STRUCTURED FLOW AUTOENCODERS: LEARNING STRUCTURED PROBABILISTIC REPRESENTATIONS WITH FLOW MATCHING

Anonymous authors
Paper under double-blind review

ABSTRACT

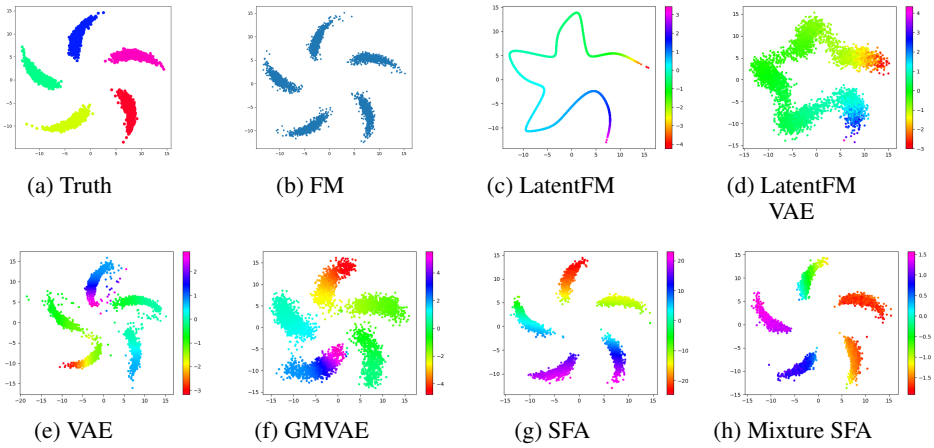
Flow matching is a powerful approach for high-fidelity density estimation, but it often fails to capture the latent structure of complex data. Probabilistic models like variational autoencoders (VAEs), on the other hand, learn structured representations but underperform in sample quality. We propose Structured Flow Autoencoders (SFA), a family of probabilistic models that augments graphical models with conditional continuous normalizing flow (CNF) likelihoods, enabling flow-matching-based structured representation learning. At the core of SFA is a novel flow matching objective that explicitly accounts for latent variables, allowing joint learning of the CNF likelihood and posterior. SFA applies broadly to graphical models with continuous and mixture latents, as well as latent dynamical systems. Empirical studies across image, video, and RNA-seq data show that SFA consistently outperforms VAEs and their structured extensions in both generation quality, representation utility, and scalability to large datasets. Compared to generative models like latent flow matching (LFM), SFA also produces more diverse samples, suggesting better coverage of the data distribution.

1 INTRODUCTION

Generative modeling has become a foundational pillar of modern machine learning, offering powerful tools for capturing complex data distributions and generating high-quality samples. Among recent advances, diffusion models (Ho et al., 2020; Nichol & Dhariwal, 2021; Song et al., 2020; 2021b;a; Austin et al., 2021) and flow-based methods (Lipman et al., 2022; Liu et al., 2022; Gat et al., 2024; Tong et al., 2024; Isobe et al., 2024) have shown remarkable performance as neural density estimators, excelling at likelihood estimation and high-fidelity sample generation. In particular, flow matching has emerged as a scalable and efficient approach, aligning vector fields of probability paths using optimal transport principles, enabling efficient and scalable generative modeling with exact likelihood evaluation (Lipman et al., 2022; Liu et al., 2022; Gat et al., 2024).

Despite their success in generation quality, neural density estimators like flow matching often fall short in *structured representation learning*, failing to capture or expose the rich latent structures underlying complex data. This limitation is especially salient in scientific and structured domains such as computational biology, where interpretable low-dimensional representations are essential for downstream tasks, analysis, and control. Recent work has revealed both empirical evidence of implicit low-dimensional structures in pretrained diffusion models (Wang & Vastola, 2023; Chen et al., 2024) and theoretical guarantees of their adaptivity to such structures (Wang et al., 2024; Li & Yan, 2024). However, these models neither explicitly model latent structure during training nor produce readily interpretable representations, limiting their utility beyond sample generation.

In contrast, probabilistic latent-variable models such as variational autoencoders (VAEs) (Kingma & Welling, 2013; Johnson et al., 2016) are explicitly designed to capture latent structure through probabilistic encoder-decoder architectures. These models learn structured probabilistic representations that can be leveraged for conditional generation and downstream tasks. However, VAEs typically underperform in data modeling and generation fidelity compared to modern flow-based models, limiting their utility in high-resolution or diverse generative tasks. This gap in generative fidelity also raises concerns about the reliability and expressiveness of their learned representations.



(a) Truth (b) FM (c) LatentFM (d) LatentFM VAE

(e) VAE (f) GMVAE (g) SFA (h) Mixture SFA

Figure 1: Generated samples on the Pinwheel dataset with 5 clusters. Color in (a) indicates class membership, which is not provided during training. Color in (c) indicates the latent distribution learned via deterministic autoencoder. Color in (d)-(h) indicates the generated posterior value $\boldsymbol{z}_1 \sim q(\boldsymbol{z}_1|\boldsymbol{x}_1)$ given the generated sample $\boldsymbol{x}_1$. We use a continuous latent variable $\boldsymbol{z}$ in (c),(d),(e),(g); and a mixture latent variable in (f),(h).

This performance gap raises the question: *Can we build models that retain the structured latent representations of VAEs while achieving the high fidelity and scalability of flow matching?*

Main idea. We propose *structured flow autoencoders (SFA)*, a new family of probabilistic models that augments graphical models with conditional Continuous Normalizing Flow (CNF) likelihoods. This family aims to combine the strengths of both approaches: the high-fidelity data modeling of neural density estimators and the structured representation learning capabilities of graphical models.

We motivate with a simple latent variable model where continuous latents $\mathbf{z} \in \mathbb{R}^p$ generate observations $\mathbf{x} \in \mathbb{R}^d$, with $0 < p < d$:

$$\mathbf{z}_i \overset{i.i.d.}{\sim} p(\mathbf{z}), \qquad \mathbf{x}_i|\mathbf{z}_i \overset{ind.}{\sim} p(\mathbf{x}|\mathbf{z}). \tag{1}$$

This standard latent variable framework enables structured representation learning through the posterior $p(\mathbf{z}|\mathbf{x})$. To enable high-fidelity data modeling in SFA, we parametrize $p(\boldsymbol{x}|\boldsymbol{z})$ using conditional CNFs, achieving the expressivity of modern neural density estimators while maintaining structured latents. However, both the likelihood and the posterior are no longer available in explicit forms. To address this challenge, we propose the *Structured Conditional Flow Matching (SCFM)* objective, a training objective that jointly learns both the conditional flow $p(\mathbf{x}|\mathbf{z})$ and an approximate posterior $q(\mathbf{z}|\mathbf{x})$. Unlike standard flow matching that only models $p(\mathbf{x})$, SCFM explicitly account for the conditional structure $p(\mathbf{x}|\mathbf{z})$ and posterior $p(\mathbf{z}|\mathbf{x})$. As illustrated in Fig. 1, this decomposition enables SFA to capture interpretable latent variables while maintaining high-fidelity generation, providing structured representation learning unavailable in standard flow matching.

