Lawrence Berkeley National Laboratory
LBL Publications

Title
Flow matching for generative modelling in bioinformatics and computational biology

Permalink
https://escholarship.org/uc/item/07b9j5th

Journal
Nature Machine Intelligence, 8(4)

ISSN
2522-5839

Authors
Morehead, Alex
Atanackovic, Lazar
Hegde, Akshata
et al.

Publication Date
2026-04-01

DOI
10.1038/s42256-026-01220-0

Copyright Information
This work is made available under the terms of a Creative Commons Attribution License, available at https://creativecommons.org/licenses/by/4.0/

Peer reviewed

eScholarship.org Powered by the California Digital Library
University of California

Flow matching for generative modeling in bioinformatics and computational biology

Alex Morehead[1*†], Lazar Atanackovic[2†], Akshata Hegde[3†], Yanli Wang[3†], Frimpong Boadu[3†], Joel Selvaraj[3†], Alexander Tong[4], Aditi Krishnapriyan[5,1], Jianlin Cheng[3*]

[1*]Lawrence Berkeley National Laboratory, Berkeley, California, USA.
[2]Electrical & Computer Engineering, Vector Institute, University of Toronto, Toronto, Canada.
[3*]Electrical Engineering & Computer Science, NextGen Precision Health, University of Missouri, Columbia, Missouri, USA.
[4]Mila - Quebec AI Institute, Université de Montréal, Montréal, Canada.
[5]Computer Science & Chemical Engineering, University of California-Berkeley, Berkeley, California, USA.

*Corresponding author(s). E-mail(s): acmwhb@lbl.gov; chengji@missouri.com;
Contributing authors: l.atanackovic@mail.utoronto.ca; aghktb@missouri.edu; yw7bh@missouri.edu; fbqc9@missouri.edu; jsbrq@missouri.edu; alexander.tong@mila.quebec; aditik1@berkeley.edu;
†These authors contributed equally to this work.

Abstract

Numerous problems in bioinformatics and computational biology can be framed as a task of learning a mapping from one state of a biological system to another relevant state or to explore novel data points across biologically constrained spaces. However, manually deriving such mappings, e.g., to transform cells in a diseased state back into a healthy state, or extrapolating from existing datasets to create new data, is often nontrivial and can require extraordinary domain expertise and resources. Fortunately, the field of generative artificial intelligence (AI) has introduced a new training paradigm referred to as (conditional) flow matching, which has emerged as a promising solution to this problem, with broad applicability in computer vision, natural language processing, and the physical and life sciences. Flow matching is a powerful and principled, data-driven

framework for efficiently learning a mapping between arbitrary pairs of high-dimensional data distributions, making it well-suited for addressing problems in molecular and cell biology. In this Review, we characterize the theoretical foundations of flow matching and its applications in biomolecular modeling for small molecules, proteins, DNA/RNA, and their interactions, as well as its uses in single/multi-cellular modeling for cell phenotyping and imaging, each contributing towards the development of an AI-based virtual cell. Lastly, this review highlights open-source flow matching methods and discusses future directions in flow-based generative modeling for bioinformatics and computational biology.

Keywords: Flow matching, generative AI, bioinformatics, computational biology

1 Introduction

The field of generative AI has developed numerous important applications for biological and biomedical data analysis to date. Notably, methods such as AlphaFold 3 [1] and RoseTTAFold-All-Atom [2] have advanced our ability to meaningfully predict and design the structure of life's molecules, and agentic approaches to running virtual research labs using large language models [3] have led to the discovery of new SARS-CoV-2 nanobodies [4]. As such, bioinformatics and computational biology are poised for transformation in the age of generative AI algorithms [5].

Highlighting specific examples of this trend, Frazer et al. [6] introduced Bayesian variational autoencoders (VAEs) [7] for evolutionary protein sequence modeling, while Cui et al. [8] created a generative pre-trained transformer (GPT)-style [3] foundation model [9] for single cell data. As first-of-their-kind results, Anand and Huang [10] generated protein structures *in silico* using a deep generative adversarial network (GAN) and further experimentally validated structures generated using an autoregressive model [11]. Interestingly, applications of generative AI in bioinformatics have also adapted ideas from non-equilibrium statistical physics to develop denoising diffusion models [12] for high-quality molecule [13] and protein [14] generation and even robust translation between biomedical imaging techniques [15]. Such techniques have also extended beyond the biosciences into the materials sciences [16].

More recently, generative AI has been experiencing a surge of research interest in developing new deep learning (DL) algorithms [41] for sampling from arbitrary (e.g., biological) data distributions (Box 1). Popular types of DL algorithms for generative modeling to date include GANs, VAEs, normalizing flows, denoising diffusion and large language models. The most recent and prominent of such methods, grounded in the principles of optimal transport theory, represent a new paradigm of generative modeling called **flow matching (FM)** [23]. At a high level, FM refers to a class of methods designed to learn a transformation that maps samples from a source distribution—which is often an easy-to-sample distribution such as a standard Gaussian—into samples from a specified target distribution. The target distribution is typically defined by an empirical dataset and may represent intricate structures or patterns present in real-world data. The overarching goal of FM is to construct a continuous, learnable

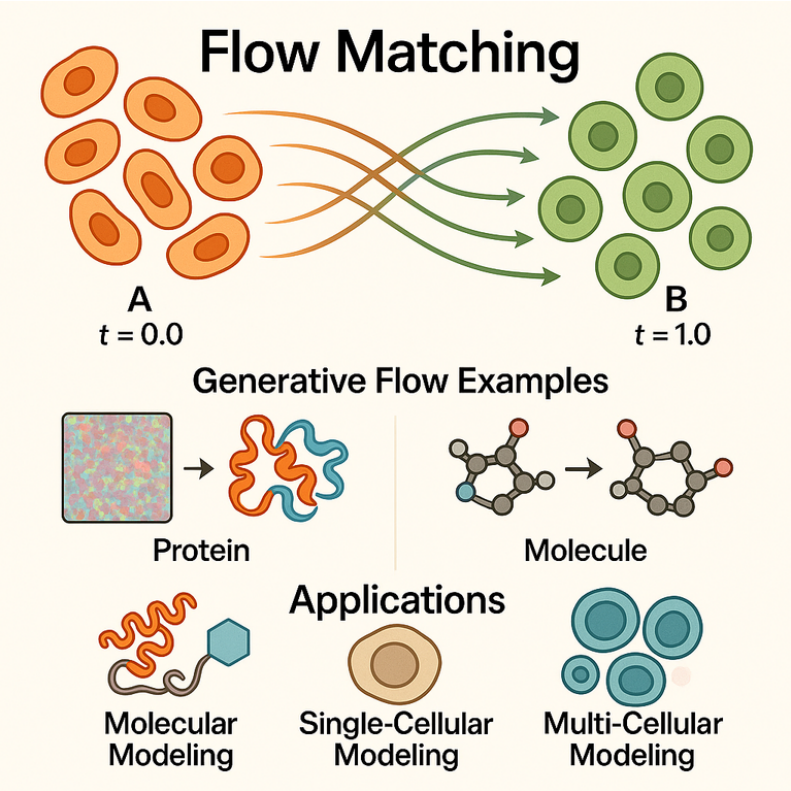


Fig. 1 An overview of flow matching and its applications in bioinformatics. Top, Generative flow matching for optimal transport of samples from a source cell population (A) to a target cell population (B) using a learned, time (t)-based velocity field. **Middle**, Examples of flow matching for generative modeling of proteins and small molecules. **Bottom**, Application areas of flow matching in bioinformatics relevant to developing an AI-based virtual cell.

transport path that progressively aligns the source distribution with the target, facilitating efficient sample generation and enabling applications in generative modeling, density estimation, and probabilistic inference (Box 2).

Developing powerful FM generative models could unlock several promising new application areas within bioinformatics and computational biology, including obtaining end-to-end learnable mappings between functional states of 3D (structural) biomolecules and single or multi-cellular systems as illustrated in Figure 1. With data-driven mappings like these, the development of an AI-based virtual cell (AIVC) [42] may be within the scientific community's grasp in the coming decade, ushering in a new era of rapid biological hypothesis generation and testing via techniques from computational molecular and cell biology. In Section 2 of this Review, we first introduce the theoretical foundations of FM and how its core techniques may be improved in the coming years. We then discuss in Section 3 how FM has already been applied within diverse biological modeling contexts to accelerate scientific discovery. Lastly, in Section 4, we provide readers with an outlook on the future of FM's role in bioinformatics.

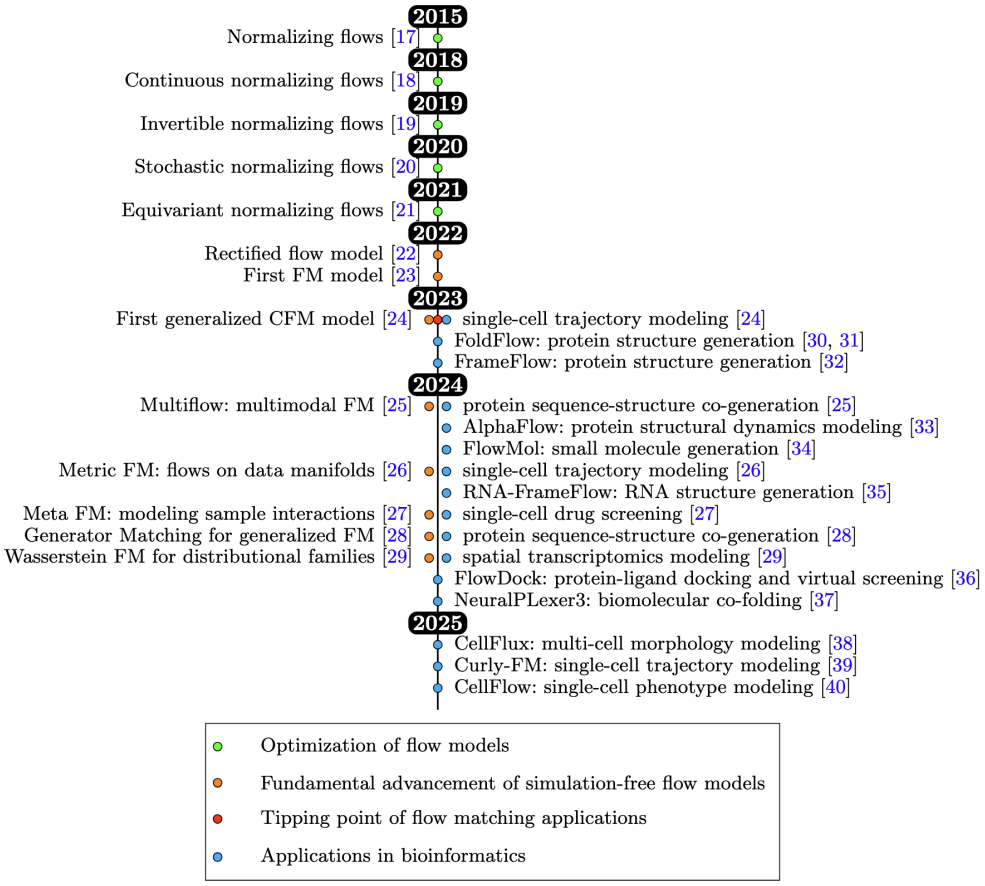


Fig. 2 A timeline illustrating the history of flow matching model development and the application of flow matching methods in bioinformatics.

2 Fundamentals of flow matching

In this subsection, we will adopt the standardized FM notation of Lipman et al. [43]. At its core, FM theory is centered on learning to transport samples using probability paths between arbitrary pairs of distributions [17, 44–46], one often referenced as a (easy-to-sample) source distribution $p_0 = p$ and the other denoted as a (empirical) target distribution $p_1 = q$, where p and q are distributions defined over $\mathbb{R}^d$. Notably, what sets FM apart from previous popular generative modeling frameworks such as diffusion models [5, 47] is that, with FM, the source distribution p_0 is not restricted to be Gaussian in nature and can instead be any distribution of one's choosing [24]. FM can thus be viewed as a generalization of diffusion models. Further, one can reduce the time required to sample new data points from the generative flow model by ensuring its probability paths promote optimal transport [23]. For example, a common way to do this is to define the model's probability paths using linear time interpolation

between the source and target distributions, akin to performing geodesic interpolation on Riemannian manifolds [48], yielding simpler, straight-line trajectories compared to those of popular diffusion probability paths [49]. In theory, this opens the possibility to use FM to learn arbitrary mappings between different states of (biological) systems to accelerate data modeling and analysis in numerous scientific domains.

2.1 History of development

As illustrated in Figure 2, the earliest works preceding FM, from 2015 to 2021, include normalizing flows [17] and their continuous variants [18, 19], which were introduced to address known limitations of variational inference for data density estimation [17, 50]. At a high level, (discrete) normalizing flows resemble a sequence of Euler update steps of an ordinary differential equation (ODE):

$$X_T = f_T \circ f_{T-1} \circ \cdots \circ f_1, \text{ with } f_t = X_t + m\phi(w^\top X_t + b), \tag{1}$$

where X_t represents a continuous random variable with time-dependent probability $p(X_t)$; $\{m \in \mathbb{R}^d, w \in \mathbb{R}^d, b \in \mathbb{R}\}$ are learnable parameters; and ϕ denotes an element-wise nonlinearity with derivative ϕ'. Notably, a limitation of (discrete) normalizing flows is the need to compute the expensive Jacobian $\frac{\partial f_t}{\partial X_t} = 1 + m^\top \phi'(w^\top X_t + b)w$ for density estimation:

$$\log p(X_{t+1}) = \log p(X_t) - \log \left| \det \frac{\partial f_t}{\partial X_t} \right|. \tag{2}$$

In 2018, Chen et al. [18] introduced continuous normalizing flows, which removed the need to calculate the Jacobian $\frac{\partial f_t}{\partial X_t}$ and relaxed the requirement that f_t be a bijective function (i.e., neural network architecture). Then, in 2022, these foundational flow methods culminated in the development of the first formalized FM model [23], which greatly improved the speed and numerical stability of flow-based generative models by no longer requiring ODE integration (i.e., simulation) during training.

