Integrated Investment and Policy Planning for Power Systems via Differentiable Scenario Generation

Robert Mieth

Rutgers University

Suggested Citation

R. Mieth, "Integrated Investment and Policy Planning for Power Systems via Differentiable Scenario Generation," in *Proceedings of the PowerUp Conference*, 2026.

BibTeX Entry

```
@inproceedings{mieth2026integratedinvestment,
  author = {Mieth, Robert},
  title = {Integrated Investment and Policy Planning for Power Systems via
      Differentiable Scenario Generation},
  booktitle = {Proceedings of the PowerUp Conference},
  year = {2026},
}
```

Integrated Investment and Policy Planning for Power Systems via Differentiable Scenario Generation

Robert Mieth†

***Abstract*—We formulate a method to co-optimize power system capacity planning decisions and policy investments that shape electricity load patterns. To this end, we leverage a gradient-based solution technique that enables the efficient solution of operation-aware planning models. To compute gradients with respect to the conditions that define daily electricity demand profiles, we introduce and formalize the concept of differentiable scenario generation and show that generative machine learning models satisfy the mathematical requirements needed to compute consistent gradients. We demonstrate the feasibility of the proposed approach through numerical experiments using a diffusion model–based scenario generator and a stylized generation and capacity expansion planning model.**

I. Introduction

Electric load patterns are changing due to ongoing electrification efforts, the deployment of behind-the-meter energy resources, and the adoption of energy management systems [1]. As a result, investments in electric power infrastructure are necessary to ensure that power supply remains secure and affordable. Traditional power system planning approaches consider load patterns as external scenarios that are simulated using predefined socio-technical assumptions, for example, the levels of building electrification or electric vehicle adoption [2]. These properties, however, are not arbitrary but can be influenced by policies like subsidies or tax credits [3]. To date, there exists no method to co-optimize investments in such policies, their costs and their effects on electric load patterns, together with physical investments in generation and transmission assets, increasing the risk of stranded assets. This paper closes this gap via load scenarios created from conditional and differentiable scenario generation models that enable integrated policy and investment planning.

It is clear that grid investments are necessary to accommodate future electric loads [4], [5]. Energy planning studies such as [4], [5] use pre-defined policy and socio-technical assumptions to create scenario sets that determine various investment pathways in generation and transmission. Analyses on investments in policy instruments that may shape these scenario sets exist largely in parallel to the energy planning and modeling community. However, it is also clear that energy technology adoption, and hence load patterns, can be influenced via policy and public investments [3], [6], [7].

The compound effect of policy, technology adoption, consumer behavior, and environmental factors (e.g., temperature) on load patterns is a complex and stochastic process. However, modern computational tools allow simulating multiple thousand devices, households, or buildings and aggregate their (net-)load patterns at the desired spatial and temporal resolution [1], [8]–[11]. Integrating such detailed simulations into energy system planning models *and* optimizing over the simulation parameters, however, is hopeless because the resulting model will be computationally intractable.

Complementary to simulating load patterns, *generative models* have emerged as a means to create time series data that can serve as an input to power and energy models. For example [12], [13] propose conditional generative adversarial models. More recently, the work in [14] proposed the use of diffusion models. Options for energy system data synthesis via generative models were also discussed in a recent EPRI report [15]. Such generative models are typically based on a neural network architecture and are trained via gradient descent algorithms [15]. This means, the generative load models are *differentiable* both in terms of their parameters, but also in terms of their *conditions*, i.e., the input that is given to such models to generate specific scenarios, for example season, day of the week, or temperature. This differentiability opens a pathway to iteratively tune the conditions of the scenario generator with the parameters of a planning model that is also solved via a gradient descent method, e.g., as proposed in [16], [17].

This paper explores this possibility of co-optimizing conditions that shape load patterns and investments in power infrastructure. We make the following contributions:

1) We modify the operation-aware grid planning problem from [16] such that it can co-optimize grid *and* policy decisions that shape load scenarios via a gradient descent solution method.
2) We formalize the concept of differentiable scenario generation and show that the resulting gradients are suitable to solve the target problem.
3) We describe and implement a proof-of-concept pipeline that simulates policy-dependent load scenarios, trains a suitable differentiable scenario generator, and applies these scenarios in a power system planning model.

II. Problem Formulation

We consider a planner with the goal of future-proofing an electric power system. As common in power system planning, the planner seeks to optimize an objective $J(\eta)$ by making investment decisions η within a set $\mathcal{I}$ of feasible investments. Objective J and planning space $\mathcal{I}$ may encode emission targets, production cost targets, electrification targets, or limits on the potential for grid and generation expansion.

Given a planning decision η, the power system will pursue its day-to-day operations. Typically, given a set of available resources and an electricity demand d, the power system operator solves a cost-minimization problem of the form

$$x^*(\eta, d) \in \arg\min_{x} \Big\{ c(x) : x \in \mathcal{X}(\eta, d) \Big\}. \tag{1}$$

†RM is with the Department of Industrial and Systems Engineering at Rutgers University, New Brunswick, USA. robert.mieth@rutgers.edu

Decisions x are the operational decisions, for example generator dispatches. Objective function c captures the cost of running the system, for example generator fuel cost. The feasible space $\mathcal{X}$ is the feasible operational space of the system defined by generator capacities, complex generator constraints (e.g., ramping), power flow, transmission line capacities, and the requirement to balance generation and demand. We note that this *operations problem* (1) may be more complex in practice and entail multi-stage dispatch processes. The general formulation in $\mathcal{X}$ allows for these complexities by defining $\mathcal{X}$ as a set that depends on other optimization problems, but we assume single-stage operations moving forward.

Optimal planning subject to the operations becomes:

$$\min_{\eta\in\mathcal{I}} \quad J(\eta) := \gamma_I^\top \eta + \mathbb{E}_{d\sim\mathbb{D}}\big[h\big(x^*(\eta, d)\big)\big] \tag{2a}$$

$$\text{s.t.} \quad x^* \in \arg\min_x \Big\{c(x) : x\in\mathcal{X}(\eta, d)\Big\}, \tag{2b}$$

where function h maps the operational decisions into the objective of the planner. This allows to capture the possibility that the planner is not strictly interested in cost-minimal dispatches but, for example, also values reliability, certain power flow patterns, or certain technology mixes in the dispatch.

The expectation in (2a) is taken over the possible realizations of (net-)demand d which we assume to follow a distribution $\mathbb{D}$. Further, we model investment cost via linear cost coefficients γ_I. We acknowledge that assuming linear investment costs may be restrictive, especially for discrete ("lumpy") investment decisions. However, we highlight that additional complexity can be added here via piecewise linear functions or higher complexity $\mathcal{I}$. For the purpose and exposition of this paper, linear investment costs are pertinent.

Traditional planning models focus on investments in physical assets, such as siting and sizing of new generation capacity, upgrading and hardening existing assets, and developing new transmission corridors. Commonly, these *tangible* investments η are optimized as a *reaction* to the net-demand distribution $\mathbb{D}$. In other words, future net-demand patterns that inform the optimal choice of η are considered external to the optimal planning model. Such patterns include emerging large loads such as data centers and, more critically, aggregated consumer behavior that results from changes in consumer technology. For example, electric vehicle charging, electric heat pumps, or electrified commercial buildings in combination with any energy management technology such as flexible smart charging or smart building energy management.

