than marginalizing over it. The resulting column-conditional approximation may be close to the true block-conditional marginal when the temporal conditional is insensitive to nearby spatial plug-in values. In the regimes we tested, the column-conditional posterior of $\boldsymbol{\eta}_k$ changed little as those values varied. The joint importance proposal sketched above would reduce reliance on this conditioning, so the proposed extensions point in the same direction.

Taken together, these limitations describe the scenarios in which the PLVA is reliable. The approximation works well when (1) cross-column coupling is weak to moderate, so that the column-conditional posterior of $\boldsymbol{\eta}_k$ is approximately invariant to the spatial weights of other columns and the pseudo-likelihood of Eq. (4.3.43) introduces little bias; (2) the column-conditional spatial posterior is unimodal, so that the Laplace expansion centered at the Newton mode captures its mass; and (3) the EIVA Gaussian proposal remains well calibrated against the column-conditional posterior of $\boldsymbol{\eta}_k$ at convergence, as monitored by the importance sampling ESS reported throughout § 4.4 and 4.5. The simulation studies and the California application of this chapter operate within these scenarios. Outside them, particularly under strong cross-column coupling or substantial multimodality of the column-conditional posterior, the block-coordinate fixed point can drift to a region of the parameter space where the EIVA proposal degrades and the variational posterior fails to represent the true posterior. In such settings the MCMC sampler of Chapter 3 remains the safer reference.

These findings support a clear practical recommendation. For final reported inference, whenever calibrated credible intervals on the spatial parameters are themselves the inferential target, and in settings with strong cross-column coupling where the conditions above may not hold, the block MCMC sampler of Chapter 3 remains the method of

choice. The PLVA credible intervals were too narrow across the replicated datasets, by a factor of 2.8 on the retention rates and 1.7 on the transmission weights. An example of an inferential target requiring MCMC would be a formal statement about the presence or absence of a transmission pathway. For fast updates during an unfolding outbreak, for short-term and medium-term forecasting, and for scenario analysis in settings where the structure cooperates, the PLVA is preferable. In the experiments considered, its predictive accuracy was close to the MCMC reference despite the under-coverage on the spatial parameters. The method is demonstrated only over the tested moderate values of S and T, and larger spatial networks or longer time series require additional computational structure noted in § 4.3.6. A practical workflow combines the two. Update at the daily cadence with the PLVA at $M = 50$, and validate periodically against a full MCMC run, with the latter serving as the validating reference whenever calibrated intervals are needed or cross-column coupling is strong.

Further directions include scaling to all 58 California counties or to larger spatial domains; adaptive choice of the number of particles per column based on the running ESS; extensions to continuous-time formulations of the network Hawkes-type model; application to other surveillance data such as influenza and dengue; and adaptation to other count processes with similar multiplicative structure, such as crime incidence and earthquake aftershocks. A complementary methodological direction is to define a normalized continuous joint proposal over the temporal, spatial, scale, and global blocks. Specifying that density together with its sampling rule would allow an IWAE objective to be evaluated and maximized as a lower bound on $\log p(y_{1:T})$, giving a globally coherent variational method. A further direction is integrating the PLVA into the **dgtf** package of Chapter 5, which currently supports only the univariate DGTF

class. The component-based interface and the unified inference dispatch developed there generalize naturally to the multivariate setting once the spatial weight matrix is added as a model component, and the structured variational algorithm developed here is the natural counterpart of the HVA that Chapter 5 already provides for the univariate class.

Chapter 5

The `dgtf` R package for modular Bayesian state-space models

§ 5.1. Introduction

Chapter 2 introduced the dynamic generalized transfer function (DGTF) model class as a unified state-space framework for count time series in which discrete-time Hawkes-type models, recursive distributed lag models, and AR count models all arise from different choices of six interchangeable components. This chapter develops the accompanying R package, **dgtf**, which makes the framework usable in applied surveillance work and supports the structural model comparison that motivated the methodology. This chapter gives a concise account of the software contribution in the context of the dissertation.

Several R packages already address related problems, and the case for a unified interface rests on what they do not jointly provide. **EpiEstim** (Cori et al., 2013) estimates time-varying reproduction numbers under a renewal equation with sliding windows, **surveillance** (Meyer et al., 2017) provides endemic-epidemic models through `hhh4()`, **tscount** (Liboschik et al., 2017) fits observation-driven count autoregressions

by conditional quasi-maximum likelihood, and **epidemia** (Scott et al., 2020) offers a regression interface for R_t within a **Stan**-based hierarchical framework. Each is powerful within its own model class, and none offers a single interface for specifying, fitting, forecasting, and comparing Hawkes, distributed lag, and autoregressive formulations under a shared Bayesian state-space framework.

The **dgtf** package addresses this gap by exposing the compositional structure of the DGTF class directly. Models are built from constructor functions for the six components, and the same model and prior objects pass to either a fast hybrid variational approximation (HVA) or an MCMC sampler without rewriting the analysis. The design has three parts. The compositional specification interface lets users assemble models from independent components and change the structure by replacing one component while leaving the rest intact. The unified inference interface separates the model and prior from the choice of inference method. An integrated evaluation toolkit provides posterior predictive checks, proper scoring rules, parameter recovery diagnostics, multi-step forecasting, and side-by-side model comparison. Together these carry a user from model construction through inference, forecasting, diagnostics, and structural comparison without switching packages.

§ 5.2 summarizes the model class as the package implements it, § 5.3 describes the interface, and § 5.4 outlines the two inference methods. § 5.5 and § 5.6 demonstrate the workflow through a simulation study and an application to COVID-19 surveillance data from Santa Cruz County, California, and § 5.7 concludes.

§ 5.2. Model framework

This section provides a concise overview of the DGTF model class. Full derivations and methodological details are given in Chapter 2.

5.2.1. General formulation

A DGTF model is a state-space model for univariate count time series, assembled from six interchangeable components: an observation distribution, a link function, a gain function, a lag distribution, a system equation, and an innovation distribution. Chapter 2 develops the class in full, and this subsection fixes only the notation that the package interface exposes.

The observed count y_t has conditional mean $\lambda_t > 0$ under a Poisson or negative binomial observation model, selected through `obs_poisson()` or `obs_nbinom()`, where the negative binomial carries an overdispersion parameter $\rho > 0$ and the Poisson is its limit as $\rho \to \infty$. A link function maps λ_t to a linear predictor that splits into an exogenous regression term $\mathbf{x}_t'\boldsymbol{\varphi}$, which carries the level of the series together with seasonal patterns and covariates, and a self-exciting transfer function whose form distinguishes the model families of § 5.2.2. Identity, log, and logistic links are available, and the mathematical log link is exposed in the package by its inverse-link constructor `link_exponential()`, alongside `link_identity()` and `link_logistic()`.

The transfer function is driven by a latent state evolving as a Gaussian random walk with innovation covariance $\mathbf{W}_t$. Past counts enter weighted by a discrete lag distribution $\{\phi_l\}$ and by a scalar latent driver ψ_t, which is mapped to a non-negative scale by a gain function $h(\cdot)$, so that $R_t = h(\psi_t)$ is the time-varying reproduction number wherever the

model admits that reading. The system component determines how past observations enter the transfer function and is the main mechanism by which the model families are obtained, with `sys_shift()` for discrete-time Hawkes-type models, `sys_nbinom()` for recursive distributed lag models, and `sys_identity()` for autoregressive count models. Shortcut constructors for common configurations are summarized in Table 5.2. The static parameters are collected in $\boldsymbol{\gamma}$ and may include ρ, the regression coefficients $\boldsymbol{\varphi}$, the system variances, and the parameters of the lag distribution.

5.2.2. Canonical models

Chapter 2 derives the three canonical families and their state-space representations. This subsection records how each is assembled from package components and what distinguishes it in use.

Discrete-time Hawkes-type models. The transfer function weights each past count by a lag probability ϕ_l and by the transmission intensity $R_{t+1-l} = h(\psi_{t+1-l})$ prevailing at the time of that count. Separating the overall intensity from the lag distribution is the main interpretive advantage of this specification. The `sys_shift()` component stores the current and recent values of the latent driver, so the lagged values are available in the state vector. The lag distribution is specified with `lag_lognormal()`, `lag_nbinom()`, or `lag_uniform()`, and the shortcut constructor `dgtf_hawkes()` combines a negative binomial observation model, identity link, softplus gain, lognormal lag distribution, `sys_shift()` system structure, and Gaussian system innovations.

The truncation length L is set by the package rather than by the user. The argument `residual_prob` of `lag_lognormal()` and `lag_nbinom()` gives a threshold, 0.005

by default, and L is the smallest integer for which the residual probability $1 - \sum_{l=1}^{L} \phi_l$ falls below it. The fitted summary reports both the threshold and the resulting L.

Distributed lag models. These represent the same cumulative effect of past cases through a recursion of infinite order instead of a stored lag window, which keeps the state vector short. **dgtf** implements the recursion through the `sys_nbinom()` system component together with `lag_nbinom()`, and the interpretation $R_t = h(\psi_t)$ is preserved. A model of order r is built by the shortcut constructor `dgtf_distributed_lag(r, kappa, W)`, and the Koyck model is the case $r = 1$, available through the shortcut `dgtf_koyck()`.

Autoregressive count models. These are implemented with the `sys_identity()` system component, and the autoregressive coefficients are obtained from the latent drivers through the gain transformation $b_{l,t} = h(\psi_{l,t})$. Because the gain function is non-negative, so are the coefficients. With $\mathbf{W}_t = \mathbf{0}$ the model reduces to a standard Poisson or negative binomial autoregression with static coefficients, and with nonzero $\mathbf{W}_t$ the coefficients vary over time.

The lag component plays a different role here. In the Hawkes and distributed lag families the lag weights form a probability distribution, which separates the temporal pattern of transmission from the intensity $R_t = h(\psi_t)$. In autoregressive models each lag has its own coefficient and the coefficients are not normalized, so these models carry no direct reproduction number interpretation. They remain useful for flexible short-term dependence and forecasting. The shortcut constructors are `dgtf_poisson_ar()` and `dgtf_nb_ar()`.

5.2.3. Seasonality and exogenous regressors

Surveillance counts often display systematic reporting patterns that are not part of the underlying transmission process. A common example is day-of-week variation, in which fewer cases are reported on weekends and compensating increases appear early in the following week. In **dgtf**, these effects can be modeled through $\mathbf{x}_t'\boldsymbol{\varphi}$. The vector $\mathbf{x}_t$ contains known regressors, and $\boldsymbol{\varphi}$ contains the corresponding regression coefficients. These coefficients are treated as static parameters and can be fixed or estimated jointly with the remaining elements of $\boldsymbol{\gamma}$.

For example, for seasonal effects with period Q, the design vector $\mathbf{x}_t$ is an indicator vector for the position of time t within the cycle. Thus, $\mathbf{x}_t = (e_{1,t}, \ldots, e_{Q,t})'$, where exactly one entry is equal to one and the remaining entries are zero. The corresponding parameter vector $\boldsymbol{\varphi} = (\varphi_1, \ldots, \varphi_Q)'$ contains one effect for each position in the seasonal cycle. For example, when $Q = 7$, the seven components represent day-of-week effects.

Because the seasonal indicators include the level of the series, no separate intercept is included when $Q > 1$. The special case $Q = 1$ reduces the regression component to a scalar intercept. In the package interface, this is implemented through `seas_period()`. For example,

```
seas_period(1, init = 2)
```

specifies a scalar intercept initialized at 2, whereas

```
seas_period(7, init = rep(2, 7))
```

specifies seven seasonal effects each initialized at 2. The function `seas_weekly()` is equivalent to `seas_period(7, ...)` and is provided for day-of-week reporting ef-

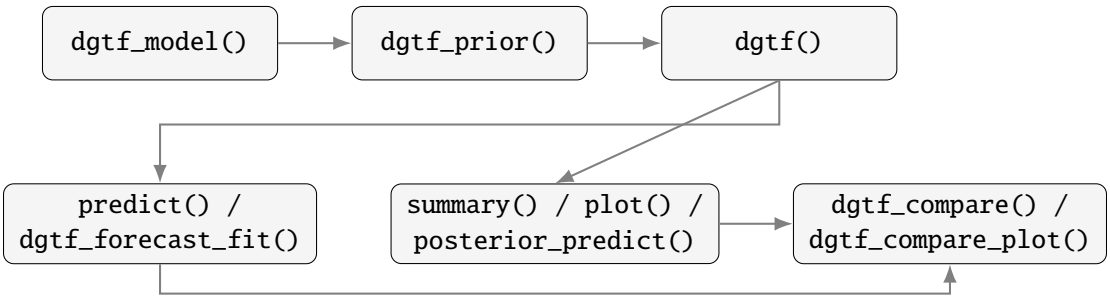


Figure 5.1.: Workflow of the **dgtf** package. A model object and a prior specification feed into `dgtf()`, which produces a fit. The fit supports forecasting, posterior inspection including posterior predictive checks, and cross-model comparison through a `dgtf_compare()` that accepts either fit objects or forecast objects.

fects. The same seasonal regression component can be used with discrete-time Hawkes-type, distributed-lag, and autoregressive count models.

§ 5.3. The dgtf package

The **dgtf** package implements the model class introduced in § 5.2 through a compositional **R** interface. The workflow has five main stages: model construction, prior specification, posterior inference, posterior inspection and forecasting, and model comparison. Figure 5.1 summarizes this workflow. A model object and a prior specification are passed to the common inference function `dgtf()`, which returns a fitted object. The fitted object can then be inspected using standard S3 methods, used to generate posterior predictive checks and forecasts, visualized through single-model or multi-model plotting functions, and compared with alternative fitted models.

Where an operation has natural variants by object type, **dgtf** uses standard **R** dispatch rather than separate function names. For example, `predict()` returns in-sample fitted

Table 5.1.: Component families in the **dgtf** package and the types used in this chapter. String shortcuts are accepted in place of the explicit constructor calls when default parameter values suffice. The observation row lists the count models used in the examples here, and the package help documents the full set of components.

Family	Constructor	Available types
Observation	`obs_*`	`poisson`, `nbinom`
Link	`link_*`	`identity`, `exponential`, `logistic`
Gain	`gain_*`	`softplus`, `exponential`, `identity`, `ramp`, `logistic`
Lag distribution	`lag_*`	`lognormal`, `nbinom`, `uniform`
System equation	`sys_*`	`shift`, `identity`, `nbinom`
Innovation noise	`err_*`	`gaussian`, `constant`

values when the forecast horizon is zero and out-of-sample forecasts when the horizon is positive; `plot()` dispatches according to the class of the object being displayed; and `dgtf_compare()` accepts fitted objects, posterior predictive objects, or forecast objects. This design allows users to move from model construction to inference, diagnostics, forecasting, and structural comparison without rewriting the analysis for each model class.

5.3.1. Model construction

Models may be assembled from the six component constructors in Table 5.1 or created using the shortcut constructors in Table 5.2. The explicit interface is useful when an individual component must be replaced, whereas the shortcuts provide concise specifications of the canonical model families. Both approaches return the same model-object class and therefore use the same subsequent workflow.

For example, a discrete-time Hawkes-type model with a scalar intercept is constructed by

Table 5.2.: Shortcut constructors in the **dgtf** package. Each pre-fills the component arguments of `dgtf_model()` to produce the named canonical model. Koyck is the $r = 1$ special case of the negative binomial distributed lag family.

Model class	Constructor signature	System	Lag structure
Discrete-time Hawkes-type	`dgtf_hawkes(meanlog, sigma2, W, seasonality)`	`sys_shift()`	Lognormal
Distributed lag	`dgtf_distributed_lag(r, kappa, W)`	`sys_nbinom()`	Negative binomial
Koyck	`dgtf_koyck(kappa)`	`sys_nbinom()`	Geometric
Poisson autoregression	`dgtf_poisson_ar(p, W)`	`sys_identity()`	Lag-specific
NegBin autoregression	`dgtf_nb_ar(p, W)`	`sys_identity()`	Lag-specific

```
mod <- dgtf_hawkes(meanlog = 1.5, sigma2 = 0.4, W = 0.005,
                   seasonality = seas_period(1, init = 2))
```

The component interface can be used instead when a nonstandard configuration is required. Complete constructor signatures and parameterization details are provided in the package documentation.

Fully specified model objects can also be passed to `dgtf_simulate_model()` to generate synthetic data, as illustrated in § 5.5.

5.3.2. Priors

Prior distributions for static parameters are specified separately from the model components. The separation is deliberate. The model object defines the structure and supplies initial or fixed values, while the prior object determines which static parameters are inferred. Any parameter given a prior is treated as unknown and inferred, and any parameter omitted is held fixed at the value used to construct the model object, so users can fit models with selected parameters fixed, for instance holding the serial interval at published values while inferring the innovation variance W and the overdispersion parameter ρ.

The prior constructors are organized by the support of the parameter, with `normal`

on the real line, `half_normal`, `half_cauchy`, `half_t`, `inv_gamma` and `dist_gamma` on the positive half line, `dist_beta` on the unit interval, and `uniform` on a bounded interval. The `dist_` prefix avoids conflicts with the base R functions `base::gamma()` and `base::beta()`.

A prior specification is constructed with `dgtf_prior(intercept, seasonality, W, rho, lag, ...)`. Each named slot accepts a single prior object for a scalar parameter or a list for a vector-valued one, so the `lag` argument takes a list when the lag distribution has more than one parameter, as with the lognormal lag of the Hawkes model.

```
prior <- dgtf_prior(W = inv_gamma(1, 1), rho = inv_gamma(1, 1),
                    lag = list(par1 = normal(0, 1),
                               par2 = inv_gamma(1, 1)))
```

Dropping the `lag` slot from this specification would leave the lognormal lag parameters fixed at the values supplied to `dgtf_hawkes()` or `dgtf_model()`.

5.3.3. Model fitting

The function `dgtf()` provides a common entry point for posterior inference. It takes the observed count series, a model object, and a prior specification, and returns an object of class `dgtf_fit`. The argument `method = "hva"` selects the hybrid variational approximation, whereas `method = "mcmc"` selects the block MCMC sampler. The two methods are summarized in § 5.4.

Method-specific settings are supplied through `vb_control()` or `mcmc_control()`. For example, an HVA fit is obtained with

```
fit_hva <- dgtf(y = sim$y,model = mod, prior = prior,
  method = "hva",control = vb_control(
    iter = 1000,n_sample = 5000, n_particle = 500))
```

The same model and prior can be fitted by MCMC by changing the `method` argument and supplying an `mcmc_control()` object. Thus, the choice of inference method does not require the model specification or prior to be rewritten. The optional `seed` argument controls reproducibility, and complete control settings are documented in the package help.

5.3.4. Forecasting

Out-of-sample forecasts are produced from a fitted model using either `predict(fit, horizon = h)` or the more explicit helper given by `dgtf_forecast_fit(fit, h, nrep, ypred_true, only_last)`. When `horizon = 0`, `predict()` returns the in-sample fitted intensity. When `horizon > 0`, it generates posterior predictive forecasts for times $T + 1, \ldots, T + h$.

A typical forecasting call is

```
fc <- dgtf_forecast_fit(fit, h = 10, ypred_true = ytest)
```

5.3.5. Inspecting a fitted model

A fitted model returned by `dgtf()` is an object of class `dgtf_fit` supporting the standard S3 methods, so that fitted models are inspected through familiar R workflows. `coef()` returns posterior medians of the inferred static parameters and `confint()` their credible intervals, `fitted()` returns the posterior median of the conditional mean λ_t,

`residuals()` returns response or Pearson residuals, and `predict()` returns in-sample fitted intensities when `horizon = 0` and out-of-sample forecasts when `horizon > 0`. The methods `print()`, `vcov()`, `nobs()` and `plot()` are also available.

Posterior predictive checks are generated with `posterior_predict(fit, nrep, Rt, level)`, which simulates replicated count series and computes the scoring metrics of § 5.3.6.

5.3.6. Model comparison

The function `dgtf_compare()` provides a common interface for comparing models, dispatching on the type of object supplied. A list of `dgtf_fit` or `dgtf_ppc` objects returns an in-sample comparison table, while a list of `dgtf_forecast` objects returns an out-of-sample table averaged across the horizon, so the same function serves both training-data and held-out comparisons.

The in-sample table reports the number of inferred static parameters, runtime summaries, and the posterior predictive metrics, namely the chi-square discrepancy, the CRPS, empirical coverage, mean interval width, and the Winkler interval score. For HVA fits it also reports the log marginal likelihood proxy from a robust tail summary of the SMC trace, reported as `NA` otherwise. The proxy is evaluated at the final variational values of the static parameters and is not integrated over their prior, so it measures how well a model fits the training counts once its parameters have been set, without accounting for parameter uncertainty or for the calibration of the predictive distribution. It is reported as a diagnostic of the variational optimization, and model selection should rest on the predictive scores in the remaining columns. The out-of-sample table reports empirical coverage, the chi-square discrepancy, and the CRPS, each averaged across the

forecast horizon.

The scoring metrics give complementary views of predictive performance in both comparisons. The CRPS combines point accuracy and dispersion into a proper scoring rule, so smaller values indicate better predictive distributions. Empirical coverage is a calibration check, reporting the fraction of observations inside the nominal predictive interval. The chi-square discrepancy compares squared residuals with predictive variance, where values near one indicate approximate variance calibration, values above one suggest underdispersion, and values below one suggest overdispersion. The mean interval width and the Winkler interval score summarize the sharpness and the calibration of the predictive intervals.

Supplying a reference R_t trajectory through `Rt_truth` adds columns measuring recovery of the latent trajectory, and the `truth` argument summarizes static parameter recovery across fits. A parameter-level comparison is available through function `dgtf_compare_params()`, which stacks the output of `param_recovery()` into a long-format table with one row per parameter and model. For an out-of-sample comparison, forecast objects are generated first and then passed to the same function.

```
fc_DH <- dgtf_forecast_fit(fit_DH, h = 14, ypred_true = ytest)
fc_DL <- dgtf_forecast_fit(fit_DL, h = 14, ypred_true = ytest)
dgtf_compare(list(DH = fc_DH, DL = fc_DL))
```

For comparison against a standard renewal equation baseline, `epiestim_baseline()` wraps **EpiEstim** (Cori et al., 2013) and returns posterior R_t bands in the format the plotting functions expect, so a baseline can be overlaid with **dgtf** fits through `dgtf_compare_plot()`. Its serial interval is supplied through a mean and standard deviation, unlike `dgtf_hawkes()`, whose lognormal lag is parameterized on the log

scale. Since **EpiEstim** is a suggested rather than an imported dependency, the function returns an informative error when it is not installed.

§ 5.4. Inference methods

The **dgtf** package provides two inference methods that share the same model and prior interface but differ in computational cost and posterior calibration. This section summarizes the methods at the level needed for the user to make an informed choice. Full algorithmic details and benchmarking are given in Chapter 2.

5.4.1. HVA

The hybrid variational approximation, selected with `method = "hva"`, combines SMC smoothing with stochastic gradient variational Bayes, and is developed in Chapter 2.

HVA suits exploratory analysis, model screening, sensitivity checks, and routine surveillance where many candidate models or repeated updates are needed. The trade-off, common to variational methods, is that posterior uncertainty can be understated relative to MCMC, especially for latent state trajectories, so MCMC may be preferable when calibrated posterior intervals are the primary goal. Post-hoc corrections (Giordano et al., 2018; Syring and Martin, 2019) can be applied when calibrated uncertainty is essential.

5.4.2. MCMC

The MCMC method (`method = "mcmc"`) targets the joint posterior $p(\boldsymbol{\gamma}, \psi_{1:T} \mid y_{1:T})$ through the block Gibbs sampler of Chapter 2, which updates the static parameters by

HMC and then the latent trajectory in a single block by a MH step, referred to as the disturbance update throughout the package. The HMC step size and number of leapfrog steps are tuned by dual averaging during warmup (Hoffman and Gelman, 2014), so the user normally does not set them. The sampler costs more than HVA but targets the exact posterior, and is the reference method for final uncertainty quantification.

5.4.3. Convergence diagnostics

The diagnostic tools depend on the inference method. For MCMC fits, `summary()` reports HMC diagnostics for the static parameter update, including the acceptance rate, leapfrog step size, number of leapfrog steps, energy diagnostics, and gradient norms, as well as MH diagnostics for the disturbance update, including mean acceptance rates and the number of low-acceptance time points. The parameter table also reports effective sample sizes and a single-chain split-$\hat{R}$, computed by splitting the retained chain into two halves. For stronger multi-chain checks, users can run independent fits and call `rhat(list(fit1, fit2, ...))`. The functions `ess()` and `rhat()` compute the effective sample size and the split-$\hat{R}$ diagnostic of Gelman and Rubin (1992), the latter accepting a numeric matrix of iterations by chains, a list of equal-length numeric vectors, or a list of independent `dgtf_fit` objects.

For HVA fits, split-$\hat{R}$ is not defined. **dgtf** instead records the SMC log marginal likelihood across variational iterations, then uses `summary()` and `plot(..., what = "marglik")` to summarize this trace through robust summaries from an early and a late window, an estimated stabilization iteration, and the last large outlier spike. These diagnostics assess the stabilization of the variational optimization, and are not Markov chain convergence diagnostics.

5.4.4. Choosing between methods

For daily monitoring, exploratory analysis, and initial model screening, HVA gives fast approximate posteriors that are typically accurate in point estimation. For final analyses in which calibrated uncertainty is the priority, MCMC provides the reference target. A practical workflow fits several candidate structures with HVA, compares them through posterior predictive checks and forecast scores, and refits the selected model with MCMC for reporting. The application below follows this workflow across three structures, while the simulation study checks the two methods against each other on a single model with a known truth. Chapter 2 documents the speed gains and the agreement of the point estimates.

§ 5.5. Simulation study

This section demonstrates a complete and reproducible **dgtf** workflow on a simulation for which the data generating process is known. The goals are to illustrate the package functions used in a typical analysis, to verify that the fitting functions recover known parameters and latent trajectories, to compare the two inference methods on the same model, and to demonstrate the posterior predictive and forecasting tools in a controlled setting. Comparison across structurally different models is demonstrated on real data in § 5.6.