Contributions. (1) We introduce Structured Flow Autoencoders (SFA), a family of generative models that augments graphical models with conditional Continuous Normalizing Flows (CNFs) likelihoods. SFA bridges the gap between high-fidelity neural density estimation and structured representation learning, improving upon both VAEs and latent flow-based models. (2) We propose Structured Conditional Flow Matching (SCFM), a novel training objective that extends flow matching to explicitly incorporate latent variables. SCFM explicitly learns the conditional probability flows in the graphical model while preserving the marginal density information. SCFM enables joint learning of the likelihood and posterior, supporting both generative modeling and structured representation learning within a unified framework. (3) We demonstrate the flexibility of SFA across diverse domains, including image, video, and RNA-seq data, and modeling scenarios with continuous, finite mixture, and dynamical latent variables. SFA achieves high-fidelity sample generation, increased sample diversity, and enhances structured representation learning, while remaining computationally efficient on high-dimensional datasets.

Related work. Simultaneous high-fidelity generation and structured representation learning has been an important task (Grathwohl et al., 2018; Mittal et al., 2023; Dao et al., 2023; Davtyan et al., 2023), drawing particular interest in scientific domains (Bashiri et al., 2021; Xu et al., 2023; Kapoor et al.,

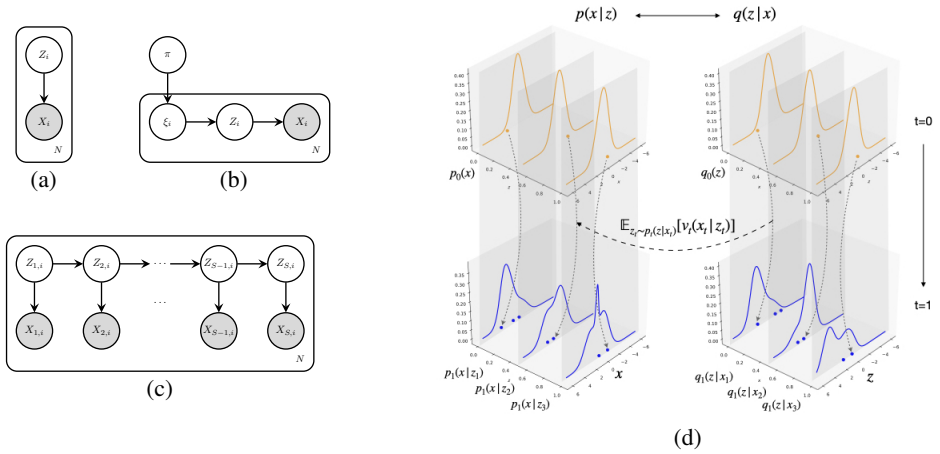


Figure 2: Overview of Structured Flow Autoencoders (SFAs). Examples of graphical models that can be incorporated into SFAs: (a) latent continuous variable model; (b) latent finite mixture model; (c) latent linear dynamical system; (d) SFA framework showing conditional probabilities for latent $\boldsymbol{z}$ and observed $\boldsymbol{x}$ with conditional CNFs. In the SCFM objective, we compute a convolution of conditional vector field v_t for $\boldsymbol{x}_t(\cdot, \boldsymbol{z}_t)$ with respect to $q_t(\boldsymbol{z}_t|\boldsymbol{x}_t)$, when $\boldsymbol{x}_t = (1-t)\boldsymbol{x}_0 + t\boldsymbol{x}_1$. For conditional generation, with a particular prior $p_1(\boldsymbol{z}_1)$, sampling follows from the graphical model $\boldsymbol{z}_1 \sim p_1(\boldsymbol{z}_1), \boldsymbol{x}_1 \sim p_1(\boldsymbol{x}_1|\boldsymbol{z}_1)$; deriving latent representation of $\boldsymbol{x}_1$ involves sampling from the posterior $\tilde{\boldsymbol{z}}_1 \sim q_1(\boldsymbol{z}_1|\boldsymbol{x}_1)$.

2024). Variational autoencoders (VAE) (Kingma & Welling, 2013) is one such probabilistic model that learns both generative model $p(\boldsymbol{x}|\boldsymbol{z})$ and inference model $p(\boldsymbol{z}|\boldsymbol{x})$ simultaneously, typically use neural networks to parameterize exponential families. Grathwohl et al. (2018); Chen et al. (2020) extended the VAE to families of normalizing flows, which improves the flexibility of density estimation together with latent space learning. While appealing, VAEs fall short of modern generative models in data modeling fidelity.

Recent work has explored combining neural density estimators with encoder-decoder frameworks (Mittal et al., 2023; Dao et al., 2023; Davtyan et al., 2023; Vahdat et al., 2021), typically mapping observations to low-dimensional latent spaces where the latent marginal distribution is learnt via flows or diffusion models before decoding back to observation space. While these neural prior methods excel at dimensionality reduction, they often constrain encoders and decoders to simple parametric families (e.g., Gaussians) that inadequately capture complex data distributions. Our approach differs by making the entire likelihood and posterior flexible through conditional flows while incorporating structured latent dependencies. Another line of work, including Wang et al. (2023); Preechakul et al. (2022), focuses on modeling the likelihood with flexible diffusion models while maintaining simple Gaussian posteriors. However, these approaches lack mechanisms for structured dependencies between latent variables, limiting interpretability in complex domains. Most closely related to our work, structured variational autoencoders (SVAEs) (Johnson et al., 2016; Lin et al., 2018) incorporate graphical model structure into VAEs to capture hierarchical dependencies. However, SVAEs are constrained by parametric assumptions that limit expressiveness. In addition, extending SVAEs to more expressive density models like CNFs faces significant challenges: direct extensions suffer from numerical instability and computational inefficiency due to the need of likelihood evaluation at every training step (Liu et al., 2022). Our flow matching approach circumvents these issues while enabling both structured representations and expressive data density modelling.

2 Preliminaries: Flow-based Generative Modelling

We start by reviewing continuous normalizing flows and the flow matching learning objective, laying the groundwork before introducing structured flow autoencoders (SFAs).

Notations. We follow the notations in Lipman et al. (2022) and denote the time indexed vector field by $v(\cdot, \cdot) : [0, 1] \times \mathbb{R}^d \to \mathbb{R}^d$ and equivalently, $v_t(\cdot) : \mathbb{R}^d \to \mathbb{R}^d$ for $t \in [0, 1]$. The path of probability densities is denoted by $p_t(\cdot) : \mathbb{R}^d \to \mathbb{R}^+$, and the flow $\phi_t(\cdot) : \mathbb{R}^d \to \mathbb{R}^d$ for $t \in [0, 1]$. In addition, $\boldsymbol{x}_1 \sim p_{data}$ represents an observed sample, and $\boldsymbol{x}_0 \sim p_0$ a sample from a chosen base distribution.