The following year soon marked the tipping point for normalizing flow and FM theory [24], as in 2023, FM enabled the development of the first generation of large-scale bioinformatics applications in molecular [30, 32] and single-cell modeling [24] with flow-based generative models. From 2024 to mid-2025, FM has continued to introduce promising new flow models in the chemical and biology sciences, including those for drug screening [27], spatial transcriptomics [29], biomolecular co-folding [37], and morphology modeling [38]. Notably, in 2024 alone, 8 key FM methods for molecular modeling were released, while in early 2025, 3 foundational methods for single-cell modeling-among other unlisted methods for molecular modeling-were introduced. This trend suggests that FM theory has already begun widely influencing the fields of molecular and single-cell modeling, and vice versa, as oftentimes fundamental advancements for flow models have been introduced alongside new FM applications. Next, to better understand how and why this trend is emerging, we will discuss how to design a FM generative model from first principles.

2.2 Workflow of flow matching

2.2.1 Defining a training objective

Concretely, to learn a generative model for data generation, FM builds a **probability path** $(p_t)_{0\leq t\leq 1}$, where each p_t also denotes a distribution over $\mathbb{R}^d$. With this path defined, FM theory turns to training a neural network to predict the **vector field** needed to convert samples from the source distribution p_0 to the target distribution p_1 along the probability path p_t. Namely, using a trained vector field network, FM enables one to generate a new data point from the target distribution $X_1 \sim q$ by randomly sampling a point from the source distribution $X_0 \sim p$ and subsequently solving the ODE that the vector field characterizes. Specifically, with neural network parameters θ, this time-dependent vector field $u^\theta : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ determines the time-dependent **flow** $\psi : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ defined as:

$$\frac{\mathrm{d}}{\mathrm{d}t}\psi_t(x) = u_t^\theta(\psi_t(x)), \text{ with } \psi_t := \psi(t,x) \text{ and } \psi_0(x) = x. \tag{3}$$

With FM, we say that the vector field u_t^θ *generates* the probability path p_t if, for its corresponding flow ψ_t, the following statement is satisfied:

$$X_t := \psi_t(X_0) \sim p_t \text{ for all } t \in [0,1]. \tag{4}$$

The **standard FM loss** is defined as:

$$\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}_{t,X_t}\|u_t^\theta(X_t) - u_t(X_t)\|^2, \text{ with } t \sim \mathcal{U}[0,1] \text{ and } X_t \sim p_t. \tag{5}$$

Unfortunately, calculating the ground-truth vector field $u_t(X_t)$ is computationally intractable, as doing so requires integrating over all possible transformations between two high-dimensional distributions. Instead, in each training iteration, we can equivalently *condition* the loss on a single (random) example X_1 from the training dataset, yielding the following **conditional FM loss**:

$$\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}_{t,X_1,X_{t|1}}\|u_t^\theta(X_{t|1}) - u_t(X_{t|1}|X_1)\|^2, \tag{6}$$

where $u_t(X_{t|1}|X_1)$ is the corresponding **conditional vector field** which *generates* the **conditional probability path** $p_{t|1}(\cdot|X_1)$ of the **conditional random variable** $X_{t|1}$. Substituting (the intractable) $\mathcal{L}_{\mathrm{FM}}(\theta)$ with (the tractable) $\mathcal{L}_{\mathrm{CFM}}(\theta)$ is valid, as previous works have demonstrated that the gradients of Equations 5 and 6 are equal [23, 24]:

$$\nabla_\theta \mathcal{L}_{\mathrm{FM}}(\theta) = \nabla_\theta \mathcal{L}_{\mathrm{CFM}}(\theta). \tag{7}$$

To realize the simplifications this conditioning provides, we must first design our probability path p_t and, thereby, our conditional random variable $X_{t|1}$. The probability path p_t can be defined as the aggregation of conditional probability paths $p_{t|1}(x|X_1)$,

which are conditioned on the training dataset's (ground-truth) data point X_1:

$$p_t(x) = \int p_{t|1}(x|X_1)q(X_1)\mathrm{d}X_1. \tag{8}$$

Several choices for $p_{t|1}(x|X_1)$ exist, as *affine* probability paths with a Gaussian source distribution (including those of diffusion models) have the form:

$$p_{t|1}(x|X_1) = \mathcal{N}(x|\alpha_t X_1, \sigma_t^2 I). \tag{9}$$

For example, **variance preserving paths** assume $(\alpha_t = 1, \sigma_0 \gg 1, \sigma_1 = 0)$, while **variance exploding paths** assume $(\alpha_t = e^{-\frac{1}{2}\beta_t}, \sigma_t = \sqrt{1 - e^{-\beta_t}}, \beta_0 \gg 1, \beta_1 = 0)$. To date, a popular choice for $p_{t|1}(x|X_1)$ is the **optimal transport (OT) path** defined by $(\alpha_t = t, \sigma_t = 1 - t)$, since it readily satisfies the boundary conditions $p_0 = p$ and $p_1 = q$. This leads to a simple definition of the conditional random variable $X_{t|1}$ in terms of linear interpolation:

$$X_{t|1} = tX_1 + (1-t)X_0, \text{ where } X_{t|1\sim}\ p_{t|1}(\cdot|X_1) = \mathcal{N}(\cdot|tX_1, (1-t)^2 I). \tag{10}$$

For such an OT path, we can solve for X_0 in Equation 10 as $X_0 = \frac{X_{t|1} - tX_1}{1-t}$. Calculating the derivative $u_t(X_{t|1}|X_1) = \frac{\mathrm{d}}{\mathrm{d}t}X_{t|1} = \frac{\mathrm{d}}{\mathrm{d}t}(tX_1 + (1-t)X_0) = X_1 - X_0$, and subsequently substituting X_0 to obtain $u_t(X_{t|1}|X_1) = X_1 - \frac{X_{t|1} - tX_1}{1-t}$, simplifies to the following (tractable) **conditional OT vector field**, which we can use in Equation 6 to train the vector field network u_t^θ:

$$u_t(X_{t|1}|X_1) = \frac{X_1 - X_{t|1}}{1-t}. \tag{11}$$

2.2.2 Sampling with a learned vector field

To sample a new data point X_1 from a starting point X_0, one can use a trained vector field network u_t^θ to solve the corresponding *push-forward* ODE. Namely, using a simple update rule of the *Euler method*:

$$X_{t+h} = X_t + hu_t^\theta(X_t), \text{ where } h = n^{-1} > 0 \text{ determines the step size,} \tag{12}$$

one can arrive at a novel sample X_1 in a fixed number of steps $n \in \mathbb{N}$ iteratively. Note that many other ODE solvers can be substituted for the Euler method [51]. In particular, many prior works on FM adopt the Dormand–Prince (DOPRI5) method as a common ODE solver for generative modeling due to its empirically-improved performance compared to an Euler solver.

2.3 Discrete flow matching models

Amino acid and nucleotide sequences are a common form of *discrete* data in the biological sciences. However, modeling such data with flow matching-among other

types of generative models-requires additional consideration, as most flow matching models to date are designed for continuous data only [52]. To support generative modeling of discrete data, flow matching models typically follow one of two approaches: (1) treating discrete data as continuous data via floating-point one-hot encodings [53] or simplex mappings [54]; or (2) representing transformations of discrete data as a Continuous Time Markov Chain (CTMC) [25]. For simplicity of presentation, we will describe the latter approach in more detail, which currently achieves the best computational performance across a range of discrete generative modeling benchmarks [55].

2.3.1 Generative modeling with CTMCs

To begin, let $\mathcal{S} = \mathcal{T}^d$ represent a discrete state space; $T = \{1, 2, \ldots, K\}$ denote a K-sized discrete vocabulary; and $x^i \in (x^1, \ldots, x^d) = x, y, z \in \mathcal{S}$ be a single token. A probability mass function (PMF) for this discrete case is then

$$\delta(x, z) = \begin{cases} 1 & \text{if } x = z, \\ 0 & \text{else.} \end{cases} \tag{13}$$

With this PMF, a probability transition kernel for a CTMC can be defined as

$$p_{t+h|t}(y|x) := \mathbb{P}(X_{t+h} = y|X_t = x) = \delta(y, x) + hu_t(y, x) + o(h), \tag{14}$$

with step size h informing $o(h)/h \to 0$ as $t \to 0$ and with the initial distribution $\mathbb{P}(X_0 = x) = p(x)$ at time $t = 0$. The rates $u_t(y, x)$, also known as discrete velocities, determine how quickly one state changes to another state, which, to ensure $p_{t+h|t}(y|x)$ yields a valid probability distribution $\mathbb{P}(X \in A \subset S)$, are subject to the following rate conditions that ensure (1) the transition rates out of state x balance each other out and (2) states cannot transition with negative velocity:

$$(1) \sum_y u_t(y, x) = 0 \text{ and } (2)\ u_t(y, x) \geq 0.0\ \forall y \neq x. \tag{15}$$

Practically speaking, to sample new discrete data X_{t+h} from the starting point $X_t = X_0 \sim p$, one can apply the following Euler method with the (learned) discrete velocities $u_t^\theta(y, X_t)$:

$$\mathbb{P}(X_{t+h} = y \mid X_t) = \delta(y, X_t) + hu_t^\theta(y, X_t). \tag{16}$$

2.3.2 Learning discrete flows

To learn the velocities $u_t^\theta(y, X_t)$, analogous to continuous FM, a principled training objective is the **conditional discrete FM loss**:

$$\mathcal{L}_{\text{CDFM}}(\theta) = \mathbb{E}_{t, X_1, X_t \sim p_{t|1}} D_{X_t}(u_t(\cdot, X_t|X_1), u_t^\theta(\cdot, X_t)), \tag{17}$$

where D_{X_t} represents a Bregman divergence with the convex function $\Phi_x : \Omega_x \to \mathbb{R}$ defined by:

$$\Omega_x = \left\{ v \in \mathbb{R}^S \,\middle|\, v(x) = -\sum_{y \neq x} v(y) \text{ and } v(y) \geq 0 \, \forall y \neq x \right\} \subset \mathbb{R}^S. \tag{18}$$

The last two components needed for training a discrete FM model are the definitions of one's noise schedule for intermediate discrete data X_t and the ground-truth velocities $u_t(\cdot, X_t|X_1)$, for which we refer interested readers to Lipman et al. [43] for full derivations. In the following subsection, we introduce readers to the notion of representing *geometric* data types with flow matching.

2.4 Geometric flow matching

2.4.1 Symmetry groups

Rigid bodies in three-dimensional space are fully described by their position and orientation, forming a configuration space structured as the Lie group $SE(3)$, which combines translations ($\mathbb{R}^3$) and rotations ($SO(3)$) [30, 31, 43]. This group is a smooth Riemannian manifold, supporting definitions of distances, angles, and geodesics essential for modeling motions and probability distributions. In biological macromolecules such as proteins, each residue can be represented as a rigid body with a specific position and orientation, and the entire backbone as a sequence of frames in $SE(3)^N$ [30, 31, 56, 57]. Chirality is preserved by restricting to rotations and translations, as reflections in certain contexts are considered biologically invalid.

2.4.2 Comparing flow matching to other generative models

Generative modeling of 3D biological molecules, such as proteins and nucleotides, thus often involves learning distributions over $SE(3)^N$, requiring models that respect the manifold's symmetries and geometry [14, 47, 58]. Traditional diffusion models, adapted from Euclidean spaces, may struggle with manifold constraints and require many inference steps [59]. FM [43, 60] often overcomes these limitations by learning vector fields that more succinctly transport samples from a source distribution to a target distribution along continuous paths on the manifold. In particular, FM offers four key advantages *simultaneously* in comparison to diffusion models: it (1) needs fewer inference steps; (2) is simpler to implement; (3) allows explicit control over distribution couplings; and (4) can optionally preserve important geometric properties such as equivariance through neural network design or training. Notably, diffusion models may possess a subset of these attributes but not each of them collectively. To more systematically characterize the advantages and disadvantages of flow matching compared to previous types of generative models, Table 1 presents a comprehensive listing of the relative strengths and weaknesses of well-known generative modeling methods.

2.5 Improving flow matching models

While FM models offer an elegant and theoretically grounded framework for generative modeling, several challenges must be addressed to improve their empirical performance, particularly in bioinformatics applications involving high-dimensional, structured, and multimodal data. To this end, recent work has introduced multiple avenues for improvement. For example, Rectified Flows [22] solve a nonlinear least squares optimization problem to improve the sampling speed of FM, in some cases enabling high-quality sample generation in a single ODE solver step, much like Mean Flows [61]. Moreover, Generator Matching [28] provides a theoretical unification of various diffusion and FM methods, which may yield new multi-modal applications of FM models in future work. As a survey of open-source methods for implementing and improving FM models, in Table 2, we offer readers a comprehensive listing.

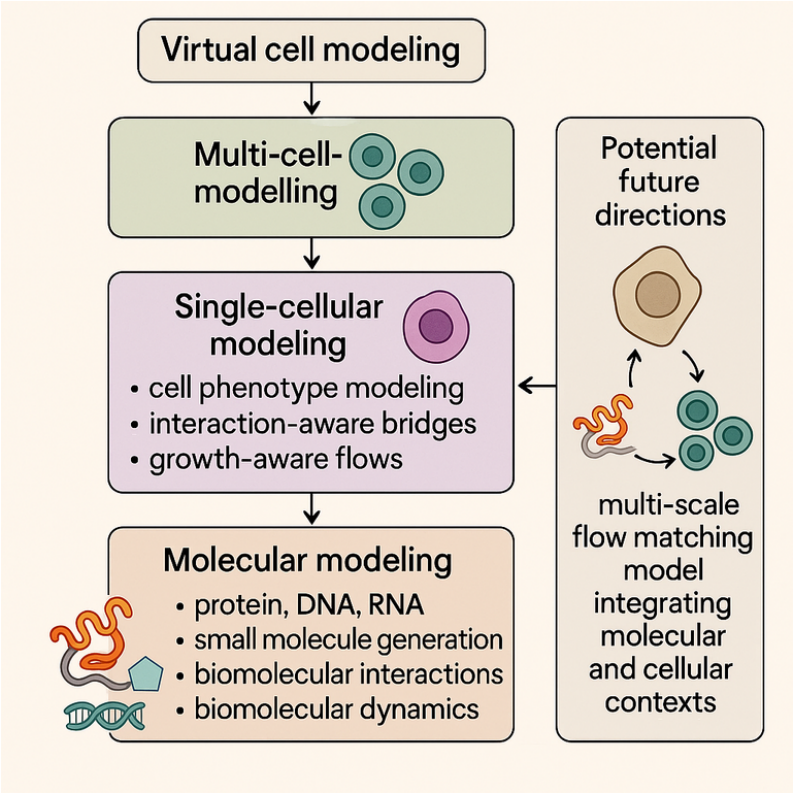


Fig. 3 A taxonomy of flow matching applications in bioinformatics relevant to virtual cell modeling. Top, Virtual cell modeling represented as the highest level of abstraction encompassing recent efforts in molecular and cellular modeling using flow matching. **Middle**, Single and multi-cellular modeling denoting the intermediate level of abstraction in virtual cell modeling with flow matching. **Bottom**, Molecular modeling signifying the foundational level of abstraction for flow matching-based virtual cell modeling. **Right**, Potential future research directions including flow matching-based integration of multiple (molecular and cellular) biological scales.