It is clear that the economy-wide adoption of such load-shaping technologies is a distributed decision made by the individual consumers. However, this decision *can* be shaped by policy decisions and *intangible investments*, such as tax credits, subsidies, or specialized tariffs, that incentivize or discourage the adoption of certain technologies at the consumer level. We capture such policies in a vector π and make the following subtle, but impactful, alteration to (2) that explicitly models the planner's ability to impact $\mathbb{D}$ via policy:

$$\min_{\eta\in\mathcal{I},\pi\in\mathcal{P}} J(\eta,\pi) := \gamma_I^\top\eta + \gamma_P^\top\pi + \mathbb{E}_{d\sim\mathbb{D}(\pi)}\big[h\big(x^*(\eta, d)\big)\big] \tag{3a}$$

$$\text{s.t. } x^* \in \arg\min_x \Big\{c(x) : x\in\mathcal{X}(\eta, d)\Big\} \tag{3b}$$

where, similar to cost related to η, we also assume a linear policy cost function with coefficients γ_P. The policy space $\mathcal{P}$ captures any constraints on the policy vector π. Most notably, the net-demand distribution $\mathbb{D}(\pi)$ now depends on π, highlighting the impact of policy on demand patterns. In the following section, we will outline how problem (3) can be solved using a first-order gradient descent method.

We note the the problem structure of (3b) is related to problems with decision-dependent uncertainty and in particular resembles *performative prediction* [18]–[21], in particular the idea of using a functional relationship between model predictions and the data distribution [22]. In contrast to [18]–[22], however, we are not interested in predictive accuracy but assume to have knowledge about $\mathbb{D}(\pi)$ as we discuss below.

III. Solution Approach

The problem in (3) cannot be solved directly. We need to deal with the complication of the expected value in objective (3a) and the inner operations problem.

A. Sample average formulation

We first deal with the expectation operator as follows. Assume that we have access to a scenario generator $D_\theta(\pi, z, \epsilon)$ that takes as input the policy vector π, some additional *context* z, for example, a temperature profile, and some random noise ϵ. Vector θ collects the parameters of the scenario generator. In the following we may drop writing some of the explicit dependencies of D on θ, z, and ϵ and mostly write $D(\pi)$. Let $\hat{\mathbb{D}}_\pi^N$ be the empirical distribution supported by N samples generated from $D(\pi)$. For a sufficiently high number of samples N the scenario generator should achieve $\hat{\mathbb{D}}_\pi^N \approx \mathbb{D}(\pi)$. Given such a generator we denote $d_\pi^{(i)}$ as a scenario drawn from $D(\pi)$ and write the sample average version of (3) as:

$$\min_{\eta\in\mathcal{I},\pi\in\mathcal{P}} \widehat{J}(\eta,\pi) := \gamma_I^\top\eta + \gamma_P^\top\pi + \frac{1}{N}\sum_{i=1}^N h\big(x^*(\eta, d_\pi^{(i)})\big) \tag{4a}$$

$$\text{s.t. } x^*(\eta, d_\pi^{(i)}) \in \arg\min_x \Big\{c(x) : x\in\mathcal{X}\big(\eta, d_\pi^{(i)}\big)\Big\}. \tag{4b}$$

B. Gradient Computations

Next, we tackle the bilevel structure that remains in (4) by leveraging established results from implicit differentiation over the inner operations problem to formulate a stochastic gradient descent algorithm. Specifically, we can write the gradients of the integrated investment and policy objective of (4) as:

$$\nabla_\eta J(\eta,\pi) = \gamma_I + \frac{1}{N}\sum_{i=1}^N \big(\nabla_\eta x_i^* \cdot \nabla_x h\big) \tag{5}$$

$$\nabla_\pi J(\eta,\pi) = \gamma_P + \frac{1}{N}\sum_{i=1}^N \big(\nabla_\pi d^{(i)} \cdot \nabla_d x_i^* \cdot \nabla_x h\big), \tag{6}$$

where we write x_i^* as shorthand for $x^*(\eta, d_\pi^{(i)})$. The gradient $\nabla_x h$ is straightforward to compute assuming that h is differentiable. The Jacobian $\nabla_\eta x_i^*$ can be computed using modern approaches from implicit function differentiation. Implicit differentiation has been discussed comprehensively, e.g., in

[23], [24]. Energy applications are discussed, e.g., in [16], [25]. We refer the reader to these references for details on the theory and only repeat the intuition of how to obtain $\nabla_\eta x_i^*$ here. In fact, a formulation for $\nabla_\eta x_i^*$ from an optimal primal-dual solution of the operations problem can be derived from the problem's KKT system using classic results from sensitivity analysis [26]. The strength of modern differentiable optimization comes from the realization that instead of first computing $\nabla_\eta x_i^*$ and *then* multiplying it with $\nabla_x h$, we can directly solve the KKT system for $(\nabla_\eta x_i^* \cdot \nabla_x h)$. This way, it is possible to exploit the structure of the KKT system to simplify obtaining necessary derivatives by backpropagating through the problem solution via automatic differentiation [23], [24].

We note that the effectiveness of implicit differentiation depends on the properties of the inner operations problem. Usually, strongly convex problems like quadratic or conic problems work best as they provide unique optimal solutions. However, our experiments with linear c and $\mathcal{X}$ were successful.

For (6), we need the Jacobian $\nabla_d x_i^*$. Generally, this Jacobian will be easier to compute than $\nabla_\eta x_i^*$ because the net-demand usually appears on the right-hand side of the constraints. This allows re-using the KKT factorization from the optimal solution [27]. As a result, right-hand side derivatives can be computed virtually for free from an optimal primal-dual solution. If, however, d also appears as a parameter in the constraint matrix, the same argument holds as for $\nabla_\eta x_i^*$.

Finally, computing a gradient $\nabla_\pi d$ implies a functional relationship between π and d. So far, we have only established this relationship for the simulator $D(\pi)$. To close the connection between the demand simulator and the differentiability of a load scenario $d_\pi^{(i)}$ we introduce the following definition:

Definition 1 (Differentiable Scenario Generator)**.** *We call a function $D_\theta(\pi, z, \epsilon)$ a* differentiable scenario generator *if it has the following properties.*

1) *Let $\pi^\star$ and $z^\star$ be given fixed parameters and let ϵ be a random variable ϵ with fixed distribution $\mathcal{Q}$, e.g., $\mathcal{Q} = \mathcal{N}(0, I)$. Further, for a set of samples $\{\epsilon^{(i)}\}_{i=1}^N$ from ϵ, write $d_{\pi^\star}^{(i)} = D_\theta(\pi^\star, z^\star, \epsilon^{(i)})$ and let $\{d_{\pi^\star}^{(i)}\}$ be the set of corresponding scenarios. For sufficiently large N the empirical distribution $\hat{\mathbb{D}}_{\pi^\star}^N := \frac{1}{N}\sum_{i=1}^N \delta_{d_{\pi^\star}^{(i)}}$ achieves $\hat{\mathbb{D}}_{\pi^\star}^N \approx \mathbb{D}(\pi^\star)$, i.e., similarity to a target distribution.*
2) *For a given fixed sample $\epsilon^\star$ of ϵ the gradient $\nabla_{\pi^\star} D_\theta(\pi^\star, z^\star, \epsilon^\star)$ exists and is computable.*

We define $\nabla_\pi d_\pi^{(i)} := \nabla_\pi D_\theta(\pi, z, \epsilon^{(i)})$ for a given sample $\epsilon^{(i)}$.

We note that Definition 1 implies that, besides the random input ϵ, the scenario generator is a deterministic function, i.e., for fixed $\pi^\star$ and $z^\star$ we have that $D(\pi^\star, z^\star, \epsilon) = D(\pi^\star, z^\star, \epsilon')$ for any $\epsilon = \epsilon'$. These properties are common in modern generative models as we show and discuss in Section IV below.

C. Stochastic Gradients

We first argue that taking the scenario sample gradients is indeed a valid estimator for the change of expected cost with regard to the distributional shift resulting from a change in π.