We further denote the conditional vector field as $u(\cdot,\cdot,\boldsymbol{z}) : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$, equivalently as $u_t(\cdot,\boldsymbol{z}) : \mathbb{R}^d \to \mathbb{R}^d$ indexed by $t \in [0,1]$. The path of conditional probability densities is denoted by $p_t(\cdot|\boldsymbol{z}) : \mathbb{R}^d \to \mathbb{R}^+$; and the conditional flow by $\phi_t(\cdot|\boldsymbol{z}) : \mathbb{R}^d \to \mathbb{R}^d$, $t \in [0,1]$.

2.1 Continuous Normalizing Flow

Continuous normalizing flows (CNFs) describe probability distributions by the evolution of some probability density path. Denote the observed data by $\boldsymbol{x} \in \mathbb{R}^d$. Further, assume there exists a time-dependent vector field $v_t : \mathbb{R}^d \to \mathbb{R}^d,\ t \in [0,1]$ that describes the evolution of a probability density path $p_t : \mathbb{R}^d \to \mathbb{R}^+$ indexed by $t \in [0,1]$; we will use v_t to describe the density of $\boldsymbol{x}$. The path then solves the continuity equation $\partial_t p_t = -\nabla \cdot (v_t p_t)$, which is the Fokker-Plank equation with zero diffusion. Due to the probabilistic representation theorem in Ambrosio et al. (2008, Theorem 8.2.1), the continuity equation admits a representation formulated as a solution of the ODE,

$$\frac{d}{dt}\phi_t(\boldsymbol{x}) = v_t(\phi_t(\boldsymbol{x})),\ \phi_0(\boldsymbol{x}) = \boldsymbol{x}_0, \tag{2}$$

where $\phi_t : \mathbb{R}^d \to \mathbb{R}^d$ is a push-forward map sending μ_0 to $\mu_t = \phi_{t,\sharp}\mu_0$. This map is called flow in the machine learning literature (Chen et al., 2018; Grathwohl et al., 2018; Lipman et al., 2022). The log likelihood $f(t) = \log p_t(\phi_t(\boldsymbol{x}))$ at any point $\boldsymbol{x}$ can be obtained by solving the instantaneous change-of-variable formula forward in time, with initial conditions $c = \log p_0(\phi_0(\boldsymbol{x}))$, $f(1) = \log p_1(\phi_1(\boldsymbol{x}))$:

$$\frac{d}{dt}\begin{pmatrix}\phi_t(\boldsymbol{x}) \\ f(t)\end{pmatrix} = \begin{pmatrix}v_t(\phi_t(\boldsymbol{x})) \\ -\nabla \cdot (v_t(\phi_t(\boldsymbol{x})))\end{pmatrix}, \quad \begin{pmatrix}\phi_0(\boldsymbol{x}) \\ f(0)\end{pmatrix} = \begin{pmatrix}\boldsymbol{x}_0 \\ c\end{pmatrix}. \tag{3}$$

2.2 Flow Matching

Liu et al. (2022) and Lipman et al. (2022) concurrently introduced a similar training objective for learning flexible flow-based generative models, with NN parameterized vector field v_t,

$$\inf_\theta \mathbb{E}_{t,p_{data}(\boldsymbol{x}_1),p_t(\boldsymbol{x}_t|\boldsymbol{x}_1)} \left\| v_t(\boldsymbol{x}_t;\theta) - u_t\left(\boldsymbol{x}_t \mid \boldsymbol{x}_1\right)\right\|^2, \tag{4}$$

where $t \sim \mathcal{U}[0,1], \boldsymbol{x}_1 \sim p_{data}\left(\boldsymbol{x}_1\right)$, and now $\boldsymbol{x} \sim p_t\left(\boldsymbol{x} \mid \boldsymbol{x}_1\right)$. We refer to this objective as Flow Matching (FM). Flow matching resembles diffusion model with score matching except that the steps of noising (with conditional vector field $u_t(\boldsymbol{x}|\boldsymbol{x}_1)$) and denoising (with marginal vector field v_t) are deterministic. Solving Eq. 2 forward in time allows for generation from the learnt model.

The family of conditional vector field u_t that governs the conditional probability path $p_t(\boldsymbol{x}_t|\boldsymbol{x}_1)$ is a design choice. Lipman et al. (2022) considered a particular example of the conditional probability path, $p_t(\boldsymbol{x}_t|\boldsymbol{x}_1) = N(\mu_t(\boldsymbol{x}_1), \sigma_t(\boldsymbol{x}_1)^2 I)$, where μ_t and σ_t are time-dependent functions, with end points $\mu_0(\boldsymbol{x}_1) = 0$ and $\sigma_0^2(\boldsymbol{x}_1) = 1$ such that $p_0(\boldsymbol{x}_0|\boldsymbol{x}_1) \overset{d}{=} N(\boldsymbol{x}_0; 0, I_p)$. Therefore, the probability path $p_t = \varphi_t \sharp p_0$ is induced by the map $\varphi_t(\boldsymbol{x}) = \mu_t(\boldsymbol{x}_1) + \sigma_t(\boldsymbol{x}_1)\boldsymbol{x}$, which is the solution of the characteristic ODE $\frac{d}{dt}\varphi_t(\boldsymbol{x}) = u_t(\varphi_t(\boldsymbol{x})|\boldsymbol{x}_1)$ A special example includes linear interpolation in the Wasserstein space, $\varphi_t(\boldsymbol{x}) = (1-t)\boldsymbol{x} + t\boldsymbol{x}_1$. For this choice, the corresponding conditional vector field is $u_t(\mathbf{x}_t|\mathbf{x}_1) = \frac{\mathbf{x}_1 - \mathbf{x}_t}{1-t}$ for $t \in [0,1)$.

3 Structured Flow Autoencoders

In this section, we augment probabilistic graphical models with CNF likelihoods to design *structured flow autoencoders (SFAs)*, a family of structured flow-based probabilistic generative models.

From marginal vector field to conditional vector field. To enable probabilistic graphical modeling using flow-based models, we rely on a key insight arising from Bayes formula: the marginal vector field can be equivalently derived as the expectation of conditional vector field $v_t(\boldsymbol{x}|\boldsymbol{z})$ over an unobserved latent variable $\boldsymbol{z}$,

$$v_t(\boldsymbol{x}) = \int v_t(\boldsymbol{x}|\boldsymbol{z}) \frac{p_t(\boldsymbol{x}|\boldsymbol{z})p_t(\boldsymbol{z})}{\int p_t(\boldsymbol{x}|\boldsymbol{z})p_t(\boldsymbol{z})d\boldsymbol{z}} d\boldsymbol{z} = \mathbb{E}_{p_t(\boldsymbol{z}|\boldsymbol{x})}[v_t(\boldsymbol{x}|\boldsymbol{z})], \tag{5}$$

which also resembles the posterior predictive distribution. We formally state this result below in Proposition 3.1, which shows that $\mathbb{E}_{p_t(\boldsymbol{z}|\boldsymbol{x})}[v_t(\boldsymbol{x}|\boldsymbol{z})]$ is indeed the vector field that generates the path of marginal probability distributions $p_t(\boldsymbol{x})$. The proof is in Appendix A.1, which proceeds by verifying that Eq. 5 satisfies the continuity equation (Lipman et al., 2022).