3 Applications of flow matching in bioinformatics

Imagine an experiment where you need to determine how a cell's state changes over a short period of time while accounting for numerous environmental, chemical, and evolutionary factors in effect. With FM, how could one go about modeling such cellular dynamics *in silico*? The first step would be to collect a sufficient quantity of paired measurements of the cell's state at (at least) two different time points. Encoding the cell's state variables with the simple yet expressive representations of a Transformer neural network [62] would be a natural next step. Then, as a starting point, using the Transformer to parametrize a conditional OT flow with an affine probability path would allow for rapid prototyping of the Transformer's architectural design, such as its width and depth. These architectural variables can be evaluated using cell state-specific reconstruction metrics, such as a mean squared error of continuous state feature vectors. Once trained, this Transformer can be used as an iterative ODE solver to transform the state of new cellular samples starting from an initial time point. Overall, this represents simply one of many ways one could computationally model this cellular experiment, given the versatility of FM.

In the remainder of this section, we outline several of such computational applications of FM in bioinformatics. Figure 3 particularly illustrates those that contribute to the development of an AIVC for multi-scale biological simulation and modeling. Importantly, given the hierarchical taxonomy of molecular and cell modeling methods depicted in Figure 3, we posit that bridging these applications will spark important new research directions in generative AI for virtual cells in the coming years. To concretely examine this trend, Section 3.1 characterizes existing FM applications for molecular modeling, while Sections 3.2, 3.3, and 3.4 describe recent applications of FM in single/multi-cellular modeling and bioimaging, respectively. As a supplement to these sections, Table 3 provides a comprehensive listing of the latest open and closed-source FM methods for bioinformatics applications, describing their supported modalities, data sources, and performance. For practical, general-purpose advice on how to build and apply FM models in real-world bioinformatics settings, we refer interested readers to Box 3.

3.1 Flow matching for molecular modeling

In this subsection, we discuss selected FM generative models for molecular modeling, as listed in Table 3 and taxonomized at the foundational level in Figure 3. Molecular modeling tasks, in this context, encompass small molecule conformer prediction and generation; sequence and structure generation of proteins/peptides and protein side-chain packing; (protein/ligand-conditioned) RNA/DNA sequence and structure generation; as well as prediction of biomolecular interactions and dynamics. Such biomolecules are inherently multimodal, often featuring (discrete) biological or chemical sequences, (discrete) atomic bonds, and (continuous) 3D atomic coordinates found in a variety of public datasets and evaluated according to domain-specific metrics (e.g., RMSD). Accordingly, in the following sub-subsections, we discuss selected generative modeling applications for small molecules (Section 3.1.1), proteins/peptides (Section

3.1.2), nucleotides (Section 3.1.3), as well as higher-order biomolecular interactions (Section 3.1.4) and dynamics (Section 3.1.5).

3.1.1 Generative modeling of small molecules

Designing 3D small molecules with high chemical fidelity is a core challenge in computational chemistry due to its direct medical and commercial applicability in drug discovery and material design [63]. In this setting, traditional generative models—such as graph-based autoencoders and diffusion models—are often hampered by slow sampling and difficulties in handling the discrete-continuous nature of molecular data. Recent FM approaches provide a promising solution by learning continuous velocity fields that efficiently map noise to valid molecular structures in a handful of steps. State-of-the-art models like SemlaFlow [64] and FlowMol [34] advance this *in silico* frontier by combining geometric deep learning, equivariant architectures, and optimal transport objectives. SemlaFlow leverages a message-passing neural network with latent attention and E(3)-equivariance to generate molecules up to 100x faster than diffusion models, while FlowMol bridges continuous and categorical data using chirality-aware Geometric Vector Perceptrons [65] and simplex-based or relaxed categorical modeling.

ADiT [58] introduces a unified Transformer-based framework for both molecular and crystalline material generation, performing denoising in latent space for efficiency and scalability without relying on a graph-specific inductive bias. ET-Flow [66] employs an equivariant transformer architecture and harmonic priors to generate low-energy molecular conformers directly from graphs, with post hoc correction for chirality and a lightweight, scalable design. Overall, these FM-based models indicate that there could be promising advances in 3D molecular generation, offering potential new possibilities for drug discovery and atomistic material design.

3.1.2 Generative modeling of proteins

Designing (large or small) proteins that satisfy geometric, physicochemical, and functional constraints is a complex challenge due to their vast conformational landscape and intricate sequence-structure dependencies. Nonetheless, achieving such aims can accelerate novel developments for atomically-precise medical, agricultural, and industrial products requiring programmable biological systems, resulting in new therapeutics, energy resources, and self-assembling nanomaterials [67]. Traditional diffusion-based models for protein design, like RFdiffusion [14], are effective but computationally intensive and often treat sequence and structure separately. Recent advances leverage FM to efficiently learn generative models on the product manifold $\mathrm{SE}(3)^N$, capturing both the geometry and symmetries of protein backbones. State-of-the-art methods—including FrameFlow [56], FoldFlow [30], Multiflow [25], and OriginFlow [68]—employ equivariant architectures, optimal-transport objectives, and discrete-state flows to accelerate protein structure generation and enable *in silico* sequence-structure co-design. FrameFlow and FoldFlow focus on fast, accurate backbone generation by learning continuous or stochastic flows on $\mathrm{SE}(3)^N$, with FoldFlow introducing optimal transport and stochasticity for improved diversity and robustness.

Multiflow extends this paradigm to joint sequence and structure generation, integrating discrete (sequence) and continuous (structure) flows for efficient, multimodal protein co-design.

OriginFlow addresses de novo protein design by combining FM with expressive neural architectures, such as Invariant Point Attention and graph neural networks, to robustly generate functional protein structures from noise for PD-L1, RBD, and VEGF targets, achieving overall expression, affinity, and solubility rates of 90%. Accordingly, it achieves remarkable experimental success rates, particularly in challenging tasks like binder design, outperforming previous models in both speed and controllability. Collectively, these FM-based approaches deliver improvements in designability, diversity, accuracy, and (in some cases) experimental validation, marking a shift from mere structure prediction to practical, AI-driven functional protein design and accelerating the integration of generative AI into real-world biopharma pipelines.

3.1.3 Generative modeling of nucleotides

Generative modeling of nucleotide sequences and structures is a central challenge in synthetic biology and RNA therapeutics, given the high therapeutic potential of RNA-based CRISPR gene editing systems [69]. With the discrete nature of DNA/RNA alphabets and the complex, high-dimensional biological constraints they encode, FM could provide an alternative to traditional autoregressive and diffusion models by enabling efficient, parallelizable, and controllable transformations from noise to structured sequences. Recent advances such as Dirichlet Flow Matching [70] and Fisher FM [71] introduce geometry-aware flows on the probability simplex, improving expressivity, stability, and performance in *in silico* DNA design tasks. For RNA, frameworks like RNACG [72] leverage Dirichlet FM and multimodal transformers for high-fidelity, conditional sequence generation, outperforming previous approaches across a range of *in silico* RNA design benchmarks.

Beyond sequence generation, FM-based models have also advanced RNA structure and sequence-structure co-design. RNAFlow [73] and RiboFlow [74] unify RNA sequence and structure generation by conditioning on protein or ligand geometry, using SE(3)-equivariant architectures and multi-scale representations to capture RNA's conformational flexibility. RNA-EFM [75] further incorporates energy-based priors for protein-conditioned co-design, while RNA-FrameFlow [35] and RNAbpFlow [76] address unconditional and base-pair-constrained 3D RNA structure generation, respectively. Collectively, these FM-based approaches establish a unified foundation for DNA/RNA sequence and structure design, and *in silico* engineering of nucleic acids.

3.1.4 Generative modeling of biomolecular interactions

Recent advances in generative modeling—especially those leveraging geometric FM and continuous normalizing flows—have enabled direct, scalable prediction of biomolecular interactions and complex structures, bypassing traditional docking and alignment-dependent pipelines in drug discovery and development [77]. Models like FlowDock [36] use conditional FM to map unbound protein structures to holo complexes, supporting simultaneous multi-ligand docking and accurate affinity prediction, and outperforming AlphaFold 3 and other baselines in certain benchmarking settings.

NeuralPLexer3 [37] employs physics-inspired normalizing flows for fast, high-resolution prediction of diverse biomolecular complexes, exceeding the accuracy of diffusion-based and docking methods across a wide range of targets. FlexDock [78] introduces Unbalanced Flow Matching to model flexible protein-ligand docking, relaxing strict mass preservation to better capture conformational changes; this two-stage approach improves docking accuracy and physical plausibility, as measured by PoseBusters and PDBBind benchmarks. Lastly, with wide applicability, UniMoMo [79] generates small molecules, peptides, and antibodies for target protein pockets using latent flow matching.

3.1.5 Generative modeling of biomolecular dynamics

Biomolecules are inherently dynamic entities, often transitioning between multiple conformations to carry out their biological functions. Capturing this structural heterogeneity is essential for understanding allostery, binding mechanisms, and functional regulation for programmable biological design [80]. Fortuitously, generative FM techniques are also transforming the modeling of biomolecular dynamics, enabling the simulation of conformational ensembles and transition pathways beyond static structure prediction. AlphaFlow [33] and related frameworks reinterpret AlphaFold and ESMFold as generative denoisers within FM, producing realistic protein conformational ensembles directly from sequence. FMRC [81] leverages FM to learn optimal reaction coordinates, outperforming traditional kinetic modeling methods and enhancing both Markov state model construction and enhanced sampling. MDGen [82] extends FM to generate full molecular dynamics trajectories, accurately capturing free-energy landscapes and kinetic properties in an SE(3)-invariant manner. Finally, Onsager-Machlup-based action minimization [83] enables efficient, zero-shot transition path sampling by optimizing generative model trajectories, offering scalable and model-agnostic solutions for simulating biomolecular motions. Overall, these advances establish FM-based generative modeling as a framework for predicting the dynamics of 3D biomolecules *in silico*, setting the stage for modeling the dynamics of cellular systems based on molecular context (Figure 3).

3.2 Flow matching for single-cellular modeling

Single-cell modeling, the next level up in the methodological hierarchy of Figure 3, carries great potential to advance our ability to efficiently analyze cellular processes *in silico*. In particular, generative modeling with FM has emerged as a powerful approach for simulating and predicting single-cell phenotypes under diverse perturbations, addressing the (perturbation/conditioning-related) combinatorial and high-dimensional challenges inherent in systems biology and precision medicine when attempting to characterize disease acquisition, progression, and treatment [84]. By learning continuous-time vector fields that transport distributions of unperturbed cellular states to those under complex perturbations, FM enables efficient and interpretable inference as well as simulation-free training across vast phenotypic landscapes. Frameworks such as CellFlow [40] and CFGen [85] integrate optimal transport, multimodal biological embeddings, and permutation-invariant architectures to model

single-cell responses. CellFlow excels at virtual phenotyping and systematic perturbation design, while CFGen extends FM to discrete count data and multi-modal settings, enabling compositional and attribute-guided simulation of cellular phenotypes.

Beyond static phenotype modeling, FM-based methods have advanced the *in silico* reconstruction of cellular dynamics and trajectories from single-cell snapshot data, overcoming the limitations of destructive assays and the simplifying assumptions of traditional approaches. Conditional Flow Matching (CFM) [24] and $[SF]^2M$ [86] provide simulation-free objectives for learning deterministic and stochastic flows, respectively, while GENOT [87] generalizes to stochastic, entropic, and unbalanced optimal transport, supporting cross-modal and heterogeneous data. Further innovations like Meta FM [27] and Wasserstein Flow Matching (WFM) [29] extend FM to population-level and distributional settings, enabling personalized trajectory modeling and the synthesis of spatial cellular niches.

Recent work has also focused on incorporating geometric, biological, and dynamical realism into FM-based cellular modeling. Metric FM [26] leverages data-induced Riemannian metrics to ensure learned trajectories remain close to the true data manifold, with a goal towards improving biological plausibility. Unbalanced Mean-Field Schrödinger Bridge and CytoBridge [88] jointly model state transitions, population growth, and cell-cell interactions, while Curly-FM [39] enables the modeling of non-gradient, cyclic dynamics such as the cell cycle by matching both distributional and velocity field constraints. Velocity-Growth Flow Matching (VGFM) [89] further extends FM to jointly infer cellular state transitions and population-level mass changes with regression-based and growth-aware training objectives. Consequently, VGFM aims to ensure scalability and training efficiency, to model development and disease progression *in silico*.

Finally, Multi-Marginal Flow Matching [90] generalizes FM to handle transitions across multiple time points, enabling globally consistent trajectory inference in complex experimental designs. Collectively, these *in silico* advances establish FM as a foundation for single-cell generative modeling-enabling virtual phenotyping, dynamic trajectory reconstruction, and the integration of biological knowledge across high-dimensional and heterogeneous datasets-laying the groundwork for *in silico* analysis of inter-cellular interactions and dynamics (Figure 3).