5.5.1. Model specification and data generation

We generate data from a discrete-time Hawkes-type model DGTF model with negative binomial observations, identity link, softplus gain, lognormal lag distribution, and

Gaussian random walk evolution. The model is specified as follows.

```
R> mod <- dgtf_model(obs = obs_nbinom(), link = link_identity(),
+    sys = sys_shift(), gain = gain_softplus(),
+    lag = lag_lognormal(meanlog = 1.35, sigma2 = 0.32),
+    err = err_gaussian(W = 0.01),
+    seasonality = seas_period(1, init = 2))
```

The scalar seasonal component `seas_period(1, init = 2)` acts as an intercept with baseline level set to 2. The lag distribution corresponds to a lognormal serial interval with mean approximately 4.7 days and standard deviation approximately 2.9 days on the original scale. The innovation variance $W = 0.01$ produces a moderately smooth R_t trajectory.

We simulate $T = 210$ observations and split the resulting series into a training set of 200 observations and a test set of 10 observations. The same simulated object also contains the latent driver ψ_t, which we transform using the softplus gain to obtain the true reproduction number trajectory $R_t = \log\{1 + \exp(\psi_t)\}$.

```
R> sim     <- dgtf_simulate_model(mod, ntime = 210, seed = 3269)

R> ytrain  <- c(sim$y)[1:200]
R> ytest   <- c(sim$y)[201:210]

R> Rt_train <- log1p(exp(c(sim$psi)))[1:200]
R> Rt_test  <- log1p(exp(c(sim$psi)))[201:210]

R>plot(sim, what = "y", split = 200)
```

```
R>plot(sim, what = "Rt", split = 200)
```

The dgtf_sim object holds the simulated counts and the latent R_t trajectory, and its plot method displays either.

5.5.2. Fitting with HVA and MCMC

We fit the model with both inference methods. The prior treats the intercept, innovation variance, overdispersion parameter, and the two lognormal lag parameters as unknown.

```
R> prior <- dgtf_prior(seasonality = normal(0, 10), W = inv_gamma(1, 1),
+    rho = inv_gamma(1, 1), lag = list(par1 = normal(0, 1),
+    par2 = inv_gamma(1, 1)))
```

Because the model argument of dgtf() both specifies the structure and supplies starting values for the inferred parameters, we deliberately initialize at values away from the truth, so that the reported recovery is not an artifact of starting there.

```
R> set.seed(123)
R> m0_init <- dgtf_hawkes(meanlog = runif(1, 1, 2),
+  sigma2 = runif(1, 0.1, 0.5), W = 1 / rgamma(1, 10, 1),
+     seasonality = seas_period(1, init = runif(1, 1, 10)))
R> fit <- dgtf(y = ytrain, model = m0_init, prior = prior, method =
↪  "hva",
+    control = vb_control(iter = 1000, n_sample = 5000, n_particle =
↪  500))
R> summary(fit)

DGTF model fit (method = HVA)
```

```
------------------------------------------------
Elapsed     : 154.35 s   (optim 104.76 s, sampling 49.56 s)


Static parameters
 parameter   mean median      sd    q025   q975
         W 0.0115 0.0114 0.00133 0.00906 0.0143
      seas   2.36    2.3   0.582    1.41   3.67
       rho   34.1   33.7    5.09    25.2     45
      par1   1.05   1.05    0.16   0.741   1.36
      par2  0.367  0.344   0.129   0.184  0.681


Convergence (HVA marglik trace)
  iterations             : 1000
  MAD ratio (late/early) : 0.161  [substantial stabilization]
  tail drift (in MADs)   : -0.05  [flat]
  stabilized             : iter ~ 359
  last large spike       : iter ~ 447
```

The convergence block summarizes the SMC log marginal likelihood trace through robust tail and head statistics, an auto-detected stabilization iteration, and the last large outlier spike. A mean absolute deviation (MAD) ratio between the late and early windows well below one indicates that the variational parameters have settled. The fit diagnostics are produced through the same plotting interface used for fitted objects elsewhere, with `plot(fit, what = "Rt", truth = Rt_train)` showing the posterior median of R_t following the simulated truth closely and `plot(fit, what = "marglik")` showing the trace stabilizing after the initial optimization phase.

The same model and prior objects are refitted by MCMC with only the method and control object changed.

```
R> set.seed(123)
R> fit_mcmc <- dgtf(y = ytrain, model = m0_init, prior = prior,
+    method = "mcmc", control = mcmc_control(iter = 5000, warmup = 2000,
+    thin = 5))
```

5.5.3. Parameter recovery and comparison

Because the data generating values are known here, `param_recovery()` scores the posterior against them, reporting bias, mean absolute error, root mean square error, CRPS, and whether the credible interval covers the truth.

```
R> truth_list <- list(W = 0.01, seas = 2, rho = 30,
+                     par1 = 1.35, par2 = 0.32)
R> param_recovery(fit, truth_list)
```

The HVA fit recovers every parameter within its 95% credible interval. For multi-fit recovery, the corresponding function is `dgtf_compare_params()`, which stacks the parameter recovery output from each fit into a long-form table. The full output is reformatted as Table 5.3.

```
R> dgtf_compare_params(list(HVA = fit, MCMC = fit_mcmc),
+                       truth = truth_list)
```

Multi-fit overlays of R_t trajectories and individual static parameters are produced by the function `dgtf_compare_plot()`. The function accepts a heterogeneous list of `dgtf_fit` objects and numeric truth vectors, with the latter auto-styled as a dashed

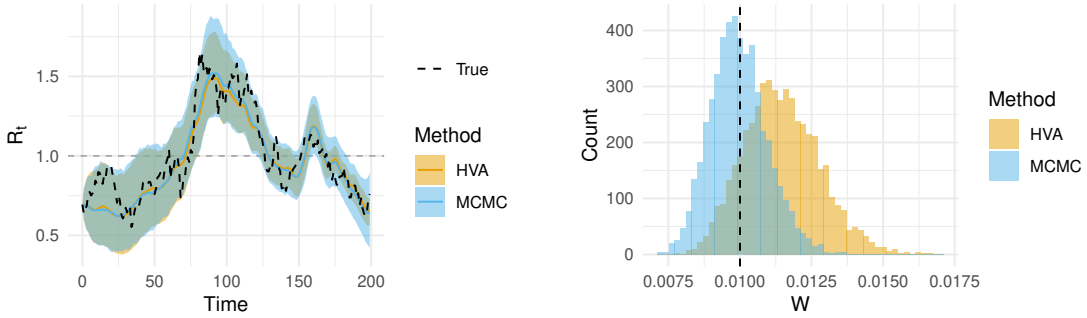


Figure 5.2.: HVA versus MCMC comparison. The left panel shows the posterior median R_t band from each method with the simulated truth overlaid. The right panel shows overlaid posterior histograms of the innovation variance W with the truth marked by a dashed vertical line.

black line for time-series targets or as a dashed vertical line for static parameters. As shown in Fig. 5.2, the HVA and MCMC posterior trajectories for R_t are nearly indistinguishable, and the posteriors of the static parameters are broadly consistent. The two posterior distributions for W have slightly different centers and shapes, for example, but both concentrate near the true value and their central credible intervals overlap.

```
R> dgtf_compare_plot(list(HVA = fit, MCMC = fit_mcmc, True = Rt_train),
+     what = "Rt")
R> dgtf_compare_plot(list(HVA = fit, MCMC = fit_mcmc, True = 0.01),
+     what = "W")
```

5.5.4. Posterior predictive checks and forecasting

The `posterior_predict()` function generates posterior predictive samples and computes scoring metrics for the observed counts and, when a reference trajectory is supplied through the `Rt` argument, for R_t as well. The returned object attaches to the fit through

Table 5.3.: Static parameter recovery for the HVA and MCMC fits, computed using function `dgtf_compare_params()` on the training data. Columns report the posterior mean, median, standard deviation, central 95% credible interval, mean absolute error, CRPS, and an indicator for whether the credible interval covers the truth.

	Method	Truth	Mean	Median	SD	2.5%	97.5%	MAE	CRPS	Cover
par1	HVA	1.35	1.05	1.05	0.160	0.741	1.36	0.303	0.213	✓
	MCMC	1.35	1.11	1.11	0.173	0.782	1.46	0.258	0.160	✓
par2	HVA	0.32	0.367	0.344	0.129	0.184	0.681	0.098	0.029	✓
	MCMC	0.32	0.502	0.462	0.205	0.226	1.010	0.199	0.090	✓
rho	HVA	30	34.1	33.7	5.09	25.2	45.0	5.10	2.24	✓
	MCMC	30	29.7	29.5	4.54	21.7	39.1	3.63	1.08	✓
seas	HVA	2	2.36	2.30	0.582	1.41	3.67	0.517	0.195	✓
	MCMC	2	2.44	2.43	0.545	1.39	3.57	0.562	0.255	✓
W	HVA	0.01	0.011	0.011	0.001	0.009	0.014	0.002	0.001	✓
	MCMC	0.01	0.010	0.010	0.001	0.008	0.012	0.001	0.000	✓

the `ppc` attribute, after which the plotting and comparison functions reuse it rather than recomputing.

```
R> attr(fit, "ppc") <- posterior_predict(fit, nrep = 100, Rt = Rt_train)
R> summary(attr(fit, "ppc"))

<dgtf posterior predictive summary, level = 95%>


Scalar metrics:
 target         metric value nominal_level
      y     chi-square 0.963
      y           CRPS  32.5
      y       coverage  0.97          0.95
     Rt            MAE 0.116
     Rt       coverage 0.884          0.95
```

The chi-square discrepancy close to one and the empirical 95% coverage close to nominal indicate good calibration of the predictive distribution. The corresponding MCMC values are 1.02 and 0.975. Scoring R_t against a known trajectory is possible only in simulation, and the coverage of 0.884 against a nominal 0.95 is the understatement of latent state uncertainty expected of a variational method.

Forecasts for the held-out window are produced and scored by `dgtf_forecast_fit()`, and `plot_dgtf_y()` renders the in-sample and out-of-sample bands on a continuous time axis. On this test window the forecast attained coverage of 1, a chi-square discrepancy of 0.094, and a CRPS of 23.5.

```
R> fc <- dgtf_forecast_fit(fit, h = 10, ypred_true = ytest)
R> plot_dgtf_y(fit, fc)
```

The remaining parts of the workflow, namely comparison across structurally different models, out-of-sample forecast scoring, and comparison against an external benchmark, are demonstrated on real data in § 5.6 and are not repeated here on simulated data. The functions involved, `dgtf_compare()`, `dgtf_forecast_fit()` and `epiestim_baseline()`, are described in § 5.3.4 and 5.3.6.

§ 5.6. Application: COVID-19 surveillance in Santa Cruz County

This section applies the **dgtf** package to daily COVID-19 case counts from Santa Cruz County, California, illustrating the model comparison workflow on real surveillance data where the true data generating process is unknown. Unlike simulation studies, the

data generating process and the true R_t trajectory are unknown so relevant questions here concern in-sample fit and out-of-sample forecast accuracy across structurally different models, along with calibration of the predictive distribution. The full 518-day series is analyzed in Chapter 2. The analysis below reproduces the workflow on a 264-day subset.

5.6.1. Data

The `covid_scz_2020` dataset bundled with the package contains $T = 518$ daily counts of confirmed COVID-19 cases from Santa Cruz County, California, drawn from the California Department of Public Health and spanning July 1, 2020 to December 1, 2021. The series covers two substantial pandemic waves and shows a pronounced day-of-week reporting periodicity attributed to testing and reporting practices. To keep the MCMC runtime tractable for a vignette demonstration, we restrict the analysis to the first 264 days of the series, splitting those into a training set of $T_{\text{train}} = 250$ observations and a test set of $h = 14$ days for forecast evaluation.

```
R> ytrain <- covid_scz_2020[1:250]
R> ytest  <- covid_scz_2020[251:264]
```

5.6.2. Model configurations

We fit three of the canonical models from Table 5.2 on the training data, namely a discrete-time Hawkes-type model (DH), a distributed lag (DL), and an autoregression (AR). All three use a negative binomial observation distribution, an identity link, a softplus gain function, and weekly seasonality entered through the regression term,

so that the comparison isolates the effect of the transfer function structure rather than confounding it with the choices at the observation level. The substantive trade-offs between these three families are discussed in § 5.2.2, and the relevant points for the present comparison are that DH separates R_t from the lag distribution, DL trades a more restrictive lag family for computational efficiency, and AR does not produce an R_t output suitable for epidemiological communication. The DL lag order is fixed at $r = 2$ and the AR order $p = 3$ are taken from Chapter 2, where they were selected by minimizing the χ^2 discrepancy across $r \in \{2, \ldots, 9\}$ and $p \in \{2, \ldots, 6\}$ on the full 518-day series. The three model specifications are below.

```
R> set.seed(123)
R> mod_DH <- dgtf_model(obs = obs_nbinom(), link = link_identity(),
+    sys = sys_shift(), gain = gain_softplus(), lag =
+    lag_lognormal(meanlog = rnorm(1, 1, 1), sigma2  = 1 / rgamma(1, 10,
↪  1)),
+    err = err_gaussian(W = 1 / rgamma(1, 10, 1)),
+    seasonality = seas_period(7, init = runif(7, 0, 10)))
R> mod_DL <- dgtf_distributed_lag(r = 2, kappa = runif(1, 0.1, 0.9),
+    seasonality = seas_weekly(init = runif(7, 0, 10)))
R> mod_AR <- dgtf_nb_ar(p = 3, W = 1 / rgamma(1, 10, 1),
+    seasonality = seas_weekly(init = runif(7, 0, 10)))
```

The three models share mildly informative priors on the components they have in common, namely the seasonality, the innovation variance W, and the overdispersion ρ, and differ only in the lag prior, since the three families parameterize the lag structure differently. The discrete-time Hawkes-type model places a normal prior on the log-

scale mean and an inverse-gamma prior on the log-scale variance of its lognormal lag. The distributed lag model replaces these with a Beta$(1, 1)$ prior on the geometric decay parameter κ, supplied through the single `lag` slot `par1`. The autoregressive model omits the `lag` entry, since its lag structure is the discrete order p rather than a continuous parameter.

```
R> prior_DH <- dgtf_prior(seasonality = normal(0, 10), W = inv_gamma(1,
↪  1),
+    rho = inv_gamma(1, 1),
+    lag = list(par1 = normal(0, 1), par2 = inv_gamma(1, 1)))
R> prior_DL <- dgtf_prior(seasonality = normal(0, 10), W = inv_gamma(1,
↪  1),
+    rho = inv_gamma(1, 1), lag = list(par1 = dist_beta(1, 1)))
R> prior_AR <- dgtf_prior(seasonality = normal(0, 10), W = inv_gamma(1,
↪  1),
+    rho = inv_gamma(1, 1))
```

5.6.3. Posterior inference

We fit the discrete-time Hawkes-type model with both inference methods, MCMC and HVA, and the distributed lag and AR models with HVA only. Comparing HVA against MCMC on the DH model is a practical diagnostic of whether the fast approximate posterior agrees with the MCMC reference on this dataset, not a formal validation of the approximation.

```
R> set.seed(123)
R> fit_DH <- dgtf(ytrain, mod_DH, prior_DH, method = "hva",
```

```
+    control = vb_control(iter = 1000, n_sample = 5000, n_particle =
↪  500))
R> fit_DH_mcmc <- dgtf(ytrain, mod_DH, prior_DH, method = "mcmc",
+    control = mcmc_control(iter = 10000, warmup = 5000, thin = 1))
R> fit_DL <- dgtf(ytrain, mod_DL, prior_DL, method = "hva",
+    control = vb_control(iter = 1000, n_sample = 5000, n_particle =
↪  500))
R> fit_AR <- dgtf(ytrain, mod_AR, prior_AR, method = "hva",
+    control = vb_control(iter = 1000, n_sample = 5000, n_particle =
↪  500))
```

The two inference methods agreed closely on the discrete-time Hawkes-type model. Static parameters, day-of-week seasonal effects, and the posterior R_t trajectory had overlapping 95% credible intervals throughout, and the in-sample posterior predictive scores and 14-day forecast scores were closely similar. HVA ran roughly 15 times faster than MCMC on this dataset. The parameter-by-parameter comparison, the seasonality and R_t plots, and the combined in-sample and forecast score table are deferred to § C.1.

The discrete-time Hawkes-type posterior predictive draws and 14-day forecast object are cached for reuse in the model comparison of § 5.6.4 and the forecasting comparison of § 5.6.5.

```
R> set.seed(123)
R> attr(fit_DH, "ppc") <- posterior_predict(fit_DH)
R> fc_DH <- dgtf_forecast_fit(fit_DH, h = 14, ypred_true = ytest)
```

The DL and AR models are fitted with HVA using analogous prior specifications and control settings, and their posterior predictive draws are cached in the same way before

Table 5.4.: Model comparison on the COVID-19 data from `dgtf_compare()`. The in-sample block reports the number of inferred static parameters, the HVA proxy for the log marginal likelihood, the CRPS, the chi-square discrepancy, the empirical 95% coverage and mean width of the posterior predictive interval, the Winkler interval score, and the total runtime. The forecast block reports coverage, the chi-square discrepancy, and the CRPS averaged across the 14-day held-out horizon.

		In-sample							14-day forecast		
Model	n_{par}	log-margl.	CRPS	χ^2	Cover	Width	IS	Time (s)	Cover	χ^2	CRPS
DH	11	5.91	9.43	1.47	0.952	62.8	84.8	264.0	0.929	0.280	3.28
DL	10	6.61	9.74	1.46	0.956	68.4	85.9	380.2	0.929	0.263	3.67
AR	9	10.70	12.40	2.38	0.936	83.5	109.0	382.3	1.000	0.264	4.00

the comparison in § 5.6.4.

5.6.4. In-sample comparison

For an in-sample comparison of the three models, we cache the distributed lag and AR posterior predictive draws (the discrete-time Hawkes-type draws were cached in § 5.6.3) and pass the three fits to `dgtf_compare()`, which assembles Table 5.4 with one row per model.

```
R> set.seed(123)
R> attr(fit_DL, "ppc") <- posterior_predict(fit_DL, nrep = 100)
R> attr(fit_AR, "ppc") <- posterior_predict(fit_AR, nrep = 100)
R> dgtf_compare(list(DH = fit_DH, DL = fit_DL, AR = fit_AR))
```

The DH and DL models perform comparably (Table 5.4). DH attains the slightly lower CRPS, 9.43 against 9.74, and the two are essentially tied on the chi-square discrepancy. The AR model is noticeably worse, with a CRPS roughly 27% larger and a chi-square of 2.38 indicating a predictive distribution too narrow for the residuals it

produces. All three chi-square values exceed one, so in-sample residuals are larger than the predictive variances throughout. The separation of DH and DL from AR matches the ordering reported in Chapter 2 for the full 518-day series.

5.6.5. Out-of-sample forecasting

The 14-day forecasts and their scoring against the held-out test set are produced for each fit through `dgtf_forecast_fit()` and collected through the same `dgtf_compare()` interface. The discrete-time Hawkes-type forecast `fc_DH` was already computed in § 5.6.3; here we add the distributed lag and AR forecasts and compare all three, producing Table 5.4. We note that Chapter 2 evaluates forecast performance through a rolling one-step-ahead procedure rather than the multi-step forecast demonstrated here. The two protocols answer different operational questions, and the multi-step variant is more informative about the package's forecasting interface. With only 14 held-out observations, the scores below illustrate that interface and serve as a short-horizon diagnostic.

```
R> fc_DL <- dgtf_forecast_fit(fit_DL, h = 14, ypred_true = ytest)
R> fc_AR <- dgtf_forecast_fit(fit_AR, h = 14, ypred_true = ytest)
R> dgtf_compare(list(DH = fc_DH, DL = fc_DL, AR = fc_AR))
```

The DH model achieves the lowest out-of-sample CRPS of 3.28, as shown in Table 5.4. Coverage should be read cautiously on a window this short, since each missed observation moves the empirical value by about seven percentage points. DH and DL share a coverage of 0.929, both missing one observation during a brief drop in the test window, but DL has the worse CRPS of 3.67, reflecting slightly wider intervals around a

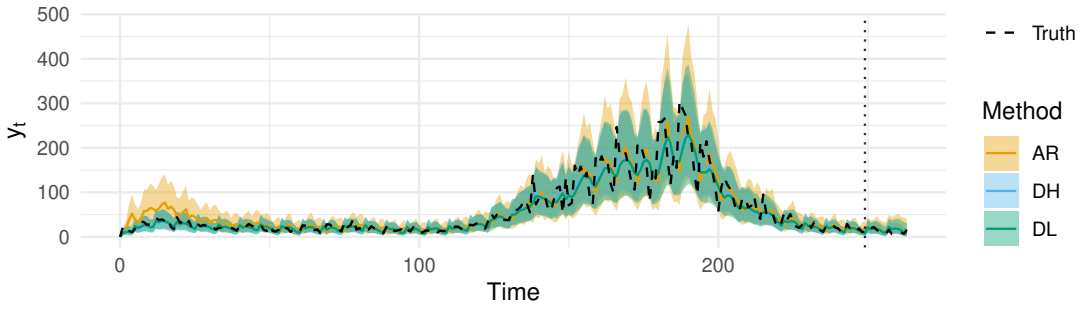


Figure 5.3.: Combined posterior predictive and forecast bands for the three model variants on the COVID-19 data. The in-sample posterior predictive band over the 250-day training window and the out-of-sample forecast band over the 14-day test window are drawn in the same color for each model, and a vertical dotted line marks the train/test boundary. The black dashed line represents observed counts.

similar median. The AR model reaches coverage of 1.0 by issuing very wide predictive intervals, which shows in its chi-square of 0.264, well below the calibrated value of one, so it is the most underconfident of the three.

The in-sample posterior predictive bands and the 14-day forecast bands for all three models are shown together in Figure 5.3, produced by passing the fitted objects and their forecasts to `plot_dgtf_y_compare()`:

```
R> fits_list <- list(DH = fit_DH, DL = fit_DL, AR = fit_AR)
R> fcs_list  <- list(DH = fc_DH,  DL = fc_DL,  AR = fc_AR)
R> plot_dgtf_y_compare(fits_list, fcs_list)
```

Taken together, the COVID-19 application shows how **dgtf** supports a real-data workflow in which structurally different models are specified, fitted, compared, visualized, and evaluated through one interface, and it separates models with a direct R_t interpretation from autoregressive count models that forecast usefully but are less directly interpretable.

§ 5.7. Concluding remarks

The **dgtf** package provides a unified R interface for Bayesian count time series models in the DGTF class, making a flexible state-space framework available through a compositional software interface. Users assemble models from modular components for the observation distribution, link function, gain function, lag distribution, system equation, and innovation noise, and then fit, forecast, diagnose, and compare them through a common workflow. The same interface specifies discrete-time Hawkes-type models, recursive distributed lag models, and AR count models, which differ in how past counts affect future incidence. The Hawkes and distributed lag formulations separate a time-varying transmission intensity R_t from the lag structure and are directly interpretable in epidemiological applications, while AR count models offer a simpler forecasting alternative whose lag-specific coefficients admit no reproduction number interpretation.

A second contribution is the unified inference interface, under which the same model and prior objects are fitted by either HVA or MCMC. HVA gives a fast approximate posterior suited to exploratory analysis, routine surveillance, and screening candidate structures, while MCMC costs more but targets the exact posterior and is better suited to final analyses where calibrated uncertainty matters. The COVID-19 application follows this workflow, screening three structures with HVA before comparing the selected one against MCMC, while the simulation study checks the two methods against each other on a single model with a known truth.

The package also provides an integrated evaluation toolkit. Posterior predictive checks, multi-step forecasts, parameter recovery diagnostics, and model comparison tables are available through standard functions, and the application shows why several

scoring rules are reported side by side, since on the held-out window the AR model attains the highest coverage while carrying the worst CRPS. The visualization functions complement these summaries by displaying latent R_t trajectories, posterior predictive bands, forecast bands, and estimated lag distributions.

Four limitations of the current implementation are worth noting. First, the package is restricted to univariate series. The network Hawkes-type extension for inter-regional transmission is developed in Chapter 3 and the variational methods that scale it to many regions in Chapter 4, and integrating both so that multivariate models share the same constructor, fitter, and evaluation toolkit is the natural next step. Second, pointwise log-likelihood values are not exposed by either inference method, which prevents generic information criteria and leave-one-out diagnostics that depend on the contribution of each observation. Third, the inferred lag distribution should not be read as a serial or generation interval when the model is fitted to reported case counts, since those counts convolve the epidemiological interval with incubation, testing, and administrative reporting delays, a point Chapter 2 develops in detail. Fourth, although HVA screens models quickly and usually gives point estimates close to MCMC, its posterior intervals can be under-dispersed, particularly for latent states, so MCMC or a post-hoc calibration is the safer route when interval calibration is central.

The illustrations in this chapter cover the core workflow. The package documentation and vignettes carry the current interface in full, together with reproducible workflows and worked examples.

Chapter 6

Discussion and future directions

The thesis has developed a unified Bayesian framework for count time series in infectious disease surveillance, organized around three threads identified in § 1.3. Chapter 2 introduced the dynamic generalized transfer function (DGTF) class as a single state-space model that contains discrete-time Hawkes-type models, Poisson autoregressions, and distributed lag models as special cases of a common conditional mean structure, and developed a hybrid variational approximation (HVA) algorithm that achieves a roughly three-fold speedup over MCMC with minimal accuracy loss across the model variants. Chapter 3 extended the framework to spatially connected populations through a multivariate Hawkes model, with a regularized horseshoe prior that learns the transmission network from data, an additive decomposition of the closed-population reproduction number into local and imported components, and a Markov-switching state for structural zeros. Chapter 4 developed the particle Laplace variational approximation (PLVA) as the variational counterpart to the block MCMC sampler of Chapter 3, combining an efficient importance variational approximation (EIVA) for the latent reproduction trajectory with a mixture of Laplace approximations for the spatial parameters, whose between-

particle component represents the ridge directions that joint Gaussian and mean-field families would collapse to a single point. Chapter 5 described the **dgtf** package for R, which exposes the six-component framework of Chapter 2 through a compositional interface aligned with the modular structure of the model class and dispatches to either inference method through a common entry point.