Proposition 3.1. *Given conditional vector field $v_t(\boldsymbol{x}|\boldsymbol{z})$ that generates the path $\{p_t(\boldsymbol{x}|\boldsymbol{z})\}$ of probability kernel for $p(\boldsymbol{z})$ a.e. in $\boldsymbol{z}$, $v_t(\boldsymbol{x})$ is the marginal vector field that generates the marginal probability path $p_t(\boldsymbol{x})$ over $t \in [0, 1]$ under regularity conditions.*

Proposition 3.1 allows us to gain flexibility and interpretability in flow-based generative modeling by introducing latent structure to the otherwise marginal vector field of data distribution. This realization is the key to uncovering rich latent structure while ensuring marginal distribution is captured faithfully.

Structured Flow Autoencoders (SFA). Proposition 3.1 further motivates us to design SFAs, which consists of co-evolving probability paths $\{p_t(\cdot|\boldsymbol{z}_t)\}$ and $p_t\{\cdot|\boldsymbol{x}_t\}$ across time $t \in [0, 1]$; these paths are connected to the observed data distribution $p_t(\boldsymbol{x}_t)$ through $\mathbb{E}_{p_t(\boldsymbol{z}|\boldsymbol{x})}[v_t(\boldsymbol{x}|\boldsymbol{z})]$ (Fig. 2d). At $t = 1$, the probability $p_1(\cdot|\boldsymbol{x}_1)$ and $p_1(\cdot|\boldsymbol{z}_1)$ corresponds to the model likelihood and posterior for the observed data; at $t = 0$, the probabilities correspond to the marginal base distributions that are easy to sample and evaluate. In addition, Proposition 3.1 allows the specification of any posterior family, which could have inbuilt structure according to a graphical model. We defer three representative examples to the next subsections. To learn SFAs, we propose to match the marginal path $p_t(\boldsymbol{x}_t)$ to a preselected path as in FM objective.

Structured Conditional Flow Matching (SCFM). Although KL divergence and ELBO are standard objectives for unsupervised distribution learning, CNF likelihood evaluation incurs significant computational overhead. Recognizing that the FM objective is fast and easy to evaluate for flow based generative models, we propose an objective that accommodates latent structure while improving computational efficiency based on FM. Specifically, Proposition 3.1 shows the marginal vector field emerges from the conditional vector field, enabling us to replace $v_t(\boldsymbol{x})$ with $\mathbb{E}_{p_t(\boldsymbol{z}|\boldsymbol{x})}[v_t(\boldsymbol{x}|\boldsymbol{z})]$ in the FM loss (Eq. 4). We formalize this approach through the *Structured Conditional Flow Matching (SCFM)* objective:

$$\inf_{\theta} \mathbb{E}_{\substack{\boldsymbol{x}_1 \sim p_{data}(\boldsymbol{x}_1) \\ \boldsymbol{x}_t \sim p_t(\boldsymbol{x}|\boldsymbol{x}_1),\ t \sim Unif[0,1]}} \left\| \mathbb{E}_{p_t(\boldsymbol{z}_t|\boldsymbol{x}_t)}[v_t(\boldsymbol{x}_t|\boldsymbol{z}_t;\theta)] - u_t\left(\boldsymbol{x}_t \mid \boldsymbol{x}_1\right) \right\|^2, \tag{6}$$

where the outer expectation is w.r.t. the flow trajectory on the marginal given observed samples $\boldsymbol{x}_1$; the inner expectation is w.r.t. the distribution trajectory of the corresponding posterior; the reference vector field u_t is chosen *a priori*, which defines the desired trajectory connecting observed data to the base distribution p_0. Intuitively, SCFM is solving a "de-mixing" problem, decomposing the observed signal to (1) the data generation model and (2) the latent structure components.

For different graphical models (c.f. Fig. 2), SCFM objective can be adapted to accommodate their specific structures. We illustrate through three examples in the next subsection, spanning continuous, finite mixture and Markov dynamic latent structure. We chose these three examples because they are (1) widely applicable across different domains, (2) representative of different dependency types (continuous, finite mixture, temporal), and (3) sufficient to demonstrate the framework's flexibility.

Posterior approximation. In practice, we approximate the expectation in Eq. 5 using samples from $p_t(\boldsymbol{z}_t|\boldsymbol{x}_t)$. As the posterior is generally intractable, we employ an approximating family $Q = \{(t, x) \mapsto q_t(\boldsymbol{z}|\boldsymbol{x}), (t, x) \in [0, 1] \times \mathcal{X}\}$ to enable sampling and evaluation at training and evaluation time. The choice of approximating family Q must be sufficiently expressive to capture the complexity of the true posterior, while not too complex that de-stabilize the training. We discuss specific choices for each latent structure in Fig. 2 in the following subsection. Now, with a learned posterior approximation, the corresponding marginal distribution (prior) in the latent space can be derived post-hoc. Motivated by empirical Bayes, this can be achieved via integration over the observation marginal: $q_t(\boldsymbol{z}_t) = \int q_t(\boldsymbol{z}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t)d\boldsymbol{x}_t$. In practice, a separate model can be used to learn the marginal after training (Wang et al., 2023; Preechakul et al., 2022).

3.1 Examples of Structured Flow Autoencoders (SFAs)

In this section, we expand on examples of SFA with continuous, finite mixture and Markov dynamic latent structures in Fig. 2. We discuss the extensions of SCFM objective functions and choices of approximation families.

3.1.1 CONTINUOUS LATENT VARIABLE MODEL

Consider the graphical model in Fig. 2a, where $\boldsymbol{z} \in \mathbb{R}^d$ and $\boldsymbol{x} \in \mathbb{R}^p$,

$$\boldsymbol{z}_i \overset{iid}{\sim} p(\boldsymbol{z}), \quad \boldsymbol{x}_i|\boldsymbol{z}_i \overset{ind}{\sim} p(\boldsymbol{x}|\boldsymbol{z}),$$

giving rise to the posterior $\boldsymbol{z}_i|\boldsymbol{x}_i \sim p(\boldsymbol{z}|\boldsymbol{x})$. We estimate both the unknown likelihood and posterior from observed data, $\boldsymbol{x}_{1,i} \sim p_{data}(\boldsymbol{x}) = \int p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z}$, under SCFM objective. Following from Proposition 3.1, the likelihood model is the conditional CNF generated by the conditional vector field $v_t(\boldsymbol{x}_t|\boldsymbol{z}_t, \theta)$, governed by the ODE:

$$\frac{d}{dt}\phi_t(\boldsymbol{x}) = v_t(\phi_t(\boldsymbol{x})|\boldsymbol{z}; \theta), \quad \phi_0(\boldsymbol{x}) = \boldsymbol{x}_0, \quad \boldsymbol{x}_0 \sim p_0(\boldsymbol{x}). \tag{7}$$

In this example, the risk function follows directly from Eq. 6. For practical implementation, the inner expectation $\mathbb{E}_{q_t(\boldsymbol{z}_t|\boldsymbol{x}_t)}[v_t(\boldsymbol{x}_t|\boldsymbol{z}_t)]$ of the objective can be approximated with a single sample $\tilde{\boldsymbol{z}} \sim q_t(\boldsymbol{z}|\boldsymbol{x})$, as commonly used in VAE (Kingma & Welling, 2013).