3.3 Flow matching for multi-cellular modeling

FM has recently enabled early experiments using generative modeling to characterize the behavior of multi-cellular systems *in silico*, with such systems being highly relevant for tissue engineering and regenerative medicine [91]. Generally speaking, extending FM to image-based multi-cellular phenotypic modeling could enable a new level of insight into cellular morphology and its response to perturbations. Nonetheless, while high-content imaging assays provide rich spatial information, they also present challenges for generative modeling due to batch effects, experimental variability, and morphological complexity. Fortunately, recent *in silico* advances, such as CellFlux [38], resolve such issues by leveraging conditional FM to simulate morphological changes at the pixel level, bridging the gap between generative modeling and high-resolution cellular imaging.

CellFlux employs a U-Net-based FM architecture to learn continuous transformations from control to perturbed cell images, incorporating classifier-free guidance and noise augmentation for improved fidelity. Evaluated across diverse datasets—including BBBC021, RxRx1, and JUMP—CellFlux achieves superior Fréchet Inception Distance and mode-of-action prediction accuracy compared to prior baselines [92, 93], and generalizes well to unseen perturbations. Notably, it corrects for batch effects by conditioning on batch-specific control images, enabling realistic and interpretable interpolation between cellular states. These *in silico* advances could position CellFlux, and FM more generally, as a promising foundation for "virtual cell" platforms in drug discovery and personalized medicine, with future potential to scale across cell types and perturbation regimes based on a molecular context (Figure 3).

3.4 Flow matching for bioimaging

Broadly speaking, FM is also transforming bioimaging by enabling scalable, probabilistic models that capture complex spatial structures and provide principled uncertainty quantification, for broad uses within both research labs and medical clinics [94]. FM frameworks learn continuous-time vector fields to map simple priors to intricate biological images, supporting flexible inference without the need for extensive fine-tuning. Recent advances highlight FM's versatility across modalities, from cryo-electron microscopy (cryo-EM) to medical image segmentation, unifying representation learning and generative modeling in high-dimensional, noisy, and structured imaging data.

CryoFM [95], an unconditional FM model, exemplifies this potential as a foundation model for cryo-EM density maps, learning the distribution of high-quality protein densities and generalizing to tasks such as denoising and 3D reconstruction without task-specific retraining. The model achieves this by using a task-specific flow posterior sampling method during inference to convert CryoFM's unconditional vector field into a conditional variant. FlowSDF [96] advances microscopic image segmentation by modeling probability flows over signed-distance functions, yielding smooth, accurate segmentation masks and providing uncertainty estimates. FlowSDF reaches or surpasses state-of-the-art segmentation accuracy in tumor cell nuclei (MoNuSeg) [97] and gland (GlaS) [98] segmentation. In biomedical imaging, FM has shown recent success in MRI image reconstruction [99, 100] and medical image synthesis [101]. Together, these methods demonstrate how FM can bridge generative modeling and classical imaging approaches, offering a set of uncertainty-aware tools for a wide range of bioimaging applications.

4 Outlook

The rapid emergence of FM as a unifying paradigm for generative modeling marks a pivotal moment in the computational life sciences. Across molecular, cellular, and imaging domains, FM has already demonstrated advantages over previous simulation-based generative models, offering simulation-free training, efficient sampling, principled incorporation of geometry, and interpretable, controllable mappings between biological states. As surveyed in this Review, FM-based models now set the

state-of-the-art in a wide array of bioinformatics tasks, as assessed by current *in silico* benchmarks and evaluation: from de novo protein and RNA design, to conditional single-cell phenotype prediction, to generative modeling of high-content imaging data. Yet, the field is only beginning to realize the full potential of flow-based generative modeling in biology.

4.1 From foundational advances to biological impact

The past two years have seen foundational advances in FM theory and practice. New formulations—such as optimal transport FM, discrete and manifold-valued flows, and simulation-free Schrödinger bridges—have expanded the scope of generative modeling to settings previously inaccessible to diffusion or normalizing flow methods. These innovations have enabled the modeling of complex, multimodal, and structured biological data, accommodating discrete sequences, geometric manifolds (e.g., SE(3) for proteins), and high-dimensional empirical distributions (e.g., single-cell omics or cellular images). Crucially, FM models have begun to bridge the gap between generative modeling and biological interpretability. By learning explicit vector fields and probability paths, FM provides a transparent lens on the transformations between biological states—be they conformational transitions in biomolecules, cell-state changes under perturbation, or morphological shifts in response to drugs. This interpretability, coupled with the ability to condition on rich biological context (e.g., protein partners, ligand structures, perturbation embeddings), positions FM as a cornerstone for the next generation of AI-driven virtual biology. A clear next step forward will be to connect these ML advances to real-world benchmarks, evaluations, and applications.

4.2 Unifying and extending the generative toolkit

A striking trend is the unification of previously disparate generative modeling approaches under the FM umbrella. Recent work has shown that FM subsumes diffusion models, continuous normalizing flows, and optimal transport as special cases, while enabling new hybrid and multimodal architectures. This theoretical convergence is mirrored by practical advances: models such as FrameFlow [56], FoldFlow [30], FlowDock [36], Multiflow [25], and CellFlow [40] demonstrate how FM can be flexibly adapted to discrete, continuous, or mixed data; to Euclidean, Riemannian, or statistical manifolds; and to both unconditional and richly conditional settings. Looking forward, we anticipate three key directions for extending the FM toolkit:

1. **Generalization to new data modalities and biological scales.** FM has already proven effective for sequences, structures, and images. Extensions to spatial omics, multi-modal single-cell data, multi-scale tissue modeling, and even population-level dynamics are natural next steps. Lifting FM to operate over distributions of distributions, as in WFM [29], or to model time-evolving population heterogeneity, as in Meta FM [27] and VGFM [89], will be essential for capturing the complexity of biological systems. Key to enabling this will be the development of new real-world benchmarks and evaluation toolkits for the machine learning and bioinformatics community, to more systematically assess the applications of such methods under different practical contexts and constraints.

2. **Principled incorporation of biological priors and constraints.** Future FM models may encode necessary domain knowledge: energetic constraints, evolutionary couplings, or experimentally derived priors, as exemplified by NeuralPLexer3 [37]. However, recent advances in developing foundation models for the physical and life sciences [58, 102] have demonstrated that incorporating such constraints must be done with care so as not to hinder model performance and scalability in large data regimes, which may become increasingly common in molecular and cell biology in the coming years.
3. **Scalable, interpretable, and uncertainty-aware modeling.** As FM models are deployed in high-stakes applications—such as therapeutic design, synthetic biology, or personalized medicine—scalability and simplicity become paramount. Efficient distillation and one-step generative flows (e.g., via Mean Flows [61] or distilled FM), uncertainty quantification (as in FlowSDF [96]), and experimental (real-world) verification of the accuracy and generalizability of learned vector fields will be critical for building trust and enabling scientific discovery.

4.3 Toward the artificial intelligence-based virtual cell

A central vision emerging from recent FM advances is the realization of an AIVC: a unified, generative framework capable of simulating the molecular, structural, and phenotypic consequences of genetic, chemical, and environmental perturbations. FM models are uniquely well-suited to this challenge. By enabling efficient, simulation-free mappings across molecular, cellular, and imaging spaces—and by supporting conditional, context-aware generation—FM provides the scaffolding for an end-to-end, multi-scale virtual cell platform. Realizing this vision will require continued progress in several areas. Integrating FM models across biological scales and modalities—e.g., coupling protein-protein interaction flows with single-cell phenotype generators, or linking nucleotide design flows with 3D chromatin structure models—will demand advances in data curation and in compositional and multi-modal FM architectures. Robust benchmarking, open-source tool development, and community standards for biological generative modeling (see Tables 2 and 3) will be essential to ensure reproducibility, interoperability, and broad adoption.

4.4 Open challenges and future research directions

Despite its promise, FM-based generative modeling faces open challenges. Theoretical questions remain regarding optimality, convergence, and expressivity of learned flows, especially in high-dimensional or data-sparse regimes. Efficient training and inference for very large or multi-modal biological datasets remains an area of active development. Furthermore, the integration of FM with experimental design—enabling closed-loop, AI-driven hypothesis generation and validation—remains largely unexplored. Key future directions include:

- **Learning on limited or noisy data:** Developing robust, data-efficient FM models that can generalize from scarce or noisy biological measurements.

- **Integration with experimental feedback:** Coupling FM-based generators with experimental platforms (e.g., high-throughput screening, CRISPR perturbation [103]) for active learning and closed-loop discovery.
- **Interpretability and validation:** Building tools for visualizing, interpreting, and experimentally validating learned flows, ensuring that generative models yield biologically meaningful and actionable hypotheses connected to real-world challenges in molecular and cell biology.
- **Ethical and societal considerations:** As generative AI becomes increasingly influential in biological design, careful consideration of ethical, safety, and regulatory implications will be critical.

Acknowledgements

This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy User Facility, using NERSC award DDR-ERCAP 0034574 awarded to AM. This work was also financially supported by the U.S. National Science Foundation [grant number DBI2308699, awarded to JC] and the U.S. National Institutes of Health [grant numbers: R01GM146340 and R01GM093123, awarded to JC]. Lastly, this manuscript was edited (and Figures 1 and 3 were partially designed) using large language models hosted on CBorg, the multi-model AI portal of Lawrence Berkeley National Laboratory.

Author contributions

- **Alex Morehead:** Investigation; writing - original draft; writing - reviewing and editing.
- **Lazar Atanackovic:** Investigation; writing - original draft; writing - reviewing and editing.
- **Akshata Hegde:** Investigation; writing - original draft; writing - reviewing and editing.
- **Yanli Wang:** Investigation; writing - original draft; writing - reviewing and editing.
- **Frimpong Boadu:** Investigation; writing - original draft; writing - reviewing and editing.
- **Joel Selvaraj:** Investigation; writing - original draft; writing - reviewing and editing.
- **Alexander Tong:** Investigation; writing - original draft; writing - reviewing and editing.
- **Aditi Krishnapriyan:** Investigation; writing - original draft; writing - reviewing and editing.
- **Jianlin Cheng:** Conceptualization; investigation; writing - challenges and future perspectives, writing - reviewing and editing.

Competing interests

The authors declare no competing interests.

Code availability statement

- An accompanying paper list as a GitHub repository: https://github.com/amorehead/awesome-generative-flows.
- Meta AI's flow matching guide and code [43]: https://ai.meta.com/research/publications/flow-matching-guide-and-code.
- A PyTorch-friendly library [104] for training conditional flow matching models [24]: https://github.com/atong01/conditional-flow-matching.
- Illustrations of generative flow matching sampling trajectories driven by the FlowDock model for protein-ligand docking [36] can be found at: https://github.com/BioinfoMachineLearning/FlowDock/tree/main/img.

Tables

Model	Sampling & Speed	Likelihood	Architecture	Manifold	Conditioning	Equivariance	Issues / Failures
Flow Matching Models	Deterministic ODE sampling (learned vector field); typically requiring fewer steps than diffusion [105].	Exact density via invertible ODE (change-of-variables).	Time-dependent network (neural ODE) to model vector field.	Assumes Euclidean support; manifold-aware variants (e.g., Riemannian or pullback flow) preserve geometry [106].	CFM handles conditional generation (flows between sample pairs).	Equivariant FM yields shorter flows and greater efficiency [107].	Potential mode collapse (rectified flows and mean flows have "model collapse") [108, 109] → decompose training objective to improve stability; highly sensitive to probability-path choice [105]; gradient-variance instabilities near endpoints [110] → consider gradient clipping.
Diffusion Models	Iterative denoising (100s–1000+ steps; slow inference) [111] (distillation reduces to ∼10–20).	No explicit likelihood (trained via score matching); exact likelihood via probability-flow ODE is expensive.	U-Net–like or similar flexible networks.	Ambient Gaussian noise; Riemannian (manifold) extensions exist but are complex.	Naturally supports class/text conditioning.	Equivariance via network design (e.g., convolutional U-Nets).	Slow sampling and high compute; compared to GANs, stabler training and higher diversity (less mode-collapse) [111].
Continuous Normalizing Flows	Continuous-flow integration required; sampling similar to FM (can be slow).	Exact log-density via change-of-variables (integrated Jacobian).	Invertible transforms (coupling/autoregressive flows or neural ODE); must maintain tractable Jacobian.	Full-support flows; Riemannian flows can encode known manifold geometry explicitly.	Conditional flows by conditioning transformations on context.	Equivariant CNFs exist (e.g., physics Boltzmann generators) [107].	Computationally heavy (Jacobian computation); scaling to very high dimensions is challenging.
Generative Adversarial Networks	One-shot generation (single forward pass; extremely fast) [111].	No likelihood (implicit model).	Flexible generator/discriminator networks (adversarial training).	Implicit manifold learning; no explicit support constraints.	Well-established conditional GAN variants (class/-text, etc.).	Equivariance via architecture (e.g., convolution exploiting translation) possible.	Mode collapse (limited diversity) and training instability [111]; no exact density.
Variational Autoencoders	One-pass decoding (fast).	Approximate likelihood via ELBO (variational bound).	Encoder–decoder with latent Gaussian; simpler architecture.	Latent-manifold representation; risk of posterior collapse (encoder ignores latent) [112].	Conditional VAEs by concatenating labels/conditions.	Equivariance via encoder/decoder design (e.g., convolutional nets) possible.	Blurry outputs; limited diversity (mode-collapse analog) [112]; posterior collapse issues [112].

Table 1 Comparison of generative model classes. Key dimensions for this comparison include likelihoods, manifold constraints, conditioning options, and possible failure modes. (C)FM, (conditional) flow matching; (C)NF, (continuous) normalizing flow; GAN, generative adversarial network; VAE, variational autoencoder; ODE, ordinary differential equation; ELBO, evidence lower bound; U-Net, U-shaped neural network.