The chapters interrelate in three ways that the thesis has used as organizing principles. The univariate model of Chapter 2 and the spatial model of Chapter 3 share the self-exciting conditional mean form $\mathrm{E}[y_t \mid \mathcal{D}_{t-1}] = \mu + R_t \sum_k \phi_k y_{t-k}$ that was introduced in § 1.2.2; the multivariate generalization replaces the scalar R_t with a vector $\{R_{k,t}\}$ and introduces a transmission matrix that allocates secondary infections across destinations. The sampler of Chapter 3 and the variational method of Chapter 4 share the column-wise conditional update structure that the column-stochastic constraint makes available; the PLVA transcribes the structure of the block sampler into a particle approximation in which each EIVA proposal targets a column-conditional law and each Laplace component targets the corresponding column-conditional spatial posterior. The univariate methodology of Chapter 2 and the software interface of Chapter 5 share the six-component decomposition that motivated both. The package's constructor families correspond one-to-one with the model components, and its inference methods are the HVA and MCMC algorithms developed in the methodological chapter.

The empirical findings reported in Chapters 2 and 3 carry several lessons for surveillance practice. The simulation study of Chapter 2 showed that the structural choice of transfer function family matters, and that a discrete-time Hawkes-type model with a lognormal lag distribution outperformed both a geometric distributed lag variant and a negative binomial autoregression on out-of-sample forecast scores from the COVID-19

application in Santa Cruz County. The Santa Cruz application also showed that explicit weekly seasonality reduces bias in the R_t estimates during low-incidence periods, and that the HVA variance underestimation typical of Gaussian variational methods is modest but real, with 95% credible interval coverage falling from 0.97 under MCMC to 0.87 under HVA. The California application of Chapter 3 showed that closed-population reproduction numbers near one can mask substantial heterogeneity across locations, with intrinsic source-specific reproduction numbers ranging from 0.55 in San Joaquin to 1.14 in Santa Clara, and that the estimated transmission network shows a source and sink structure in which Bay Area employment counties appear as net sources and Central Valley residential counties as net sinks. None of these patterns are recoverable under the closed-population assumption that drives standard reproduction number estimators, and they suggest that decomposing the observed reproduction number into local and imported components changes the actionable interpretation of a surveillance signal.

Several limitations of the framework deserve emphasis. The variational approximation of Chapter 4 rests on a local unimodality assumption for the conditional posterior of the spatial block that the algorithm cannot verify internally, and the accuracy of the importance step depends on the effective sample size of the EIVA particle set, which can degrade when the working linearization is poor. More broadly, both variational methods developed in this thesis underestimate posterior uncertainty relative to MCMC. The HVA of Chapter 2 undercovers the latent R_t intervals, at about 0.87 against 0.97 for MCMC, and the PLVA of Chapter 4 produces intervals on the spatial parameters that are narrower than the MCMC reference by a factor of 2.8 on the retention rates and 1.7 on the transmission weights. In both cases the point estimates and the predictive distributions track MCMC closely. This is a central limitation of the variational route.

Where calibrated posterior intervals are themselves the inferential target, MCMC remains the reference, and the variational methods are best used for fast point estimation, forecasting, and model comparison, or paired with periodic MCMC validation. The spatial weights in Chapter 3 are treated as static, so the model does not capture temporal changes in transmission pathways over the course of an outbreak, and validation of the inferred pathways is necessarily indirect, since the model identifies lagged statistical dependence in incidence rather than physical transmission. The **dgtf** package of Chapter 5 is currently restricted to univariate time series and does not yet expose the multivariate extension. None of these limitations is structural to the framework, but each defines a direction for future work.

Methodological extensions span several axes. Continuous-time formulations of both the DGTF class and the network Hawkes model would accommodate surveillance data reported on irregular schedules and would connect the discrete-time models developed here to the continuous-time Hawkes literature. Time-varying spatial weights would relax the static $\boldsymbol{\alpha}$ assumption of Chapter 3 at the cost of additional identifiability constraints and computational overhead, and offer a natural place to incorporate auxiliary data on mobility patterns evolving over time. For the PLVA in particular, the under-coverage on the spatial parameters points to two refinements that fit within the block-coordinate structure: a joint importance proposal over the temporal and spatial blocks within each column, and a heavier-tailed mixture family in place of the Gaussian Laplace components. A complementary direction is reformulating the PLVA on a coherent importance-weighted variational bound (IWAE) bound, trading the analytical EIVA calibration for a single coherent variational objective. Loss-based variational objectives (Frazier et al., 2025) would tailor the inference procedure to specific epidemiological

loss functions, such as those that penalize false negatives during outbreak detection or that target asymmetric costs of over-restriction and under-restriction. Adaptation to count processes with similar multiplicative structure outside epidemiology, such as crime incidence and earthquake aftershocks, is a promising direction for transferring the methodology to other applied domains.

Computational extensions track the dominant scaling regimes. For online surveillance in which data arrive sequentially, recursive updating (Tomasetti et al., 2022; Mastrototaro and Olsson, 2024) and nested sequential Monte Carlo (Naesseth et al., 2019) would enable fully online inference at the cost of a single particle update per observation rather than re-estimating the model from scratch. For larger spatial scales, divide-and-conquer sequential Monte Carlo (Lindsten et al., 2017) would enable parallel processing of column-wise updates across spatial subregions, and would allow the framework to scale to surveillance domains with hundreds of locations. Adaptive control of the EIVA particle count per column based on effective sample size, rather than a fixed value across all columns, would route computational effort to columns where the linearization is least accurate. Natural gradient updates (Zhang et al., 2024) could improve convergence of the variational optimization in regimes where the posterior is highly anisotropic.

The **dgtf** package is a natural vehicle for disseminating the methodology developed here, and its extension to the multivariate setting is the most concrete next step on the software side. The component-based interface and the unified inference dispatch of Chapter 5 generalize naturally once the spatial weight matrix is added as a model component, and the PLVA of Chapter 4 is the natural variational counterpart of the HVA that the package already provides for the univariate class. The R package design also lends itself to integration into operational surveillance pipelines, where the relevant

interfaces are not interactive but programmatic. Beyond the package, public release of the simulation and application code as a reproducibility supplement is planned, alongside the development of vignettes that walk a user through the full workflow on representative datasets. Such infrastructure has been central in turning related Bayesian methodologies into routine surveillance practice, and the framework developed in this thesis is well placed to follow that path.

The thesis has approached infectious disease surveillance as a structured statistical problem in which model unification, spatial extension, and scalable computation are mutually reinforcing rather than separable concerns. The unified state-space framework of Chapter 2 clarifies which structural choices matter; the spatial extension of Chapter 3 shows that those choices interact non-trivially with the geographic structure of transmission; the structured variational approximation of Chapter 4 shows that the resulting inference problem is tractable at operationally relevant scales; and the package of Chapter 5 provides a vehicle for these methods to leave the dissertation and reach surveillance practice. The COVID-19 pandemic that motivated much of the work documented in this thesis is unlikely to be the last outbreak in which decisions are made under data that are noisy, partial, and arriving daily. The framework developed here is one contribution toward making those decisions on a firmer statistical footing.

Appendix

A. Supplement to Chapter 2

This appendix collects the additional simulation details, derivations, convergence diagnostics, and sensitivity analyses that support the results reported in Chapter 2.

§ A.1. Simulation from discrete-time Hawkes-type model

A.1.1. Model specification

First, we specify the data generating process. The model $\mathcal{M}_0$, from which the data is simulated, is defined as follows:

$$y_t \mid \lambda_t \sim \mathrm{NB}(\lambda_t, \rho),\ \rho > 0, \quad \lambda_t = \sum_{l=1}^{L} \phi_l\, h(\psi_{t+1-l})\, y_{t-l}, \quad \psi_t = \psi_{t-1} + w_t,\ w_t \sim N(0, W),$$

where $\rho = 30$, $W = 0.01$, and $h(\psi_t) = \log(\exp(\psi_t) + 1)$. The lags $\{\phi_l\}$ are PMF of a discretized log-normal distribution,

$$\phi_l = \Phi\left(\frac{\log(l) - \mu}{\sigma}\right) - \Phi\left(\frac{\log(l-1) - \mu}{\sigma}\right),\ l = 2, \ldots, L.$$

For $l = 1$, we define $\phi_1 = \Phi(-\mu/\sigma)$. The truncation parameter L is selected to ensure $\sum \phi_l \geq 0.99$. We set logarithmic mean $\mu = 1.3863$ and logarithmic variance

$\sigma^2 = 0.3226$. Its expectation is approximately 4.7 and the standard deviation is 2.9. The static parameters in this specific model are $\boldsymbol{\gamma} = (\rho, \mu, \sigma^2, W)'$. Except for μ, which can span values across the entire real line, the remaining parameters must be positive. This model uses an identical link such that $\eta_t = \lambda_t$. This model has a DGTF representation as follows:

$$y_t \mid \lambda_t \sim \text{NB}(\lambda_t, \rho),\ \rho > 0, \quad \lambda_t = f_t(\boldsymbol{\theta}_t, \mathbf{y}_{t-1}), \quad \boldsymbol{\theta}_t = \mathbf{G}\boldsymbol{\theta}_{t-1} + \mathbf{w}_t, \quad \mathbf{w}_t \sim N(\mathbf{0}, \mathbf{W}),$$

where $\boldsymbol{\theta}_t = (\psi_t, \ldots, \psi_{t+1-L})'$,

$$\mathbf{G} = \begin{pmatrix} \mathbf{e}_1 & 0 \\ \mathbf{I} & \mathbf{0} \end{pmatrix},\ \mathbf{e}_1 = (1, 0, \ldots, 0), \quad \mathbf{w}_t = (w_t, 0, \ldots, 0), \quad \mathbf{W} = \text{diag}(W, 0, \ldots, 0).$$

A.1.2. Gradient of the log joint probability for HVA and HMC

In the discrete-time Hawkes-type model, we have a set of static parameters $\boldsymbol{\gamma} = (\rho, \boldsymbol{\varphi}, \mu, \sigma^2, W)$. We will map all entries to the whole real line and use $\check{\boldsymbol{\gamma}} = (\check{\rho}, \check{\boldsymbol{\varphi}}, \check{\mu}, \check{\sigma}^2, \check{W})$ to represent the static parameters after this transformation, where

$$\check{\rho} = \log(\rho),\ \check{\boldsymbol{\varphi}} = \log(\boldsymbol{\varphi}),\ \check{\mu} = \mu,\ \check{\sigma}^2 = \log(\sigma^2),\ \check{W} = \log(W).$$

Further, we specify the following priors: $W \sim \text{IG}(a_w, b_w)$, $\rho \sim \text{IG}(a_\rho, b_\rho)$, $\sigma^2 \sim \text{IG}(a_\sigma, b_\sigma)$, $\mu \sim N(a_\mu, b_\mu)$, and for $\boldsymbol{\varphi}$ we use a normal prior on the logarithm scale: $\check{\boldsymbol{\varphi}} \sim N(\boldsymbol{a}_\varphi, \mathbf{B}_\varphi)$. We use $\pi(\check{\boldsymbol{\gamma}})$ to represent the prior distribution w.r.t $\check{\boldsymbol{\gamma}}$, and for the discrete-time Hawkes-type model, we essentially have:

$$\pi(\check{\boldsymbol{\gamma}}) = \pi(\check{\rho})\pi(\check{\boldsymbol{\varphi}})\pi(\check{\mu})\pi(\check{\sigma}^2)\pi(\check{W}).$$

The log joint probability $\log p(y_{1:T}, \boldsymbol{\theta}_{1:T}, \check{\boldsymbol{\gamma}})$, which we will denote as $\log p$ for brevity, is given by:

$$\log p = \log \pi(\check{\boldsymbol{\gamma}}) + \sum_{t=1}^{T} \log p(y_t \mid \lambda_t, \boldsymbol{\gamma}) + \sum_{t=1}^{T} \log p(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{t-1}, W).$$

Note that here we consider $\boldsymbol{\gamma}$ as a function of $\check{\boldsymbol{\gamma}}$. For gradient-based inference methods like HVA and HMC, we need to compute:

$$\nabla_{\check{\boldsymbol{\gamma}}} \log p = \left(\frac{\partial \log p}{\partial \check{\rho}}, \frac{\partial \log p}{\partial \check{\boldsymbol{\varphi}}}', \frac{\partial \log p}{\partial \check{\mu}}, \frac{\partial \log p}{\partial \check{\sigma}^2}, \frac{\partial \log p}{\partial \check{W}} \right)'.$$

Next we will derive each component of this gradient vector.

Partial derivative w.r.t. $\check{\rho}$ Since $\rho = \exp(\check{\rho})$, we have $\mathrm{d}\rho/\mathrm{d}\check{\rho} = \exp(\check{\rho})$. The prior density for $\check{\rho}$, derived from the inverse-gamma prior on ρ, is:

$$\pi(\check{\rho}) = \pi(\rho) \left| \frac{\mathrm{d}\rho}{\mathrm{d}\check{\rho}} \right| \propto \exp(-a_\rho \check{\rho} - b_\rho \exp(-\check{\rho})).$$

This gives us:

$$\frac{\mathrm{d} \log \pi(\check{\rho})}{\mathrm{d}\check{\rho}} = -a_\rho + b_\rho \exp(-\check{\rho}).$$

For a negative-binomial distribution, the log-likelihood is:

$$\log p(y_t \mid \lambda_t, \boldsymbol{\gamma}) = \log \Gamma(y_t + \rho) - \log \Gamma(\rho) - \log(y_t!) + \rho \log \left(\frac{\rho}{\lambda_t + \rho} \right) + y_t \log \left(\frac{\lambda_t}{\lambda_t + \rho} \right).$$

Differentiating with respect to ρ:

$$\frac{\partial \log p(y_t \mid \lambda_t, \boldsymbol{\gamma})}{\partial \rho} = \Psi(y_t + \rho) - \Psi(\rho) + \log \left(\frac{\rho}{\lambda_t + \rho} \right) + \frac{\lambda_t - y_t}{\lambda_t + \rho}.$$

Here, $\Psi(\cdot)$ is the digamma function. Combining these results via the chain rule, we have:

$$\frac{\partial \log p}{\partial \check{\rho}} = \frac{\mathrm{d} \log \pi(\check{\rho})}{\mathrm{d}\check{\rho}} + \sum_{t=1}^{T} \frac{\partial \log p(y_t \mid \lambda_t, \boldsymbol{\gamma})}{\partial \rho} \frac{\mathrm{d}\rho}{\mathrm{d}\check{\rho}}.$$

Partial derivative w.r.t. $\check{\boldsymbol{\varphi}}$ The partial derivative of log joint probability w.r.t. $\check{\boldsymbol{\varphi}}$ can be decomposed as follows:

$$\frac{\partial \log p}{\partial \check{\boldsymbol{\varphi}}} = \frac{\mathrm{d} \log \pi(\check{\boldsymbol{\varphi}})}{\mathrm{d}\check{\boldsymbol{\varphi}}} + \sum_{t=1}^{T} \frac{\partial \log p(y_t \mid \lambda_t, \boldsymbol{\gamma})}{\partial \lambda_t} \frac{\partial \lambda_t}{\partial \boldsymbol{\varphi}} \frac{\mathrm{d}\boldsymbol{\varphi}}{\mathrm{d}\check{\boldsymbol{\varphi}}}. \tag{A.1.1}$$

Note that here we use $\mathrm{d}\boldsymbol{\varphi}/\mathrm{d}\check{\boldsymbol{\varphi}}$ to specifically represent an element-wise operation with $\mathrm{d}\boldsymbol{\varphi}/\mathrm{d}\check{\boldsymbol{\varphi}} = (\mathrm{d}\varphi_k/\mathrm{d}\check{\varphi}_k)$ and $\mathrm{d}\varphi_k/\mathrm{d}\check{\varphi}_k = \exp(\check{\varphi}_k)$. We assume $\check{\boldsymbol{\varphi}}$ follows a normal prior: $\check{\boldsymbol{\varphi}} \sim N(\mathbf{a}_\varphi, \mathbf{B}_\varphi)$, and thus we have:

$$\frac{\mathrm{d} \log \pi(\check{\boldsymbol{\varphi}})}{\mathrm{d}\check{\boldsymbol{\varphi}}} = \mathbf{B}_\varphi^{-1}(\mathbf{a}_\varphi - \check{\boldsymbol{\varphi}}).$$

The remaining components in Eq. (A.1.1) are:

$$\frac{\partial \log p(y_t \mid \lambda_t, \boldsymbol{\gamma})}{\partial \lambda_t} = \frac{y_t}{\lambda_t} - \frac{y_t + \rho}{\lambda_t + \rho}, \ \frac{\partial \lambda_t}{\partial \boldsymbol{\varphi}} = \mathbf{x}_t.$$

Partial derivative w.r.t. $\check{\mu}$ In this model, we use a log-normal distribution $LN(\mu, \sigma^2)$ as the lag distribution with the lag ϕ_l defined as follows:

$$\phi_l = \Phi\left(\frac{\log(l) - \mu}{\sigma}\right) - \Phi\left(\frac{\log(l-1) - \mu}{\sigma}\right), \ l \in \mathbb{N}^+. \tag{A.1.2}$$

Here $\Phi(\cdot)$ represents the CDF of the standard normal distribution $N(0, 1)$. Since μ is already on the real line, no transformation is needed ($\check{\mu} = \mu$). The log derivative of the

normal prior density w.r.t $\check{\mu}$ is:

$$\frac{\mathrm{d}\log\pi(\check{\mu})}{\mathrm{d}\check{\mu}} = \frac{a_\mu - \mu}{b_\mu}.$$

Then, the partial derivative w.r.t $\check{\mu}$ can be decomposed as follows:

$$\frac{\partial \log p}{\partial \check{\mu}} = \frac{\mathrm{d}\log\pi(\check{\mu})}{\mathrm{d}\check{\mu}} + \sum_{t=1}^{T} \frac{\partial \log p(y_t \mid \lambda_t, \boldsymbol{\gamma})}{\partial \lambda_t} \frac{\partial \lambda_t}{\partial \mu} \frac{\mathrm{d}\mu}{\mathrm{d}\check{\mu}}.$$

The conditional intensity λ_t is:

$$\lambda_t = \mathbf{x}_t'\boldsymbol{\varphi} + \sum_{l=1}^{L} \phi_l h(\psi_{t+1-l}) y_{t-l}.$$

Its partial derivative w.r.t μ is:

$$\frac{\partial \lambda_t}{\partial \mu} = \sum_{l=1}^{L} \frac{\partial \phi_l}{\partial \mu} h(\psi_{t+1-l}) y_{t-l}, \text{ where } \frac{\partial \phi_l}{\partial \mu} = \frac{1}{\sigma}\left[\phi\left(\frac{\log(l-1)-\mu}{\sigma}\right) - \phi\left(\frac{\log(l)-\mu}{\sigma}\right)\right].$$

Here, $\phi(\cdot)$ represents the standard normal PDF, not to be confused with the lag weights ϕ_l.

Partial derivative w.r.t. $\check{\sigma}^2$ We can treat σ as a function of $\check{\sigma}^2$, $\sigma = \exp(\check{\sigma}^2/2)$, and the derivative is $\mathrm{d}\sigma/\mathrm{d}\check{\sigma}^2 = \exp(\check{\sigma}^2/2)/2$. With an inverse-gamma distribution IG(a_σ, b_σ) on σ^2,we have:

$$\frac{\partial \log\pi(\check{\sigma}^2)}{\partial \check{\sigma}^2} = -a_\sigma + b_\sigma \exp(-\check{\sigma}^2).$$

The partial derivative of ϕ_l w.r.t. $\check{\sigma}^2$ is:

$$\frac{\partial \phi_l}{\partial \check{\sigma}^2} = \frac{1}{2}\left[\phi\left(\frac{\log(l-1)-\mu}{\exp(\check{\sigma}^2/2)}\right)\frac{\log(l-1)-\mu}{\exp(\check{\sigma}^2/2)} - \phi\left(\frac{\log(l)-\mu}{\exp(\check{\sigma}^2/2)}\right)\frac{\log(l)-\mu}{\exp(\check{\sigma}^2/2)}\right].$$

Finally, we can plug in the above derivatives into the partial derivative of log joint probability w.r.t. $\check{\sigma}^2$:

$$\frac{\partial \log p}{\partial \check{\sigma}^2} = \frac{\mathrm{d} \log \pi(\check{\sigma}^2)}{\mathrm{d}\check{\sigma}^2} + \sum_{t=1}^{T} \frac{\partial \log p(y_t \mid \lambda_t, \boldsymbol{\gamma})}{\partial \lambda_t} \frac{\partial \lambda_t}{\partial \check{\sigma}^2},$$

where

$$\frac{\partial \lambda_t}{\partial \check{\sigma}^2} = \sum_{l=1}^{L} \frac{\partial \phi_l}{\partial \check{\sigma}^2} h(\psi_{t+1-l}) y_{t-l}.$$

Partial derivative w.r.t. $\check{W}$ With $\check{W} = \log(W)$ and $W \sim \mathrm{IG}(a_w, b_w)$, we have $\mathrm{d}W/\mathrm{d}\check{W} = \exp(\check{W})$ and

$$\frac{\partial \log \pi(\check{W})}{\partial \check{W}} = -a_w + b_w \exp(-\check{W}).$$

The partial derivative of the log joint probability w.r.t. $\check{W}$ is:

$$\frac{\partial \log p}{\partial \check{W}} = \frac{\partial \log \pi(\check{W})}{\partial \check{W}} + \sum_{t=1}^{T} \frac{\log p(\psi_t \mid \psi_{t-1}, W)}{\partial W} \frac{\mathrm{d}W}{\mathrm{d}\check{W}},$$

where

$$\frac{\log p(\psi_t \mid \psi_{t-1}, W)}{\partial W} = -\frac{1}{2W} + \frac{(\psi_t - \psi_{t-1})^2}{2W^2}.$$

This partial derivative w.r.t. $\check{W}$ is used in the HVA algorithm. For MCMC, W can be sampled directly via a Gibbs sampler or jointly with other static parameters using HMC in the MH step.

Table A.1.: Model-specific settings and hyperparameter choices for the five DGTF models compared in the simulation study. The table shows the model structure, key hyperparameters, prior specifications for lag distribution parameters, and variational rank k used in HVA inference.

	$\mathcal{M}_0$ Discretized Hawkes	$\mathcal{M}_1$ Discounted Discretized Hawkes	$\mathcal{M}_2$ Distributed Lag	$\mathcal{M}_3$ NB-AR	$\mathcal{M}_4$ Poisson AR
Hyperparameter	$L = 30$	$L = 30$, $\hat{\delta} = 0.9$	$\hat{r} = 4$	$\hat{p} = 3$	$\hat{p} = 5$
Lag Prior	$\mu \sim N(0, 1)$	$\mu \sim N(0, 1)$	$\kappa \sim \text{Beta}(1, 1)$		
	$\sigma^2 \sim \text{IG}(1, 1)$	$\sigma^2 \sim \text{IG}(1, 1)$			
Variational rank	$k = 5$	$k = 4$	$k = 4$	$k = 3$	$k = 3$

Table A.2.: Common inference settings applied consistently across all DGTF models in simulation. Prior specifications are shown for parameters shared across models, along with algorithm-specific settings for HVA (two-filter smoother and gradient ascent optimization) and MCMC sampling.

Prior specifications	ρ and W	φ_i
	IG(1, 1)	$N(0, 10)\delta(\varphi_i > 0)$
HVA	Iterations	Posterior samples
	1,000	5,000
Two-filter smoother	Particles	Resampling ratio
	500	0.5
Gradient ascent	Initial step size	ADADELTA learning rate
	10^{-5}	0.02
MCMC	Burnin-in	Final samples
	2,000	5,000 (thinning = 2)
HMC - Leapfrog	Integration steps	Step size
	50	0.005

A.1.3. Implementation

Having specified the data generating process in the previous section, we now detail the implementation settings for all models in our comparison study. Table A.1 presents the model-specific settings across all five models considered. For the true model $\mathcal{M}_0$ (discrete-time Hawkes-type model), we set the truncation parameter $L = 30$ to ensure adequate capture of the lag distribution tail. The lag distribution followed a discretized log-normal with prior specifications $\mu \sim N(0, 1)$ and $\sigma^2 \sim \text{IG}(1, 1)$. For HVA inference, we used a variational rank of $k = 5$ to match the number of static parameters.

Model $\mathcal{M}_1$ extended the discrete-time Hawkes-type model by incorporating time-varying system variance through a discount factor $\hat{\delta} = 0.9$, maintaining the same lag structure but with reduced variational rank $k = 4$. The distributed lag model $\mathcal{M}_2$ used a negative binomial lag distribution with order $\hat{r} = 4$ and $\kappa \sim \text{Beta}(1, 1)$ prior. The autoregressive models $\mathcal{M}_3$ and $\mathcal{M}_4$ were implemented with orders $\hat{p} = 3$ and $\hat{p} = 5$ respectively, both using variational rank $k = 3$.

Table A.2 summarizes the common inference settings applied consistently across all models. We specified inverse-gamma IG(1,1) priors for the dispersion parameter ρ and system variance W, while regression coefficients φ_i received truncated normal priors $N(0, 10)\delta(\varphi_i > 0)$ to ensure positivity. For HVA, we generated 5,000 posterior samples after convergence (1,000 samples for simulation studies). The two-filter smoother used 500 particles with a resampling ratio of 0.5, balancing computational efficiency with approximation accuracy. Gradient ascent optimization began with an initial step size of 10^{-5} and used ADADELTA with a learning rate of 0.02.

For MCMC comparison on model $\mathcal{M}_0$, we ran 12,000 total iterations with 2,000 burn-in samples. After thinning by a factor of 2, we retained 5,000 posterior samples

for analysis. The HMC component used 50 leapfrog integration steps with step size 0.005, parameters chosen through preliminary runs to achieve acceptance rates between 60-80%.