There are different choices of approximation family of posterior Q and each with their own trade-offs. With conditional CNFs, it amounts to modelling the parameterized conditional vector field $r_t(\boldsymbol{z}|\boldsymbol{x}; \theta)$ for $\boldsymbol{x}$ a.e. During training, we evaluate the inner expectation in Eq. 6 with samples from q_t by solving the ODE system from $t = 0$ to $t = 1$. The gradient computation then follows from reparameterization trick of q_t. Specifically, the sample from conditional CNF family requires backpropagation through the adjoint ODE steps (Chen et al., 2018), which adds to instability and computational burden.

$$\frac{d}{ds}\psi_s(\boldsymbol{z}) = r_s(\psi_s(\boldsymbol{z})|\boldsymbol{x}_t; \theta), \quad \psi_0(\boldsymbol{z}) = \boldsymbol{z}_0,\ \boldsymbol{z}_0 \sim q_0(\boldsymbol{z}),$$

Alternatively, Q can be chosen as parametric families, with parameters indexed by t and $\boldsymbol{x}$. For continuous latent, a simple choice is Gaussian family, $Q = \{(t, x) \mapsto N(\mu_\theta(t, \boldsymbol{x}), \sigma^2_\theta(t, \boldsymbol{x})I_d)\}$. As the gradient computation follows directly from the standard reparameterization trick, it offers computational efficiency and stability advantages. This approximation family evolving across $t \in [0, 1]$ also provides flexibility beyond fixed-time counterparts.

3.1.2 LATENT FINITE MIXTURE MODEL

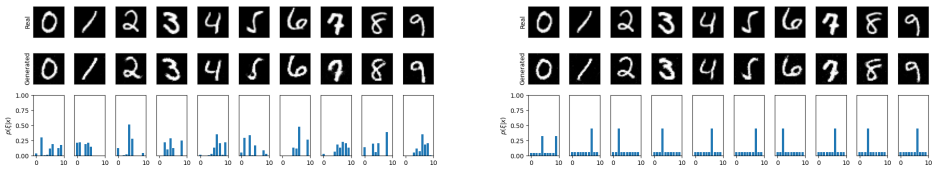
In this section, we consider the generative model in Fig. 2b, where the latent variable $\boldsymbol{z}$ follows a finite mixture distribution with the number of classes K. This graphical model takes into account of latent class $\xi \in [K]$, where $p(\xi_i = k|\pi) = \pi_k$ for each sample $\boldsymbol{x}$. It gives rise to posteriors on the local class label ξ_i, the continuous latent $\boldsymbol{z}$, and the global class proportion π, as detailed under the inference model.

Generative Model	**Inference Model**
$\pi \sim p(\pi),\ \xi_i\|\pi \overset{iid}{\sim} \text{Cat}(\pi),$	$\xi_i\|\boldsymbol{x}_i, \pi \sim \text{Cat}(p(\xi_i\|\boldsymbol{x}_i, \pi)),$
$\boldsymbol{z}_i\|\xi_i \sim p(\boldsymbol{z}\|\xi_i),\ \boldsymbol{x}_i\|\boldsymbol{z}_i \overset{ind}{\sim} p(\boldsymbol{x}\|\boldsymbol{z}_i),$	$\boldsymbol{z}_i\|\boldsymbol{x}_i,\ \xi_i \sim p(\boldsymbol{z}_i\|\boldsymbol{x}_i, \xi_i), \pi\|\boldsymbol{z}_{[n]} \sim p(\pi\|\boldsymbol{z}_{[n]}).$

When $\pi \sim Dir(\alpha)$, the posterior for overall proportions $p(\pi|z_{[n]})$ has a closed form $Dir(\tilde{\alpha})$ with $\tilde{\alpha}_k = \alpha_k + \sum_{i=1}^n \mathbf{1}\{\xi_i = k\}$. The local label $\xi, \boldsymbol{z}$ are of major interest for drawing inference on the latent class assignment and value. Next, we adapt SCFM for latent finite mixture model: both ξ and $\boldsymbol{z}$ are now integrated out in the inner expectation. Applying Proposition 3.1 gives rise to Eq. 8.

$$\inf_{q\in Q, \theta\in\Theta} \mathbb{E}_{\substack{\boldsymbol{x}_1\sim p_{data}(\boldsymbol{x}_1) \\ \boldsymbol{x}\sim p_t(\boldsymbol{x}|\boldsymbol{x}_1) \\ t\sim Unif[0,1]}} \left\| \mathbb{E}_{q_t(\xi_t|\boldsymbol{x}_t)q_t(\boldsymbol{z}_t|\boldsymbol{x}_t,\xi_t)}[v_t(\boldsymbol{x}_t|\boldsymbol{z}_t; \theta)] - u_t\left(\boldsymbol{x}_t \mid \boldsymbol{x}_1\right) \right\|^2. \tag{8}$$

The design of likelihood model follows similarly as in Eq. 7, which is a CNF conditioned on $\boldsymbol{z}$ only. The approximation family for $p_t(\xi_i|\boldsymbol{x}_i)$ could be chosen as a Gumbel-Softmax distribution with time-dependent parameters, alternatively constant across t to reduce the complexity of the model. The approximation family for $p_t(\boldsymbol{z}|\boldsymbol{x}, \xi)$, should be chosen as conditional CNF or parametric distribution indexed by $t, \boldsymbol{x}, \xi$. As $\boldsymbol{z}$ is unconstrained, a Gaussian approximation family can be posited similarly as in Section 3.1.1.



(a) GMVAE (b) Mixture-SFA

Figure 3: Comparison of GMVAE and Mixture-SFA on MNIST dataset. The first row displays the posterior predictive $\boldsymbol{x}_i \sim \int p_1(\boldsymbol{x}|\boldsymbol{z}) q_1(\boldsymbol{z}|\boldsymbol{x}_{1,i}) d\boldsymbol{z}$ and latent class assignment probability $\xi_i \sim q_1(\xi|\boldsymbol{x}_{1,i})$ for a test data $\boldsymbol{x}_{1,i}$. The second row shows the latent representation learned for digits, where each point is sampled from $\boldsymbol{z}_i \sim q_1(\boldsymbol{z}|\boldsymbol{x}_{1,i})$ for a test data sample $\boldsymbol{x}_{1,i}$.