Method name	Year	Brief description	Source Code
Neural Autoregressive Flows [113]	2018	Foundational model	GitHub
Neural (latent) ODE [18]	2018	Foundational model	GitHub
GLOW [114]	2018	Foundational generative flows	GitHub
FFJORD [115]	2019	Improves computational efficiency of [18]	GitHub
Augmented Neural ODEs [116]	2019	Foundational neural ODE	GitHub
Residual Flows [19]	2019	Invertible generative flows	GitHub
Graph Normalizing Flows [117]	2019	Generative geometric flows	Github
Stochastic Normalizing Flows [20, 118]	2020	Generative stochastic flows	GitHub
Equivariant Normalizing Flows [21]	2021	Generative equivariant flows	GitHub
Diffusion Normalizing Flow [119]	2021	Generative stochastic flows	GitHub
Monte Carlo Flows [120]	2022	Flows for MCMC	GitHub
Divergence-free Neural Conservation Laws [121]	2022	Flows with divergence-free neural networks	GitHub
Rectified Flows [22]	2022	Straight and fast flow trajectories	GitHub
Flow Matching for Generative Modeling [23, 43]	2022	Generative flows with Gaussian probability paths	GitHub
Stochastic Interpolants (InterFlow) [60, 122, 123]	2022	Building flows with stochastic interpolants & diffusion	GitHub
RIVER [124]	2022	Flow matching for video prediction	GitHub
Point Straight Flow [125]	2022	Generating point clouds with Straight and fast flows	GitHub
Action Matching [126]	2023	Generative optimal transport with arbitrary paths	GitHub
OT Conditional Flow Matching [24]	2023	Generative (conditional) continuous flows	GitHub
Fast-ODE [127]	2023	Minimizing trajectory curvature of ODE-based generative models	GitHub
Latent Flow Matching [128]	2023	Flow matching in the latent space	GitHub
$[SF]^2M$ [86]	2023	Simulation-free stochastic dynamics	GitHub
Riemannian Flow Matching [48]	2023	Generative geometric flows	GitHub
Equivariant Flow Matching [107]	2023	Generative equivariant flows	GitHub
Wasserstein Lagrangian Flows [129]	2023	Unified flows and optimal transport	GitHub
FM Boosting [130]	2023	Boosting latent diffusion models with flow matching	GitHub
Motion Flow Matching [131]	2023	Flow matching for human motion synthesis and editing	GitHub
InstaFlow [132]	2023	Flows for accelerating text-to-image generation	GitHub
Multimodal Flow Matching [25]	2024	Continuous-discrete data generation	GitHub
Functional Flow Matching [133]	2024	Flow matching for infinite dimensional spaces	GitHub
Dirichlet Flow Matching [70]	2024	Discrete data generation	GitHub
Discrete Flow Matching [52]	2024	Discrete data generation	GitHub
Fisher Flow Matching [71]	2024	Discrete data generation	GitHub
Bellman Optimal Stepsize Straightening [134]	2024	Distilling and fine-tuning flow matching generative models	GitHub
Consistency Flow Matching [135]	2024	Learning strait flows with self-consistency in velocity fields	GitHub
Metric Flow Matching [26]	2024	Flows on the data manifold	GitHub
Depth FM [136]	2024	Flow matching for depth estimation	GitHub
Probabilistic Forecasting with Interpolants [137]	2024	Stochastic interpolants for probabilistic forecasting of dynamical systems	GitHub
SemFlow [138]	2024	Flows for semantic segmentation & semantic image synthesis	GitHub
Preference Flow Matching [139]	2024	Flow matching for preference-based reinforcement learning	GitHub
COT-FM [140]	2024	Conditional optimal transport with simulation-free flows	GitHub
FlowSeq [141]	2024	Flow matching for conditional text generation	GitHub
Statistical Flow Matching [142]	2024	Flows on the manifold of parameterized probability measures	GitHub
Optimal Flow Matching [49]	2024	Shorter flow trajectories	GitHub
Discrete Guidance [143]	2024	Guidance of discrete state space flow and diffusion models	GitHub
Meta Flow Matching [27]	2024	Flows across measures and modeling sample interactions	GitHub
Trajectory Flow Matching [144]	2024	Simulation-free modeling of stochastic time-series	GitHub
Generator Matching [28]	2024	Generalized multi-modal flow matching	GitHub
Wasserstein Flow Matching [29]	2024	Flows for families of distributions	GitHub
Multi-Marginal Flow Matching [90]	2025	Flows with smooth spline-based interpolation	GitHub
Contrastive Flow Matching [145]	2025	Unique conditional flows	GitHub
TFG-Flow [146]	2025	Continous-discrete flow guidance	GitHub
SDE Matching [147]	2025	Simulation-free latent SDEs	GitHub

Table 2 Open-source methods for implementing and improving flow matching models. O(S)DE, ordinary (stochastic) differential equation; MCMC, Markov chain Monte Carlo.

Research area	Applications	Method name	Flow conditioning	Network architecture	Modalities	Data	Performance	Improvement	Source Code
Molecular modeling	Small molecule generation	EquiFM [146]	Unconditioned	Geometric GNN	Sequence, 3D	GEOM-Drugs	98.9% (Validity)	+7%	GitHub
	Small molecule generation	SemlaFlow [64]	Unconditioned	Geometric GNN	Sequence, graph, 3D	GEOM-Drugs	93.9% (Validity)	-1%	GitHub
	Small molecule generation	FlowMol [34]	Unconditioned	Geometric GNN	Sequence, graph, 3D	GEOM-Drugs	51.2% (Validity)	-33%	GitHub
	Small molecule conformer prediction	ET-Flow [96]	Conditioned	Transformer	3D	GEOM-Drugs	75.2% (Precision-Coverage)	+13%	GitHub
	Small molecule & materials generation	ADIT [58]	Unconditioned	All-atom transformer	Sequence, 3D	GEOM-Drugs	95.3% (Validity)	+1%	GitHub
	Protein sequence generation	PenFlow [149]	Conditioned	Transformer	Sequence	SwissProt	4.24 (scPerplexity)	+35%	-
	Protein structure generation	FrameFlow [32]	Unconditioned	SE(3)-equivariant transformer	3D	SCOPe	0.81 (Designability)	+93%	GitHub
	Protein structure generation	FoldFlow [30, 31]	Unconditioned	SE(3)-equivariant transformer	3D	PDB	0.82 (Designability)	+34%	GitHub
	Protein side-chain packing	FlowPacker [57]	Conditioned	SE(3)-equivariant transformer	3D	PDB	0.76Å (RMSD)	+10%	GitHub
	Protein sequence & structure generation	Multiflow [25]	Unconditioned	SE(3)-equivariant transformer	Sequence, 3D	PDB	0.86 (Designability)	+37%	GitHub
	Protein sequence & structure generation	CoFlow [150]	Conditioned	Transformer	Sequence, 3D	MGnify30	3.10Å (scRMSD)	+87%	GitHub
	Protein sequence & structure generation	OriginFlow [68]	Unconditioned/Conditioned	Transformer	3D	PDB	0.45 (FM-Diversity)	+6%	GitHub
	Antibody protein sequence & structure generation	FlowDesign [151]	Conditioned	SE(3)-equivariant transformer	Sequence, 3D	SAbDab	30% (IMP-RI3)	+11%	GitHub
	Peptide protein sequence & structure generation	D-Flow [152]	Conditioned	Transformer	Sequence, 3D	PDB	1.03Å (RMSD)	+21%	GitHub
	Peptide protein sequence & structure generation	PepFlow [153]	Conditioned	SE(3)-equivariant transformer	Sequence, 3D	PDB	2.07Å (RMSD)	+37%	GitHub
	Peptide protein sequence & structure generation	PPFlow [154]	Conditioned	SE(3)-equivariant transformer	Sequence, 3D	PDB	35.43% (IMP%-B)	+23%	GitHub
	DNA sequence generation	Dirichlet FM [70]	Unconditioned	Transformer with FM on simplex	Sequence	lncRNA	1.98 (Perplexity)	+12%	GitHub
	DNA sequence generation	Fisher-Flow [71]	Unconditioned	Hyperspherical (Fisher-Rao) FM	Sequence	lncRNA	1.40 (Perplexity)	+29%	GitHub
	RNA sequence generation	RNACG [72]	Conditioned	Multimodal Diffusion Transformer with Dirichlet FM	Sequence	PDB	69.1% (Recall-All)	+2%	-
	RNA sequence & structure generation	RNAFlow [73]	Conditioned	Geometric GNN	Sequence	PDB	37.0% (Recovery)	+37%	GitHub
	Ligand-conditioned RNA sequence & structure generation	RiboFlow [74]	Conditioned	SE(3) flow with torsion angle modeling	Sequence, 3D	PDB	34.7% (Validity)	+13%	-
	Protein-conditioned RNA sequence & structure generation	RNA-EFM [75]	Conditioned	Energy-based SE(3) flow with protein contacts	Sequence	PDB	40.0% (Recovery)	+8%	GitHub
	RNA structure generation	RNA-FrameFlow [35]	Unconditioned	Torsion-based SE(3)-equivariant transformer	3D	PDB	25.0% (Validity)	First kind	GitHub
	RNA structure prediction	RNAbpFlow [37]	Conditioned	Base-pair-augmented SE(3)-equivariant transformer	3D	PDB	0.38 (TM-score)	+19%	GitHub
	Biomolecular interactions	FlowDock [38]	Conditioned	SE(3)-equivariant transformer	3D	PDB	50.7% (RMSD < 2Å & PB)	+18%	GitHub
	Biomolecular interactions	NeuralPLexer3 [37]	Conditioned	All-atom transformer	3D	PDB	77.9% (RMSD < 2Å & PB)	+7%	GitHub
	Biomolecular interactions	FlexDock [78]	Conditioned	SE(3)-equivariant GNN	3D	PDB	40.0% (RMSD < 2Å & PB)	+800%	GitHub
	Biomolecular interactions	UniMoMo [79]	Conditioned	Equivariant transformer	Sequence, graph, 3D	PepBench/ProtFrag, SAbDab, CrossDocked2020	10.38 (Weighted Score)	+8%	GitHub
	Biomolecular dynamics	AlphaFlow [33]	Conditioned	SE(3)-equivariant transformer	3D	PDB	0.48 (Pairwise RMSD r)	+216%	GitHub
	Biomolecular dynamics	FlowBC [81]	Conditioned	Fully-connected neural network	3D	PDB	0.45 (Mean eigenvalue:0.38)	+5%	GitHub
	Biomolecular dynamics	MDGen [82]	Conditioned	Transformer	3D	MD	0.48 (Pairwise RMSD r)	+0%	GitHub
	Biomolecular dynamics	OM-TPS [83]	Conditioned	Diffusion or flow-based neural network	3D	MD	0.35 (JSD)	+40%	GitHub
	Biomolecular dynamics	UniSim [155]	Conditioned	Geometric GNN	3D	MD	0.33 (JSD, pairwise)	+9%	GitHub
Single-cellular modeling	Cell phenotype modeling	CellFlow [40]	Conditioned	Residual neural network	Sequence	PBMC	0.60 (R^2-DEG)	+200%	GitHub
	Phenotype generation	CFGen [85]	Compositional (multi-attribute)	Flow matching with discrete likelihoods	Sequence	HLCA	10.7 (2-WD)	+9%	GitHub
	Simulation-free flows	CFM [24]	Conditioned	Conditional FM (regression)	Sequence	Multi	0.94 (1-WD)	+13%	GitHub
	Stochastic bridges	SF^{2}M [86]	Unconditioned	Score-FM	Sequence	Multi	0.93 (1-WD)	+1%	GitHub
	Stochastic flow alignment	GENOT [87]	Unconditioned	Entropic Gromov–Wasserstein flows	Sequence	Tree Match	0.16 (FOSCTTM)	+48%	GitHub
	Population-conditioned flow modeling	Meta FM [27]	Population-conditioned	FM-based GNN	Sequence	PDO	4.10 (1-WD)	+3%	GitHub
	Distribution-level modeling	Wasserstein FM [29]	Distributional	Barre–Wasserstein FM	Sequence	MERFISH	54.5 (1-WD)	First kind	GitHub
	Manifold-constrained flows	Metric FM [26]	Metric-conditioned	Geodesic interpolant + flow field	Sequence	Multi (100D)	50.9 (1-WD)	+3%	GitHub
	Interaction-aware bridges	CytoBridge [88]	Conditioned (mean-field)	Unbalanced mean-field Schrödinger bridge	Sequence	Mouse	6.64 (1-WD, $t \simeq 2$)	+2%	GitHub
	Rotational dynamics	Curly-FM [59]	Velocity-conditioned	Schrödinger bridge with drift	Sequence	Erythroid	0.49 (Centre distance, $d = 50$)	+1%	GitHub
	Growth-aware flows	VGFM [89]	Conditioned (growth + velocity)	Flow and growth field regression	Sequence	EB	0.68 (1-WD)	+5%	GitHub
	Multi-context flow learning	MMFM [90]	Conditioned (time + condition)	Global vector field + spline guidance	Sequence	Multi (100D)	79.4 (1-WD)	-38%	GitHub
Multi-cellular modeling	Cell imaging	CellFlux [39]	Conditioned	U-Net	Image	JUMP	84.4 (FID$_c$)	+16%	GitHub
Bioimaging	Cryo-EM denoising & 3D reconstruction	CryoFM [105]	Conditioned	Transformer	3D	EMDB	0.00 (FR)	+100%	-
	Microscopic Image Segmentation	FlowSAM [90]	Conditioned	U-Net	Image	MoNuSeg	81.22 (F1 score)	+6%	GitHub
	Biomedical Image Reconstruction	GTF^{2}M [99]	Unconditioned	U-Net	Image	fastMRI	0.80 (SSIM, 4x)	+1%	-
	Biomedical Image Reconstruction	MMSFlow [100]	Unconditioned	U-Net	Image	fastMRI	0.64 (SSIM, 4x)	+2%	-
	Biomedical Image Synthesis	MOTFM [101]	Conditioned	U-Net	Image	3D MSD MRI Brain	7.93 (3D-FID)	+73%	GitHub

Table 3 Flow matching methods for bioinformatics. A comprehensive taxonomy of molecular, single-cellular, multi-cellular, and bioimaging applications based on flow matching. The data, performance, and improvement columns denote on which dataset a method was trained and/or evaluated, what performance it achieved with that data, and the relative performance improvement of the method compared to a corresponding state-of-the-art method for the application domain. SE(3), the special Euclidean group in $\mathbb{R}^3$; GNN, graph neural network; FM, flow matching; U-Net, U-shaped neural network; SC, self-consistency; PDB, Protein Data Bank; Å, Angstrom; RMSD, root mean square deviation; TM, template modeling score or TM-score; IMP, energy improvement percentage; PB, validity according to the PoseBusters software suite; r, Pearson correlation; MD, molecular dynamics; JSD, Jensen-Shannon divergence; DEG, differentially expressed genes; WD, Wasserstein distance; FID$_c$, conditional Fréchet inception distance; EMDB, Electron Microscopy Data Bank; FR, fail rate; SSIM, structural similarity index measure.