A.1.4. Convergence diagnostics

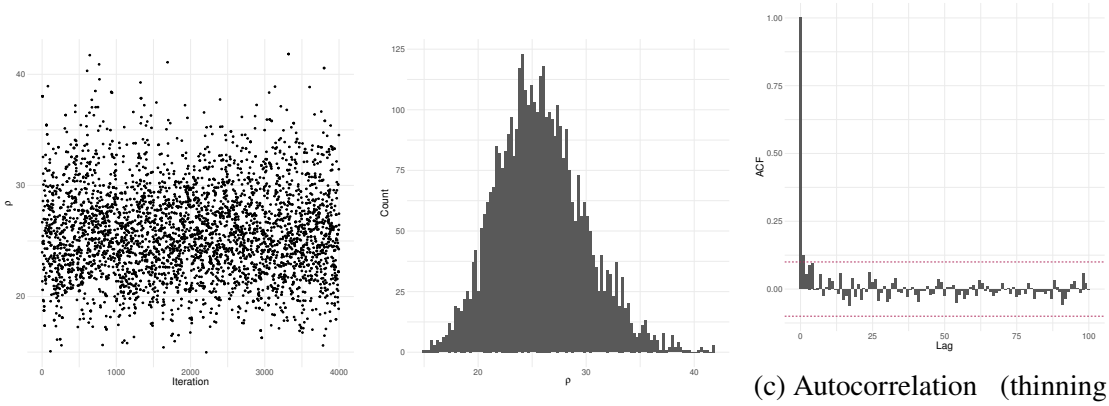
Convergence of MCMC We assessed MCMC convergence using multiple diagnostics. Table A.3 shows that Gelman-Rubin statistics approach 1.0 and the predictive performance for y_t and R_t plateau by iteration 2,000, indicating convergence across all chains.

Figures A.1 to A.5 present trace plots, posterior densities, and autocorrelation functions for all static parameters. The results are based on iterations 2,001–5,000, after discarding the first 2,000 iterations as burn-in. The trace plots indicate that the MCMC chains mix well and converge rapidly to the posterior distribution. The posterior densities align closely with the true parameter values used in data generation. The autocorrelation functions decay rapidly below 0.1 (marked by the red dashed line), confirming efficient mixing and low serial dependence.

Convergence of HVA For HVA, we monitored the conditional marginal likelihood $p(y_{1:T} \mid \boldsymbol{\gamma})$ throughout the first 1,000 iterations to assess convergence. As illustrated in Fig. A.6, the HVA algorithm achieves convergence by 1,000 iterations, with the conditional marginal log-likelihood stabilizing at this point. Also, Table A.4 confirms that both the χ^2 discrepancy of y_t and the MAE of R_t show minimal changes after the first 1,000 iterations, corresponding to the stabilization of the conditional marginal loglikelihood.

Table A.3.: Convergence diagnostics and accuracy metrics for different numbers of iterations in MCMC. The metrics include the Gelman-Rubin statistics for convergence assessment, the χ^2 discrepancy for the posterior predictive distribution of y_t, and the MAE for the posterior distribution of R_t. Results show that MCMC requires approximately 2,000 iterations for full convergence.

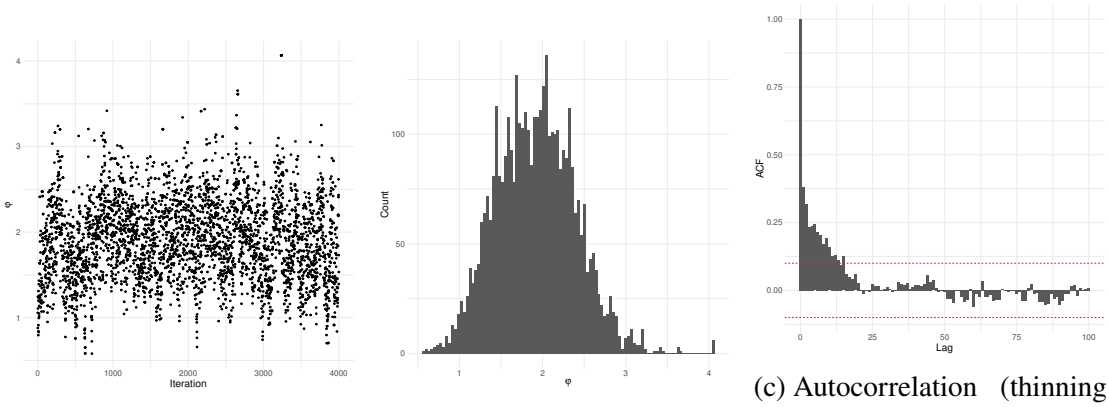
	Gelman-Rubin Statistics						
Iterations	W	ρ	φ	μ	σ^2	χ^2 (y_t)	MAE (R_t)
100	1.69	1.61	3.87	1.51	3.09	1.9	0.3
200	1.22	1.59	2.14	1.58	1.83	1.9	0.23
500	1.12	1.15	1.29	1.2	1.47	1.31	0.14
1,000	1.02	1.09	1.22	1.26	1.31	1.23	0.14
2,000	1.05	1.04	1.19	1.12	1.14	1.18	0.12
5,000	1.06	1.03	1.02	1.1	1.16	1.19	0.12
10,000	1.04	1.02	1.06	1.05	1.12	1.18	0.12



(a) Trace plot (b) Posterior density (c) Autocorrelation (thinning = 3)

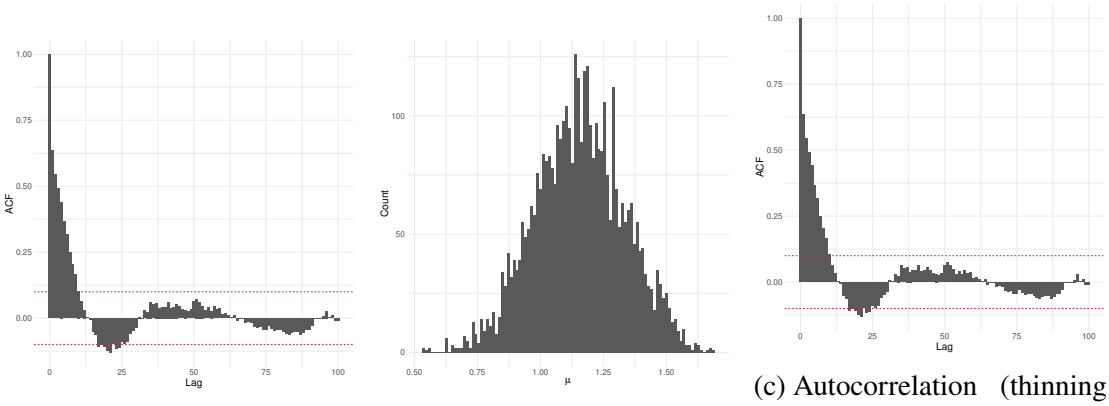
Figure A.1.: Convergence diagnostics for the dispersion parameter ρ showing iterations from 2,001 to 5,000 using MCMC.

However, HVA shows consistently lower 95% CI coverage for R_t compared to MCMC, indicating that the variational approximation may underestimate posterior uncertainty. This is a known limitation of variational methods, which often trade off some accuracy in uncertainty quantification for computational speed. Overall, these results suggest



(a) Trace plot (b) Posterior density (c) Autocorrelation (thinning = 3)

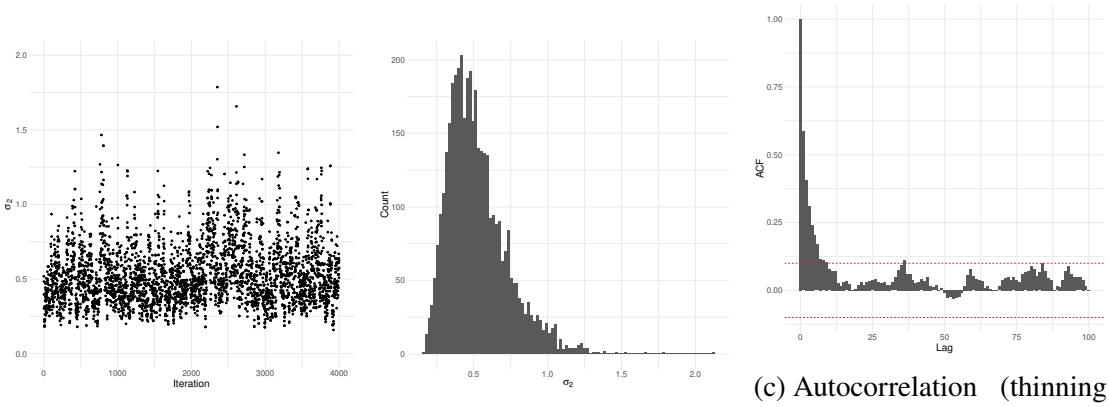
Figure A.2.: Convergence diagnostics for the baseline intensity φ showing iterations from 2,001 to 5,000 using MCMC.



(a) Trace plot (b) Posterior density (c) Autocorrelation (thinning = 3)

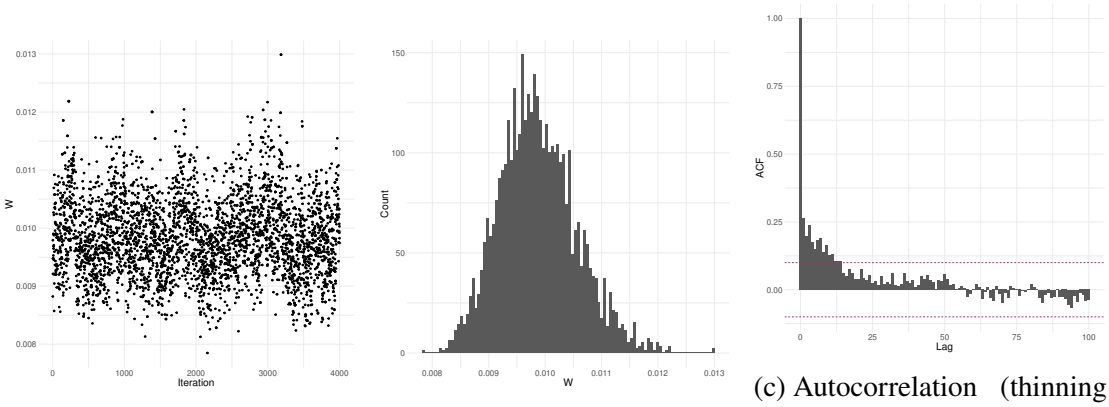
Figure A.3.: Convergence diagnostics for the log-normal mean parameter μ showing iterations from 2,001 to 5,000 using MCMC.

that HVA provides a fast and reasonably accurate alternative to MCMC for inference in the discrete-time Hawkes-type model, though care should be taken when interpreting uncertainty estimates.



(a) Trace plot (b) Posterior density (c) Autocorrelation (thinning = 3)

Figure A.4.: Convergence diagnostics for log-normal variance parameter σ^2 showing iterations from 2,001 to 5,000 using MCMC.



(a) Trace plot (b) Posterior density (c) Autocorrelation (thinning = 3)

Figure A.5.: Convergence diagnostics for the system variance W showing iterations from 2,001 to 5,000 using MCMC.

A.1.5. Particle sensitivity analysis for HVA

To determine the appropriate number of particles N for our HVA implementation, we conducted a sensitivity analysis using the same simulated data from the discrete-time Hawkes-type model $\mathcal{M}_0$. We tested particle counts ranging from 50 to 1,000 particles. The HVA algorithm ran for 1,000 iterations for each configuration, with

Table A.4.: Convergence diagnostics and accuracy metrics for different numbers of iterations in HVA. The metrics include the conditional marginal log likelihood for convergence assessment, the χ^2 discrepancy and 95% CI coverage for the posterior predictive distribution of y_t, and the MAE and 95% CI coverage for the posterior distribution of R_t. Results demonstrate that HVA achieves convergence by 1,000 iterations with accuracy comparable to MCMC at 2,000 iterations.

		y_t Metrics		R_t Metrics	
Iterations	Marginal Loglike	χ^2	Coverage	MAE	Coverage
100	-105.42	3.76	1	0.95	1
200	-62.22	1.4	0.97	0.15	0.87
500	-60.69	1.22	0.96	0.14	0.9
1,000	-59.24	1.18	0.97	0.12	0.85

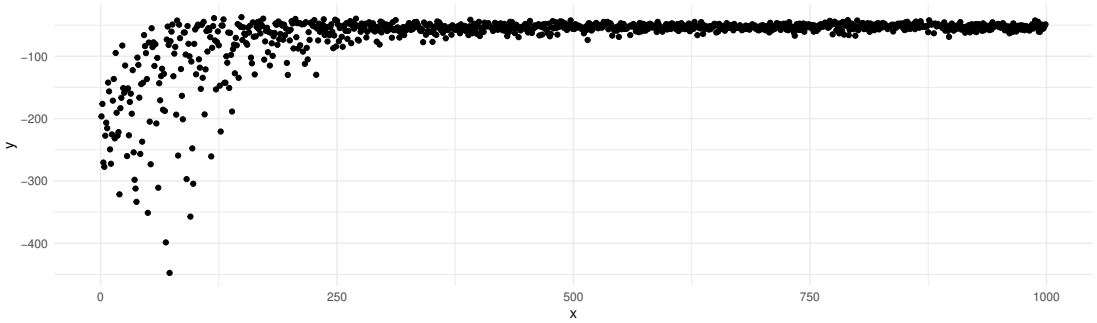


Figure A.6.: Evolution of the conditional marginal likelihood $p(y_{1:T} \mid \boldsymbol{\gamma})$ during the first 1,000 iterations of HVA, using simulated data from the discrete-time Hawkes-type model $\mathcal{M}_0$. The plot shows stabilization of the objective function by iteration 1,000.

results summarized in Table A.5. All metrics were averaged across the 200 time points and five independent runs with different initial values.

The results reveal clear trade-offs between computational cost and statistical accuracy. The χ^2 discrepancy decreases substantially as particle count increases, dropping from 1.85 with 50 particles to 1.19 with 500 particles, with only marginal improvement to 1.18 when using 1,000 particles. Similarly, the CRPS improves from 32.52 to 27.16

at 500 particles, with minimal further gains beyond this point. These patterns suggest diminishing returns for particle counts above 500.

Coverage for y_t remains stable around 95-96% across all particle counts, indicating robust uncertainty quantification for observed counts. In contrast, coverage for R_t begins at 97% with 50 particles but stabilizes at around 85% as the number of particles increases, which is a characteristic behavior of variational approximations.

Computational time increases approximately linearly with particle count, from 14.32 seconds for 50 particles to 208.64 seconds for 1,000 particles. This linear scaling makes the computational burden predictable but substantial for large particle counts.

Based on these findings, we selected $N = 500$ particles as our default setting. This choice represents a pragmatic balance: the χ^2 discrepancy of 1.19 and CRPS of 27.16 fall within 1% of the best values achieved with 1,000 particles, while computational time remains manageable at approximately 100 seconds. The 96% coverage for y_t indicates well-calibrated predictive intervals, though practitioners should note the 87% coverage for R_t when interpreting uncertainty bounds for latent states. Doubling the particle count to 1,000 would double the computational time while providing minimal accuracy gains.

A.1.6. Prior sensitivity analysis for HVA and MCMC

To evaluate the robustness of our inference to different prior specifications, we performed an extensive sensitivity analysis. In the manuscript we fit the discrete-time Hawkes-type model, which is the true data generating process, to the simulated data and inferred the latent states along with the unknown static parameters, $\boldsymbol{\gamma} = (\rho, \varphi, \mu, \sigma^2, W)$. The default prior distributions are: $\rho, W, \sigma^2 \sim \text{IG}(1, 1)$, $\mu \sim N(0, 1)$, and $\varphi \sim N(0, 10)$ truncated

Table A.5.: Performance metrics for different particle counts in the HVA algorithm applied to simulated data from the discretized Hawkes model $\mathcal{M}_0$. Metrics are averaged across 200 time points and five independent runs. Runtime measured in seconds on an Apple M1 Pro processor (10 cores, 8 threads). The table shows diminishing returns in accuracy beyond 500 particles, while computational cost continues to increase linearly.

			y_t Metrics		R_t Metrics	
Particles	Runtime (s)	χ^2	CRPS	Coverage	MAE	Coverage
50	14.32 ± 0.82	1.85	32.52	0.96	0.23	0.97
100	25.43 ± 2.52	1.39	30.96	0.95	0.16	0.93
200	40.95 ± 0.48	1.25	29	0.96	0.13	0.86
300	61.32 ± 0.65	1.26	28.08	0.95	0.13	0.88
400	79.86 ± 0.84	1.19	29.1	0.95	0.12	0.86
500	99.95 ± 1.05	1.19	27.16	0.96	0.12	0.87
600	123.82 ± 9.56	1.19	27.16	0.96	0.12	0.82
1000	208.64 ± 18.47	1.18	27.12	0.95	0.12	0.84

to $(0, \infty)$.

We considered a range of alternative prior specifications for each static parameter. For the dispersion ρ of the negative-binomial likelihood, variance σ^2 of the log-normal lag distribution, and system variance W, we consider three weakly informative inverse-gamma priors: IG(0.01, 0.01), IG(0.1, 0.1), and IG(1, 1). For μ, the mean of the log-normal lag distribution, we consider three normal priors: N(0, 1), N(0, 10), N(0, 100). We also test these three normal priors on the intercept φ, truncated to $(0, +\infty)$. For parameter $\kappa \in (0, 1)$ in the distributed lag model, we test four Beta priors: Beta(1, 1), Beta(2, 2), Beta(2, 5), and Beta(5, 2).

A concern raised in the literature is that inverse-gamma priors do not place density at zero and may artificially inflate variance estimates when the true variance is small (Gelman, 2006). To address this, we additionally evaluated MCMC inference for W

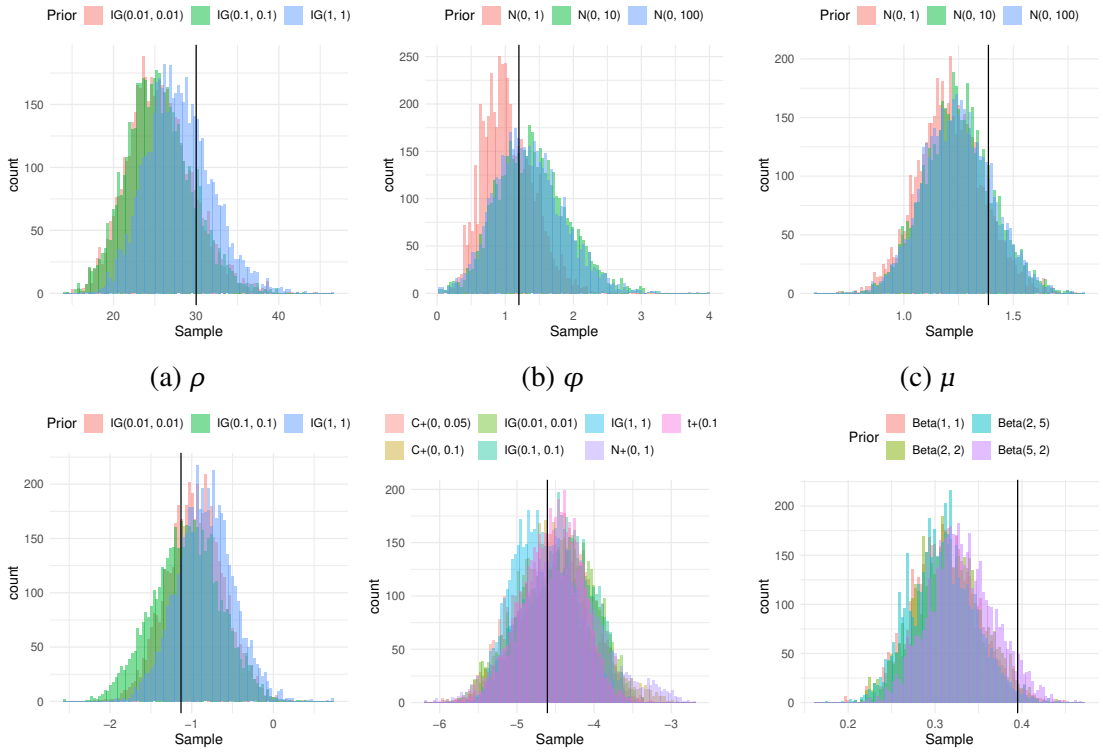
under four priors that place positive density at zero: a half-normal prior with scale $s = 1$ on the standard deviation $\sqrt{W}$, a half-t prior with scale $s = 0.1$ and $\nu = 3$ degrees of freedom, and two half-Cauchy priors with scales $s = 0.1$ and $s = 0.05$. These priors span a range from diffuse (half-normal with $s = 1$) to concentrated near the true value (half-Cauchy with $s = 0.05$), and all allow substantial prior mass near $W = 0$.

MCMC results The posterior distributions produced by MCMC are presented in Fig. A.7. Across all inverse-gamma specifications, the posteriors remain consistent and concentrate around the true parameter values, though the constant baseline φ is biased toward smaller values when using the relatively informative prior N(0,1).

For the system variance W, Fig. A.7e shows that the posteriors under the half-normal, half-t, and half-Cauchy priors are closely aligned with those obtained under the three inverse-gamma priors. This agreement across two fundamentally different prior families confirms that the posterior for W is driven by the likelihood rather than by the prior's behavior near zero, and that the IG(1,1) specification does not artificially inflate the variance estimate.

HVA results The posterior distributions inferred by HVA using different inverse-gamma, normal, and Beta priors are presented in Fig. A.8, where different priors do not significantly affect the posteriors and they all concentrate around the true values.

HVA inference is conducted with inverse-gamma priors for all variance parameters. This choice provides a natural stabilizing barrier in the stochastic optimization: the gradient of the log prior with respect to $\log W$ includes a β/W term that diverges as $W \to 0$, preventing the variational mean from collapsing during early iterations when the particle-based gradient estimates are noisy. Priors that place density at zero, such as

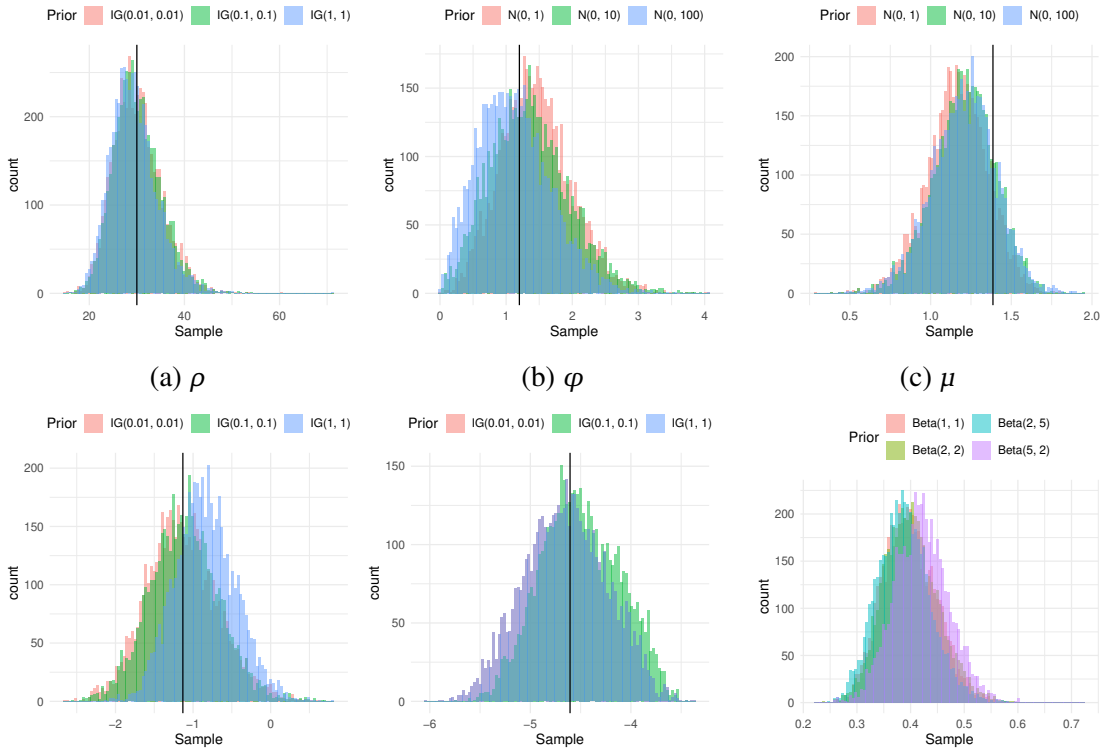


(a) ρ (b) φ (c) μ

(d) $\log \sigma^2$ (e) $\log W$ (f) κ

Figure A.7.: MCMC posteriors of the static parameters $\boldsymbol{\gamma} = (\rho, \varphi, \mu, \sigma^2, W)$ in the discrete-time Hawkes-type model, with data simulated from the same model. The default prior settings are: $\rho, W, \sigma^2 \sim \text{IG}(1, 1)$, $\mu \sim N(0, 1)$, $\varphi \sim N(0, 10)$ truncated to $(0, \infty)$, and $\kappa \sim \text{Beta}(1, 1)$. Each panel shows the effect of altering the prior distribution for a single parameter while keeping the priors for all other parameters fixed at their default values. For W (panel e), the comparison includes priors that place positive density at zero: half-normal ($\text{N}^+(s = 1)$), half-t ($t^+(s = 0.1, \nu = 3)$), and half-Cauchy ($\text{C}^+(s = 0.1)$ and $\text{C}^+(s = 0.05)$).

the half-normal and half-t, lack this barrier and can destabilize the optimization. Since the MCMC results confirm that the inverse-gamma prior does not distort the posterior relative to priors that place density at zero, retaining it for HVA is a computational choice that does not compromise statistical validity.



(a) ρ (b) φ (c) μ

(d) $\log \sigma^2$ (e) $\log W$ (f) κ

Figure A.8.: HVA posterior distributions of the static parameters $\boldsymbol{\gamma} = (\rho, \varphi, \mu, \sigma^2, W)$ in the discrete-time Hawkes-type model, with data simulated from the same model. The default prior settings are: $\rho, W, \sigma^2 \sim \text{IG}(1, 1)$, $\mu \sim N(0, 1)$, $\varphi \sim N(0, 10)$ truncated to $(0, \infty)$, and $\kappa \sim \text{Beta}(1, 1)$. Each panel shows the effect of altering the prior distribution for a single parameter while keeping the priors for all other parameters fixed at their default values.

A.1.7. Model selection

After examining the prior sensitivity, we compared different model specifications to determine the most appropriate one for our data. In addition to the true model $\mathcal{M}_0$, we considered four variants of DGTF: a generalized discrete-time Hawkes-type model with time-varying system variance, denoted by $\mathcal{M}_1$; a distributed lag model $\mathcal{M}_2$; a negative-binomial autoregression $\mathcal{M}_3$ and a Poisson autoregression $\mathcal{M}_4$.