3.2 Latent Dynamic System

We consider discrete-time sequential generation with continuous latent states following the graphical model in Fig. 2c. For index $s \in [S]$, the generative model is formalized as conditional independent observation $\boldsymbol{x}^s$ given latent state $\boldsymbol{z}^s$; the inference model is focused on the full posterior of latent trajectory $\boldsymbol{z}^{[S]}$ given the observation sequence $\boldsymbol{x}^{[S]}$, which can be factorized into full conditionals at each index s given all previous history.

$$\textbf{Generative Model} \qquad \boldsymbol{z}_i^s|\boldsymbol{z}_i^{s-1} \sim p(\boldsymbol{z}|\boldsymbol{z}^{s-1}), \quad \boldsymbol{x}_i^s|z_i^s \sim p(\boldsymbol{x}^s|\boldsymbol{z}_i^s).$$

$$\textbf{Inference Model} \qquad \boldsymbol{z}_i^{[S]}|\boldsymbol{x}_i^{[S]} \sim p(\boldsymbol{z}^{[S]}|\boldsymbol{x}^{[S]}) = \prod_{s\in[S]} p(\boldsymbol{z}^s|\boldsymbol{z}^{[s-1]}, \boldsymbol{x}^{[S]}).$$

Given observed sample sequences $\{\boldsymbol{x}_i^{[S]}\}_{i=1}^n$, the SCFM objective can be shown to have the form in Eq. 9. Accompanying theoretical results and numerical derivations are detailed in Theorem A.2, which follows similar arguments for Proposition 3.1.

$$\inf_{q\in Q, \theta\in\Theta} \mathbb{E}_{\substack{\boldsymbol{x}_1\sim p_{data}, \boldsymbol{x}_t\sim p_t(\boldsymbol{x}_t|\boldsymbol{x}_1) \\ t\sim \text{Unif}[0,1]}} \left\| \sum_{s\in[S]} \mathbb{E}_{p_t(\boldsymbol{z}_t^{[S]}|\boldsymbol{x}_t^{[S]})} \left[v_t(\boldsymbol{x}_t^s|\boldsymbol{z}_t^s;\theta) \right] - u_t\left(\boldsymbol{x}_t^s \mid \boldsymbol{x}_1^s\right) \right\|^2. \tag{9}$$

Here the sum over $s \in [S]$ captures the sequential dependencies inherent in the model structure. Following the assumed conditional independence, we parameterize the likelihood using conditional CNFs for each $\boldsymbol{x}^s, s \in [S]$ according to Eq. 7. To approximate the posterior $p_t(\boldsymbol{z}^{[S]}|\boldsymbol{x}^{[S]})$, we employ a parametric family with its parameters indexed by $t \in [0, 1]$, previous states $\boldsymbol{z}^{[s-1]}$, and the full observation sequence $\boldsymbol{x}^{[S]}$. Details of implementations can be found in App. B.3.

4 Empirical Studies

In this section, we evaluate the proposed SFA across a range of tasks and data modalities: (a) conditional density estimation for Pinwheel dataset; (b) latent clustering on MNIST dataset; (c) gene expression modelling on single-cell RNA-seq data; (d) sequence modelling with Pendulum trajectory video dataset. We compare our method mainly to VAE counterparts, including SVAE (Johnson et al., 2016), VampVAE(Tomczak & Welling, 2018), βVAE(Higgins et al., 2017), GMVAE (Lin et al., 2018) and Latent Flow Matching with deterministic autoencoders (LatentFM) (Dao et al., 2023) and probabilistic autoencoders (LatentFM w/VAE).

For a fair comparison, we restrict the prior to be fixed in SVAE and only focus on modeling the conditional probabilities. All the metrics are evaluated based on samples held out from training. More experimental details and supporting visualizations are hosted in App. B. We draw comparisons in the quality of (1) generated posterior samples (or latent distribution samples for Latent FM) and (2) generated posterior predictive samples, and downstream tasks, such as latent space clustering.

For (1), given a large training dataset, the posterior distribution should be able to provide an accurate representation of the latent distribution. Therefore, under simulation settings where latent ground truth is known, we are able to evaluate the discrepancy between learned latent representation and the truth, provided $\boldsymbol{z} \sim q_1(\boldsymbol{z}|\boldsymbol{x})$ with $\boldsymbol{x} \sim p_{data}(\boldsymbol{x})$.

Table 1: Comparing generated samples to data samples with W_1 metric (Earth Mover's Distance). W_1 metric is evaluated with samples from marginal data distribution $p(\boldsymbol{x}_1)$ and that generated from $\tilde{p}_1(\boldsymbol{x}_1) = \int p_1(\boldsymbol{x}_1|\boldsymbol{z}_1)q_1(\boldsymbol{z}_1)d\boldsymbol{z}_1$. SFA and FM achieves comparable performance on marginal density estimations.

	VAE	VampVAE	GMVAE	FM	LatentFM(w/VAE)	SFA	Mixture-SFA
$\hat{W}_1(p, \tilde{p}) \downarrow$	0.119	0.081	0.457	**0.025**	0.496(0.145)	**0.024**	0.046

Table 2: Subspace clustering on MNIST with latent mixtures models GMVAE and Mixture-SFA. Evaluated on a held-out set of size 1000.

	$\log p(x\|z) \uparrow$	$\log p(z\|x) \uparrow$	SSIM $\uparrow$	softNMI $\uparrow$	NMI $\uparrow$	ARI $\uparrow$
GMVAE	-1133	0.667	0.634	0.698	0.161	0.0716
Mixture-SFA	$\mathbf{-905.803}$	**725.232**	**0.779**	**0.728**	**0.489**	**0.332**

For (2), we conduct posterior predictive check to evaluate the discrepancy of the samples from $p_{data}(\boldsymbol{x})$ and $p_{pred}(\tilde{\boldsymbol{x}}|\boldsymbol{x}) = \int p_1(\tilde{\boldsymbol{x}}|\boldsymbol{z})p(\boldsymbol{z}|\boldsymbol{x})d\boldsymbol{z}$. To sample from the latter, we follow

$$\boldsymbol{x} \sim p_{data}(\boldsymbol{x}), \quad \boldsymbol{z}|\boldsymbol{x} \sim q_1(\boldsymbol{z}|\boldsymbol{x}), \quad \tilde{\boldsymbol{x}}|\boldsymbol{z} \sim p_1(\boldsymbol{x}|\boldsymbol{z}).$$

We evaluate the diversity of the generated samples using Vendi score (Friedman & Dieng, 2022); quality of image generation with SSIM (Wang et al., 2004); quality of latent clustering with ARI (Hubert & Arabie, 1985), NMI (Strehl & Ghosh, 2002) and probabilistic version softNMI (Eq. 13).

Summary of findings. In conditional density modeling (Pinwheel), SFA consistently outperforms LatentFM and VAE-based models, showing better data density reconstruction and better latent space modelling. To assess scalability, we apply SFA to a single-cell RNA-seq dataset, where it effectively models high-dimensional gene expression data and outperforms VAEs in reconstruction quality. On image data (MNIST), both SFA and its mixture extension (Mixture-SFA) learn meaningful latent representations, generate high-fidelity samples, and perform well on latent-space clustering tasks. Finally, we highlight SFA's versatility on sequential data using the pendulum video dataset, where it successfully captures the low-dimensional periodic structure of the underlying physical system.