Boxes

Box 1

Deep learning and generative modeling

Deep learning (DL) [41] represents a subset of artificial intelligence and machine learning research focused on developing many-layer (i.e., deep) artificial neural networks. Each neuron in an artificial neural network receives input data features and provides output data representations to a subsequent layer of interconnected neurons. Through their hierarchical structure, deep neural networks can learn increasingly abstract yet informative representations of the network's input data, which can be used for predicting properties of unseen data or generating new data entirely.

In the realm of generative modeling, DL underpins advanced techniques such as diffusion models [47] and flow matching [23]. Diffusion models generate data by simulating a process that gradually transforms simple noise into complex structures, achieving impressive results in image, text, and graph synthesis. Flow matching, a related approach, trains models to learn simpler transport paths between data distributions, offering efficient and scalable solutions for generating high-quality samples. These methods demonstrate the versatility of DL in capturing and reproducing the underlying distributions of complex datasets. Some of the most well-known examples of successful generative DL applications include GPT-4o for multi-modal natural language understanding and reasoning [156], Stable Diffusion for image generation [157], and AlphaFold 3 for biomolecular 3D structure prediction [1].

Box 2

Key concepts related to flow matching models

Generative models [158]: Models that learn to generate new data samples from a learned distribution, often by transforming noise into structured outputs such as images, sequences, or graphs. These form the basis of most generative AI applications to date.

Optimal transport [159]: The study of transforming probability distributions by minimizing a cost function over mappings, often used to compare or interpolate between distributions.

Continuous normalizing flows [18] (CNFs): Generative models that transform a simple base distribution into a target distribution using invertible flows [17, 44, 45] parameterized by neural ordinary differential equations, allowing exact likelihood computation.

Conditional flow matching [43] (CFM): A simulation-free method for training continuous normalizing flows by regressing a vector field to known conditional transitions, enabling efficient and scalable generative modeling.

Rectified flow [22]: A simplified flow model first popularized in the field of computer vision, where data points follow straight-line paths in latent space, which improves sample efficiency and aligns with optimal transport objectives.

Vector field parametrization [43]: A core modeling component in flow matching methods, where a neural network is trained to predict the time-dependent velocity vector that transports data points along learned trajectories from a source to the target distribution.

Diffusion models [12]: A class of generative models that learn to reverse a diffusion (noise injection) process, closely related to flow matching (FM) methods but typically trained with score-based objectives over noisy data.

Transformers [62]: Neural network architectures based on self-attention mechanisms, widely used in sequence, graph, and image modeling, including in flow-based generative tasks due to their scalability and expressiveness.

Graph neural networks [160] (GNNs): Architectures that operate on graph-structured data by iteratively updating node representations via message passing, commonly used when modeling social, physical, or biomolecular structures.

SE(3)-equivariant networks [161]: Neural network models that preserve geometric equivariance under the special Euclidean group in $\mathbb{R}^3$ (i.e., the group SE(3)) consisting of 3D rotations and translations.

Box 3

A pragmatic guide for developing and applying flow matching models in bioinformatics

1. Define the mapping task. Clearly identify the biological states or data distributions between which you aim to learn a mapping (e.g., perturbed vs. unperturbed cellular states, random vs. ground-truth 3D structures, or latent phenotype trajectories). Flow matching is best suited for problems that can be expressed as transporting samples between two distributions in $\mathbb{R}^d$.

2. Choose appropriate data representations. Select feature encodings that capture relevant biological information while being compatible with neural network inputs. For molecules, this may include graph-based representations or 3D atomic coordinates; for cellular data, it may involve gene expression vectors, imaging embeddings, or latent phenotypes.

3. Select a model class and vector field parameterization. Some examples of architectures that have been used previously include SE(3)-equivariant or (rotation-augmented) transformer-based networks for 3D molecular modeling, convolutional or transformer-based encoders for high-dimensional cellular data, and GNNs for biomolecular interaction tasks. Consider ensuring the vector field respects certain (non-trivial) symmetries when necessary, which can be done either through the architecture's design or through input data augmentation.

4. Define the source distribution p_0 and sampling strategy. While a standard Gaussian is often used for p_0, FM permits more biologically meaningful priors (e.g., empirical distributions or structured noise priors) when justified by domain knowledge.

5. Design the training objective and probability path. Choose a probability path p_t (e.g., linear interpolation, optimal transport geodesic) and loss formulation (e.g., squared vector field deviation or dual potential-based objectives) appropriate for the application and model capacity.

6. Validate with domain-relevant metrics. Evaluate generated samples using biologically meaningful metrics, such as structural accuracy [37] (e.g., RMSD, TM-score), functional annotations, or predictive consistency with downstream tasks.

7. Integrate constraints when necessary. For improved biological fidelity, it may be useful to incorporate inductive biases such as geometric constraints, known interaction priors, or energy functions, either directly in the model or during post-processing. However, note that doing so may make model training more complex or challenging [162].

8. Use and contribute to open-source FM libraries. Several frameworks support FM training, including Meta AI's beginner-friendly `flow_matching` GitHub repository. Starting with open-source implementations can greatly accelerate model development and reproducibility.

References

[1] Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., Ronneberger, O., Willmore, L., Ballard, A.J., Bambrick, J., *et al.*: Accurate structure prediction of biomolecular interactions with alphafold 3. Nature **630**(8016), 493–500 (2024)

[2] Krishna, R., Wang, J., Ahern, W., Sturmfels, P., Venkatesh, P., Kalvet, I., Lee, G.R., Morey-Burrows, F.S., Anishchenko, I., Humphreys, I.R., *et al.*: Generalized biomolecular modeling and design with rosettafold all-atom. Science **384**(6693), 2528 (2024)

[3] Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023)

[4] Swanson, K., Wu, W., Bulaong, N.L., Pak, J.E., Zou, J.: The virtual lab of ai agents designs new sars-cov-2 nanobodies. Nature, 1–3 (2025)

[5] Guo, Z., Liu, J., Wang, Y., Chen, M., Wang, D., Xu, D., Cheng, J.: Diffusion models in bioinformatics and computational biology. Nature reviews bioengineering **2**(2), 136–154 (2024)

[6] Frazer, J., Notin, P., Dias, M., Gomez, A., Min, J.K., Brock, K., Gal, Y., Marks, D.S.: Disease variant prediction with deep generative models of evolutionary data. Nature **599**(7883), 91–95 (2021)

[7] Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: International Conference on Learning Representations (2014)

[8] Cui, H., Wang, C., Maan, H., Pang, K., Luo, F., Duan, N., Wang, B.: scgpt: toward building a foundation model for single-cell multi-omics using generative ai. Nature Methods **21**(8), 1470–1480 (2024)

[9] Bommasani, R., Hudson, D.A., Adeli, E., Altman, R., Arora, S., Arx, S., Bernstein, M.S., Bohg, J., Bosselut, A., Brunskill, E., et al.: On the opportunities and risks of foundation models. arXiv preprint arXiv:2108.07258 (2021)

[10] Anand, N., Huang, P.: Generative modeling for protein structures. Advances in neural information processing systems **31** (2018)

[11] Anand, N., Eguchi, R., Mathews, I.I., Perez, C.P., Derry, A., Altman, R.B., Huang, P.-S.: Protein sequence design with a learned potential. Nature communications **13**(1), 746 (2022)

[12] Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020)

[13] Hoogeboom, E., Satorras, V.G., Vignac, C., Welling, M.: Equivariant diffusion for molecule generation in 3d. In: International Conference on Machine Learning, pp. 8867–8887 (2022). PMLR

[14] Watson, J.L., Juergens, D., Bennett, N.R., Trippe, B.L., Yim, J., Eisenach, H.E., Ahern, W., Borst, A.J., Ragotte, R.J., Milles, L.F., *et al.*: De novo design of protein structure and function with rfdiffusion. Nature **620**(7976), 1089–1100 (2023)

[15] Shome, D., Sarkar, P., Etemad, A.: Region-disentangled diffusion model for high-fidelity ppg-to-ecg translation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, pp. 15009–15019 (2024)

[16] Miller, B.K., Chen, R.T.Q., Sriram, A., Wood, B.M.: FlowMM: Generating materials with Riemannian flow matching. In: Proceedings of the 41st International Conference on Machine Learning, vol. 235, pp. 35664–35686 (2024)

[17] Rezende, D., Mohamed, S.: Variational inference with normalizing flows. In: International Conference on Machine Learning, pp. 1530–1538 (2015). PMLR

[18] Chen, R.T., Rubanova, Y., Bettencourt, J., Duvenaud, D.K.: Neural ordinary differential equations. Advances in neural information processing systems **31** (2018). **This article introduces the theoretical foundations necessary for generative flow matching of differential systems.**

[19] Chen, R.T., Behrmann, J., Duvenaud, D.K., Jacobsen, J.-H.: Residual flows for invertible generative modeling. Advances in Neural Information Processing Systems **32** (2019)

[20] Wu, H., Köhler, J., Noé, F.: Stochastic normalizing flows. Advances in neural information processing systems **33**, 5933–5944 (2020)

[21] Garcia Satorras, V., Hoogeboom, E., Fuchs, F., Posner, I., Welling, M.: E (n) equivariant normalizing flows. Advances in Neural Information Processing Systems **34**, 4181–4192 (2021)

[22] Liu, X., Gong, C., *et al.*: Flow straight and fast: Learning to generate and transfer data with rectified flow. In: The Eleventh International Conference on Learning Representations (2023). **This article presents the first few-step formulation of generative flow matching.**

[23] Lipman, Y., Chen, R.T., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. In: The Eleventh International Conference on Learning Representations (2023). **This article reports the first modern formulation of flow matching for generative AI.**

[24] Tong, A., Fatras, K., Malkin, N., Huguet, G., Zhang, Y., Rector-Brooks, J.,

Wolf, G., Bengio, Y.: Improving and generalizing flow-based generative models with minibatch optimal transport. Transactions on Machine Learning Research (2024). **This article introduces the first generally tractable training objective for flow matching in generative AI.**

[25] Campbell, A., Yim, J., Barzilay, R., Rainforth, T., Jaakkola, T.: Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. In: International Conference on Machine Learning, pp. 5453–5512 (2024). PMLR. **This article was the first to generalize flow matching to discrete data types.**

[26] Kapusniak, K., Potaptchik, P., Reu, T., Zhang, L., Tong, A., Bronstein, M., Bose, J., Di Giovanni, F.: Metric flow matching for smooth interpolations on the data manifold. Advances in Neural Information Processing Systems **37**, 135011–135042 (2024)

[27] Atanackovic, L., Zhang, X., Amos, B., Blanchette, M., Lee, L.J., Bengio, Y., Tong, A., Neklyudov, K.: Meta flow matching: Integrating vector fields on the wasserstein manifold. In: The Thirteenth International Conference on Learning Representations (2024)

[28] Holderrieth, P., Havasi, M., Yim, J., Shaul, N., Gat, I., Jaakkola, T., Karrer, B., Chen, R.T.Q., Lipman, Y.: Generator matching: Generative modeling with arbitrary markov processes. In: The Thirteenth International Conference on Learning Representations (2025). **This article comprehensively generalizes flow matching across modalities and data types.** https://openreview.net/forum?id=RuP17cJtZo

[29] Haviv, D., Pooladian, A.-A., Pe'er, D., Amos, B.: Wasserstein flow matching: Generative modeling over families of distributions. arXiv preprint arXiv:2411.00698 (2024)

[30] Bose, J., Akhound-Sadegh, T., Huguet, G., Fatras, K., Rector-Brooks, J., Liu, C.-H., Nica, A.C., Korablyov, M., Bronstein, M.M., Tong, A.: Se(3)-stochastic flow matching for protein backbone generation. In: The Twelfth International Conference on Learning Representations (2024). **This article reports one of the first applications of geometric generative flow matching in bioinformatics.**

[31] Huguet, G., Vuckovic, J., Fatras, K., Thibodeau-Laufer, E., Lemos, P., Islam, R., Liu, C., Rector-Brooks, J., Akhound-Sadegh, T., Bronstein, M., *et al.*: Sequence-augmented se (3)-flow matching for conditional protein generation. Advances in neural information processing systems **37**, 33007–33036 (2024)

[32] Yim, J., Campbell, A., Foong, A.Y., Gastegger, M., Jiménez-Luna, J., Lewis, S., Satorras, V.G., Veeling, B.S., Barzilay, R., Jaakkola, T., et al.: Fast protein backbone generation with se (3) flow matching. arXiv preprint arXiv:2310.05297

(2023)

[33] Jing, B., Berger, B., Jaakkola, T.: Alphafold meets flow matching for generating protein ensembles. In: Forty-first International Conference on Machine Learning (2024)

[34] Dunn, I., Koes, D.R.: Mixed continuous and categorical flow matching for 3d de novo molecule generation. ArXiv, 2404 (2024)

[35] Anand, R., Joshi, C.K., Morehead, A., Jamasb, A.R., Harris, C., Mathis, S.V., Didi, K., Hooi, B., Lio, P.: RNA-frameflow: Flow matching for de novo 3d RNA backbone design. Transactions on Machine Learning Research (2025)

[36] Morehead, A., Cheng, J.: Flowdock: Geometric flow matching for generative protein-ligand docking and affinity prediction. In: Intelligent Systems for Molecular Biology (ISMB) (2025)

[37] Qiao, Z., Ding, F., Dresselhaus, T., Rosenfeld, M.A., Han, X., Howell, O., Iyengar, A., Opalenski, S., Christensen, A.S., Krishna Sirumalla, S., et al.: Neuralplexer3: Physio-realistic biomolecular complex structure prediction with flow models. arXiv e-prints, 2412 (2024)