Proposition 1. *Write $f(d) = h(x(\eta, d))$ as a shorthand. For a given differentiable scenario generator $D(\pi, \epsilon)$ (as per Definition 1) the gradient $\frac{1}{N}\sum_{i=1}^N \nabla_\pi f\big(D(\pi, \epsilon^{(i)})\big)$ is an unbiased estimator of $\nabla_\pi \mathbb{E}_{d\sim\mathbb{D}(\pi)}\big[f(d)\big]$.*

Proof. The proof follows the argument from [28, Sec. 5.2] and requires $D(\pi, \epsilon)$ to be differentiable or, to use standard machine learning language, have a *differentiable sampling path*. This assumption holds as per Definition 1. We introduce $\omega(x, \pi)$ as the density function of $\mathbb{D}(\pi)$ and for this proof use the symbol x for scenarios sampled from $D(\pi, \epsilon)$ to avoid confusion with the differential operator and to remain consistent with most literature on generative models. Further, we write $q(\epsilon)$ as the density function of the distribution $\mathcal{Q}$ of the random variable ϵ. We can now write

$$\begin{aligned}\nabla_\pi \mathbb{E}_{x\sim\mathbb{D}(\pi)}\big[f(x)\big] &= \nabla_\pi \int \omega(x, \pi) f(x) dx \\ &= \nabla_\pi \int q(\epsilon) f\big(D(\pi, \epsilon)\big) d\epsilon \\ &= \mathbb{E}_{\epsilon\sim\mathcal{Q}}\Big[\nabla_\pi f\big(D(\pi, \epsilon)\big)\Big].\end{aligned}$$

It is clear that the final expectation can be estimated through $\frac{1}{N}\sum_{i=1}^N \nabla_\pi f\big(D(\pi, \epsilon^{(i)})\big)$ which is exactly the structure required for the gradient in (6). $\square$

D. Gradient descent algorithm

With the gradients available, we can solve (4) using a stochastic projected gradient descent approach [16]. For an iteration counter τ and a batch of N_{B} samples we can write the stochastic estimate of the gradients from (5) and (6) as

$$\Delta_\eta^{(\tau)} = \gamma_I + \frac{1}{N_{\mathrm{B}}}\sum_{i=1}^{N_{\mathrm{B}}} \big(\nabla_\eta x_i^* \cdot \nabla_x h\big) \tag{7}$$

$$\Delta_\pi^{(\tau)} = \gamma_P + \frac{1}{N_{\mathrm{B}}}\sum_{i=1}^{N_{\mathrm{B}}} \big(\nabla_\pi d^{(i)} \cdot \nabla_d x_i^* \cdot \nabla_x h\big). \tag{8}$$

For a given learning rate λ, we can perform the gradient steps

$$\eta^{(\tau+1)} = \mathrm{proj}_{\mathcal{I}}\left(\eta^{(\tau)} - \lambda\Delta_\eta^{(\tau)}\right) \tag{9}$$

$$\pi^{(\tau+1)} = \mathrm{proj}_{\mathcal{P}}\left(\pi^{(\tau)} - \lambda\Delta_\pi^{(\tau)}\right), \tag{10}$$

until a convergence criterion has been reached. The operator $\mathrm{proj}_{\mathcal{X}}$ denotes the projection onto set $\mathcal{X}$. The projection ensures feasibility of the investments in each step.

IV. Numerical Experiments

We conduct numerical experiments to establish a proof-of-concept for using differentiable scenario generation. For this purpose we study the dependence of daily load patterns as a function of the policy vector $\pi = (\pi_{\mathrm{adopt}}^{\mathrm{EV}}, \pi_{\mathrm{flex}}^{\mathrm{EV}}, \pi_{\mathrm{adopt}}^{\mathrm{HP}}, \pi_{\mathrm{eff}}^{\mathrm{HP}})$:

- $\pi_{\mathrm{adopt}}^{\mathrm{EV}} \in [0, 1]$ describes the level of EV adoption and resulting charging demand.
- $\pi_{\mathrm{flex}}^{\mathrm{EV}} \in [0, 1]$ describes the level of adoption of EV smart charging and flexibility technologies and captures if EV charging demand is shifted away from peak demand hours.
- $\pi_{\mathrm{adopt}}^{\mathrm{HP}} \in [0, 1]$ describes the level of heat pump (HP) adoption and building electrification. It captures the sensitivity of the load to colder temperatures alongside a general shift of base and peak demand.

- $\pi^{\text{HP}}_{\text{eff}} \in [0, 1]$ describes the level of building efficiency and reduces direct temperature sensitivity.

A. Simulation setup

The objective of the simulation is to create a set of sample tuples $\mathcal{S}_M = \{(d^{(m)}, \pi^{(m}, z^{(m)})\}_{m=1}^{M}$ that we can use to train the parameters θ of scenario generator $D_\theta(\pi, z, \epsilon)$. This simulator can be arbitrarily complex and does not have to meet any requirements on differentiability. Detailed models that simulate individual loads and their behavioral, contextual (e.g., temperature-dependent), and otherwise stochastic patterns are available and well-studied [1], [8]–[11].

For the purpose of this paper we created a simple scenario generator that qualitatively captures the EV and heat-pump driven load patterns reported in [1]. As a data basis, we use three years (from 2023, 2024, 2025) of daily 24h load patterns obtained from PJM [29] alongside hourly temperature data at the center of the respective PJM zone. We obtained temperature data from the Copernicus ERA5 reanalysis data [30]. We call the real-world load data *base load*.

We created a total of $M = 10,000$ training samples. For each $m \in \{1, ..., M\}$ we selected a random base load alongside its corresponding temperature time series. Next, we sampled a random $\pi^{(m)}$ by drawing uniformly from the unit hypercube $[0, 1]^4$. For the random policy $\pi^{(m)}$ and the temperature profile, which we capture in the context vector $z^{(m)}$, we compute the policy-dependent load profile as follows.

1) EV adoption ($\pi^{\text{EV}}_{\text{adopt}}$): Increasing EV adoption will add peak load to the base load of up to 75% of the current yearly peak load [1]. We simulate this by adding bell curve-shaped load on top of the base load profile centered at the 18-th hour of the day with a standard deviation of 1h. The height of the bell curve is scaled continuously between 0 ($\pi^{\text{EV}}_{\text{adopt}} = 0$) and 0.75 ($\pi^{\text{EV}}_{\text{adopt}} = 1$), given relative to the yearly peak load. For each sample the exact center of the peak and the exact standard deviation of the bell curve is further distorted by multiplication with random noise with standard deviation 0.05 and 0.1, respectively. In addition, $\pi^{\text{EV}}_{\text{adopt}}$ creates additional uniform peak and off-peak load drawn from $\mathcal{N}(0.1, 0.01)$.

2) EV flexibility ($\pi^{\text{EV}}_{\text{flex}}$): Increased EV flexibility widens the bell curve that models EV load and moves the peak load occurrence away from the peak at the 18-th hour. At $\pi^{\text{EV}}_{\text{flex}} = 1$ the standard deviation of the added load is widened to 4h and the peak is moved at most 5h from the 18-th hour. The same random noise factor as described above applies to the flexibility-scaled load. We note that any load addition that would occur past midnight is applied to the morning hours of the same day to ensure consistency.

3) HP Adoption ($\pi^{\text{HP}}_{\text{adopt}}$): HP adoption adds load as a function of the temperature profile up to an additional peak load of 2 relative to the yearly peak load [1]. We compute the electricity demand from heating for each hour of the day by taking the weighted average of the hourly temperature (80% weight) and the daily mean temperature (20% weight). The heat demand signal is then computed to rise smoothly whenever the temperature signal T_t for each hour t falls below $T_{\text{th}} = 19°\text{C}$ via $\tau \log\left(1 + \exp\left(\frac{T_{\text{th}} - T_t}{\tau}\right)\right)$, with $\tau = 1.3$. The resulting heating demand is then scaled with the level of HP adoption $\pi^{\text{HP}}_{\text{adopt}}$. We introduce random behavior by drawing a random smoothing factor from $\mathcal{N}(0.05, 0.07)$. The smoothing factor computes the convex combination of the hourly heat demand signal and its daily mean. If the smoothing factor is equal to 1, then the signal is equal to its mean.