SVAE and SFA comparison. The SVAE baseline uses β-VAE (Higgins et al., 2017) with a regularization parameter to balance likelihood and KL terms during training. This is crucial for balancing the likelihood and posterior components in training. SFA is much more stable in jointly learning the conditional probabilities, without the generation-latent learning trade-off often encountered in VAE training. This is owing to the fact that SFA latent accounts for meaningful structure in the trajectory of $p_t(x_t|z_1)$, which stabilizes the path and provides accurate reconstruction to the marginal distribution. We refer to App. B.4 for more details. When the latent is lower-dimensional, a smaller posterior model is sufficient relative to the model size needed for the likelihood. When learning multiple components jointly, simpler parametric approximation family are preferred over conditional CNFs for training stability. Computation-wise, SFA (2.4M parameters) requires 13.220 ± 1.848 seconds per epoch, comparable to VAE's 12.789 ± 2.011 seconds. With CNF as the posterior of SFA, we observe a drastic increase in the runtime to 167.460 ± 176.817 seconds. This is owing to the extra time arised from sampling CNF during training, which requires solving a ODE at each gradient evaluation step.

4.1 Mixture Modeling: Pinwheel Data

We first illustrate the ability of SFA in learning conditional distributions using the toy example of the pinwheel dataset, with five clusters each having the shape of a blade (Johnson et al., 2016). The class membership is not provided during training. the goal is to evaluate if the posterior is able to uncover the latent structure of the data, and whether the model is able to capture the observed data distribution. In Fig. 1, we visualize the generated data together with their representation coded in 1D colorbar. SFA is able to reconstruct the support of the ground truth distribution, in addition to capturing a meaningful latent representation for the angular rotation. In contrast, both the Latent FM and VAE-estimated density does not have well-separated components. As shown in Table 1, SFA based methods achieve similar density estimation quality as FM and comparable to ground truth, while SVAE based methods fail to model the density accurately.

Table 3: Comparison of metrics across different datasets and methods. (a) Kang HVG dataset evaluated on a held-out set of size 500. The observation has dimension 5000, due to the size, the log likelihood for CNF cannot be directly computed by solving adjoint-ODE, therefore left out of the comparison. (b) GLDS dataset over posterior samples of observed ($\texttt{RMSE}_x$), and latent ($\texttt{RMSE}_z$). Evaluated on a held-out set of size 300.

(a) HVG

	$\log p(z\|x) \uparrow$	Vendi ($\boldsymbol{x}$) $\uparrow$	NMI $\uparrow$	ARI $\uparrow$
VAE	-40.040	26.580	0.412	0.257
LatentFM	-	5.801	0.617	0.457
SFA	**384.137**	**737.728**	**0.633**	**0.460**

(b) Pendulum

	$\texttt{RMSE}_x \downarrow$	$\texttt{RMSE}_z \downarrow$
GLDSVAE	4.574	8.090
LDS-SFA	**3.233**	**1.526**

4.2 Image Modeling: MNIST Data

Next, we consider MNIST dataset LeCun et al. (2010), where we aim to recover the probabilistic assignment of each image to the 10 classes, and learn a low dimensional feature representation at the same time. We first consider the graphical model with continuous latent, and compare SFA to VAE and Latent FM on the latent space clustering task. In addition, to check if the learned latent space captures desirable structures in the data, such as stroke and abstract shape, we sample from the latent distribution encoding Out-of-Distribution (OOD) data in the EMNIST dataset Cohen et al. (2017). Result and comparisons are summarized in Table 2 and Fig. 6. Notably, VAE has a posterior collapse, resulting in unstructured latent space. On the other hand, both Latent FM and SFA learn meaningful representations that generalize to OOD samples, with SFA achieving better clustering performance and higher diversity (Vendi score). Performance using mixture graphical model are organized in Fig. 3 and Fig. 6. GMVAE improves generation quality and latent class separation over VAE. However, Mixture-SFA still achieves better clustering quality as shown in Table 2.

4.3 Gene Expression Modeling: Single-cell RNA-seq Data

The dataset obtained from Lotfollahi et al. (2023) includes PBMCs from eight patients with Lupus. The data consists of 7 cell types, and treated and control with IFN-β (Kang et al., 2018). The observed count is normalized and $\log(x+1)$ transformed, then $5,000$ HVGs are selected. We apply continuous latent to learn the low-dim representation of the high-dimensional differential expression data. Fig. 4 indicates that both the SFA and Latent FM are able to produce meaningful clusters of the cell type in the latent space. While both methods has good accuracy in the downstream clustering task, SFA has a larger Vendi score in the generated samples (Table 3a), indicating better diversity.

4.4 Sequential Modeling: Pendulum Trajectory Video Data

We choose a pendulum trajectory dataset for LDS example. The dynamics is driven by the pendulum physical system modeled as a damped harmonic oscillator. The latent trajectory is 2 dimensional, consisting of angle and angular velocity. The observation is a video with discrete time frames mapped from latent trajectory. Additional details on model implementations are in App. B.3. We compare SVAE with SFA in Table 3b, where we measure the discrepancy between the generated and ground truth of both observed and latent dynamics using RMSE. LDS-SFA outperforms in both aspects.

5 Discussion

In this work, we introduced structured flow autoencoders (SFA), a framework that integrates continuous normalizing flows (CNFs) with probabilistic graphical models (PGMs) to achieve both high-fidelity generation and structured latent representation learning. At the core of SFA is our proposed structured conditional flow matching (SCFM) objective, which extends flow matching by explicitly modeling latent variables, enabling the estimation of both the generative likelihood $p(\boldsymbol{x}|\boldsymbol{z})$ and the posterior $p(\boldsymbol{z}|\boldsymbol{x})$. This approach improves upon existing methods like Variational Autoencoders (VAEs), which rely on restrictive parametric assumptions, and flow-based models, which often lack structured interpretability. By leveraging the flexibility of CNFs while maintaining the interpretability of PGMs, SFA provides a principled and expressive framework for learning complex data distributions. Empirical results demonstrate the effectiveness of SFA in capturing both marginal densities and structured latent dependencies, outperforming existing generative models in

density estimation and representation learning. This work highlights the potential of bridging neural density estimation with structured probabilistic modeling, paving the way for more interpretable and scalable generative frameworks.

REFERENCES

Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media, 2008.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. *arXiv preprint arXiv:2107.03006*, 2021.

Mohammad Bashiri, Edgar Walker, Konstantin-Klemens Lurz, Akshay Jagadish, Taliah Muhammad, Zhiwei Ding, Zhuokun Ding, Andreas Tolias, and Fabian Sinz. A flow-based latent state generative model of neural population responses to natural images. *Advances in Neural Information Processing Systems*, 34:15801–15815, 2021.