[38] Zhang, Y., Su, Y., Wang, C., Li, T., Wefers, Z., Nirschl, J., Burgess, J., Ding, D., Lozano, A., Lundberg, E., *et al.*: Cellflux: Simulating cellular morphology changes via flow matching. In: Proceedings of the 42nd International Conference on Machine Learning (2025)

[39] Petrović, K., Atanackovic, L., Kapusniak, K., Bronstein, M.M., Bose, J., Tong, A.: Curly flow matching for learning non-gradient field dynamics. In: ICLR 2025 Workshop on Machine Learning for Genomics Explorations (2025)

[40] Klein, D., Fleck, J.S., Bobrovskiy, D., Zimmermann, L., Becker, S., Palma, A., Dony, L., Tejada-Lapuerta, A., Huguet, G., Lin, H.-C., et al.: Cellflow enables generative single-cell phenotype modeling with flow matching. bioRxiv, 2025–04 (2025)

[41] LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. nature **521**(7553), 436–444 (2015). **This article characterizes foundational algorithms and applications of deep learning in artificial intelligence and beyond.**

[42] Bunne, C., Roohani, Y., Rosen, Y., Gupta, A., Zhang, X., Roed, M., Alexandrov, T., AlQuraishi, M., Brennan, P., Burkhardt, D.B., *et al.*: How to build the virtual cell with artificial intelligence: Priorities and opportunities. Cell **187**(25), 7045–7063 (2024). **This article presents the first formulation of virtual cell modeling with artificial intelligence.**

[43] Lipman, Y., Havasi, M., Holderrieth, P., Shaul, N., Le, M., Karrer, B., Chen,

R.T., Lopez-Paz, D., Ben-Hamu, H., Gat, I.: Flow matching guide and code. arXiv preprint arXiv:2412.06264 (2024)

[44] Tabak, E.G., Vanden-Eijnden, E.: Density estimation by dual ascent of the log-likelihood. Commun. Math. Sci. **8**(1), 217–233 (2010)

[45] Tabak, E.G., Turner, C.V.: A family of nonparametric density estimation algorithms. Communications on Pure and Applied Mathematics **66**(2), 145–164 (2013)

[46] Peluchetti, S.: Non-denoising forward-time diffusions. arXiv preprint arXiv:2312.14589 (2023)

[47] Yang, L., Zhang, Z., Song, Y., Hong, S., Xu, R., Zhao, Y., Zhang, W., Cui, B., Yang, M.-H.: Diffusion models: A comprehensive survey of methods and applications. ACM Computing Surveys **56**(4), 1–39 (2023)

[48] Chen, R.T.Q., Lipman, Y.: Flow matching on general geometries. In: The Twelfth International Conference on Learning Representations (2024). **This article reports a theoretical framework crucial to developing flow matching models for geometric data.** https://openreview.net/forum?id=g7ohDlTITL

[49] Kornilov, N., Mokrov, P., Gasnikov, A., Korotin, A.: Optimal flow matching: Learning straight trajectories in just one step. Advances in Neural Information Processing Systems **37**, 104180–104204 (2024)

[50] Hoffman, M.D., Blei, D.M., Wang, C., Paisley, J.: Stochastic variational inference. the Journal of machine Learning research **14**(1), 1303–1347 (2013)

[51] Iserles, A.: A First Course in the Numerical Analysis of Differential Equations. Cambridge University Press, Cambridge (2009)

[52] Gat, I., Remez, T., Shaul, N., Kreuk, F., Chen, R.T., Synnaeve, G., Adi, Y., Lipman, Y.: Discrete flow matching. Advances in Neural Information Processing Systems **37**, 133345–133385 (2024)

[53] Dunn, I., Koes, D.R.: Exploring discrete flow matching for 3d de novo molecule generation. ArXiv, 2411 (2024)

[54] Stark, H., Jing, B., Barzilay, R., Jaakkola, T.: Harmonic self-conditioned flow matching for joint multi-ligand docking and binding site design. In: Forty-first International Conference on Machine Learning (2024)

[55] Peng, F.Z., Bezemek, Z., Rector-Brooks, J., Zhang, S., Zhang, A.R., Bronstein, M., Bose, A.J., Tong, A.: Planner aware path learning in diffusion language models training. arXiv preprint arXiv:2509.23405 (2025)

[56] Yim, J., Trippe, B.L., De Bortoli, V., Mathieu, E., Doucet, A., Barzilay, R., Jaakkola, T.: Se(3) diffusion model with application to protein backbone generation. arXiv preprint arXiv:2302.02277 (2023)

[57] Lee, J.S., Kim, P.M.: Flowpacker: protein side-chain packing with torsional flow matching. Bioinformatics **41**(3), 010 (2025)

[58] Joshi, C.K., Fu, X., Liao, Y.-L., Gharakhanyan, V., Miller, B.K., Sriram, A., Ulissi, Z.W.: All-atom diffusion transformers: Unified generative modelling of molecules and materials. In: International Conference on Machine Learning (2025). PMLR

[59] Yim, J., Stärk, H., Corso, G., Jing, B., Barzilay, R., Jaakkola, T.S.: Diffusion models in protein structure and docking. Wiley Interdisciplinary Reviews: Computational Molecular Science **14**(2), 1711 (2024)

[60] Albergo, M.S., Vanden-Eijnden, E.: Building normalizing flows with stochastic interpolants. arXiv preprint arXiv:2209.15571 (2022)

[61] Geng, Z., Deng, M., Bai, X., Kolter, J.Z., He, K.: Mean flows for one-step generative modeling. arXiv preprint arXiv:2505.13447 (2025)

[62] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)

[63] Wu, K., Karapetyan, E., Schloss, J., Vadgama, J., Wu, Y.: Advancements in small molecule drug design: A structural perspective. Drug Discovery Today **28**(10), 103730 (2023)

[64] Irwin, R., Tibo, A., Janet, J.P., Olsson, S.: Semlaflow–efficient 3d molecular generation with latent attention and equivariant flow matching. arXiv preprint arXiv:2406.07266 (2024)

[65] Jing, B., Eismann, S., Suriana, P., Townshend, R.J.L., Dror, R.: Learning from protein structure with geometric vector perceptrons. In: International Conference on Learning Representations (2021)

[66] Hassan, M., Shenoy, N., Lee, J., Stärk, H., Thaler, S., Beaini, D.: Et-flow: Equivariant flow-matching for molecular conformer generation. In: Advances in Neural Information Processing Systems, vol. 37, pp. 128798–128824 (2024)

[67] Huang, P.-S., Boyken, S.E., Baker, D.: The coming of age of de novo protein design. Nature **537**(7620), 320–327 (2016)

[68] Yan, J., Cui, Z., Yan, W., Chen, Y., Pu, M., Li, S., Ye, S.: Robust and Reliable de novo Protein Design: A Flow-Matching-Based Protein Generative Model

Achieves Remarkably High Success Rates. bioRxiv. preprint, 2025-04 (2025)

[69] Barrangou, R., Doudna, J.A.: Applications of crispr technologies in research and beyond. Nature biotechnology **34**(9), 933–941 (2016)

[70] Stark, H., Jing, B., Wang, C., Corso, G., Berger, B., Barzilay, R., Jaakkola, T.: Dirichlet flow matching with applications to dna sequence design. In: International Conference on Machine Learning, pp. 46495–46513 (2024). PMLR

[71] Davis, O., Kessler, S., Petrache, M., Ceylan, I., Bronstein, M., Bose, J.: Fisher flow matching for generative modeling over discrete data. Advances in Neural Information Processing Systems **37**, 139054–139084 (2024)

[72] Gao, L., Lu, Z.J.: Rnacg: A universal rna sequence conditional generation model based on flow-matching. arXiv preprint arXiv:2407.19838 (2024)

[73] Nori, D., Jin, W.: RNAFlow: RNA Structure & Sequence Design via Inverse Folding-based Flow Matching. arXiv preprint arXiv:2405.18768 (2024)

[74] Ma, R., Zhang, Z., Wang, Z., Hua, C., Zhou, Z., Cao, F., Rao, J., Zheng, S.: Riboflow: Conditional de novo rna sequence-structure co-design via synergistic flow matching. arXiv preprint arXiv:2503.17007 (2025)

[75] Abir, A.R., Zhang, L.: Rna-efm: Energy based flow matching for protein-conditioned rna sequence-structure co-design. bioRxiv, 2025–02 (2025)

[76] Tarafder, S., Bhattacharya, D.: Rnabpflow: Base pair-augmented se (3)-flow matching for conditional rna 3d structure generation. bioRxiv (2025)

[77] Morehead, A., Giri, N., Liu, J., Cheng, J.: Deep learning for protein-ligand docking: Are we there yet? In: ICML AI for Science Workshop (2024)

[78] Corso, G., Somnath, V.R., Getz, N., Barzilay, R., Jaakkola, T., Krause, A.: Composing unbalanced flows for flexible docking and relaxation. In: The Thirteenth International Conference on Learning Representations (2025). https://openreview.net/forum?id=gHLWTzKiZV

[79] Kong, X., Zhang, Z., Zhang, Z., Jiao, R., Ma, J., Huang, W., Liu, K., Liu, Y.: Unimomo: Unified generative modeling of 3d molecules for de novo binder design. In: Forty-second International Conference on Machine Learning (2025). https://openreview.net/forum?id=KUN7A7Okb6

[80] Mortier, J., Rakers, C., Bermudez, M., Murgueitio, M.S., Riniker, S., Wolber, G.: The impact of molecular dynamics on drug design: applications for the characterization of ligand–macromolecule complexes. Drug discovery today **20**(6), 686–702 (2015)

[81] Zhang, M., Zhang, Z., Wu, H., Wang, Y.: Flow matching for optimal reaction coordinates of biomolecular systems. Journal of Chemical Theory and Computation **21**(1), 399–412 (2024)

[82] Jing, B., Stärk, H., Jaakkola, T., Berger, B.: Generative modeling of molecular dynamics trajectories. Advances in Neural Information Processing Systems **37**, 40534–40564 (2024)

[83] Raja, S., Sipka, M., Psenka, M., Kreiman, T., Pavelka, M., Krishnapriyan, A.S.: Action-minimization meets generative modeling: Efficient transition path sampling with the onsager-machlup functional. In: Forty-second International Conference on Machine Learning (2025)

[84] Yofe, I., Dahan, R., Amit, I.: Single-cell genomic approaches for developing the next generation of immunotherapies. Nature medicine **26**(2), 171–177 (2020)

[85] Palma, A., Richter, T., Zhang, H., Lubetzki, M., Tong, A., Dittadi, A., Theis, F.J.: Multi-modal and multi-attribute generation of single cells with CFGen. In: The Thirteenth International Conference on Learning Representations (2025). https://openreview.net/forum?id=3MnMGLctKb

[86] Tong, A., Malkin, N., Fatras, K., Atanackovic, L., Zhang, Y., Huguet, G., Wolf, G., Bengio, Y.: Simulation-free schrödinger bridges via score and flow matching. In: AISTATS, pp. 1279–1287 (2024). https://proceedings.mlr.press/v238/y-tong24a.html

[87] Klein, D., Uscidda, T., Theis, F., Cuturi, M.: Genot: Entropic (gromov) wasserstein flow matching with applications to single-cell genomics. Advances in Neural Information Processing Systems **37**, 103897–103944 (2024)

[88] Zhang, Z., Wang, Z., Sun, Y., Li, T., Zhou, P.: Modeling cell dynamics and interactions with unbalanced mean field schr\" odinger bridge. arXiv preprint arXiv:2505.11197 (2025)

[89] Wang, D., Jiang, Y., Zhang, Z., Gu, X., Zhou, P., Sun, J.: Joint velocity-growth flow matching for single-cell dynamics modeling. arXiv preprint arXiv:2505.13413 (2025)

[90] Rohbeck, M., De Brouwer, E., Bunne, C., Huetter, J.-C., Biton, A., Chen, K.Y., Regev, A., Lopez, R.: Modeling complex system dynamics with flow matching across time and conditions. In: The Thirteenth International Conference on Learning Representations (2025)

[91] Kim, S.-j., Kim, E.M., Yamamoto, M., Park, H., Shin, H.: Engineering multicellular spheroids for tissue engineering and regenerative medicine. Advanced Healthcare Materials **9**(23), 2000608 (2020)

[92] Bourou, A., Boyer, T., Gheisari, M., Daupin, K., Dubreuil, V., De Thonel, A., Mezger, V., Genovesio, A.: Phendiff: Revealing subtle phenotypes with diffusion models in real images, pp. 358–367. Springer, Berlin, Heidelberg (2024). https://doi.org/10.1007/978-3-031-72384-1_34

[93] Palma, A., Theis, F.J., Lotfollahi, M.: Predicting cell morphological responses to perturbations using generative modeling. Nat. Commun. **16**(1), 505 (2025)

[94] Weissleder, R., Nahrendorf, M.: Advancing biomedical imaging. Proceedings of the National Academy of Sciences **112**(47), 14424–14428 (2015)

[95] Zhou, Y., Li, Y., Yuan, J., Gu, Q.: CryoFM: A flow-based foundation model for cryo-EM densities. In: The Thirteenth International Conference on Learning Representations (2025). https://openreview.net/forum?id=T4sMzjy7fO

[96] Bogensperger, L., Narnhofer, D., Falk, A., Schindler, K., Pock, T.: Flowsdf: Flow matching for medical image segmentation using distance transforms. International Journal of Computer Vision **133**(7), 4864–4876 (2025) https://doi.org/10.1007/s11263-025-02373-y

[97] Kumar, N., Verma, R., Sharma, S., Bhargava, S., Vahadane, A., Sethi, A.: A dataset and a technique for generalized nuclear segmentation for computational pathology. IEEE Transactions on Medical Imaging **36**(7), 1550–1560 (2017) https://doi.org/10.1109/TMI.2017.2677499