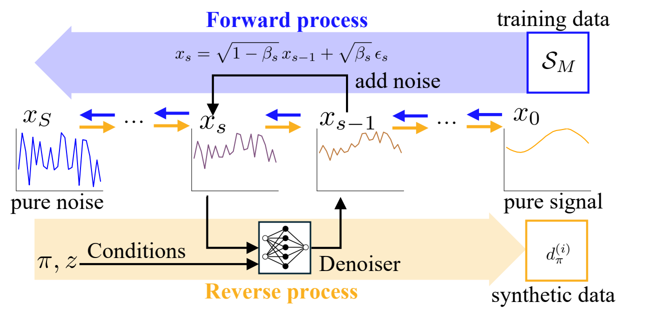

Fig. 1: Overview of the forward and reverse diffusion processes. The forward process adds noise over S diffusion steps on the available training data. A denoising model is trained to learn how to remove noise from a given noisy signal using given additional conditions and the current step index s.

4) Building efficiency ($\pi^{\text{HP}}_{\text{eff}}$): Increasing building efficiency reduces peak demand from heating and shifts the demand dependency from hourly temperature to daily mean temperature. We model this by continuously increasing the mean and standard deviation of the heat demand smoothing factor up to 0.8 and 0.3, respectively, for $\pi^{\text{HP}}_{\text{eff}} = 1$.

Fig. 2 shows some exemplary simulations for various settings of π. We highlight that the central purpose of this simulation setup is *not* to realistically model policy-dependent demand behavior. Better tools are available and should be used for that. The purpose of our simulation setup was to create a low-complexity but still meaningful scenario generator to test the feasibility of differentiable scenario generation.

B. Differentiable scenario generator

We use a diffusion model (DM) as an effective way to train a differentiable scenario generator. We outline the relevant mechanics of the DM below and also refer to [14], [31] for more details on using DMs for time series synthesis.

The fundamental idea of a DM is to train a *denoiser model* to effectively remove noise from a noisy signal until all that is left is signal. See Fig. 1 for an overview. The noise removal is step-wise over a number of $s = 0, ..., S$ steps. To this end, a *forward process* corrupts a given signal x_s via $x_s = \sqrt{1 - \beta_s} x_{s-1} + \sqrt{\beta_s} \epsilon_s$, where x_0 is the pure starting signal, x_S is the final corrupted signal close to pure noise, and ϵ_s is random gaussian noise. We note that for the discussion in this section we also use symbol x for the time series vector to remain in line with standard DM language. Parameters β_s define the diffusion *schedule* and are chosen such that noise is not added too quickly in training.

In the reverse process, the denoiser model takes newly sampled noise ϵ as an input, alongside conditions π, z, and information on the step number s in the denoising process. After S applications of the denoiser model on the noisy signal, the final result is a synthetic data sample. The reverse process is the same for training and for sampling new scenarios. Algorithms 1 and 2 in Appendix A detail the training and sampling procedures, respectively. We highlight that our target scenario generator $D_\theta(\pi, z, \epsilon)$ is the *entire* sampling process as shown in Algorithm 2. This involves applying the trained denoiser model S times. In our case study, we use a multi-layer perceptron with a single hidden layer of width 128.

Because our goal is to effectively compute gradients over the sampling condition π (i.e., $\nabla_\pi d_\pi$ as discussed in Section III-B above), we slightly depart from traditional DM design where new noise is added in each reverse step s. Specifically, we adopt the approach from [32] where the reverse process is a deterministic path from the initial noise sample (x_S) to the final target scenario (x_0). This way, we can compute well-behaved gradients $\nabla_\pi d_\pi$ and, as a side-effect, sample faster.

Finally, we note that for our case study we trained the DM on load residuals. This means in addition to the temperature profile, context vector z also contains a base load profile. The DM then learns how this base load profile is altered to reflect the policy-induced patterns. Because we are simulating the load patterns using real-world temperature and load data, this approach was pertinent and provided much better results in the scenario generation than directly learning the load patterns. We also highlight that all data was normalized before training and needs to be de-normalized after data generation. This is a standard approach in machine learning pipelines.

C. Differentiable scenarios

Fig. 2 shows an overview of some exemplary scenarios, both *simulated* with the simulation setup described in Section IV-A and *generated* with the scenario DM-based scenario generator described in Section IV-B. All plots in Fig. 2(a)–(f) have been created for the same day, i.e., the same baseline load and temperature profile. The temperature profile is plotted in Fig. 4. Each subplot shows the simulated and generated scenarios for a different target policy vector. We observe that the generated profiles closely match the simulated samples. We note that the scenarios in Fig. 2 are "out-of-sample" in the sense that both the simulation and the scenario generation was performed after training with new inputs and new noise.

Figs. 3 and 4 show the gradient of the scenarios with respect to a subset of the components of the policy vector π. We observe that the gradients obtained from this single sampling instance correctly reflect how load is reshaped through π in the simulations. For example, in Fig. 3 we see that an increase of EV adoption ($\pi^{\text{EV}}_{\text{adopt}}$) will increase peak load at the 18-th hour and increase ramping steepness in the hours leading up to the peak hour. Likewise, an increase of EV flexibility ($\pi^{\text{EV}}_{\text{flex}}$) will decrease the load at the peak hour and move load towards the off-peak night hours. The analysis of the gradients related to HP adoption and building efficiency ($\pi^{\text{HP}}_{\text{adopt}}$, $\pi^{\text{HP}}_{\text{eff}}$) in Fig. 4 correctly captures the opposing effect of these two parameters on the temperature sensitivity of the load. We highlight again that this information is completely encoded in a single scenario $d^{(i)}_\pi$, i.e., in a single parametrization of $D_\theta(\pi, z, \epsilon)$.

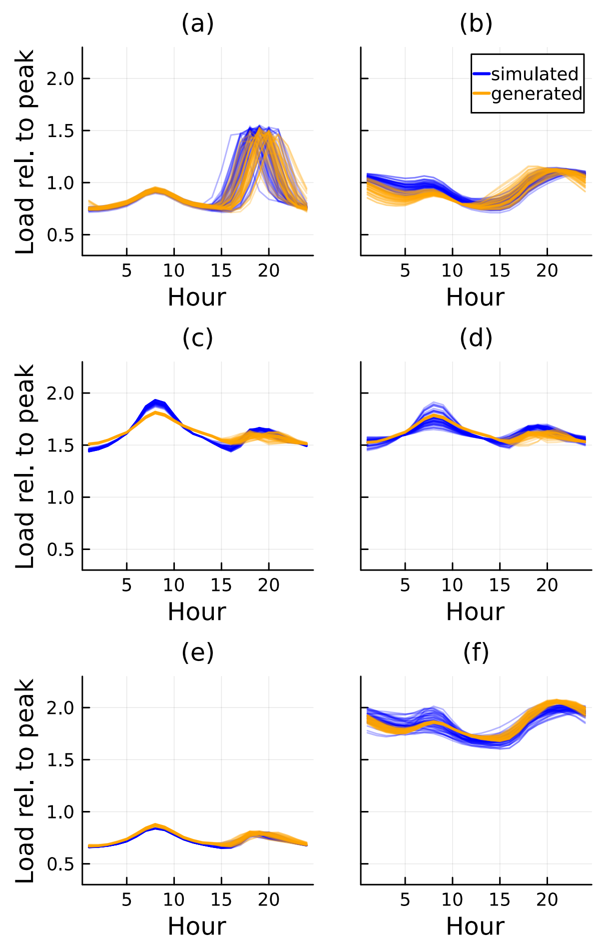

Fig. 2: Simulated and generated scenarios for the same day (i.e., for a given load baseline and winter temperature profile as in Fig. 4) for different policy vectors ($\cdot = 0$). (a) $\pi = [1, \cdot, \cdot, \cdot]$ (max. EV adoption). (b) $\pi = [1, 1, \cdot, \cdot]$ (max. EV adoption with max. flexibility). (c) $\pi = [\cdot, \cdot, \cdot, 1]$ (max. HP adoption). (d) $\pi = [\cdot, \cdot, 1, 1]$ (max. HP adoption with max. building efficiency). (e) $\pi = [\cdot, \cdot, \cdot, \cdot]$ (baseline). (f) $\pi = [1, 1, 1, 1]$ (max. electrification with max. flexibility and efficiency).