While the main manuscript focuses on the comparison of these four models, this section presents the numerical details in the selection of key hyperparameters for each of these models. Specifically, we needed to determine the appropriate discount factor for the generalized discrete-time Hawkes-type model ($\mathcal{M}_1$), the optimal order of distributed lag for $\mathcal{M}_2$, and the autoregressive orders for both the negative-binomial ($\mathcal{M}3$) and Poisson ($\mathcal{M}4$) autoregressions. Using HVA for inference, we conducted a grid search and chose values that minimized both the χ^2 discrepancy for y_t and MAE for R_t. The results are presented in Table A.6.

Based on the grid search results in Table A.6, we selected optimal parameters for each model. For the generalized discrete-time Hawkes-type model ($\mathcal{M}_1$), a discount factor of $\delta = 0.9$ minimized both error metrics. For the distributed lag model ($\mathcal{M}_2$), we chose $r = 4$ as it achieved the lowest χ^2 discrepancy (1.218). The optimal autoregressive order is determined as $\hat{p} = 3$ for the negative-binomial autoregression $\mathcal{M}_3$ and $\hat{p} = 5$ for the Poisson autoregression $\mathcal{M}_4$.

Table A.6.: Selection of the discount factor δ of $\mathbf{W}_t$ in $\mathcal{M}_1$, the order of distributed lags r in $\mathcal{M}_2$, and the autoregressive orders p in $\mathcal{M}_3$ and $\mathcal{M}_4$, based on the χ^2 discrepancy of y_t and MAE of R_t. The data is simulated from a discretized Hawkes model with constant system variance.

	δ	0.87	0.88	0.89	**0.90**	0.91	0.92	0.93	0.94	0.95
y_t	χ^2	1.237	1.235	1.226	**1.222**	1.227	1.231	1.239	1.258	1.262
	CRPS	28.76	28.42	28.2	**27.92**	29.08	29.32	29.7	30.04	30.44
R_t	MAE	0.139	0.138	0.136	**0.134**	**0.134**	0.138	0.141	0.148	0.157

(a) $\mathcal{M}_1$: Discount factor δ accounts for the effect of an unknown, time-varying full-rank variance-covariance matrix $\mathbf{W}_t$, the full-rank discretized Hawkes model where the latent state is driven $\boldsymbol{\theta}_t \sim N(\mathbf{g}_t(\boldsymbol{\theta}_{t-1}), \mathbf{W}_t)$.

	r	2	3	**4**	5	6	7
y_t	χ^2	1.285	1.228	**1.218**	1.224	1.265	1.312
	CRPS	27.78	27.57	**27.18**	27.31	27.27	27.33
R_t	MAE	0.138	0.132	**0.130**	**0.130**	0.136	0.134

(b) $\mathcal{M}_2$: r is order of distributed lags.

	p	1	2	**3**	4	5	6
y_t	χ^2	3.969	1.564	**1.416**	1.563	1.915	2.687
	CRPS	41.83	33	**30.25**	30.45	31.6	41.12

(c) $\mathcal{M}_3$: p is autoregressive order in a negative-binomial autoregression with time-varying autoregressive coefficients.

	p	1	2	3	4	**5**	6
y_t	χ^2	6.491	2.908	2.067	1.781	**1.632**	1.638
	CRPS	45.52	34.08	31.3	32.48	**30.02**	**30.02**

(d) $\mathcal{M}_4$: p is autoregressive order in a Poisson autoregression with time-varying autoregressive coefficients.

§ A.2. Simulation from distributed lag model

The main simulation study evaluated model performance when data was generated from a discrete-time Hawkes-type model. To provide a complementary perspective and assess model robustness, we conducted a second simulation where data originates from a distributed lag model. This reciprocal design tests whether the choice of lag distribution family substantially affects inference when models share similar mechanistic structure, and helps identify when structurally similar models can substitute for each other.

A.2.1. Model specification

We simulated a count time series with 200 time points from a distributed lag model with lag order $r = 6$ and decay parameter $\kappa = 0.395$. The data generating process follows the DGTF specification with negative-binomial observations $y_t \sim \text{NB}(\lambda_t, \rho)$, where $\rho = 30$ captures realistic overdispersion. The intensity λ_t uses an identity link with constant baseline $\varphi = 1$ and covariate $x_t = 1$, while the gain function takes the softplus form $h(\psi_t) = \log(\exp(\psi_t) + 1)$ to ensure positivity. The latent process evolves as a univariate random walk $\psi_t \sim N(\psi_{t-1}, W)$ with $W = 0.01$, allowing gradual changes in transmission intensity.

For model selection, we consider the following candidates:

- $\mathcal{M}_0$: Discrete-time Hawkes-type model with a negative-binomial likelihood and univariate white noise $w_t \sim N(0, W)$. The unknown parameters to be inferred are $\boldsymbol{\gamma} = (\rho, W, \mu, \sigma^2, \varphi)$.
- $\mathcal{M}_1$: Discrete-time Hawkes-type model with latent state following a random walk $\boldsymbol{\theta}_t \sim N(\mathbf{g}_t(\boldsymbol{\theta}_{t-1}), \mathbf{W}_t)$, with $\mathbf{W}_t$ is unknown and handled via a discount factor δ.

The unknown parameters are $\boldsymbol{\gamma} = (\rho, \mu, \sigma^2, \varphi)$ and $\delta \in (0, 1)$.

- $\mathcal{M}_2$: Distributed lag model with unknown parameters $\boldsymbol{\gamma} = (\rho, W, \kappa, \varphi)$ and $r \in \{2, 3, \ldots, 8\}$.

- $\mathcal{M}_3$: Negative-binomial autoregression with time-varying autoregressive coefficients. The unknown parameters are $\boldsymbol{\gamma} = (\rho, W, \varphi)$ and autoregressive order $p \in \{2, 3, \ldots, 8\}$.

- $\mathcal{M}_4$: Poisson autoregression with time-varying autoregressive coefficients. The unknown parameters are $\boldsymbol{\gamma} = (W, \varphi)$ and autoregressive $p \in \{2, 3, \ldots, 8\}$.

A.2.2. Hyperparameter selection

We fitted all five candidate models using HVA with a two-filter smoother containing 500 particles and a resampling ratio of 0.5. For gradient ascent optimization, we set the ADADELTA learning rate to 0.02, and the rank k of the variational proposal distribution as follows: $k = 4$ for $\mathcal{M}_0$ and $\mathcal{M}_2$; $k = 3$ for $\mathcal{M}_1$ and $\mathcal{M}_3$; $k = 2$ for $\mathcal{M}_4$. We ran 1,000 iterations of gradient ascent and verified convergence through trace plots, then drew 5,000 samples from the converged variational proposal distribution.

Hyperparameter tuning through grid search (Table A.7) identified optimal configurations: discount factor $\hat{\delta} = 0.9$ for $\mathcal{M}_1$, lag order $\hat{r} = 6$ for $\mathcal{M}_2$ (matching the true value), and autoregressive orders $\hat{p} = 3$ for $\mathcal{M}_3$ and $\hat{p} = 4$ for $\mathcal{M}_4$. These selections minimized in-sample estimation errors for y_t (measured by χ^2 discrepancy and CRPS) and, where applicable, the MAE of R_t.

Table A.7.: Hyperparameter selection through grid search for models fitted to data simulated from the distributed lag model $\mathcal{M}_2$ ($r = 6$, $\kappa = 0.395$). Optimal values minimize in-sample estimation errors measured by χ^2 discrepancy and CRPS for y_t and the MAE of R_t if applicable. All models use HVA inference with 500 particles and 1,000 gradient ascent iterations.

	δ	0.88	**0.90**	0.92	0.94	0.96	0.98
y_t	χ^2	1.318	**1.307**	1.307	1.317	1.324	1.348
	CRPS	2.437	**2.309**	2.373	2.395	2.478	2.529
R_t	MAE	0.155	**0.142**	0.156	0.188	0.229	0.281

(a) $\mathcal{M}_1$: Optimal discount factor δ for the generalized discretized Hawkes model with time-varying system variance $\mathbf{W}_t$. The selected value $\hat{\delta} = 0.9$ minimizes predictive accuracy (χ^2, CRPS) and reproduction number estimation (MAE of R_t).

	r	2	3	4	5	**6**	7
y_t	χ^2	1.27	1.269	1.255	1.258	**1.238**	1.242
	CRPS	2.337	2.326	2.317	2.312	**2.296**	2.299
R_t	MAE	0.146	0.141	0.141	0.144	**0.144**	0.146

(b) $\mathcal{M}_2$: Optimal lag order r for distributed lag model. The grid search correctly identifies $\hat{r} = 6$ as optimal, matching the true data-generating value.

	p	2	3	4	5	6	7
$\mathcal{M}_3$	χ^2	1.43	**1.35**	1.444	2.254	3.993	6.572
	CRPS	2.728	**2.705**	3.056	4.967	8.640	13.207
$\mathcal{M}_4$	χ^2	2.19	2.013	**1.982**	2.185	2.693	3.622
	CRPS	2.851	2.818	**2.885**	3.203	4.004	5.212

(c) Optimal autoregressive orders p for negative-binomial ($\mathcal{M}_3$) and Poisson ($\mathcal{M}_4$) autoregressions. The negative-binomial AR achieves best performance at $\hat{p} = 3$, while Poisson AR performs best at $\hat{p} = 4$.

A.2.3. Model comparison

Table A.8.: Performance comparison of five DGTF models fitted to data generated from distributed lag model $\mathcal{M}_2$. Models are evaluated on predictive accuracy (χ^2 discrepancy and CRPS), uncertainty calibration (95% CI coverage and width for observed counts yt), and reproduction number estimation accuracy (MAE, 95% CI coverage, and width for R_t). Models $\mathcal{M}_3$ and $\mathcal{M}_4$ do not estimate R_t directly.

		$\mathcal{M}_0$	$\mathcal{M}_1$ ($\hat{\delta} = 0.9$)	$\mathcal{M}_2$ ($\hat{r} = 6$)	$\mathcal{M}_3$ ($\hat{p} = 3$)	$\mathcal{M}_4$ ($\hat{p} = 4$)
y_t	χ^2	1.276	1.307	**1.238**	1.35	1.982
	CRPS	2.32	2.309	**2.296**	2.705	2.885
	Coverage	0.99	0.99	0.99	1	0.9
	Width	18.25	18.36	18.14	22.77	13.45
R_t	MAE	0.145	**0.142**	0.144		
	Coverage	0.95	0.96	0.96		
	Width	0.48	0.53	0.49		

Table A.8 compares model performance across multiple metrics. As expected, the true model $\mathcal{M}_2$ performs best with $\chi^2 = 1.238$ and CRPS = 2.296. The discrete-time Hawkes-type models provide close approximations: $\mathcal{M}_0$ achieves $\chi^2 = 1.276$ and $\mathcal{M}_1$ achieves $\chi^2 = 1.307$, with CRPS values of 2.32 and 2.309 respectively. This pattern resembles the main simulation where $\mathcal{M}_2$ successfully approximated data generated from the discrete-time Hawkes-type model ($\chi^2 = 1.22$ versus the true model's 1.19), though each model performs slightly better when correctly specified.

However, autoregressive models show substantial performance degradation: $\mathcal{M}_3$ reaches $\chi^2 = 1.35$ while $\mathcal{M}_4$ deteriorates to $\chi^2 = 1.982$. This mirrors the main simulation where autoregressive models also underperformed, consistent with autoregressive structures capturing the self-exciting dynamics less well in these simulations.

Beyond predictive accuracy, uncertainty quantification reveals consistent differences

between model classes. The first three models achieve 99% coverage for y_t with nearly identical interval widths (18.14 to 18.36), indicating well-calibrated but slightly conservative uncertainty estimates. In contrast, $\mathcal{M}_3$ produces excessively wide intervals (22.77) due to its underestimated ρ (Fig. A.12b), while $\mathcal{M}_4$ shows particularly poor coverage at 90% with overly narrow intervals (13.45) because it cannot account for overdispersion. These miscalibration patterns (Fig. A.10) persist across both simulations, indicating model-specific rather than data-specific limitations.

For reproduction number estimation, $\mathcal{M}_0$, $\mathcal{M}_1$, and $\mathcal{M}_2$ again perform comparably with MAE values between 0.142 and 0.145, and appropriate coverage (95% to 96%) as shown in Fig. A.9. The generalized Hawkes model $\mathcal{M}_1$ achieves marginally the best performance (MAE = 0.142), though differences of this magnitude likely reflect Monte Carlo variation rather than true superiority. Autoregressive models cannot produce mechanistically interpretable R_t estimates because their time-varying coefficients conflate transmission intensity with temporal structure.

Parameter estimation reveals both similarities and differences across model classes. The implied lag distributions (Fig. A.12a), computed from posterior medians of lag parameters (Figs. A.11a to A.11c), show that despite different functional forms, all three models capture similar lag structure peaking at $l = 3$. The negative-binomial distribution has a shorter tail and larger mass at $l = 1$ compared to the log-normal, yet both successfully approximate the true lag pattern. Meanwhile, dispersion and intercept estimates (Fig. A.12) show that the first three models produce similar φ posteriors closely matching the true value, whereas both autoregression models underestimate φ. All models estimate smaller ρ than the true value, with $\mathcal{M}_3$ showing the most severe underestimation.

These results complement the main simulation by demonstrating that misspecification between Hawkes and distributed lag models produces smaller performance degradation than misspecification involving autoregressive models. The similar performance of $\mathcal{M}_0$, $\mathcal{M}_1$, and $\mathcal{M}_2$ in both simulations, contrasted with persistent miscalibration of $\mathcal{M}_3$ and $\mathcal{M}_4$, highlights the importance of capturing self-exciting dynamics for reliable inference. When the true data generating process involves transmission patterns with lagged effects, practitioners can expect reasonable approximation quality even when the specific lag parameterization differs from the true data generating mechanism.

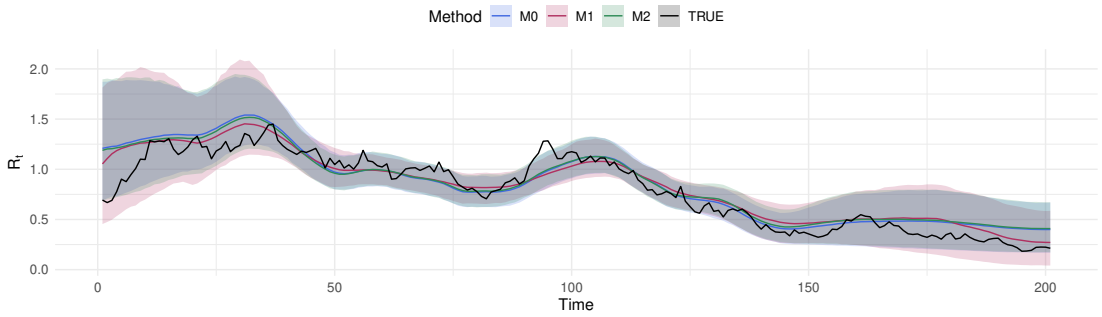
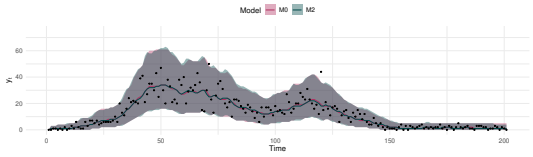
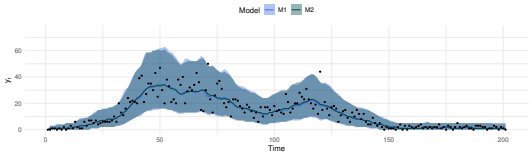


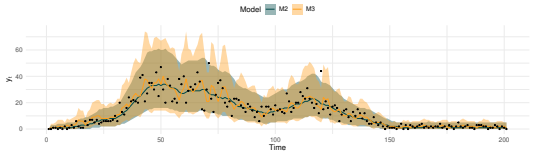
Figure A.9.: Posterior distributions of the reproduction number R_t estimated by three DGTF models for data simulated from the distributed lag model $\mathcal{M}_2$. Solid lines show posterior medians, shaded regions indicate 95% CIs, and black lines mark true R_t values. The discrete-time Hawkes-type models ($\mathcal{M}_0$, $\mathcal{M}_1$) and true distributed lag model ($\mathcal{M}_2$) produce similar estimates despite different lag parameterizations.



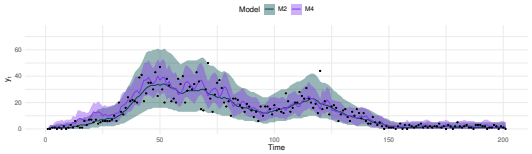
(a) Distributed lag model $\mathcal{M}_2$ ($\hat{r} = 6$) compared to discrete-time Hawkes-type model $\mathcal{M}_0$. Despite different lag parameterizations (negative-binomial vs. log-normal), both models produce similar predictive distributions.



(b) Distributed lag model $\mathcal{M}_2$ ($\hat{r} = 6$) compared to generalized discrete-time Hawkes-type model $\mathcal{M}_1$ with discount factor $\hat{\delta} = 0.9$. The time-varying variance assumption in $\mathcal{M}_1$ provides additional flexibility without substantially changing predictive distributions.

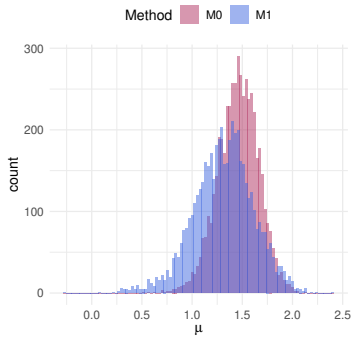


(c) Distributed lag model $\mathcal{M}_2$ ($\hat{r} = 6$) compared to negative-binomial autoregression $\mathcal{M}_3$ ($\hat{p} = 3$). The AR model shows inflated uncertainty (wider prediction intervals) but maintains reasonable coverage.

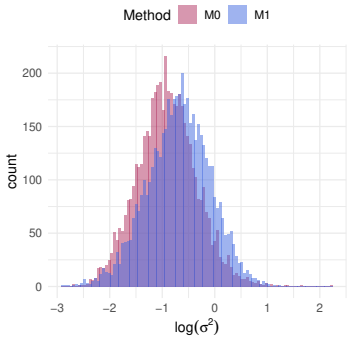


(d) Distributed lag model $\mathcal{M}_2$ ($\hat{r} = 6$) compared to Poisson autoregression $\mathcal{M}_4$ ($\hat{p} = 4$). The Poisson specification cannot capture overdispersion, resulting in overly narrow prediction intervals and poor coverage.

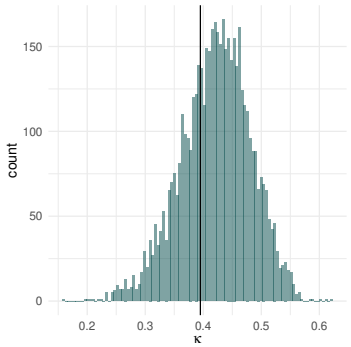
Figure A.10.: Posterior predictive distributions $p(\mathbf{y}_t^{\text{rep}} \mid \mathbf{y}_t)$ for observed counts from data generated by the distributed lag model $\mathcal{M}_2$ (true model), compared against estimates from four competing models. Black dots show observed values, solid lines represent posterior medians, and shaded ribbons show 95% prediction intervals.



(a) $\hat{\mu}$: 1.46 (CI: 1.05-1.83) for $\mathcal{M}_0$ and 1.32 (CI: 0.68-1.88) for $\mathcal{M}_1$.

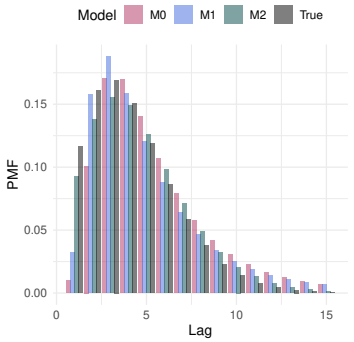


(b) $\hat{\sigma}^2$: 0.4 (CI: 0.14-1.29) for $\mathcal{M}_0$ and 0.51 (CI: 0.15-1.54) for $\mathcal{M}_1$.

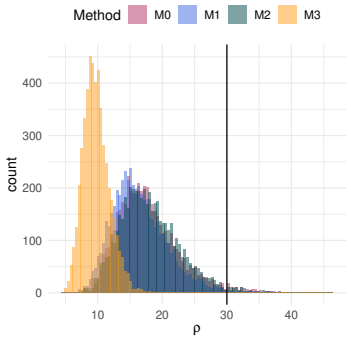


(c) $\hat{\kappa}$: 0.42 (CI: 0.3-0.53) for $\mathcal{M}_2$.

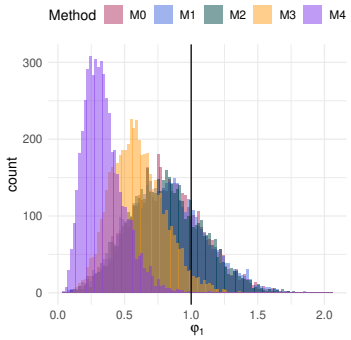
Figure A.11.: Posterior distributions of lag distribution parameters μ, σ^2, and κ estimated by competing models fitted to data generated from the distributed lag model $\mathcal{M}_2$. Black vertical line marks the true value $\kappa = 0.395$. Point estimates use posterior medians with 95% CIs shown.



(a) Implied lag distributions.



(b) Posteriors of dispersion ρ.



(c) Posteriors of baseline intensity φ.

Figure A.12.: Comparison of lag distributions and parameter estimates across models fitted to data generated from distributed lag model $\mathcal{M}_2$. (a) Implied lag distributions using posterior median parameter estimates: log-normal distributions from discrete-time Hawkes-type models $\mathcal{M}_0$ and $\mathcal{M}_1$, and negative-binomial distribution from the true distributed lag model $\mathcal{M}_2$; (b-c) Posterior distributions of the dispersion ρ and the baseline intensity φ.

§ A.3. Application to COVID-19 case counts in Santa Cruz County

After validating our methods through simulation studies, we turned to real data analysis. Specifically, we examined COVID-19 case counts from Santa Cruz County, California spanning July 2020 to November 2021.

A.3.1. Implementation

Table A.9.: Model-specific settings for the three DGTF variants applied to COVID-19 data from Santa Cruz County. Hyperparameters were selected through grid search to minimize χ^2 discrepancy. The variational rank k reflects the number of static parameters including seasonal components.

	Discretized Hawkes	DL(2)	NB-AR(3)
Hyperparameter	$L = 30$	$\hat{r} = 2$	$\hat{p} = 3$
Lag Prior	$\mu \sim N(0, 1)$ $\sigma^2 \sim \text{IG}(1, 1)$	$\kappa \sim \text{Beta}(1, 1)$	
Variational rank	$k = 10$	$k = 9$	$k = 8$

Having validated our methods through simulation, we now detail the implementation settings for analyzing COVID-19 case counts from Santa Cruz County. Table A.9 presents the model-specific configurations for the three DGTF variants compared in this application. The discrete-time Hawkes-type model maintained the same structure as in our simulation study, with truncation parameter $L = 30$ and log-normal lag distribution priors $\mu \sim N(0, 1)$ and $\sigma^2 \sim \text{IG}(1, 1)$. However, the increased complexity from incorporating weekly seasonality required a higher variational rank of $k = 10$.

The distributed lag model used the optimal order $\hat{r} = 2$ determined through grid

Table A.10.: Common inference settings maintained across all models in real data analyses. The table shows prior specifications for shared parameters and algorithmic settings.

Prior specifications	ρ and W	φ_i
	IG(1, 1)	$N(0, 10)\delta(\varphi_i > 0)$
HVA	Iterations	Posterior samples
	5,000	5,000
Two-filter smoother	Particles	Resampling ratio
	500	0.5
Gradient ascent	Initial step size	ADADELTA learning rate
	10^{-5}	0.02
MCMC	Burnin-in	Posterior samples
	5,000	5,000 (thinning = 2)
HMC - Leapfrog	Integration steps	Step size
	50	0.005

search minimizing the χ^2 discrepancy, with the negative binomial proportion parameter receiving a $\kappa \sim \text{Beta}(1, 1)$ prior. Similarly, the negative-binomial autoregression model used the optimal order $\hat{p} = 3$. These reduced-order models required variational ranks of $k = 9$ and $k = 8$ respectively, reflecting their fewer static parameters compared to the discrete-time Hawkes-type specification.

Table A.10 summarizes the inference settings maintained across all models. The prior specifications and algorithmic parameters remained identical to those used in simulation, ensuring consistency in our computational approach. Notably, while simulation studies generated 1,000 posterior samples for efficiency, the real data analysis used 5,000 samples to ensure robust uncertainty quantification for policy-relevant estimates.

For the HVA versus MCMC comparison on the discrete-time Hawkes-type model, we extended the MCMC runtime substantially. The chain consisted of 15,000 total iterations with 5,000 burn-in samples, retaining 5,000 posterior samples after thinning

by a factor of 2. This extended sampling ensured reliable convergence diagnostics given the increased model complexity from seasonal components.

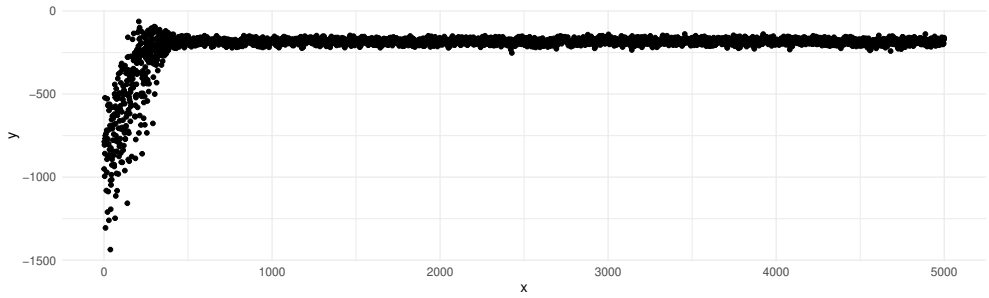
All three DGTF models incorporated weekly seasonality through a 7-dimensional parameter vector $\boldsymbol{\varphi} = (\varphi_1, \ldots, \varphi_7)'$ with cycling design matrix $\mathbf{X}$, as detailed in the main text. This direct incorporation of reporting patterns distinguished our approach from the benchmark models, which required pre-processing the data using day-of-week adjustments.