[98] Sirinukunwattana, K., Pluim, J.P.W., Chen, H., Qi, X., Heng, P.-A., Guo, Y.B., Wang, L.Y., Matuszewski, B.J., Bruni, E., Sanchez, U., Böhm, A., Ronneberger, O., Cheikh, B.B., Racoceanu, D., Kainz, P., Pfeiffer, M., Urschler, M., Snead, D.R.J., Rajpoot, N.M.: Gland segmentation in colon histology images: The glas challenge contest. Medical Image Analysis **35**, 489–502 (2017) https://doi.org/10.1016/j.media.2016.08.008

[99] Luo, X., Li, Y., Qin, C.: Unsupervised accelerated mri reconstruction via ground-truth-free flow matching. In: International Conference on Information Processing in Medical Imaging, pp. 127–141 (2025). Springer

[100] Zhang, D., Han, Q., Xiong, Y., Du, H.: Mutli-modal straight flow matching for accelerated mr imaging. Computers in Biology and Medicine **178**, 108668 (2024) https://doi.org/10.1016/j.compbiomed.2024.108668

[101] Yazdani, M., Medghalchi, Y., Ashrafian, P., Hacihaliloglu, I., Shahriari, D.: Flow matching for medical image synthesis: Bridging the gap between speed and quality. arXiv preprint arXiv:2503.00266 (2025)

[102] Qu, E., Krishnapriyan, A.: The importance of being scalable: Improving the speed and accuracy of neural network interatomic potentials across chemical

domains. Advances in Neural Information Processing Systems **37**, 139030–139053 (2024)

[103] Doudna, J.A., Charpentier, E.: The new frontier of genome engineering with crispr-cas9. Science **346**(6213), 1258096 (2014)

[104] Paszke, A.: Pytorch: An imperative style, high-performance deep learning library. arXiv preprint arXiv:1912.01703 (2019)

[105] Lim, S.H., Wang, Y., Yu, A., Hart, E., Mahoney, M.W., Li, S., Erichson, N.B.: Elucidating the design choice of probability paths in flow matching for forecasting. Transactions on Machine Learning Research (2025)

[106] Kruiff, F., Bekkers, E., Öktem, O., Schönlieb, C.-B., Diepeveen, W.: Pullback flow matching on data manifolds. arXiv preprint arXiv:2410.04543 (2024)

[107] Klein, L., Krämer, A., Noé, F.: Equivariant flow matching. Advances in Neural Information Processing Systems **36**, 59886–59910 (2023)

[108] Zhu, H., Wang, F., Ding, T., Qu, Q., Zhu, Z.: Analyzing and improving model collapse in rectified flow models. arXiv e-prints, 2412 (2024)

[109] Zhang, H., Siarohin, A., Menapace, W., Vasilkovsky, M., Tulyakov, S., Qu, Q., Skorokhodov, I.: Alphaflow: Understanding and improving meanflow models. arXiv preprint arXiv:2510.20771 (2025)

[110] Reu, T., Dromigny, S., Bronstein, M.M., Vargas, F.: Flows don't cross in high dimension. In: ICLR Workshop on Deep Generative Model in Machine Learning: Theory, Principle and Efficacy (2025)

[111] Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. Advances in neural information processing systems **34**, 8780–8794 (2021)

[112] Yacoby, Y., Pan, W., Doshi-Velez, F.: Failure modes of variational autoencoders and their effects on downstream tasks. arXiv preprint arXiv:2007.07124 (2020)

[113] Huang, C.-W., Krueger, D., Lacoste, A., Courville, A.: Neural autoregressive flows. In: International Conference on Machine Learning, pp. 2078–2087 (2018). PMLR

[114] Kingma, D.P., Dhariwal, P.: Glow: Generative flow with invertible 1x1 convolutions. Advances in neural information processing systems **31** (2018)

[115] Grathwohl, W., Chen, R.T., Bettencourt, J., Sutskever, I., Duvenaud, D.: Ffjord: Free-form continuous dynamics for scalable reversible generative models. In: International Conference on Learning Representations (2019)

[116] Dupont, E., Doucet, A., Teh, Y.W.: Augmented neural odes. Advances in neural

information processing systems **32** (2019)

[117] Liu, J., Kumar, A., Ba, J., Kiros, J., Swersky, K.: Graph normalizing flows. Advances in Neural Information Processing Systems **32** (2019)

[118] Hodgkinson, L., Heide, C., Roosta, F., Mahoney, M.W.: Stochastic continuous normalizing flows: training sdes as odes. In: Uncertainty in Artificial Intelligence, pp. 1130–1140 (2021). PMLR

[119] Zhang, Q., Chen, Y.: Diffusion normalizing flow. Advances in neural information processing systems **34**, 16280–16291 (2021)

[120] Gabrié, M., Rotskoff, G.M., Vanden-Eijnden, E.: Adaptive monte carlo augmented with normalizing flows. Proceedings of the National Academy of Sciences **119**(10), 2109420119 (2022)

[121] Richter-Powell, J., Lipman, Y., Chen, R.T.: Neural conservation laws: A divergence-free perspective. Advances in Neural Information Processing Systems **35**, 38075–38088 (2022)

[122] Albergo, M.S., Boffi, N.M., Vanden-Eijnden, E.: Stochastic interpolants: A unifying framework for flows and diffusions. arXiv preprint arXiv:2303.08797 (2023)

[123] Albergo, M.S., Goldstein, M., Boffi, N.M., Ranganath, R., Vanden-Eijnden, E.: Stochastic interpolants with data-dependent couplings. In: International Conference on Machine Learning, pp. 921–937 (2024). PMLR

[124] Davtyan, A., Sameni, S., Favaro, P.: Efficient video prediction via sparsely conditioned flow matching. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 23263–23274 (2023)

[125] Wu, L., Wang, D., Gong, C., Liu, X., Xiong, Y., Ranjan, R., Krishnamoorthi, R., Chandra, V., Liu, Q.: Fast point cloud generation with straight flows. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9445–9454 (2023)

[126] Neklyudov, K., Brekelmans, R., Severo, D., Makhzani, A.: Action matching: Learning stochastic dynamics from samples. In: International Conference on Machine Learning, pp. 25858–25889 (2023). PMLR

[127] Lee, S., Kim, B., Ye, J.C.: Minimizing trajectory curvature of ode-based generative models. In: International Conference on Machine Learning, pp. 18957–18973 (2023). PMLR

[128] Dao, Q., Phung, H., Nguyen, B., Tran, A.: Flow matching in latent space. arXiv preprint arXiv:2307.08698 (2023)

[129] Neklyudov, K., Brekelmans, R., Tong, A., Atanackovic, L., Liu, Q., Makhzani, A.: A computational framework for solving wasserstein lagrangian flows. arXiv preprint arXiv:2310.10649 (2023)

[130] Fischer, J.S., Gui, M., Ma, P., Stracke, N., Baumann, S.A., Ommer, B.: Boosting latent diffusion with flow matching. arXiv preprint arXiv:2312.07360 (2023)

[131] Hu, V.T., Yin, W., Ma, P., Chen, Y., Fernando, B., Asano, Y.M., Gavves, E., Mettes, P., Ommer, B., Snoek, C.G.: Motion flow matching for human motion synthesis and editing. arXiv preprint arXiv:2312.08895 (2023)

[132] Liu, X., Zhang, X., Ma, J., Peng, J., *et al.*: Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In: The Twelfth International Conference on Learning Representations (2023)

[133] Kerrigan, G., Migliorini, G., Smyth, P.: Functional flow matching. In: International Conference on Artificial Intelligence and Statistics, pp. 3934–3942 (2024). PMLR

[134] Nguyen, B., Nguyen, B., Nguyen, V.A.: Bellman optimal stepsize straightening of flow-matching models. arXiv preprint arXiv:2312.16414 (2023)

[135] Yang, L., Zhang, Z., Zhang, Z., Liu, X., Xu, M., Zhang, W., Meng, C., Ermon, S., Cui, B.: Consistency flow matching: Defining straight flows with velocity consistency. arXiv preprint arXiv:2407.02398 (2024)

[136] Gui, M., Schusterbauer, J., Prestel, U., Ma, P., Kotovenko, D., Grebenkova, O., Baumann, S.A., Hu, V.T., Ommer, B.: Depthfm: Fast generative monocular depth estimation with flow matching. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 39, pp. 3203–3211 (2025)

[137] Chen, Y., Goldstein, M., Hua, M., Albergo, M.S., Boffi, N.M., Vanden-Eijnden, E.: Probabilistic forecasting with stochastic interpolants and föllmer processes. In: Proceedings of the 41st International Conference on Machine Learning, pp. 6728–6756 (2024)

[138] Wang, C., Li, X., Qi, L., Ding, H., Tong, Y., Yang, M.-H.: Semflow: Binding semantic segmentation and image synthesis via rectified flow. Advances in Neural Information Processing Systems **37**, 138981–139001 (2024)

[139] Kim, M., Lee, Y., Kang, S., Oh, J., Chong, S., Yun, S.-Y.: Preference alignment with flow matching. Advances in Neural Information Processing Systems **37**, 35140–35164 (2024)

[140] Kerrigan, G., Migliorini, G., Smyth, P.: Dynamic conditional optimal transport through simulation-free flows. Advances in Neural Information Processing Systems **37**, 93602–93642 (2024)

[141] Hu, V., Wu, D., Asano, Y., Mettes, P., Fernando, B., Ommer, B., Snoek, C.: Flow matching for conditional text generation in a few sampling steps. In: Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 380–392 (2024)

[142] Cheng, C., Li, J., Peng, J., Liu, G.: Categorical flow matching on statistical manifolds. Advances in Neural Information Processing Systems **37**, 54787–54819 (2024)

[143] Nisonoff, H., Xiong, J., Allenspach, S., Listgarten, J.: Unlocking guidance for discrete state-space diffusion and flow models. In: The Thirteenth International Conference on Learning Representations (2025)

[144] Zhang, X.N., Pu, Y., Kawamura, Y., Loza, A., Bengio, Y., Shung, D., Tong, A.: Trajectory flow matching with applications to clinical time series modelling. Advances in Neural Information Processing Systems **37**, 107198–107224 (2024)

[145] Stoica, G., Ramanujan, V., Fan, X., Farhadi, A., Krishna, R., Hoffman, J.: Contrastive flow matching. arXiv preprint arXiv:2506.05350 (2025) arXiv:2506.05350 [cs.CV]

[146] Lin, H., Li, S., Ye, H., Yang, Y., Ermon, S., Liang, Y., Ma, J.: Tfg-flow: Training-free guidance in multimodal generative flow. arXiv preprint arXiv:2501.14216 (2025)

[147] Bartosh, G., Vetrov, D., Naesseth, C.A.: Sde matching: Scalable and simulation-free training of latent stochastic differential equations. arXiv preprint arXiv:2502.02472 (2025)

[148] Song, Y., Gong, J., Xu, M., Cao, Z., Lan, Y., Ermon, S., Zhou, H., Ma, W.-Y.: Equivariant flow matching with hybrid probability transport for 3d molecule generation. Advances in Neural Information Processing Systems **36**, 549–568 (2023)

[149] Kong, Z., Zhu, Y., Xu, Y., Zhou, H., Yin, M., Wu, J., Xu, H., Hsieh, C.-Y., Hou, T., Wu, J.: ProtFlow: Fast Protein Sequence Design via Flow Matching on Compressed Protein Language Model Embeddings. arXiv preprint arXiv:2504.10983 (2025)

[150] Yang, S., Ju, L., Cheng, P., Zhou, J., Cai, Y., Feng, D.: Co-design protein sequence and structure in discrete space via generative flow. Bioinformatics **41**(5), 248 (2025)

[151] Wu, J., Kong, X., Sun, N., Wei, J., Shan, S., Feng, F., Wu, F., et al.: Flowdesign: Improved design of antibody cdrs through flow matching and better prior distributions. Cell Systems **16**(6) (2025)

[152] Wu, F., Xu, T., Jin, S., Tang, X., Xu, Z., Zou, J., Hie, B.: D-Flow: Multi-modality Flow Matching for D-peptide Design. arXiv preprint arXiv:2411.10618 (2024)

[153] Li, J., Cheng, C., Wu, Z., Guo, R., Luo, S., Ren, Z., Peng, J., Ma, J.: Full-atom peptide design based on multi-modal flow matching. In: Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., Berkenkamp, F. (eds.) Proceedings of the 41st International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 235, pp. 27615–27640. PMLR, ??? (2024). https://proceedings.mlr.press/v235/li24o.html

[154] Lin, H., Zhang, O., Zhao, H., Jiang, D., Wu, L., Liu, Z., Huang, Y., Li, S.Z.: PPFLOW: Target-aware peptide design with torsional flow matching. In: Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., Berkenkamp, F. (eds.) Proceedings of the 41st International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 235, pp. 30510–30528. PMLR, ??? (2024). https://proceedings.mlr.press/v235/lin24z.html

[155] Yu, Z., Huang, W., Liu, Y.: Unisim: A unified simulator for time-coarsened dynamics of biomolecules. In: Forty-second International Conference on Machine Learning (2025). https://openreview.net/forum?id=W598uW37zt

[156] Hurst, A., Lerer, A., Goucher, A.P., Perelman, A., Ramesh, A., Clark, A., Ostrow, A., Welihinda, A., Hayes, A., Radford, A., et al.: Gpt-4o system card. arXiv preprint arXiv:2410.21276 (2024)

[157] Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., *et al.*: Scaling rectified flow transformers for high-resolution image synthesis. In: Forty-first International Conference on Machine Learning (2024)

[158] Salakhutdinov, R.: Learning deep generative models. Annual Review of Statistics and Its Application **2**(1), 361–385 (2015)

[159] Peyré, G., Cuturi, M., *et al.*: Computational optimal transport: With applications to data science. Foundations and Trends® in Machine Learning **11**(5-6), 355–607 (2019)

[160] Corso, G., Stark, H., Jegelka, S., Jaakkola, T., Barzilay, R.: Graph neural networks. Nature Reviews Methods Primers **4**(1), 17 (2024)

[161] Geiger, M., Smidt, T.: e3nn: Euclidean neural networks. arXiv preprint arXiv:2207.09453 (2022)

[162] Krishnapriyan, A., Gholami, A., Zhe, S., Kirby, R., Mahoney, M.W.: Characterizing possible failure modes in physics-informed neural networks. Advances in neural information processing systems **34**, 26548–26560 (2021)