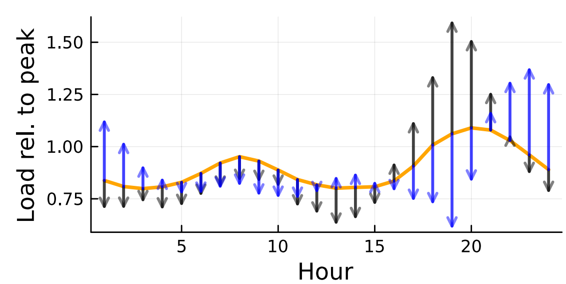

Fig. 3: One generated scenario for $\pi = [0.5, 0.5, 0.1, 0.1]$ and the gradients corresponding to the components $\pi^{\text{EV}}_{\text{adopt}}$ (black arrows) and $\pi^{\text{EV}}_{\text{flex}}$ (blue arrows). The scenario gradient correctly reflects the simulated behavior that an increase of $\pi^{\text{EV}}_{\text{adopt}}$ will increase peak demand while an increase of $\pi^{\text{EV}}_{\text{flex}}$ will reduce peak demand and shift demand towards the night.

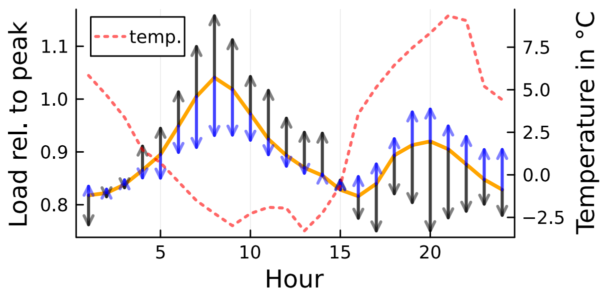

Fig. 4: One generated scenario for $\pi = [0.1, 0.1, 0.5, 0.5]$ and the gradients corresponding to the components $\pi^{\mathrm{HP}}_{\mathrm{adopt}}$ (black arrows) and $\pi^{\mathrm{eff}}_{\mathrm{HP}}$ (blue arrows) alongside the temperature profile of that day (red dashed line). The scenario gradient correctly captures the simulated opposing impact of $\pi^{\mathrm{HP}}_{\mathrm{adopt}}$ and $\pi^{\mathrm{eff}}_{\mathrm{HP}}$ on the temperature dependency of the load.

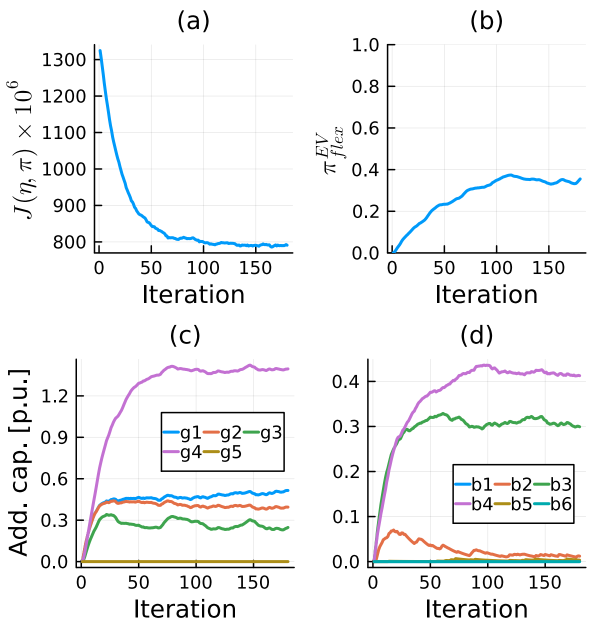

Fig. 5: Results from solving the planning model in (11) via gradient descent (Sec. III-D). Plots show the trajectories of (a) the planning objective, (b) policy $\pi^{\mathrm{EV}}_{\mathrm{flex}}$, (c) generator capacity additions, and (d) transmission (branch) capacity additions.

D. Planning application

We now use our trained scenario generator in a stylized capacity expansion planning model. The goal is to accommodate 100% EV adoption ($\pi^{\mathrm{EV}}_{\mathrm{adopt}} = 1$), which requires generation and transmission capacity investments due to the resulting load increase. To this end, the decision maker balances investments in generation capacity and transmission line upgrades and investments in enabling EV flexibility (given by $\pi^{\mathrm{EV}}_{\mathrm{flex}}$). Model (11) in Appendix B shows the detailed formulation.

We use the PJM 5 bus model with the same parametrization as in [33] based on the MATPOWER `case5` data set [34]. We use the single-period load vector from the dataset and multiply it with the generated load scenarios scaled to the interval $[0, 1.3]$. The operation model is run for 24-hour scenarios in batches of 5 days sampled randomly. If the available generation and transmission capacities prohibit a feasible load-serving dispatch, generation and transmission constraints are relaxed at a high penalty. By investing in additional capacity, the model can avoid this penalty. We set annualized cost of generation and transmission upgrades of \$1m/MW. Achieving $\pi^{\mathrm{EV}}_{\mathrm{flex}} = 1$ costs \$1,100m. We set the learning rate to $\lambda = 10^{-9}$ and initial investments and the initial $\pi^{\mathrm{EV}}_{\mathrm{flex}}$ to zero.

Fig. 5 shows the results. The model converges after 180 iterations in under one minute. Fig. 5(a) shows the planning objective. The final optimal level of $\pi^{\mathrm{EV}}_{\mathrm{flex}}$ is 36% as per Fig. 5(b). Figs. 5(c) and (d) show the resulting additional capacity investments for each generator (g1–g5) and each transmission branch (b1–b6).

E. Implementation and code availability

We implemented all necessary code for our experiments in Julia. For the DM and its training algorithm we used the `Flux.jl` [35] package. For implementing and solving the optimization problem we used the `JuMP` [36] package and for differentiation of the optimization model we used the `DiffOpt.jl` [37] package. All computations were performed on an Apple MacBook Pro with an Apple M3 Pro processor and 36 GB of memory.

Our data and code are available open-source at:

[*github.com/ropes-lab/differentiable_scenario_panning*].

V. Conclusion and Discussion

This paper introduced differentiable load scenarios to co-optimize load shape (e.g., determined by policy) and other planning decisions (e.g., capacity investments) and demonstrated the feasibility of the concept. To this end, we formulated a suitable planning problem, formalized the concept of a differentiable scenario generation, and discussed necessary mathematical properties. Our numerical experiments showcased a lightweight simulation and conditional scenario generation pipeline using a diffusion model architecture. The scenarios *and* their resulting gradients showed the expected behavior and recovered the complex underlying dependencies between scenario conditions and the load shape itself. Finally, we applied the concept to a stylized capacity expansion planning problem and showed the feasibility of the approach.

We highlight again that this work is a proof-of-concept. More research is needed to develop a framework that allows application to quantitative planning models. First, a more detailed simulator should be used. Such simulators are available but still require a carefully crafted data pipeline to lend themselves to training data generation. Second, our proposed scenario generator is low complexity. Generative models with architectures that are more carefully tailored for time series have been discussed in the literature [31]. A systematic study on the tradeoff between scenario accuracy and differentiability for different model architectures is needed. Finally, more research on scalability is required. Our case study solved quickly and was implemented without meaningful optimization or parallel computation and gradient-based planning models have been shown to scale [16]. However, more experiments and research are needed to identify optimally scalable formulations and implementations of our approach.

APPENDIX

A. Training and sampling algorithms

Algorithms 1 and 2 show details on the DM training and the sampling procedure.