A.3.2. Convergence of HVA algorithms

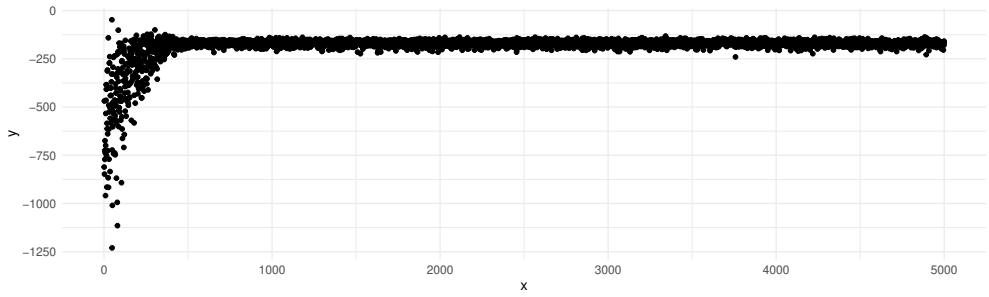
We first examined the convergence properties of our HVA algorithm across different model specifications. As shown in Fig. A.13, the HVA algorithm demonstrated efficient convergence, stabilizing after approximately 500 iterations for all model variants.

A.3.3. Model selection

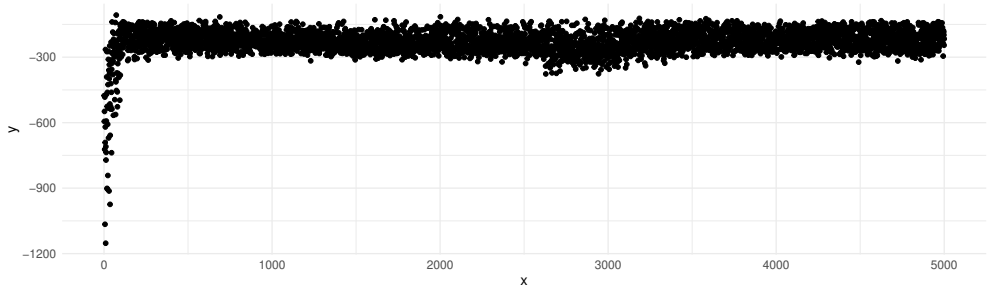
For the distributed lag model and negative-binomial autoregression, we conducted a grid search to identify optimal parameters. As shown in Table A.11, the distributed lag model with order $r = 2$ (DL(2)) achieved the lowest χ^2 discrepancy at 2.639 among all tested configurations ($r \in 2, 3, 4, 5, 6, 7$). For the negative-binomial autoregression models with varying orders ($p \in 1, 2, 3, 4, 5, 6$), the third-order specification (NB-AR(3)) performed best with a χ^2 discrepancy of 3.064. Nevertheless, this value remains higher than that of the optimal DL model, suggesting that the distributed lag model fits these COVID-19 counts better than the autoregression.



(a) NB-AR(3)



(b) Discrete-time Hawkes-type model



(c) DL(2)

Figure A.13.: Evolution of the conditional marginal likelihood $p(y_{1:T} \mid \boldsymbol{\gamma})$ in the first 5,000 iterations of HVA. We fitted the COVID-19 case counts in Santa Cruz County to three different models: (a) a third-order negative-binomial autoregression; (b) a discrete-time Hawkes-type model; (c) a second-order distributed lag model.

A.3.4. Weekly seasonality on COVID count time series

A key feature of COVID-19 case reporting is its weekly pattern, primarily driven by differences in reporting practices between weekdays and weekends. To examine

Table A.11.: Optimal lag order r for the distributed lags model and optimal autoregressive order p for negative-binomial autoregression, measured by χ^2 discrepancy, for the COVID count time series of Santa Cruz county from July 1, 2020 to Nov 29, 2021.

r	2	3	4	5	6	7
DL(r)	**2.639**	2.895	3.016	2.752	2.913	2.791

(a) Optimal lag order r for the distributed lags model.

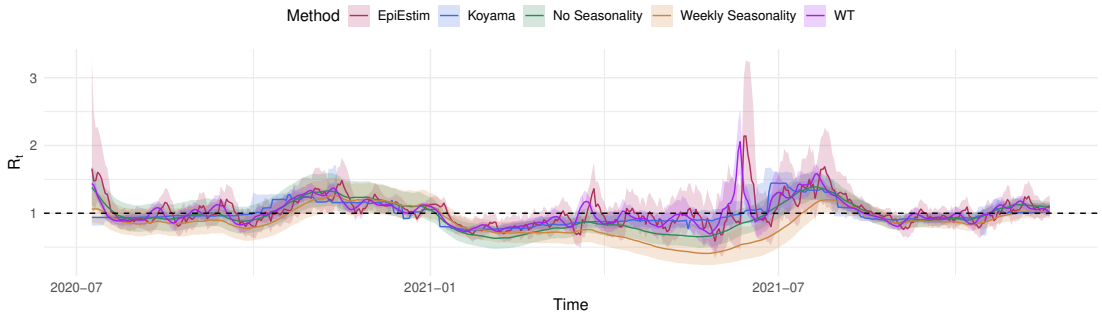
p	1	2	3	4	5	6
NB-AR(p)	4.047	3.267	**3.064**	3.252	3.894	4.737

(b) Optimal autoregressive order p for negative-binomial autoregression.

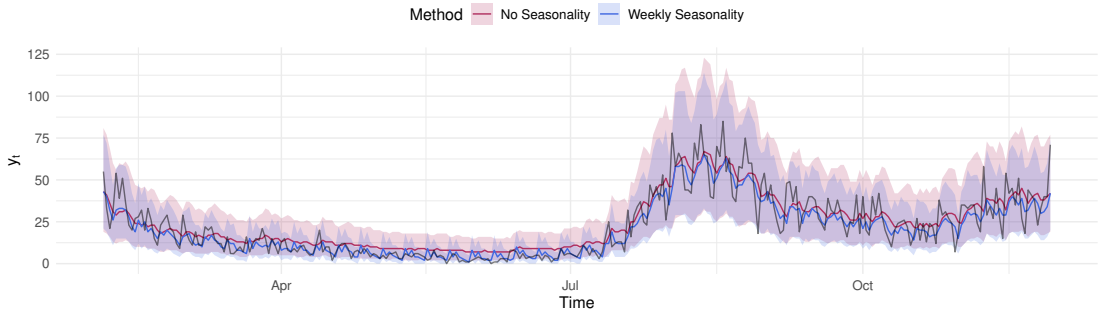
how incorporating weekly seasonality affected our results, we compared discrete-time Hawkes-type models both with and without seasonality terms, alongside three benchmark models applied to the raw count data. The benchmark model configurations remained consistent with those described in the main text.

The discrete-time Hawkes-type model with seasonal terms followed the same configuration as described in the main text. The non-seasonal variant of our model contained unknown static parameters $\boldsymbol{\gamma} = (\rho, W, \mu, \sigma^2)$. We maintained the same prior distributions for these parameters as its seasonal counterpart. For this variant, the variational proposal used $k = 4$. Other than that, the HVA configuration for these two variants remained consistent with our primary analysis. With convergence already achieved for both models after 5,000 iterations, we generated 5,000 posterior samples using the optimized parameters.

The posterior distributions of R_t estimated by the benchmark methods and our model under two different settings are shown in Fig. A.14a. The R_t estimates from our model without seasonality closely align with those from the benchmark methods. However,



(a) R_t from July 1, 2020 to Nov 29, 2021.



(b) $p(y_t^{\text{rep}} \mid y_t)$ from Feb 4, 2021 to Nov 29, 2021.

Figure A.14.: (a) The posterior distributions of R_t, from July 1, 2020 to Nov 29, 2021, for Santa Cruz County are estimated by WT, EpiEstim, and our models both with and without weekly seasonality; (b) The posterior predictive distributions $p(y_t^{\text{rep}} \mid y_t)$ for models with and without seasonality from Feb 4, 2021 to Nov, 2021.

incorporating weekly seasonality components produced clear differences during low-count periods (approximately February 4, 2021, to July 1, 2021). During these intervals, the model with seasonality yielded smaller R_t values compared to both the benchmarks and the non-seasonal model variant.

The inclusion of seasonality components also improved model fit, especially during low-count periods, as evidenced in the posterior predictive distributions of y_t shown in Fig. A.14b. This improvement was further confirmed when assessing posterior predictive loss through χ^2 discrepancy calculations between posterior predictive samples

y_t^{rep} and observed values. The model with seasonality achieved a lower χ^2 discrepancy of 2.338, compared to 2.518 for the model without seasonality, indicating a better overall fit to the data.

B. Supplement to Chapter 3

This appendix collects the posterior-sampling derivations, additional simulation results, and the application sensitivity analysis that support Chapter 3. Notation follows Chapter 3.

§ B.1. Posterior sampling details

The MCMC algorithm iterates over four blocks per sweep. First, the zero-inflation indicators $\{z_{s,t}\}$ are sampled via forward-filtering backward-sampling (FFBS). Second, the zero-inflation regression coefficients (β_0, β_1) are updated via Hamiltonian Monte Carlo (HMC). Third, for each source column $k = 1, \ldots, S$, the spatial network parameters $\{\varepsilon_{\cdot,k}, \gamma_{\cdot,k}, \tau_k, \omega_k\}$ are updated jointly via HMC, and the reproduction number disturbances $\{\eta_{k,t}\}_{t=0}^{T-1}$ are updated via a Metropolis–Hastings (MH) step with a Gaussian proposal. Finally, the baseline intensity μ is updated via HMC. The remainder of this section presents the sampling details for each block.

B.1.1. Zero-inflation indicators and coefficients

The binary indicators $\{z_{s,t}\}$ follow a first-order Markov chain through the autoregressive logistic model for $p_{s,t}$. Conditional on the data and all other parameters, the indicators

for each location s are independent across locations and can be sampled via forward-filtering backward-sampling (FFBS). The forward pass computes $p(z_{s,t} \mid y_{s,1:t}, \beta_0, \beta_1)$ recursively, and the backward pass draws $z_{s,T}, z_{s,T-1}, \ldots, z_{s,1}$ from the conditional distributions $p(z_{s,t} \mid z_{s,t+1}, y_{s,1:t}, \beta_0, \beta_1)$. Since $z_{s,t} \in \{0, 1\}$, each filtering and sampling step involves only two-point distributions and is computationally inexpensive.

Given the sampled indicators, we update β_0 and β_1 via HMC on the unconstrained scale. The log posterior target is

$$\log p(\beta_0, \beta_1 \mid \{z_{s,t}\}) \propto \log N(\beta_0 \mid a_{\beta_0}, b_{\beta_0}^2) + \log N(\beta_1 \mid a_{\beta_1}, b_{\beta_1}^2) + \sum_{s,t} \left[z_{s,t} \log p_{s,t} + (1 - z_{s,t}) \log(1 - p_{s,t}) \right],$$

where $\text{logit}(p_{s,t}) = \beta_0 + \beta_1 z_{s,t-1}$. The required gradients follow directly from the logistic likelihood.

B.1.2. Column-wise HMC for horseshoe parameters

We use a non-centered parameterization for the off-diagonal deviations, writing $\theta_{s,k} = \tau_k \tilde{\gamma}_{s,k} \varepsilon_{s,k}$ with $\varepsilon_{s,k} \sim N(0, 1)$, where $\tilde{\gamma}_{s,k}$ is the regularized local shrinkage scale defined in the main text. HMC operates on the unconstrained vector

$$\boldsymbol{\xi}_k = (\{\varepsilon_{s,k}\}_{s \neq k},\ \text{logit}(w_k),\ \{\log \gamma_{s,k}\}_{s \neq k},\ \log \tau_k)',$$

which has dimension $2(S - 1) + 2 = 2S$. Diagonal entries $\varepsilon_{k,k}$ and $\log \gamma_{k,k}$ are excluded because $\alpha_{k,k} = w_k$ is parameterized directly.

The log-posterior target is

$$\log p(\boldsymbol{\xi}_k \mid \cdot) \propto \sum_{s,t} \log \mathrm{Pois}(y_{s,t} \mid \lambda_{s,t}) + a_w \log w_k + b_w \log(1 - w_k)$$
$$+ \sum_{s \neq k} \left[\log N(\varepsilon_{s,k} \mid 0, 1) + \log p(\gamma_{s,k}) + \log \gamma_{s,k} \right] + \log p(\tau_k) + \log \tau_k,$$

where a_w and b_w are the shape parameters of the $\mathrm{Beta}(a_w, b_w)$ prior on w_k, the terms $\log \gamma_{s,k}$ and $\log \tau_k$ are Jacobian corrections arising from the log transformations, and $p(\gamma_{s,k})$ and $p(\tau_k)$ are the half-Cauchy densities

$$p(\gamma_{s,k}) = \frac{2}{\pi(1 + \gamma_{s,k}^2)}, \qquad p(\tau_k) = \frac{2}{\pi\tau_0(1 + \tau_k^2/\tau_0^2)}.$$

The w_k prior term above absorbs both the Beta density and the Jacobian of the logit transformation. Specifically, the $\mathrm{Beta}(a_w, b_w)$ density on w_k contributes $(a_w - 1) \log w_k + (b_w - 1) \log(1 - w_k)$, and the change of variables from w_k to $\mathrm{logit}(w_k)$ adds $\log w_k + \log(1 - w_k)$, yielding $a_w \log w_k + b_w \log(1 - w_k)$ in total.

We now derive the gradient components. Define $\kappa = 1/(S - 1)$. Differentiating the centered deviation $\tilde{\theta}_{s,k} = \theta_{s,k} - \bar{\theta}_k$ with respect to $\theta_{s,k}$ gives

$$\frac{\partial \tilde{\theta}_{j,k}}{\partial \theta_{s,k}} = -\kappa, \quad s \neq j \neq k,$$
$$\frac{\partial \tilde{\theta}_{s,k}}{\partial \theta_{s,k}} = 1 - \kappa, \quad s = j \neq k.$$

It follows that

$$\frac{\partial \alpha_{s,k}}{\partial \theta_{s,k}} = (1 - w_k) \frac{u_{s,k}(U_k - u_{s,k})}{U_k^2},$$
$$\frac{\partial \alpha_{j,k}}{\partial \theta_{s,k}} = -(1 - w_k) \frac{u_{j,k}\, u_{s,k}}{U_k^2}, \quad j \neq s.$$

These expressions enter the likelihood gradient through

$$\frac{\partial \lambda_{s,t}}{\partial \alpha_{s,k}} = \sum_{l=1}^{t-1} R_{k,l}\, \phi_{t-l}\, y_{k,l}.$$

Gradient with respect to $\varepsilon_{s,k}$ For $s \neq k$, applying the chain rule with $\partial \theta_{s,k}/\partial \varepsilon_{s,k} = \tau_k \tilde{\gamma}_{s,k}$ gives

$$\frac{\partial \log p(\boldsymbol{\xi}_k \mid \cdot)}{\partial \varepsilon_{s,k}} = \frac{\partial \log N(\varepsilon_{s,k} \mid 0, 1)}{\partial \varepsilon_{s,k}} + \sum_{j,t} \frac{\partial \log \mathrm{Pois}(y_{j,t} \mid \lambda_{j,t})}{\partial \lambda_{j,t}} \frac{\partial \lambda_{j,t}}{\partial \alpha_{j,k}} \frac{\partial \alpha_{j,k}}{\partial \theta_{s,k}} \tau_k \tilde{\gamma}_{s,k}.$$

Gradient with respect to $\log \gamma_{s,k}$ The chain from $\log \gamma_{s,k}$ to $\theta_{s,k}$ passes through the regularized scale $\tilde{\gamma}_{s,k}$, with

$$\frac{\partial \theta_{s,k}}{\partial \tilde{\gamma}_{s,k}} = \tau_k \varepsilon_{s,k},$$
$$\frac{\partial \tilde{\gamma}_{s,k}}{\partial \gamma_{s,k}} = \frac{c^3}{(c^2 + \tau_k^2 \gamma_{s,k}^2)^{3/2}}.$$

The gradient of the log-posterior with respect to $\log \gamma_{s,k}$ is then

$$\frac{\partial \log p(\boldsymbol{\xi}_k \mid \cdot)}{\partial \log \gamma_{s,k}} = \frac{\partial \log p(\gamma_{s,k})}{\partial \gamma_{s,k}} \gamma_{s,k} + \frac{1 - \gamma_{s,k}^2}{1 + \gamma_{s,k}^2} + \gamma_{s,k} \sum_{j,t} \frac{\partial \log \mathrm{Pois}(y_{j,t} \mid \lambda_{j,t})}{\partial \lambda_{j,t}} \frac{\partial \lambda_{j,t}}{\partial \alpha_{j,k}} \frac{\partial \alpha_{j,k}}{\partial \theta_{s,k}} \frac{\partial \theta_{s,k}}{\partial \tilde{\gamma}_{s,k}} \frac{\partial \tilde{\gamma}_{s,k}}{\partial \gamma_{s,k}}.$$

Here the second term arises from differentiating the log of the transformed half-Cauchy density on $\log \gamma_{s,k}$. Specifically, the half-Cauchy density for $\gamma_{s,k}$ expressed on the log scale takes the form $p(\log \gamma_{s,k}) \propto \gamma_{s,k}/(1 + \gamma_{s,k}^2)$, so that $\partial \log p(\log \gamma_{s,k})/\partial \log \gamma_{s,k} = (1 - \gamma_{s,k}^2)/(1 + \gamma_{s,k}^2)$.

Gradient with respect to $\log \tau_k$ Because $\theta_{s,k} = c\,\tau_k\,\gamma_{s,k}\,\varepsilon_{s,k}\,(c^2 + \tau_k^2\gamma_{s,k}^2)^{-1/2}$, define

$$\theta'_{s,k} := \frac{\partial \theta_{s,k}}{\partial \tau_k} = \frac{c^3 \gamma_{s,k}\,\varepsilon_{s,k}}{(c^2 + \tau_k^2\gamma_{s,k}^2)^{3/2}}.$$

The centered deviations then satisfy

$$\frac{\partial \tilde{\theta}_{s,k}}{\partial \tau_k} = \theta'_{s,k} - \frac{1}{S-1}\sum_{j \neq k} \theta'_{j,k},$$

and the spatial weight derivatives are

$$\frac{\partial \alpha_{j,k}}{\partial \tau_k} = (1 - w_k)\frac{u_{j,k}}{U_k}\left(\theta'_{j,k} - \sum_{i \neq k} \frac{u_{i,k}}{U_k}\theta'_{i,k}\right).$$

The half-Cauchy density for τ_k on the log scale takes the form $p(\log \tau_k) \propto (\tau_k/\tau_0)/(1 + \tau_k^2/\tau_0^2)$, giving $\partial \log p(\log \tau_k)/\partial \log \tau_k = (1 - \tau_k^2/\tau_0^2)/(1 + \tau_k^2/\tau_0^2)$.

The full gradient with respect to $\log \tau_k$ then combines this prior contribution with the likelihood terms summed over all destinations j and time points t, multiplied by τ_k for the log-transformation Jacobian.

Gradient with respect to $\operatorname{logit}(w_k)$ Writing $\tilde{w}_k = \operatorname{logit}(w_k)$, the gradient decomposes as

$$\begin{aligned}\frac{\partial \log p(\boldsymbol{\xi}_k \mid \cdot)}{\partial \tilde{w}_k} &= a_w(1 - w_k) - b_w\,w_k \\ &\quad + \sum_{s,t} \frac{\partial \log \operatorname{Pois}(y_{s,t} \mid \lambda_{s,t})}{\partial \lambda_{s,t}} \left[\sum_{l=1}^{t-1} \frac{\partial \alpha_{s,k}}{\partial w_k} R_{k,l}\,\phi_{t-l}\,y_{k,l}\right] w_k(1 - w_k),\end{aligned}$$

where the first term is the derivative of $a_w \log w_k + b_w \log(1 - w_k)$ with respect to $\tilde{w}_k$, and

$$\frac{\partial \alpha_{s,k}}{\partial w_k} = \delta(s = k) - \delta(s \neq k)\frac{u_{s,k}}{U_k}.$$

B.1.3. HMC for baseline intensity

The baseline intensity μ enters $\lambda_{s,t}$ additively and affects all locations and time points. HMC operates on $\tilde{\mu} = \log \mu$, with prior $\tilde{\mu} \sim N(a_\mu, b_\mu^2)$. The log-posterior target is

$$\log p(\tilde{\mu} \mid \cdot) \propto \log N(\tilde{\mu} \mid a_\mu, b_\mu^2) + \sum_{s,t} \log \text{Pois}(y_{s,t} \mid \lambda_{s,t}),$$

with the Jacobian correction implied by $\mu = \exp(\tilde{\mu})$.

B.1.4. Block update for reproduction number disturbances

The reproduction number trajectory at location k is recovered from the disturbance vector $\boldsymbol{\eta}_k = (\eta_{k,0}, \eta_{k,1}, \ldots, \eta_{k,T-1})'$ via $r_{k,t} = \sqrt{W} \sum_{j=0}^{t} \eta_{k,j}$. We update $\boldsymbol{\eta}_k$ jointly using a MH step whose proposal is constructed from a Gaussian approximation to the Poisson likelihood.

Linearizing $h(r_{k,l})$ around the current iterate $r_{k,l}^{(i)}$ gives

$$h(r_{k,l}) \approx h(r_{k,l}^{(i)}) + h'(r_{k,l}^{(i)})(r_{k,l} - r_{k,l}^{(i)}).$$

Substituting into $\lambda_{s,t}$ and using a Gaussian moment-matching approximation $y_{s,t} \sim N(\lambda_{s,t}, \lambda_{s,t})$ to the Poisson likelihood, the intensity becomes linear in $\boldsymbol{\eta}_k$. Define

$$g_{s,t}^{(i)} = \mu + \sum_{k'=1}^{S} \sum_{l=1}^{t-1} \alpha_{s,k'} \left[h(r_{k',l}^{(i)}) - h'(r_{k',l}^{(i)}) r_{k',l}^{(i)} \right] \phi_{t-l}\, y_{k',l}.$$

Isolating the contribution from source k and substituting $r_{k,l} = \sqrt{W} \sum_{j=0}^{l} \eta_{k,j}$ yields

$$\lambda_{s,t}^{(i)} = \mu_{s,t}^{(i)} + \sqrt{W}\, \alpha_{s,k} \sum_{j=0}^{t-1} c_{k,j,t}^{(i)}\, \eta_{k,j},$$

where $\mu_{s,t}^{(i)}$ collects all terms not involving $\boldsymbol{\eta}_k$ and

$$c_{k,j,t}^{(i)} := \sum_{l=j}^{t-1} h'(r_{k,l}^{(i)})\, \phi_{t-l}\, y_{k,l}.$$

Stacking over time points $t = 2, \ldots, T$ gives the matrix form

$$\boldsymbol{\lambda}_s^{(i)} = \boldsymbol{\mu}_s^{(i)} + \sqrt{W}\, \alpha_{s,k}\, \mathbf{C}_k^{(i)}\, \boldsymbol{\eta}_k,$$

where $\mathbf{C}_k^{(i)}$ is a $(T-1) \times T$ lower-triangular matrix with entries $[\mathbf{C}_k^{(i)}]_{t-1,j+1} = c_{k,j,t}^{(i)}$.

Combining the standard normal prior $\boldsymbol{\eta}_k \sim N(\mathbf{0}, \mathbf{I})$ with the Gaussian approximate likelihood across all S destinations gives the Gaussian proposal

$$\boldsymbol{\eta}_k \mid \cdot \;\sim\; N\!\left(\boldsymbol{u}_k^{(i)},\; \left[\boldsymbol{\Omega}_k^{(i)}\right]^{-1}\right),$$

with precision matrix and mean

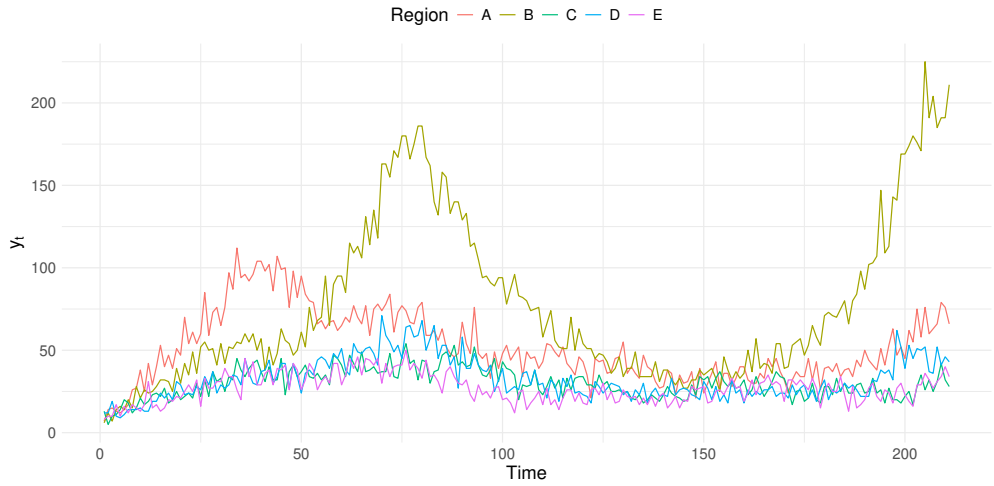
$$\boldsymbol{\Omega}_k^{(i)} = \mathbf{I} + W \sum_{s=1}^{S} \alpha_{s,k}^2\, \mathbf{C}_k^{(i)\prime} \left[\boldsymbol{\Lambda}_s^{(i)}\right]^{-1} \mathbf{C}_k^{(i)},$$

$$\boldsymbol{u}_k^{(i)} = \sqrt{W}\, \left[\boldsymbol{\Omega}_k^{(i)}\right]^{-1} \sum_{s=1}^{S} \alpha_{s,k}\, \mathbf{C}_k^{(i)\prime} \left[\boldsymbol{\Lambda}_s^{(i)}\right]^{-1} \boldsymbol{d}_s^{(i)},$$

where $\boldsymbol{\Lambda}_s^{(i)} = \mathrm{diag}(\lambda_{s,2}^{(i)}, \ldots, \lambda_{s,T}^{(i)})$ and $\boldsymbol{d}_s^{(i)} = \boldsymbol{y}_s - \boldsymbol{\mu}_s^{(i)}$. A candidate $\boldsymbol{\eta}_k^*$ is drawn from this proposal and accepted or rejected by a MH step targeting the exact Poisson posterior.