Algorithm 1: Training process

Require: Training dataset $\mathcal{S}_M$, learning rate λ, schedule $\{\beta_s\}_{s=1}^S$, batch size N_B
1: Define $\alpha_s = 1 - \beta_s$ and $\bar{\alpha}_s = \prod_{j=1}^s \alpha_j$
2: Initialize denoiser parameters θ
3: **while** not converged **do**
4: Sample minibatch $\{(x^{(i)}, \pi^{(i)}, z^{(i)})\}_{i=1}^B$ from $\mathcal{S}_M$
5: **for** $i = 1, \dots, N_\mathrm{B}$ **do**
6: Sample random diffusion step $s_i \sim \mathcal{U}\{1, ..., S\}$
7: Sample noise $\epsilon_i \sim \mathcal{N}(0, I)$
8: Forward diffuse: $x_{s_i}^{(i)} = \sqrt{\bar{\alpha}_{s_i}}\, x_0^{(i)} + \sqrt{1-\bar{\alpha}_{s_i}}\, \varepsilon_i$
9: Predict signal: $\hat{x}_{0,\theta}^{(i)} = \mu_\theta\left(x_{s_i}^{(i)}, \pi^{(i)}, z^{(i)}, s_i\right)$
10: **end for**
11: Compute minibatch loss: $\mathcal{L}(\theta) = \frac{1}{B}\sum_{i=1}^B \left\| x_0^{(i)} - \hat{x}_{0,\theta}^{(i)} \right\|_2^2$
12: Update parameters: $\theta \leftarrow \theta - \lambda \nabla_\theta \mathcal{L}(\theta)$
13: **end while**
14: **return** trained denoiser μ_θ

Algorithm 2: Sampling process with fixed noise

Require: Trained denoiser μ_θ, conditions (π, z), noise schedule $\{\beta_s\}_{s=1}^S$
1: Define $\alpha_s = 1 - \beta_s$ and $\bar{\alpha}_s = \prod_{j=1}^s \alpha_j$
2: Draw noise ϵ where $\epsilon \sim \mathcal{N}(0, I)$
3: Initialize $x_S \leftarrow \epsilon$
4: **for** $s = S, S-1, \dots, 1$ **do**
5: Reverse process: $\hat{x}_0 = \mu_\theta(x_s, \pi, z, s)$
6: Compute reverse update: $x_{s-1} = \frac{1}{\sqrt{\alpha_s}}\left(x_s - \frac{1-\alpha_s}{\sqrt{1-\bar{\alpha}_s}}(x_s - \sqrt{\bar{\alpha}_s}\,\hat{x}_0)\right)$
7: **end for**
8: **return** x_0

B. Detailed planning model

The planning model that we used for the experiments in Section IV-D follows the form of (4). In detail:

$$\min_{\eta_\mathrm{G}, \eta_L, \pi_\mathrm{flex}^\mathrm{EV}} \gamma_\mathrm{I,G}^\top \eta_\mathrm{G} + \gamma_\mathrm{I,L}^\top \eta_\mathrm{L} + \gamma_\mathrm{P}(\pi_\mathrm{flex}^\mathrm{EV}) + 365\frac{1}{N}\sum_{i=1}^N c(x_i^*) \tag{11a}$$

$$\text{s.t. } 0 \le \pi_\mathrm{flex}^\mathrm{EV} \le 1 \tag{11b}$$

$$\eta^G, \eta^L \ge 0 \tag{11c}$$

$$x_i^* \in \Big\{ \arg\min_x \sum_{t=1}^{24} \left(c^\top p_t + \rho_\mathrm{G}^\top s_\mathrm{G,t} + \rho_\mathrm{F}^\top s_\mathrm{F,t}\right) \tag{11d}$$

$$\mathbb{1}^\top p_t = \mathbb{1}^\top d_{\pi,t}^{(i)} \quad \forall t \tag{11e}$$

$$p_t \le p^\mathrm{max} + \eta_\mathrm{G} + s_{G,t} \quad \forall t \tag{11f}$$

$$f_t = B(p_t - d_{\pi,t}^{(i)}) \quad \forall t \tag{11g}$$

$$|f_t| \le f^\mathrm{max} + \eta_\mathrm{L} + s_{F,t} \quad \forall t \Big\}. \tag{11h}$$

Coefficients $\gamma_{I,G}$, and $\gamma_{I,L}$ are the cost for upgrading generation and transmission capacity. Cost γ_P capture the necessary investments to enable EV flexibility. Operational costs are taken as samples over operation days and scaled up to be comparable with investment cost which are computed as annual payments. Constraints (11b) and (11c) are basic limits on the investments. The objective of the operations problem (11d) is to minimize the cost of generation plus the cost of exceeding generation and transmission capacities. Cost of generation and penalties for constraint violations are given by c and ρ_G, ρ_F respectively. For each time step t variables p_t, $s_\mathrm{G,t}$, $s_\mathrm{F,t}$ collect generator production and constraint exceedance. Constraint (11e) enforces the balance of production and demand at each time step. Constraint (11f) softly enforces generation capacity limits for each generator which are given by the base capacities p^max and the investment decision η_G. Equation (11g) computes power flow through a linear mapping (DC power flow via PTDF matrix B). Finally, constraint (11h) enforces the soft constraint on the power flow limit given by the base capacities f^max and the investment decision η_L

AI DISCLOSURE

During the preparation of this work, the authors used generative AI tools to support some coding implementation and grammar and spelling. After any AI use, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article. No AI-generated content or code was used "as is" and no AI has been used on the reference section.

REFERENCES

[1] C. Crozier, K. Baker, K. J. Kircher, *et al.*, "Distribution grids may be a barrier to residential electrification," *Cell Reports Sustainability*, vol. 2, no. 11, 2025.

[2] N. E. Koltsaklis and A. S. Dagoumas, "State-of-the-art generation expansion planning: A review," *Applied energy*, vol. 230, pp. 563–589, 2018.

[3] S.-L. Penttinen, "Regulatory and policy instruments to promote decarbonization in the energy sector," in *Research Handbook on Climate Change Mitigation Law*, pp. 338–362, Edward Elgar Publishing, 2022.

[4] W. J. Cole, D. Greer, P. Denholm, A. W. Frazier, S. Machen, T. Mai, N. Vincent, and S. F. Baldwin, "Quantifying the challenge of reaching a 100% renewable energy power system for the united states," *Joule*, vol. 5, no. 7, pp. 1732–1748, 2021.

[5] J. D. Jenkins, E. N. Mayfield, E. D. Larson, S. W. Pacala, and C. Greig, "Mission net-zero america: The nation-building path to a prosperous, net-zero emissions economy," *Joule*, vol. 5, no. 11, pp. 2755–2761, 2021.

[6] S. Nadel, "Electrification in the transportation, buildings, and industrial sectors: a review of opportunities, barriers, and policies," *Current Sustainable/Renewable Energy Reports*, vol. 6, no. 4, pp. 158–168, 2019.

[7] J. Li, J. Jiao, and Y. Tang, "An evolutionary analysis on the effect of government policies on electric vehicle diffusion in complex network," *Energy policy*, vol. 129, pp. 1–12, 2019.

[8] T. T. Mai, P. Jadun, J. S. Logan, C. A. McMillan, M. Muratori, D. C. Steinberg, L. J. Vimmerstedt, B. Haley, R. Jones, and B. Nelson, "Electrification futures study: Scenarios of electric technology adoption and power consumption for the united states," tech. rep., National Renewable Energy Lab.(NREL), Golden, CO (United States), 2018.

[9] P. Priyadarshan, E. Pergantis, C. Crozier, K. Baker, and K. Kircher, "EDGEi: A simulation test-bed for investigating the impacts of building and vehicle electrification on distribution grids," in *Hawaii International Conference on System Sciences (HICSS-57)*, 2024.

[10] L. Earle, J. Maguire, P. Munankarmi, and D. Roberts, "The impact of energy-efficiency upgrades and other distributed energy resources on a residential neighborhood-scale electrification retrofit," *Applied Energy*, vol. 329, p. 120256, 2023.

[11] S. D. Nazemi, M. A. Jafari, and A. Matta, "Active inference for energy control and planning in smart buildings and communities," in *2025 IEEE 21st International Conference on Automation Science and Engineering (CASE)*, pp. 2369–2375, IEEE, 2025.

[12] Y. Chen, Y. Wang, D. Kirschen, and B. Zhang, "Model-free renewable scenario generation using generative adversarial networks," *IEEE Transactions on Power Systems*, vol. 33, no. 3, pp. 3265–3275, 2018.

[13] Z. Liang, R. Mieth, and Y. Dvorkin, "Operation-adversarial scenario generation," *Electric Power Systems Research*, vol. 212, p. 108451, 2022.

[14] N. Lin, P. Palensky, and P. P. Vergara, "Energydiff: Universal time-series energy data generation using diffusion models," *IEEE Transactions on Smart Grid*, 2025.

[15] Electric Power Research Institute, "Synthetic customer load data: Approaches, limitations, and the role of generative ai," White Paper 3002034348, Electric Power Research Institute (EPRI), Jan. 2026.

[16] A. Degleris, A. El Gamal, and R. Rajagopal, "Gradient methods for bilevel electricity grid expansion planning," *Applied Energy*, vol. 418, p. 128044, 2026.

[17] M. Ghazanfariharandi and R. Mieth, "Uncertainty-aware power system planning via gradient descent," *arXiv preprint arXiv:2606.27602*, 2026.

[18] J. Perdomo, T. Zrnic, C. Mendler-Dünner, and M. Hardt, "Performative prediction," in *International Conference on Machine Learning*, pp. 7599–7609, PMLR, 2020.

[19] M. Hardt and C. Mendler-Dünner, "Performative prediction: Past and future," *Statistical Science*, vol. 40, no. 3, pp. 417–436, 2025.

[20] G. Piliouras and F.-Y. Yu, "Multi-agent performative prediction: From global stability and optimality to chaos," in *Proceedings of the 24th ACM Conference on Economics and Computation*, pp. 1047–1074, 2023.

[21] C. Mendler-Dünner, J. Perdomo, T. Zrnic, and M. Hardt, "Stochastic optimization for performative prediction," *Advances in Neural Information Processing Systems*, vol. 33, pp. 4929–4939, 2020.

[22] C. Mendler-Dünner, F. Ding, and Y. Wang, "Anticipating performativity by predicting from predictions," *Advances in neural information processing systems*, vol. 35, pp. 31171–31185, 2022.

[23] B. Amos and J. Z. Kolter, "Optnet: Differentiable optimization as a layer in neural networks," in *International conference on machine learning*, pp. 136–145, PMLR, 2017.

[24] M. Besançon, J. Dias Garcia, B. Legat, and A. Sharma, "Flexible differentiable optimization via model transformations," *INFORMS Journal on Computing*, vol. 36, no. 2, pp. 456–478, 2024.

[25] P. Donti, B. Amos, and J. Z. Kolter, "Task-based end-to-end model learning in stochastic optimization," *Advances in neural information processing systems*, vol. 30, 2017.

[26] E. Castillo, A. Conejo, C. Castillo, and R. Mínguez, "Closed formulas in local sensitivity analysis for some classes of linear and non-linear problems," *Top*, vol. 15, no. 2, pp. 355–371, 2007.

[27] F. Pacaud, "Sensitivity analysis for parametric nonlinear programming: A tutorial," *arXiv preprint arXiv:2504.15851*, 2025.

[28] S. Mohamed, M. Rosca, M. Figurnov, and A. Mnih, "Monte carlo gradient estimation in machine learning," *Journal of Machine Learning Research*, vol. 21, no. 132, pp. 1–62, 2020.

[29] PJM, "PJM Data Miner 2." [Online] Available: https://dataminer2.pjm.com/list.

[30] Copernicus Climate Change Service (C3S), "Era5 hourly time-series data on single levels from 1940 to present." Copernicus Climate Change Service Climate Data Store (CDS), 2025.

[31] Y. Yang, M. Jin, H. Wen, C. Zhang, Y. Liang, L. Ma, Y. Wang, C. Liu, B. Yang, Z. Xu, *et al.*, "A survey on diffusion models for time series and spatio-temporal data," *ACM Computing Surveys*, 2024.

[32] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," *arXiv preprint arXiv:2010.02502*, 2020.

[33] R. Mieth and H. V. Poor, "Prescribed robustness in optimal power flow," *Electric Power Systems Research*, vol. 235, p. 110704, 2024.

[34] MATPOWER, "CASE5 Power flow data," 2014.

[35] M. Innes, E. Saba, K. Fischer, D. Gandhi, M. C. Rudilosso, N. M. Joy, T. Karmali, A. Pal, and V. Shah, "Fashionable modelling with flux," *CoRR*, vol. abs/1811.01457, 2018.

[36] I. Dunning, J. Huchette, and M. Lubin, "Jump: A modeling language for mathematical optimization," *SIAM Review*, vol. 59, no. 2, pp. 295–320, 2017.

[37] M. Besançon, J. Dias Garcia, B. Legat, and A. Sharma, "Flexible differentiable optimization via model transformations," *INFORMS Journal on Computing*, vol. 36, no. 2, pp. 456–478, 2023.

I. Review #87A

A. Paper summary

The work presents a differentiable pipeline for co-optimization of generation and transmission investments with policy investments such as subsidies and tax credits. Traditionally, policy inputs enter as fixed deterministic inputs into investment planning. The key idea of this work is to recognize complementaries between generation investment and policy decisions, and that both can be co-optimized in the interest of least-cost investment planning in power systems. To facilitate co-optimization, the work makes a very powerful observation: the generative models that produce future scenarios (e.g., loads that depend on policy decisions) are differentiable themselves, leading to an end-to-end differentiable pipeline that optimizes both investment and policy decisions. In the opinion of this reviewer, this powerful observation breaks the existing status quo in investment optimization, enabling exploration of the synergies between policy design and investment planning in power systems.

B. Comments for authors

C1. The narrative does not detail the cost function of the inner problem. If operational costs are quadratic, this renders the inner problem smooth and the primal solution is likely unique, which makes it more "convenient" to differentiate through. If the cost functions are linear (as in many planning software), this yields a non-smooth problem, which presents challenges for differentiating (e.g., non-unique solution for a given policy decision). While it is not necessarily a big bottleneck for implicit differentiation (the authors outsource this to an off-the-shelf package in Julia), the narrative should still expand on the choice of operating cost functions and whether the inner problem requires regularization for non-smooth dispatch problems.

C2. Formulation (3) confines the policy decision to a policy space, which can be large and lead to large variance in the differentiable scenarios. The reviewer believes that introducing some form of regularization for the policy decisions can help reduce the variance of the scenarios, and thus produce more stable co-optimization outcomes. Specifically, one would like to avoid solutions at the boundaries of the policy space to avoid overly pessimistic or overly optimistic planning. The narrative should expand on whether this is indeed an issue and needs some form of regularization.

C3. The work employs a diffusion model for scenario generation, where load patterns are conditioned on policy decisions (e.g., EV adoption). Training this model requires either credible historical data correlating policy decisions with load patterns, or models that explicitly capture the dependency of loads on policy decisions. The narrative should detail which approach is taken, rather than outsourcing this discussion to external references.

C4. The reviewer recommends outlining the plan for integrating these ideas within existing planning models, such as GenX, Temoa, Switch, etc. As a Julia-native package, the former can naturally enjoy the proposed pipeline which rests on JuMP, Flux and DiffOpt packages in Julia. However, the typical problem dimensions are of a large scale, e.g., hundreds of nodes even in the reduced models, year-long operational horizons with intertemporal constraints. They typically require days of high-performance computing to yield optimal investments for given policy inputs. This raises the question of tractability: how does the proposed differentiable pipeline scale to such settings without rendering runtimes prohibitive?

C. Summarize your recommendation in one to two sentences

The paper adheres to the spirit of PowerUp, proposing, in the opinion of this reviewer, a very fresh look into long-term power system planning, where investment decisions are co-optimized with policy decisions, potentially saving billions in policy and investment planning. I strongly recommend the acceptance.