§ B.2. Simulation study 1: additional results

Figure B.1 shows the simulated case counts and spatial weight matrix for Study 1. The design uses $S = 5$ locations, $T = 200$ time points, and heterogeneous retention rates $w_k \in \{0.9, 0.7, 0.5, 0.3, 0.3\}$ for locations A through E.



(a) Simulated case counts $y_{s,t}$ across five locations over 200 time points.

	A	B	C	D	E
A	0.90	0.12	0.20	0.16	0.24
B	0.01	0.70	0.13	0.17	0.12
C	0.00	0.03	0.50	0.18	0.18
D	0.01	0.13	0.10	0.30	0.17
E	0.09	0.03	0.07	0.19	0.30

(b) Spatial weight matrix $\boldsymbol{\alpha} = (\alpha_{s,k})$.

Figure B.1.: Simulation setup for Study 1. Panel (a) shows the simulated case counts for five locations over $T = 200$ time points with retention rates $\omega_k \in \{0.9, 0.7, 0.5, 0.3, 0.3\}$. Panel (b) shows the spatial weight matrix, where diagonal entries represent retention rates and off-diagonal entries represent cross-location transmission probabilities.

Table B.1.: Posterior summaries for the retention rates w_k in Study 1. All true values lie within the 95% credible intervals.

Location	True w_k	Posterior mean	95% CrI
A	0.90	0.85	[0.73, 0.97]
B	0.70	0.72	[0.63, 0.85]
C	0.50	0.48	[0.30, 0.76]
D	0.30	0.42	[0.18, 0.79]
E	0.30	0.36	[0.23, 0.59]

Table B.1 reports posterior inference for the diagonal retention rates w_k. All true values fall within their 95% credible intervals. Agreement is strongest for locations B, C, and E ($\hat{w}_B = 0.72$ vs. 0.70, $\hat{w}_C = 0.48$ vs. 0.50, $\hat{w}_E = 0.36$ vs. 0.30), while location A is mildly underestimated ($\hat{w}_A = 0.85$ vs. 0.90), consistent with finite-sample shrinkage toward intermediate values.

§ B.3. Simulation study 2: robustness results

Table B.2 reports the full results on robustness to misspecification summarized in the main paper. Including mobility covariates when they are absent from the data generating process causes little harm. Across retention levels, the horseshoe models with and without mobility covariates produce similar reproduction number estimates and forecasting performance. Replacing the horseshoe prior with a Gaussian prior affects forecast calibration more than point estimation. The Gaussian prior performs comparably at high retention, but at lower retention it yields worse CRPS, suggesting that adaptive shrinkage is most valuable when cross-location transmission is stronger and the network is harder to identify. Frobenius norm differences between the two

priors are small and do not consistently favor either approach, indicating that with $S = 5$ locations the primary benefit of the horseshoe prior is in forecast calibration rather than in network recovery.

Table B.2.: Robustness to misspecification ($S = 5$, $T = 200$). The data generating process has horseshoe-distributed deviations and no mobility covariate. Including mobility causes little harm, while a Gaussian prior mainly worsens forecast calibration at low retention.

Retention	Model	MAE($R_{k,t}$)	CRPS_1	CRPS_5	CRPS_{10}	Frobenius
30%	Univariate	**0.156**	5.53	7.32	8.96	—
	Horseshoe	0.166	**5.22**	5.89	7.50	0.347
	Horseshoe + Mob.	0.191	5.26	**5.87**	**7.26**	0.353
	Gaussian	0.173	6.87	6.95	8.22	**0.341**
70%	Univariate	2.724	**7.28**	9.80	13.58	—
	Horseshoe	**0.072**	7.62	**8.11**	**10.28**	0.209
	Horseshoe + Mob.	0.075	7.63	8.37	10.65	**0.204**
	Gaussian	0.075	7.67	8.39	10.60	0.206
95%	Univariate	**0.033**	**9.15**	8.77	12.41	—
	Horseshoe	0.049	9.37	8.76	11.65	**0.413**
	Horseshoe + Mob.	0.049	9.43	8.78	11.71	0.436
	Gaussian	0.061	9.21	**8.49**	**11.17**	0.454

§ B.4. Simulation study for zero-inflation component

The simulation studies in the main paper generate data without zero inflation. Here we assess the zero-inflation component through two complementary scenarios. The first generates data with zero inflation and compares models that include or exclude the zero-inflation mechanism. The second generates data without zero inflation and fits a model that includes it, testing whether the additional component causes harm when unnecessary. Both scenarios use $S = 5$ locations with a uniform retention rate of 70%

and no mobility covariate.

For the zero-inflated scenario, data are generated from the network Hawkes model with zero-inflation indicators governed by $\text{logit}(p_{s,t}) = \beta_0 + \beta_1 z_{s,t-1}$, with $\beta_0 = -1$ and $\beta_1 = 5$. These coefficients produce a persistent zero-inflation process in which, once a location enters the zero state, it remains there for several consecutive time points before reactivating. This pattern mimics both reporting interruptions (such as weekends or holidays) and the early stages of an epidemic when a location has not yet been seeded, where counts remain at zero for short runs rather than isolated single days. When $z_{s,t} = 0$, the observed count is set to zero regardless of the underlying intensity. The remaining parameters match the Study 2 design ($T = 200$, $W = 0.01$, $\mu = 2$, discretized log-normal serial interval). For the non-zero-inflated scenario, we use the same data as Study 2 at 70% retention.

Table B.3.: Simulation with zero-inflated data ($S = 5$, $T = 200$, 70% retention). Including the zero-inflation component improves both reproduction number estimation and forecast calibration.

Model	MAE($R_{k,t}$)	CRPS_1	CRPS_5	CRPS_{10}
Univariate	0.381	6.15	7.79	9.90
Horseshoe	0.064	5.60	7.62	8.42
Horseshoe + Zero	**0.046**	**5.43**	**6.60**	**7.32**

Table B.3 shows that including the zero-inflation component improves reproduction number estimation (MAE reduced by 28%, from 0.064 to 0.046) and forecast calibration, with the benefit concentrated at longer horizons. The one-step CRPS improves modestly by 3%, while the five-step and ten-step improvements grow to 13%. Without the zero-inflation mechanism, the model interprets structural zeros as periods of suppressed transmission, biasing reproduction numbers downward during reporting gaps. Over

Table B.4.: Simulation without zero-inflated data ($S = 5$, $T = 200$, 70% retention). Including the zero-inflation component when it is absent from the data generating process causes minimal harm.

Model	MAE($R_{k,t}$)	$CRPS_1$	$CRPS_5$	$CRPS_{10}$	Frobenius
Univariate	2.724	**7.28**	9.80	13.58	—
Horseshoe	**0.072**	7.62	**8.11**	**10.28**	0.209
Horseshoe + Mob.	0.075	7.63	8.37	10.65	**0.204**
Gaussian	0.075	7.67	8.39	10.60	0.206
Horseshoe + Zero	0.073	7.51	8.23	10.42	0.206

multiple forecast steps, these biased trajectories produce increasingly miscalibrated predictive distributions.

Table B.4 shows the reverse scenario. Including the zero-inflation component when it is absent from the data generating process causes negligible harm, with MAE increasing by less than 2% and CRPS values remaining within 2% of the baseline horseshoe model at all horizons. The Frobenius error on the transmission matrix is essentially unchanged, at 0.206 against 0.209 for the baseline horseshoe model, so adding the component leaves network recovery intact. The model effectively switches off the zero-inflation mechanism by learning a low activation probability for the zero state. These results mirror the robustness pattern observed for covariate misspecification in Study 2b, indicating that, in these simulations, including the component when it is unnecessary causes little harm.

§ B.5. Sensitivity analyses on simulated data

The analyses in this section report how the outputs of the network Hawkes model change when the components that are fixed or tuned rather than estimated are varied:

the lag kernel $\{\phi_h\}$, the innovation variance W of the random walk, the priors, and the gain function. All use the Study 1 data generating process of § 3.4, with $S = 5$ locations over $T = 200$ days and 10 held-out days, softplus productivity on a Gaussian random walk, a column-stochastic excitation matrix with self-retention weights $w = (0.9, 0.7, 0.5, 0.3, 0.3)$, and baseline rate $\mu = 2$. In each analysis the model is fit across a grid of the setting under study with the rest of the specification held fixed, and each fit is compared to a baseline fit that uses the data generating setting.

Each analysis is summarized by the change in the model's reported quantities relative to the baseline fit, using the metrics of § 3.4. For the transmission matrix we report the root mean square of the changes in its off-diagonal weights, $\|\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}_0\|_F / \sqrt{S(S-1)}$, where $\boldsymbol{\alpha}_0$ is the baseline matrix. For the retention weights we report the largest absolute change, $\max_k |\hat{w}_k - w_{k,0}|$, and for the network reproduction number the mean absolute change over time. Where the reproduction trajectories are the target, we report their mean absolute error against the truth, the empirical coverage of the 95% credible interval, and the mean interval width. The source and sink structure is summarized by the ordering of the locations by net flow.

B.5.1. Sensitivity to the lag distribution

We vary the serial interval mean over $\{2.5, 3.5, 4.7, 6.0, 7.5\}$ days at the data generating coefficient of variation, and its standard deviation over $\{1.5, 2.9, 4.5\}$ days at fixed mean 4.7, taking the kernel with mean 4.7 as the baseline. The excitation structure changes little across the grid (Table B.5). The root mean square change in the off-diagonal weights stays at or below 0.027, the network reproduction number changes by at most 0.083, and the retention weights change by at most 0.085. The ordering of locations by

net flow changes by at most a single adjacent transposition among the five locations, at the shortest lag and the widest dispersion. The local and imported reproduction numbers are deterministic functions of the transmission matrix and the reproduction trajectories, so their variation over the grid is of the same order.

Table B.5.: Sensitivity to the lag distribution, relative to the kernel with mean 4.7. Columns report the root mean square change in the off-diagonal transmission weights, the largest absolute change in the retention weights, and the mean absolute change in the network reproduction number.

Setting	RMS weight change	Max retention change	R^{net} change
mean 2.5	0.027	0.085	0.061
mean 3.5	0.015	0.044	0.034
mean 4.7 (base)	0.000	0.000	0.000
mean 6.0	0.009	0.021	0.038
mean 7.5	0.024	0.055	0.083
sd 1.5	0.014	0.035	0.012
sd 4.5	0.010	0.027	0.016

B.5.2. Sensitivity to the innovation variance

We fix W at each of $\{0.001, 0.005, 0.01, 0.05, 0.1\}$, with the data generating value $W = 0.01$ as the baseline, and assess recovery of the reproduction trajectories and the decomposition (Table B.6). The variance W controls the smoothness of the reproduction trajectories and the width of their credible intervals. The interval width increases monotonically with W, from 0.265 at $W = 0.001$ to 1.271 at $W = 0.1$, and the coverage rises from 0.46 under heavy smoothing toward the nominal level near the data generating value. The mean absolute error of $R_{k,t}$ is smallest at the data generating value, at 0.132. The ordering of locations by net flow changes by at most a single adjacent transposition across the grid, and the local and imported reproduction numbers change least near

the data generating value. Forecasting performance as a function of W is examined separately on the application data in § B.6.1.

Table B.6.: Sensitivity to the innovation variance W, with the data generating value $W = 0.01$ as the baseline. Columns report the mean absolute error, the empirical coverage of the 95% credible interval, and the mean interval width for $R_{k,t}$, and the mean absolute change in the local and imported reproduction numbers relative to the baseline.

W	MAE(R)	Coverage(R)	CI width(R)	Local change	Import change
0.001	0.173	0.463	0.265	0.101	0.205
0.005	0.158	0.625	0.321	0.058	0.117
0.01 (base)	0.132	0.887	0.604	0.000	0.000
0.05	0.177	0.977	0.992	0.101	0.140
0.1	0.334	0.853	1.271	0.182	0.396

B.5.3. Prior sensitivity

We vary one prior hyperparameter at a time around the default configuration, across the Beta prior on the retention weights ω_k, the horseshoe slab scale c^2, the global shrinkage scale τ_0, the prior on the log baseline intensity $\log \mu$, and the prior on the activation coefficients. The reported quantities change little under the priors we varied (Table B.7). The ordering of locations by net flow is unchanged in every scenario, the network reproduction number moves by at most 0.016, and the one-step forecast score stays within a few percent of the default. The posterior mean of the baseline intensity moves by at most 0.058 on a scale where $\mu = 2$. The two quantities that a prior parameterizes directly respond to it. The retention weights move by up to 0.13 under the Beta prior on ω_k, while any other prior change leaves them within 0.051 of the default, and the global shrinkage scale tracks τ_0, with the posterior median of τ_k

falling to 0.68 of its default value when τ_0 is halved and rising to twice its default when τ_0 is doubled. Away from the retention prior, which rescales the off-diagonal weights through the factor $1 - \omega_k$, the off-diagonal weights change by at most 0.072 in root mean square.

Table B.7.: Prior sensitivity, relative to the default prior configuration. Columns report the root mean square change in the off-diagonal transmission weights, the largest absolute change in the retention weights, the ratio of the posterior median of the global shrinkage scale to its default, the absolute change in the posterior mean of the baseline intensity μ, the mean absolute change in the network reproduction number, and the ratio of the one-step forecast CRPS to the default.

Scenario	RMS weight change	Max retention change	τ ratio	μ change	R^{net} change	CRPS_1 ratio
baseline	0.000	0.000	1.00	0.000	0.000	1.00
retention: uniform	0.208	0.130	1.51	0.028	0.016	0.94
retention: symmetric	0.216	0.123	1.74	0.031	0.012	0.93
slab: tight	0.029	0.022	1.17	0.012	0.004	1.00
slab: loose	0.033	0.019	1.06	0.015	0.004	1.00
τ_0: halved	0.034	0.022	0.68	0.005	0.005	1.00
τ_0: doubled	0.047	0.028	2.00	0.014	0.005	0.95
μ: tight	0.036	0.019	1.21	0.058	0.003	1.00
μ: wide	0.072	0.051	1.09	0.001	0.002	1.06
activation: tight	0.032	0.024	1.03	0.003	0.003	1.01
activation: wide	0.029	0.017	1.11	0.018	0.003	1.01

B.5.4. Sensitivity to the gain function

The reproduction number enters the intensity through the gain function, $R_{k,t} = h(r_{k,t})$, where $r_{k,t}$ is the latent random walk. We use the softplus gain function, $h(r) = \log(1+e^r)$, throughout, and compare it here with the exponential gain function $h(r) = e^r$. We prefer softplus for numerical stability, since the exponential gain function can produce very large reproduction numbers over a long series.

We assess the choice of gain function with two experiments on the Study 1 design ($S = 5$, $T = 200$). The first tests sensitivity to the gain function used at estimation.

Data are generated under the softplus gain function and the model is fit under each gain function. Fitting under the exponential gain function recovers the ordering of locations by net flow (rank correlation 1) and the reproduction trajectories about as well as the correctly specified softplus fit, with mean absolute error 0.125 against 0.117 and coverage 0.95 against 0.91, and recovers the transmission matrix a little less accurately, with Frobenius error to the truth 0.311 against 0.218. The main conclusions are therefore insensitive to the gain function used at estimation.

The second experiment generates and fits data under the same gain function, to check that each gain function is internally consistent (Table B.8). Both gain functions recover their own truth to a comparable degree, with Frobenius error to the truth 0.218 and 0.239, net-flow rank correlation 0.9, and $R_{k,t}$ coverage near 0.92. In this simulation the exponential process was generated with a smaller innovation variance than the softplus one, $W = 0.001$ against 0.01, because the exponential gain function otherwise produces very large reproduction numbers over the study window.

Table B.8.: Sensitivity to the gain function. Recovery of each gain function against its own data generating truth when data are generated and fit under the same gain function, showing the innovation variance W used to generate each process, the Frobenius error of the transmission matrix against the truth, the Spearman correlation of the ordering by net flow against the truth, the mean absolute error and empirical 95% coverage of $R_{k,t}$, and the mean network reproduction number.

Gain	W	Frobenius to truth	Rank corr. vs truth	MAE(R)	Coverage(R)	Mean R^{net}
softplus	0.010	0.218	0.90	0.117	0.912	0.835
exponential	0.001	0.239	0.90	0.111	0.917	1.012

§ B.6. Sensitivity analyses on the application data

This section collects the sensitivity and robustness analyses on the ten-county California application. We first vary the innovation variance W and assess its effect on forecasting, then examine whether the static excitation matrix varies across epidemic regimes, whether the residuals retain periodic structure that the model does not represent, and whether the zero-inflation component is needed for these counties. The last three analyses share a common fitting configuration. They use the mobility covariate, the horseshoe prior, and the same shrinkage calibration as the main application, with the innovation variance W fixed at 0.005, within the range over which forecasting performance is stable in § B.6.1, the zero-inflation component omitted except where it is the object of study, and each fit using 20,000 burn-in and 10,000 retained iterations thinned to every sixth draw.

B.6.1. Sensitivity to the innovation variance

The random walk innovation variance W controls the smoothness of the reproduction number trajectories $R_{k,t}$. Smaller values produce smoother trajectories that respond slowly to changes in transmission, while larger values allow rapid fluctuations that track short-term variation but may overfit noise. We assess sensitivity to this choice by fitting the network Hawkes model to the California COVID-19 data under six values of W spanning two orders of magnitude ($W \in \{0.001, 0.005, 0.01, 0.05, 0.1, 0.5\}$) and evaluating out-of-sample forecasting performance at horizons of 1, 5, and 10 days.

For each value of W, forecasting performance was evaluated using a rolling-window design over seven consecutive forecast origins that precede the fourteen-origin evalua-

tion window of Table 3.4, so the choice of W does not reuse the data on which forecast performance is reported. At each origin, the model was fit with one additional day of data and probabilistic forecasts were generated at horizons $h \in \{1, 5, 10\}$ days. The reported CRPS values are averaged across all forecast origins and all ten counties.

Table B.9.: Sensitivity of forecasting performance to the random walk innovation variance W. Values are mean CRPS across seven rolling forecast origins and ten counties. The choice $W = 0.005$ achieves the best overall performance across forecast horizons.

W	CRPS_1	CRPS_5	CRPS_{10}
0.001	2.25	2.58	3.56
0.005	2.27	2.41	3.09
0.01	2.23	2.55	3.20
0.05	2.31	2.99	3.65
0.1	2.14	3.43	4.85
0.5	1.97	11.56	25.41

Table B.9 reveals a clear bias-variance tradeoff across forecast horizons. At the one-step horizon, performance is relatively stable for $W \leq 0.1$ (CRPS ranging from 2.14 to 2.31) and improves slightly at $W = 0.5$ (CRPS = 1.97), where highly flexible trajectories can closely track recent observations. However, this flexibility comes at a steep cost at longer horizons. At ten steps ahead, $W = 0.5$ produces a CRPS of 25.41, roughly eight times worse than the best-performing values, because the rapid fluctuations that benefit one-step prediction amplify over multi-step simulation and generate highly dispersed predictive distributions.

For moderate values ($W \in \{0.005, 0.01, 0.05\}$), performance is stable across all horizons. Among these, $W = 0.005$ achieves the best five-step CRPS (2.41) and the best ten-step CRPS (3.09), while its one-step performance (2.27) remains comparable

to neighboring values. The slight one-step cost relative to $W = 0.01$ (2.27 versus 2.23) is offset by meaningful gains at longer horizons, where forecasting skill matters most for surveillance planning. The very smooth trajectories under $W = 0.001$ perform comparably at short horizons but show worse ten-step CRPS (3.56), suggesting that over-smoothed trajectories fail to capture genuine shifts in transmission that accumulate over longer horizons. Based on these results, we set $W = 0.005$ for the application in the manuscript.

Overall, the main conclusions of the application are robust to the choice of W within the range 0.005 to 0.05, and $W = 0.005$ provides a favorable balance between responsiveness to genuine transmission changes and stability in multi-step forecasting.

B.6.2. Subperiod robustness

The excitation matrix is estimated as static over the fitting window. To examine how it varies across epidemic regimes, we fit the model on the full window and on three contiguous subperiods chosen to correspond to distinct phases of the epidemic: a summer 2020 lull (2 July to 15 October 2020, 105 days), a winter 2020 to 2021 surge (15 November 2020 to 15 March 2021, 120 days), and the Delta period (15 June to 1 December 2021, 169 days). For each fit we record the posterior mean of the excitation matrix, the ordering of the ten counties by net offspring flow, and the network reproduction number, and compare each subperiod to the fit on the full window through the root mean square change in the off-diagonal weights defined in § B.5 and the Spearman correlation of the ordering by net flow (Table B.10).

The root mean square change in the off-diagonal weights from the fit on the full window is 0.163 in summer 2020, 0.169 in winter 2021, and 0.138 in the Delta period.

Table B.10.: Subperiod robustness of the fitted network. Each subperiod is compared to the fit on the full window through the root mean square change in the off-diagonal transmission weights and the Spearman correlation of the ordering of the ten counties by net flow. The mean network reproduction number is reported for each window.

Subperiod	Window	T	RMS weight change	Rank corr. vs full	Mean R^{net}
full	2020-07-02 to 2021-12-01	517	0.000	1.000	0.720
summer 2020	2020-07-02 to 2020-10-15	105	0.163	0.818	0.437
winter 2021	2020-11-15 to 2021-03-15	120	0.169	0.806	0.594
Delta 2021	2021-06-15 to 2021-12-01	169	0.138	0.709	0.885

These changes are larger than those produced by the lag and prior variations in § B.5, indicating that the excitation matrix does move across regimes rather than staying fixed. The ordering of counties by net flow is broadly preserved, with Spearman correlation 0.82, 0.81, and 0.71 against the fit on the full window, so the ranking of counties as net sources and net sinks is similar across windows but not identical. The mean network reproduction number tracks the epidemic phase, rising from 0.44 in the summer lull to 0.59 in the winter surge and 0.89 in the Delta period, and stays below one in every window. Read together, these results support treating the excitation matrix from the full window as an average connectivity structure over the study period, since the source and sink ranking it implies is representative of the regimes while the edge magnitudes and the overall growth rate vary with the epidemic phase.

B.6.3. Residual seasonality

The model contains no explicit seasonal or day-of-week term, so any periodic structure in the series is either carried by the time-varying reproduction trajectories or left in the residuals. We fit the model once on the full window and form standardized posterior

predictive residuals for each county as the observed count minus the mean of the posterior predictive replicates, divided by their standard deviation. For each county we take the dominant period and its spectral power from the periodogram, and the residual autocorrelation at lags of one week, two weeks, one month, and one year (Table B.11).

Table B.11.: Residual seasonality on the application data. For each county, the dominant period of the standardized posterior predictive residuals from the periodogram, and the residual autocorrelation at lags of one week, two weeks, and one year.

County	Peak period	ACF (lag 7)	ACF (lag 14)	ACF (lag 365)
Alameda	7.0	0.698	0.591	0.049
Contra Costa	7.0	0.570	0.491	0.040
Merced	7.0	0.538	0.518	−0.001
Sacramento	7.0	0.522	0.511	0.048
San Francisco	7.0	0.279	0.232	0.022
San Joaquin	7.0	0.514	0.459	0.027
San Mateo	7.0	0.552	0.444	0.049
Santa Clara	7.0	0.651	0.558	0.042
Santa Cruz	7.0	0.547	0.477	−0.007
Stanislaus	7.0	0.245	0.202	−0.015

The dominant residual period is one week for every county. The residual autocorrelation at a lag of one week ranges from 0.245 to 0.698, and at a lag of two weeks from 0.202 to 0.591, both positive, so a weekly cycle remains in the residuals. The autocorrelation at a lag of one month is small and negative for every county, between −0.185 and −0.023, and at a lag of one year it is near zero, between −0.015 and 0.049; the annual lag is in any case only weakly estimable over a window of 517 days. The residuals therefore show a weekly cycle with no monthly or annual structure. The weekly cycle is consistent with day-of-week reporting patterns, weekend under-reporting followed by catch-up, rather than with weekly variation in transmission, and the reproduction

trajectories, smoothed by the small innovation variance $W = 0.005$, do not absorb it. Its most direct consequence is at the observation level, where the Poisson model understates the predictive dispersion at short horizons, so the prediction intervals at short horizons are slightly too narrow. This weekly cycle is faster than the transmission dynamics that the excitation matrix and the reproduction trajectories represent, so we expect it to have limited bearing on the source and sink ordering, the network reproduction number, and the local and imported decomposition; we do not, however, fit a model with a seasonal term, so this expectation is not verified directly. Incorporating a day-of-week or seasonal term in the intensity, or aggregating to weekly counts, would remove the residual weekly structure and is left to future work.

B.6.4. Zero inflation

The zero-inflation component is designed for surveillance data with runs of structural zeros, which the California county counts do not have. Only 0.2% of the observed counts are zero, and those zeros are isolated rather than sustained. When the full model is fit with the component, the activation process switches itself off. The estimated coefficients place the activation probability near one from either state, at 0.99 from the inactive state and above 0.999 from the active state, the implied probability that an observation is a structural zero is 2.5×10^{-5}, and no observation is flagged as a structural zero. The component therefore behaves as the simulation without structural zeros in § B.4 predicts, learning a high activation probability and reducing to a plain Poisson observation model. The latent activation indicators here summarize a reporting state rather than the presence or absence of disease.

To check how much the component affects the epidemiological conclusions, we fit

the model to the ten-county data with and without it, under the same configuration as the subperiod and residual checks above and with only the activation block toggled, and compare the two fits (Table B.12). The network reproduction number is 0.720 under the plain Poisson model against 0.719 with zero inflation, the ordering of counties by net flow is essentially unchanged, with Spearman correlation 0.99, and the forecast scores are nearly identical at all three horizons. The transmission matrix differs by 0.11 in root mean square, of the same order as the lag and prior variations in § B.5 and smaller than the shifts across epidemic subperiods in § B.6.2, and this difference leaves the source and sink ranking and the network reproduction number intact. Zero inflation is therefore not needed for the California application, and including it leaves the main conclusions unchanged, consistent with the negligible harm it causes in the simulation without structural zeros.

Table B.12.: Zero-inflated against plain Poisson observation model on the ten-county application. The transmission matrix and the ordering of counties by net flow are compared to the zero-inflated fit through the root mean square change in the matrix weights and the Spearman correlation of the ordering; the mean network reproduction number and the forecast CRPS at horizons of one, five, and ten days are reported for each.