— Vladimir Dvorkin

II. Review #79B

A. Paper summary

The manuscript proposes a framework for jointly optimizing power system investments and policy decisions by introducing differentiable scenario generation.

The key idea is to make the demand distribution an explicit, differentiable function of policy parameters by training a conditional diffusion model (DDIM) on simulated policy-dependent load data, then backpropagating planning gradients through the generative model via the reparameterization trick and through the operations problem via implicit differentiation of KKT conditions.

B. Comments for authors

Strengths:

1) The idea of co-optimizing policy instruments and physical infrastructure investments through differentiable scenario generation is novel in the way that is presented in this paper. The observation that policy variables influence load distributions, and that this feedback should be internalized in planning models, is well-motivated and practically significant. The paper opens a promising research direction.
2) The integration of generative AI into a mathematical programming loop via automatic differentiation is novel.
3) Formalizing the concept of a ”differentiable scenario generator” (Definition 1) provides a strong mathematical bridge between machine learning and power systems planning.

Weaknesses:

1) The numerical experiment only demonstrates that the gradient descent algorithm converges. There is no comparison to traditional two-stage stochastic programming with endogenous uncertainty.
2) A DC-OPF formulation is used on the PJM 5-bus with24-hour periods decoupled for operational model, soft generation/transmission constraints with penalty. The 5-bus PJM system is exceptionally small. Implicit KKT differentiation requires solving dense linear systems, which scales non-linearly. The claim of efficient solutions is weakly supported by a 5-bus test.
3) The paper heavily leverages on [16] Degleris. The differentiation between their physical grid planning and this paper's combined policy-grid planning should be highlighted structurally in the introduction.

C. Summarize your recommendation in one to two sentences

The paper introduces a novel formulation for co-optimizing policy and infrastructure investments via differentiable scenario generation, but the experimental evaluation lacks baselines, quantitative metrics, and statistical analysis needed to substantiate the claims.

— David Pozo

III. Review #87C

A. Paper summary

The paper proposes an integrated investment and policy planning framework that incorporates policy decision-dependent uncertainty in the future demand scenario. A first-order gradient-based optimization approach is developed that uses sample approximation for uncertainty modeling, sensitivity computation for lower-level optimization problems, and a differentiable scenario generator. The key contribution is towards the differentiable scenario generator that allows the computation of the sensitivity of demand scenarios with respect to policy decisions.

B. Comments for authors

The use of the differentiable scenario generator for estimating distribution shift due to policy decisions is innovative. The paper contributes to the growing body of research on decision-dependent uncertainty. Addressing the following aspects would have made the manuscript stronger:

1) The setup of problem (3) fits well in the framework of decision-dependent uncertainty modeling. The paper would have benefited from making the connections, reviewing the relevant literature, and providing technical/empirical comparisons.
2) The proof of Proposition 1 reads reasonably. However, the ”gradient of scenarios with respect to policy input” may not be intuitive to the readers at first glance. Adding a discussion directed towards establishing such intuition would be helpful.
3) The setup of problem (3) is interesting and useful. However, the application to power-system policy planning does not come across as a natural fit due to the following reasons:
   a) It feels incredibly hard to model $D(\pi)$ in power systems. If one aims for the presented approach of training a scenario generator, getting reasonably accurate large training datasets for poicy decisions that have not yet been taken is nontrivial.
   b) Examples of policy decisions would be providing tax reliefs, subsidies, establishing trade agreements, etc. An outcome of such decisions would be, eg. greater adoption of EVs, which would then affect the distribution of demands. The numerical tests consider the intermediate step as policy decisions. It is unclear how a planner decides the adoption rates of EVs or heat pumps directly.
   c) It is not clear how a planner would quantify the cost of a policy decision in a way that is comparable to building a line.

   The above limitations are not to criticize the proposed approach but rather to examine the applicability in power systems planning.

4) From the algorithm's description, its scalability is not evident. The included test on the 5-bus system creates additional room for confusion. The paper should have better acknowledged and discussed the computational challenges; the approach is novel and useful even if the vanilla version is computationally prohibitive.
5) Potential typos/notational confusions: a. In Definition 1, item 2, is the gradient supposed to be with respect to ϵ or π? b. What are the dimensions of quantities in (11)? The notations appear to be for scalars, while Figure 1 indicates vectors.

C. Summarize your recommendation in one to two sentences

The paper presents a novel and interesting idea of using differentiable scenario generators for solving planning problems with decision-dependent uncertainty using gradient-based methods. The literature review, fit of the chosen application, and scalability analysis could have been improved.

— Anonymous reviewer

IV. Brief response to reviewers (optional)

The author thanks the editorial team for handling the review process and the two reviewers for their positive and constructive comments. The author provides the following responses to the reviewers.

Response to Reviewer 1

Regarding Comment 1 on the structure of the inner problem: Indeed, in the original version of our paper, we did not specify the properties of the inner problem in the problem description. We have closed this gap in the final version and added a short paragraph highlighting that, as also mentioned by the reviewer, quadratic or conic problems that lead to unique solutions and smooth differentiability are generally preferable to linear problems that exhibit "jumpy" or non-unique solutions. We also point out that our experiments with linear worked well with DiffOpt.jl.

Regarding Comment 2 on policy regularization: In the implementation discussed in this paper, policy decisions are implicitly regularized by the cost term in (3a). We agree with the reviewer that a cost function with stronger regularization properties, e.g., quadratic costs, may help guide the solution algorithm. This requires further experiments.

Regarding Comment 3 on diffusion model training, we would like to highlight that, for this paper, we used a low-complexity scenario sample simulator, which we outline in Section IV.A. We do not rely on any external packages for the experiments shown here. We mention third-party simulation tools as a pathway to applying our proposed method to more practical problems.

Regarding Comment 4 on scalability: Iterative, gradient-based approaches to solving planning models allow model decoupling, which avoids building and solving large-scale optimization problems and points toward scalability. See also the results in [a],[b]. Additionally, the training of the differentiable scenario generator can be done completely offline and independently of the target planning problem.

Response to Reviewer 2

Regarding the comment on convergence and uncertainty: We agree with the reviewer that more rigorous analyses of the mathematical properties of the proposed differentiable scenario generator and its behavior in a gradient descent algorithm are needed. This is the subject of ongoing work. The reviewer also mentions a comparison with problems with endogenous uncertainty. This is possible, but opens up some complexities that we discuss in our recent work in [b].

Regarding Comment 2 on scalability, the author refers to [a], who report promising scalability for the same problem. A related open question that we are actively investigating is the complexity of differentiating the scenario generator when scenario outputs and policy vectors become more complex. However, the maturity of current computational tools for computing gradients over machine learning models suggests good potential for scalability.

Regarding Comment 3 on the relation to [a], we have edited the paper to better reflect our contribution.

[a] A. Degleris, A. El Gamal, and R. Rajagopal, "Gradient methods for bilevel electricity grid expansion planning," Applied Energy, vol. 418, p. 128044, 2026.

[b] M. Ghazanfariharandi and R. Mieth, "Uncertainty-aware power system planning via gradient descent," arXiv preprint arXiv:2606.27602, 2026.

V. Brief summary of changes made to the final paper

Besides a few minor edits to improve upon the paper's language, the following changes have been made relative to the original submission (order based on the occurence of the change in the paper):

- We slightly edited contribution 1) to better highlight the difference of this paper relative to [16].
- In Eq. (3b), the constraint is now written out. The math itself was not changed.
- The final paragraph of Section II was added to better connect the work of this paper with related literature.
- In Section III.A a missing explanation of vector θ was added.

- In Section III.B we added a short paragraph to discuss the effectiveness of implicit differentiation on linear problems.
- In Definition 1 we fixed a critical typo in 2) (the gradient should be with respect to $\pi^\star$ and not with respect to $\epsilon^\star$), and another typo in the last sentence of the definition ($\epsilon^{(i)}$ was missing its superscript).
- In the last paragraph of Section IV.D we corrected a typo in the figure reference and the numerical result (36% instead of 46%).