Model	RMS weight change	Rank corr. vs ZI	Mean R^{net}	CRPS_1	CRPS_5	CRPS_{10}
Zero-inflated	0.000	1.000	0.719	3.13	2.42	2.04
Poisson	0.106	0.988	0.720	3.11	2.41	2.04

C. Supplement to Chapter 5

§ C.1. Application: HVA vs MCMC comparison for the discrete-time Hawkes-type model

This appendix provides the parameter-by-parameter posterior summaries, the seasonality and R_t plots, and the combined in-sample and forecast score table that support the brief statement in § 5.6.3 that HVA and MCMC agreed closely on the discrete-time Hawkes-type fit to the Santa Cruz County COVID-19 data. The two fits assumed for this appendix are the `fit_DH` and `fit_DH_mcmc` objects produced in § 5.6.3.

The MCMC fit converged with split-$\hat{R}$ values below 1.01 for all eleven static parameters, with an average effective sample size of 298. Acceptance rates were 0.69 for the HMC step and the MH update of the latent state. The corresponding HVA fit also showed stable behavior: the SMC log marginal likelihood trace stabilized after approximately 323 iterations, with a tail-to-early MAD ratio of 0.039, indicating substantial stabilization.

The static parameters can be compared side by side with `dgtf_compare_params()`, which, called without a `truth` argument, reports the posterior mean, median, standard deviation, and 95% credible interval for each inferred parameter under each method:

```
R> dgtf_compare_params(list(HVA = fit_DH, MCMC = fit_DH_mcmc))
```

The two inference methods agreed closely on the static parameters in the discrete-time Hawkes-type model. The observational negative binomial overdispersion parameter ρ had posterior median 18.7 under HVA and 17.3 under MCMC, the random walk variance W had posterior median 0.009 under HVA and 0.008 under MCMC, and the lognormal lag distribution mean parameter had a posterior median of 0.49 under HVA and 0.53 under MCMC, while its variance had posterior median 2.6 under HVA and 2.82 under MCMC. Each of the four scalar parameters lay comfortably inside the credible interval produced by the other method.

The seasonal estimates can be interpreted as day-of-week effects. Since July 1, 2020 was a Wednesday, the components `seas[1]` through `seas[7]` index Wednesday through Tuesday. The HVA posterior medians $(11.9, 11.2, 7.81, 3.39, 2.71, 17.2, 9.99)$ and the MCMC posterior medians $(11.5, 10.7, 7.46, 2.61, 2.09, 15.8, 9.61)$ both show a pronounced Monday peak and a weekend drop, consistent with the weekly reporting pattern found in the analysis of the full series in Chapter 2. The two sets of seasonal posterior summaries are similar, with overlapping 95% credible intervals for every day of the week, as shown in Fig. C.1b:

```
R> dgtf_compare_plot(list(HVA = fit_DH, MCMC = fit_DH_mcmc), what =
↪  "seas")
```

The posterior R_t trajectories from the two methods are also visually similar in the training window (Fig. C.1a):

```
R> dgtf_compare_plot(list(HVA = fit_DH, MCMC = fit_DH_mcmc), what =
↪  "Rt")
```

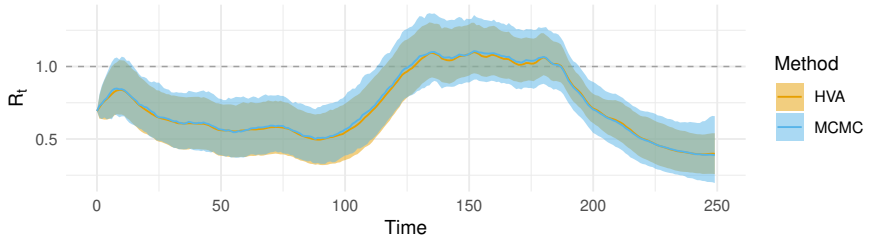
Table C.1 combines the in-sample posterior predictive scores with the 14-day forecast scores for the two methods. Both blocks are produced by `dgtf_compare()`: the in-sample block from the fitted objects after caching their posterior predictive draws, and the forecast block from forecast objects returned by `dgtf_forecast_fit()`. The HVA side of each is already cached in § 5.6.3; the corresponding MCMC caches are produced here.

```
R> set.seed(123)
R> attr(fit_DH_mcmc, "ppc") <- posterior_predict(fit_DH_mcmc)
R> dgtf_compare(list(HVA = fit_DH, MCMC = fit_DH_mcmc))
R> fc_DH_mcmc <- dgtf_forecast_fit(fit_DH_mcmc, h = 14, ypred_true =
↪  ytest)
R> dgtf_compare(list(HVA = fc_DH, MCMC = fc_DH_mcmc))
```

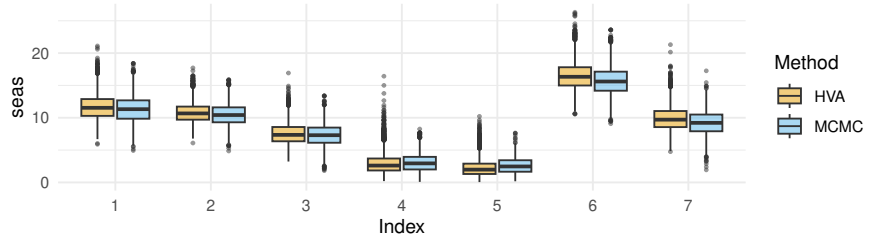
The in-sample posterior predictive scores agree closely, confirming that the small differences in the latent decomposition do not propagate to the predictive distribution, and the 14-day forecast scores agree within Monte Carlo error on CRPS and the chi-square discrepancy. The MCMC fit covers all fourteen held-out observations, while the HVA misses one, consistent with the slightly narrower predictive intervals produced by the variational approximation. The wall-clock runtimes in Table C.1 show HVA to be noticeably faster than MCMC.

Table C.1.: Comparison of HVA and MCMC for the discrete-time Hawkes-type model on the COVID-19 training data, combining in-sample posterior predictive scores from `dgtf_compare()` with 14-day forecast scores from `dgtf_forecast_fit()`. Reported metrics for the in-sample window are the CRPS, the chi-square discrepancy, the empirical coverage and mean width of the 95% credible interval, the Winkler interval score, and the total runtime in seconds. For the forecast horizon, we report empirical coverage, the chi-square discrepancy, and the CRPS averaged across the horizon.

	In-sample fit						14-day forecast		
Method	CRPS	χ^2	Cover	Width	IS	Time (s)	Cover	χ^2	CRPS
HVA	9.43	1.47	0.952	62.8	84.8	264.0	0.929	0.267	3.31
MCMC	9.51	1.57	0.96	65.5	84.8	3910.4	1.000	0.333	3.27



(a) Posterior median R_t trajectory with 95% credible intervals.



(b) Posterior distributions of the seven weekly seasonal components, indexed `seas[1]` through `seas[7]` for Wednesday through Tuesday.

Figure C.1.: Posterior comparison of the discrete-time Hawkes-type model fitted by HVA and MCMC. The two methods produce similar R_t trajectories across the training window, and the seasonality posteriors overlap on every day of the week.

Bibliography

A. Agosto and P. Giudici. A Poisson autoregressive model to understand COVID-19 contagion dynamics. *Risks*, 8(3):77, 2020.

M. Alene, L. Yismaw, M. A. Assemie, D. B. Ketema, W. Gietaneh, and T. Y. Birhan. Serial interval and incubation period of COVID-19: a systematic review and meta-analysis. *BMC Infectious Diseases*, 21(1):257, 2021.

L. J. S. Allen and P. van den Driessche. Relations between deterministic and stochastic thresholds for disease extinction in continuous- and discrete-time infectious disease models. *Mathematical Biosciences*, 243(1):99–108, 2013.

M. B. Alves, D. Gamerman, and M. A. R. Ferreira. Transfer functions in dynamic generalized linear models. *Statistical Modelling*, 10(1):3–40, 2010.

A. L. Bertozzi, E. Franco, G. Mohler, M. B. Short, and D. Sledge. The challenges of modeling and forecasting the spread of COVID-19. *Proceedings of the National Academy of Sciences*, 117(29):16732–16738, 2020.

J. Besag. Statistical analysis of non-lattice data. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 24(3):179–195, 1975.

M. Betancourt. A conceptual introduction to Hamiltonian Monte Carlo. *arXiv preprint arXiv:1701.02434*, 2018.

M. J. Betancourt and M. Girolami. Hamiltonian Monte Carlo for hierarchical models. *arXiv preprint arXiv:1312.0906*, 2013.

J. Bracher and L. Held. Endemic-epidemic models with discrete-time serial interval distributions for infectious disease prediction. *International Journal of Forecasting*, 38(3):1221–1233, 2022.

M. Briers, A. Doucet, and S. Maskell. Smoothing algorithms for state–space models. *Annals of the Institute of Statistical Mathematics*, 62(1):61–89, 2010.

R. Browning, D. Sulem, K. Mengersen, V. Rivoirard, and J. Rosseau. Simple discrete-time self-exciting models can describe complex dynamic processes: a case study of COVID-19. *PLOS ONE*, 16(4):e0250015, 2021.

Y. Burda, R. Grosse, and R. Salakhutdinov. Importance weighted autoencoders. In *International Conference on Learning Representations*, 2016.

R. Cabral, D. Bolin, and H. Rue. Fitting latent non-Gaussian models using variational Bayes and Laplace approximations. *Journal of the American Statistical Association*, 119(548):2983–2995, 2024. ISSN 1537-274X.

C. M. Carvalho, M. S. Johannes, H. F. Lopes, and N. G. Polson. Particle learning and smoothing. *Statistical Science*, 25(1):88–106, 2010a.

C. M. Carvalho, N. G. Polson, and J. G. Scott. The horseshoe estimator for sparse signals. *Biometrika*, 97(2):465–480, 2010b.

S. Chang, E. Pierson, P. W. Koh, J. Gerardin, B. Redbird, D. Grusky, and J. Leskovec. Mobility network models of COVID-19 explain inequities and inform reopening. *Nature*, 589(7840):82–87, 2021.

W.-H. Chiang, X. Liu, and G. Mohler. Hawkes process modeling of COVID-19 with mobility leading indicators and spatial covariates. *International Journal of Forecasting*, 38(2):505–520, 2022.

A. Cori, N. M. Ferguson, C. Fraser, and S. Cauchemez. A new framework and software to estimate time-varying reproduction numbers during epidemics. *American Journal of Epidemiology*, 178(9):1505–1512, 2013.

R. A. Davis, K. Fokianos, S. H. Holan, H. Joe, J. Livsey, R. Lund, V. Pipiras, and N. Ravishanker. Count time series: a methodological review. *Journal of the American Statistical Association*, 116(535):1533–1547, 2021.

O. Diekmann, J. A. P. Heesterbeek, and J. A. J. Metz. On the definition and the computation of the basic reproduction ratio R_0 in models for infectious diseases in heterogeneous populations. *Journal of Mathematical Biology*, 28(4):365–382, 1990.

A. Doucet and A. M. Johansen. A tutorial on particle filtering and smoothing: fifteen years later. In D. Crisan and B. Rozovskii, editors, *The Oxford Handbook of Nonlinear Filtering*, pages 656–704. Oxford University Press, 2011.

D. Douwes-Schultz and A. M. Schmidt. Zero-state coupled Markov switching count models for spatio-temporal infectious disease spread. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 71(3):589–612, 2022.

D. Douwes-Schultz, A. M. Schmidt, L. P. Freitas, and M. S. Carvalho. Markov switching zero-inflated space-time multinomial models for comparing multiple infectious diseases. *Biostatistics*, 26(1):kxaf034, 2025.

J. Durbin and S. J. Koopman. *Time Series Analysis by State Space Methods*. Oxford University Press, 2012.

P. Fearnhead, D. Wyncoll, and J. Tawn. A sequential smoothing algorithm with linear computational cost. *Biometrika*, 97(2):447–464, 2010.

K. Fokianos, R. Fried, Y. Kharin, and V. Voloshko. Statistical analysis of multivariate discrete-valued time series. *Journal of Multivariate Analysis*, 188:104805, 2022.

C. Fraser. Estimating individual and household reproduction numbers in an emerging epidemic. *PLOS ONE*, 2(8):e758, 2007.

D. T. Frazier, R. Loaiza-Maya, and G. M. Martin. Variational Bayes in state space models: inferential and predictive accuracy. *Journal of Computational and Graphical Statistics*, 32(3):793–804, 2023. ISSN 1537-2715.

D. T. Frazier, R. Loaiza-Maya, G. M. Martin, and B. Koo. Loss-based variational Bayes prediction. *Journal of Computational and Graphical Statistics*, 34(1):84–95, 2025. ISSN 1537-2715.

D. Gamerman. Markov chain Monte Carlo for dynamic generalised linear models. *Biometrika*, 85(1):215–227, 1998.

D. Gamerman and H. S. Migon. Dynamic hierarchical models. *Journal of the Royal Statistical Society: Series B (Methodological)*, 55(3):629–642, 1993.

D. Gamerman, C. A. Abanto-Valle, R. S. Silva, T. G. Martins, R. Davis, S. Holan, R. Lund, and N. Ravishanker. Dynamic Bayesian models for discrete-valued time series. In *Handbook of Discrete-Valued Time Series*, pages 165–186. Chapman and Hall/CRC, 2016.

M. Gatto, E. Bertuzzo, L. Mari, S. Miccoli, L. Carraro, R. Casagrandi, and A. Rinaldo. Spread and dynamics of the COVID-19 epidemic in Italy: effects of emergency containment measures. *Proceedings of the National Academy of Sciences*, 117(19): 10484–10491, 2020.

A. Gelman. Prior distributions for variance parameters in hierarchical models (comment on article by Browne and Draper). *Bayesian Analysis*, 1(3):515–534, 2006.

A. Gelman and D. B. Rubin. Inference from Iterative Simulation Using Multiple Sequences. *Statistical Science*, 7(4):457 – 472, 1992.

A. Gelman, X.-L. Meng, and H. Stern. Posterior predictive assessment of model fitness via realized discrepancies. *Statistica Sinica*, 6(4):733–760, 1996. ISSN 10170405, 19968507.

R. Giordano, T. Broderick, and M. I. Jordan. Covariances, robustness, and variational Bayes. *Journal of Machine Learning Research*, 19(51):1–49, 2018.

P. Giudici, B. Tarantino, and A. Roy. Bayesian time-varying autoregressive models of COVID-19 epidemics. *Biometrical Journal*, 65(1):e2200054, 2023.

N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F (Radar and Signal Processing)*, 140(2):107–113, 1993.

W. D. Green, N. M. Ferguson, and A. Cori. Inferring the reproduction number using the renewal equation in heterogeneous epidemics. *Journal of The Royal Society Interface*, 19(188):20210429, 2022.

A. G. Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90, 1971.

A. G. Hawkes and D. Oakes. A cluster process representation of a self-exciting process. *Journal of Applied Probability*, 11(3):493–503, 1974.

L. Held, M. Höhle, and M. Hofmann. A statistical framework for the analysis of multivariate infectious disease surveillance counts. *Statistical Modelling*, 5(3):187–199, 2005.

M. D. Hoffman and A. Gelman. The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15: 1593–1623, 2014.

M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *Journal of machine learning research*, 2013.

A. J. Holbrook, X. Ji, and M. A. Suchard. Bayesian mitigation of spatial coarsening for a Hawkes model applied to gunfire, wildfire and viral contagion. *arXiv preprint arXiv:2010.02994*, 2020a.

A. J. Holbrook, C. E. Loeffler, S. R. Flaxman, and M. A. Suchard. Scalable Bayesian inference for self-excitatory stochastic processes applied to big American gunfire data. *arXiv preprint arXiv:2005.10123*, 2020b.

J. Huggins, M. Kasprzak, T. Campbell, and T. Broderick. Validated variational inference via practical posterior error bounds. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, 2020.

M. Kim, D. Paini, and R. Jurdak. Modeling stochastic processes in disease spread across a heterogeneous social system. *Proceedings of the National Academy of Sciences*, 116(2):401–406, 2019.

M. Kirchner. Hawkes and INAR(∞) processes. *Stochastic Processes and their Applications*, 126(8):2494–2525, 2016.

M. Kirchner. An estimation procedure for the Hawkes process. *Quantitative Finance*, 17(4):571–595, 2017.

G. Kitagawa. The two-filter formula for smoothing and an implementation of the Gaussian-sum smoother. *Annals of the Institute of Statistical Mathematics*, 46(4): 605–623, 1994.

G. Kitagawa. Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5(1):1–25, 1996.

S. Ko, M. A. Suchard, and A. J. Holbrook. Scaling Hawkes processes to one million COVID-19 cases. *arXiv preprint arXiv:2407.11349*, 2024.

S. Koyama, T. Horie, and S. Shinomoto. Estimating the time-varying reproduction number of COVID-19 with a state-space method. *PLOS Computational Biology*, 17 (1):e1008679, 2021.

L. M. Koyck. *Distributed Lags and Investment Analysis*. North-Holland Pub. Co., Amsterdam, 1954.

D. Lambert. Zero-inflated Poisson regression, with an application to defects in manufacturing. *Technometrics*, 34(1):1–14, 1992.

E. Lasalle and B. Pascal. Joint reproduction number and spatial connectivity structure estimation via graph sparsity-promoting penalized functional. *arXiv preprint arXiv:2509.20034*, 2025.

P. J. Laub, Y. Lee, P. K. Pollett, and T. Taimre. Hawkes models and their applications. *Annual Review of Statistics and Its Application*, 12:233–258, 2025.

T. Liboschik, K. Fokianos, and R. Fried. **tscount**: an R package for analysis of count time series following generalized linear models. *Journal of Statistical Software*, 82 (5):1–51, 2017.

F. Lindsten, A. M. Johansen, C. A. Naesseth, B. Kirkpatrick, T. B. Schön, J. A. D. Aston, and A. Bouchard-Côté. Divide-and-conquer with sequential Monte Carlo. *Journal of Computational and Graphical Statistics*, 26(2):445–458, 2017.

R. Loaiza-Maya and D. Nibbering. Efficient importance variational approximations for state space models. *Journal of Business & Economic Statistics*, 43(4):794–806, 2025.

R. Loaiza-Maya, M. S. Smith, D. J. Nott, and P. J. Danaher. Fast and accurate variational inference for models with many latent variables. *Journal of Econometrics*, 230(2): 339–362, 2022.

Z. J. Madewell, Y. Yang, I. M. Longini, M. E. Halloran, and N. E. Dean. Household transmission of SARS-CoV-2: a systematic review and meta-analysis. *JAMA Network Open*, 3(12):e2031756, 2020.

E. Makalic and D. F. Schmidt. A simple sampler for the horseshoe estimator. *IEEE Signal Processing Letters*, 23(1):179–182, 2016.

A. Mastrototaro and J. Olsson. Online variational sequential Monte Carlo. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *ICML'24*, pages 35039–35062. JMLR.org, 2024.

S. Meyer, L. Held, and M. Höhle. Spatio-temporal analysis of epidemic phenomena using the R package **surveillance**. *Journal of Statistical Software*, 77(11):1–55, 2017.

T. P. Minka. Expectation propagation for approximate Bayesian inference. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, pages 362–369. Morgan Kaufmann Publishers Inc., 2001.

X. Miscouridou, S. Bhatt, G. O. Mohler, S. Flaxman, and S. Mishra. Cox-Hawkes: doubly stochastic spatiotemporal Poisson processes. *ArXiv*, abs/2210.11844, 2022.

C. A. Naesseth, F. Lindsten, and T. B. Schön. High-dimensional filtering using nested sequential Monte Carlo. *IEEE Transactions on Signal Processing*, 67(16):4177–4188, 2019.

R. M. Neal. MCMC using Hamiltonian dynamics. In S. Brooks, A. Gelman, G. Jones, and X.-L. Meng, editors, *Handbook of Markov Chain Monte Carlo*. Chapman and Hall/CRC, 2011.

M. Paul, L. Held, and A. M. Toschke. Multivariate modelling of infectious disease surveillance data. *Statistics in Medicine*, 27(29):6250–6267, 2008.

J. Piironen and A. Vehtari. Sparsity information and regularization in the horseshoe and other shrinkage priors. *Electronic Journal of Statistics*, 11(2):5018–5051, 2017.

M. K. Pitt and N. Shephard. Filtering via simulation: auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–599, 1999.

R. Prado, M. A. R. Ferreira, and M. West. *Time Series: Modeling, Computation and Inference*. Chapman and Hall/CRC, 2nd edition, 2021. ISBN 9781351259422.

M. Quiroz, D. J. Nott, and R. Kohn. Gaussian Variational Approximations for High-dimensional State Space Models. *Bayesian Analysis*, 18(3):989 – 1016, 2023.

R. R. Ravines, A. M. Schmidt, and H. S. Migon. Revisiting distributed lag models through a Bayesian perspective. *Applied Stochastic Models in Business and Industry*, 22(2):193–210, 2006.

B. J. Reich, S. Yang, Y. Guan, A. B. Giffin, M. J. Miller, and A. Rappold. A review of spatial causal inference methods for environmental and epidemiological applications. *International Statistical Review*, 89(3):605–634, 2021.

M.-A. Rizoiu, S. Mishra, Q. Kong, M. Carman, and L. Xie. SIR-Hawkes: linking epidemic models and Hawkes processes to model diffusions in finite populations. In *Proceedings of the 2018 World Wide Web Conference*, pages 419–428, 2018.

T. W. Russell, J. T. Wu, S. Clifford, W. J. Edmunds, A. J. Kucharski, and M. Jit. Effect of internationally imported cases on internal spread of COVID-19: a mathematical modelling study. *The Lancet Public Health*, 6(1):e12–e20, 2021.

T. H. Rydberg and N. Shephard. BIN models for trade-by-trade data. modelling the number of trades in a fixed interval of time. Econometric Society World Congress 2000 Contributed Papers 0740, Econometric Society, 2000.

N. Santitissadeekorn, M. Short, and D. J. B. Lloyd. Influence network reconstruction from discrete time-series of count data modelled by multidimensional Hawkes processes. *Physica D: Nonlinear Phenomena*, 477:134705, 2025.

J. A. Scott, A. Gandy, S. Mishra, J. Unwin, S. Flaxman, and S. Bhatt. ***epidemia**: modeling of epidemics using hierarchical Bayesian models*, 2020. R package version 1.0.0.

L. Shlomovich, E. A. K. Cohen, N. Adams, and L. Patel. A Monte Carlo EM algorithm for the parameter estimation of aggregated Hawkes processes. *arXiv preprint arXiv:2001.07160*, 2020.

R. M. Solow. On a family of lag distributions. *Econometrica*, 28(2):393–406, 1960.

D. Sulem, V. Rivoirard, and J. Rousseau. Scalable and adaptive variational Bayes methods for Hawkes processes. *Journal of Machine Learning Research*, 26(217): 1–102, 2025.

N. Syring and R. Martin. Calibrating general posterior credible regions. *Biometrika*, 106(2):479–486, 2019.

M. Tang and R. Prado. Fast approximate posterior inference for modeling disease dynamics via state-space models. *Computational Statistics & Data Analysis*, 221: 108368, 2026.

H. A. Thompson, A. Mousa, A. Dighe, H. Fu, A. Arnedo-Pena, P. Barrett, J. Bellido-Blasco, Q. Bi, A. Caputi, L. Chaw, L. De Maria, M. Hoffmann, K. Mahapure, K. Ng, J. Raghuram, G. Singh, B. Soman, V. Soriano, F. Valent, L. Vimercati, L. E. Wee, J. Wong, A. C. Ghani, and N. M. Ferguson. Severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) setting-specific transmission rates: a systematic review and meta-analysis. *Clinical Infectious Diseases*, 73(3):e754–e764, 2021.

N. Tomasetti, C. Forbes, and A. Panagiotelis. Updating variational Bayes: fast sequential posterior inference. *Statistics and Computing*, 32(1), 2022. ISSN 1573-1375.

T. K. Tsang, P. Wu, E. H. Y. Lau, and B. J. Cowling. Accounting for imported cases in estimating the time-varying reproductive number of coronavirus disease 2019 in Hong Kong. *The Journal of Infectious Diseases*, 224(5):783–787, 2021.

H. J. T. Unwin, I. Routledge, S. Flaxman, M.-A. Rizoiu, S. Lai, J. Cohen, D. J. Weiss, S. Mishra, and S. Bhatt. Using Hawkes Processes to model imported and local malaria cases in near-elimination settings. *PLoS Computational Biology*, 17(4):e1008830, 2021.

J. Wallinga and P. Teunis. Different epidemic curves for severe acute respiratory syndrome reveal similar impacts of control measures. *American Journal of Epidemiology*, 160(6):509–516, 2004.

M. P. Wand. Fast approximate inference for arbitrarily large semiparametric regression models via message passing. *Journal of the American Statistical Association*, 112 (517):137–168, 2017.

H. Wang, A. Bhattacharya, D. Pati, and Y. Yang. Structured variational inference in Bayesian state-space models. In G. Camps-Valls, F. J. R. Ruiz, and I. Valera, editors, *Proceedings of The 25th International Conference on Artificial Intelligence*

and Statistics, volume 151 of *Proceedings of Machine Learning Research*, pages 8884–8905. PMLR, 2022.

L. J. Welty, R. D. Peng, S. L. Zeger, and F. Domini. Bayesian distributed lag models: estimating effects of particulate matter air pollution on daily mortality. *Biometrics*, 65(1):282–291, 2009.

M. West and P. J. Harrison. *Bayesian forecasting and dynamic models*. Springer, 2 edition, 1997.

M. West, P. J. Harrison, and H. Migon. Dynamic generalized linear models and Bayesian forecasting. *Journal of the American Statistical Association*, 80(389):73–83, 1985.

H. Xuan, L. Maestrini, F. Chen, and C. Grazian. Stochastic variational inference for GARCH models. *Statistics and Computing*, 34(1), 2024. ISSN 1573-1375.

W. Zhang, M. Smith, W. Maneesoonthorn, and R. Loaiza-Maya. Natural gradient hybrid variational inference with application to deep mixed models. *Statistics and Computing*, 34(6), 2024. ISSN 1573-1375.

L. Zhou and G. Papadogeorgou. Bayesian inference for aggregated Hawkes processes. *Bayesian Analysis*, 1(1):1–29, 2025.