Jianfei Chen, Cheng Lu, Biqi Chenli, Jun Zhu, and Tian Tian. Vflow: More expressive generative flows with variational data augmentation. In *International Conference on Machine Learning*, pp. 1660–1669. PMLR, 2020.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Siyi Chen, Huijie Zhang, Minzhe Guo, Yifu Lu, Peng Wang, and Qing Qu. Exploring low-dimensional subspaces in diffusion models for controllable image editing. *arXiv preprint arXiv:2409.02374*, 2024.

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. Emnist: an extension of mnist to handwritten letters. *arXiv preprint arXiv:1702.05373*, 2017.

Quan Dao, Hao Phung, Binh Nguyen, and Anh Tran. Flow matching in latent space. *arXiv preprint arXiv:2307.08698*, 2023.

Aram Davtyan, Sepehr Sameni, and Paolo Favaro. Efficient video prediction via sparsely conditioned flow matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 23263–23274, 2023.

Dan Friedman and Adji Bousso Dieng. The vendi score: A diversity evaluation metric for machine learning. *arXiv preprint arXiv:2210.02410*, 2022.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky TQ Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. *arXiv preprint arXiv:2407.15595*, 2024.

Will Grathwohl, Ricky TQ Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. Ffjord: Free-form continuous dynamics for scalable reversible generative models. *arXiv preprint arXiv:1810.01367*, 2018.

Irina Higgins, Loic Matthey, Arka Pal, Christopher P Burgess, Xavier Glorot, Matthew M Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR (Poster)*, 3, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

Noboru Isobe, Masanori Koyama, Kohei Hayashi, and Kenji Fukumizu. Extended flow matching: a method of conditional generation with generalized continuity equation. *arXiv preprint arXiv:2402.18839*, 2024.

Matthew J Johnson, David K Duvenaud, Alex Wiltschko, Ryan P Adams, and Sandeep R Datta. Composing graphical models with neural networks for structured representations and fast inference. *Advances in neural information processing systems*, 29, 2016.

Hyun Min Kang, Meena Subramaniam, Sasha Targ, Michelle Nguyen, Lenka Maliskova, Elizabeth McCarthy, Eunice Wan, Simon Wong, Lauren Byrnes, Cristina M Lanata, et al. Multiplexed droplet single-cell rna-sequencing using natural genetic variation. *Nature biotechnology*, 36(1): 89–94, 2018.

Jaivardhan Kapoor, Auguste Schulz, Julius Vetter, Felix Pei, Richard Gao, and Jakob H Macke. Latent diffusion for neural spiking data. *arXiv preprint arXiv:2407.08751*, 2024.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013. URL https://api.semanticscholar.org/CorpusID:216078090.

Yann LeCun, Corinna Cortes, and Christopher JC Burges. Mnist handwritten digit database. http://yann.lecun.com/exdb/mnist/, 2010.

Gen Li and Yuling Yan. Adapting to unknown low-dimensional structures in score-based diffusion models. *arXiv preprint arXiv:2405.14861*, 2024.

Wu Lin, Nicolas Hubacher, and Mohammad Emtiyaz Khan. Variational message passing with structured inference networks. *arXiv preprint arXiv:1803.05589*, 2018.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Mohammad Lotfollahi, Anna Klimovskaia Susmelj, Carlo De Donno, Leon Hetzel, Yuge Ji, Ignacio L Ibarra, Sanjay R Srivatsan, Mohsen Naghipourfar, Riza M Daza, Beth Martin, Jay Shendure, Jose L McFaline-Figueroa, Pierre Boyeau, F Alexander Wolf, Nafissa Yakubova, Stephan Günnemann, Cole Trapnell, David Lopez-Paz, and Fabian J Theis. Predicting cellular responses to complex perturbations in high-throughput screens. *Molecular Systems Biology*, 19(6):e11517, 2023. doi: https://doi.org/10.15252/msb.202211517. URL https://www.embopress.org/doi/abs/10.15252/msb.202211517.

Sarthak Mittal, Korbinian Abstreiter, Stefan Bauer, Bernhard Schölkopf, and Arash Mehrjou. Diffusion based representation learning. In *International Conference on Machine Learning*, pp. 24963–24982. PMLR, 2023.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. *arXiv preprint arXiv:2102.09672*, 2021.

Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10619–10629, 2022.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. *Advances in neural information processing systems*, 34:1415–1428, 2021a.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2021b.

Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 3(Dec):583–617, 2002.

Jakub Tomczak and Max Welling. Vae with a vampprior. In *International conference on artificial intelligence and statistics*, pp. 1214–1223. PMLR, 2018.

Alexander Tong, Kilian FATRAS, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=CD9Snc73AW. Expert Certification.

Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *Advances in neural information processing systems*, 34:11287–11302, 2021.

Binxu Wang and John J Vastola. Diffusion models generate images like painters: an analytical theory of outline first, details later. *arXiv preprint arXiv:2303.02490*, 2023.

Peng Wang, Huijie Zhang, Zekai Zhang, Siyi Chen, Yi Ma, and Qing Qu. Diffusion models learn low-dimensional distributions via subspace clustering. *arXiv preprint arXiv:2409.02426*, 2024.

Yingheng Wang, Yair Schiff, Aaron Gokaslan, Weishen Pan, Fei Wang, Christopher De Sa, and Volodymyr Kuleshov. Infodiffusion: Representation learning using information maximizing diffusion models. In *International conference on machine learning*, pp. 36336–36354. PMLR, 2023.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Minkai Xu, Alexander S Powers, Ron O Dror, Stefano Ermon, and Jure Leskovec. Geometric latent diffusion models for 3d molecule generation. In *International Conference on Machine Learning*, pp. 38592–38610. PMLR, 2023.

A THEORETICAL DETAILS

We formally present the probabilistic representation of solutions to the continuity equation when the vector field fails to be Lipchitz w.r.t $\boldsymbol{x}$. In this case, the solution to the characteristic ODE (flow ODE) is not unique. When using neural nets to parameterize the vector field, we want to verify that the solution of the ODE indeed induces a solution to the continuity equation.

Firstly, we denote $\mu_t : [0,1] \to \mathcal{P}(\mathbb{R}^d)$ as the path of probability indexed by t, $AC^p(0,1;\mathbb{R}^d)$ as the space of absolutely continuous curves $\gamma : [0,1] \to \mathbb{R}^d$ with finite p energy, i.e. $|\gamma'| \in L^p(0,1)$. Denote Γ as the space of continuous map $\gamma : [0,1] \to \mathbb{R}^d$. Let $e_t : (\boldsymbol{x}, \gamma) \mapsto \gamma(t)$ as the evaluation map. Then define the curve of probability measure induced by the evaluation map as

$$\mu_t^\eta = e_t \sharp \eta, \ t \in [0,1]$$

where by definition,

$$\int \psi(\boldsymbol{x}) d\mu_t^\eta(\boldsymbol{x}) = \int_{\mathbb{R}^d \times \Gamma} \psi(\gamma(t)) d\eta(\boldsymbol{x}, \gamma), \ \forall \psi \in C_b^0(\mathbb{R}^d), \ t \in [0,1].$$

Then finally recall that the continuity equation

$$\partial_t \mu_t + \nabla \cdot (vt\mu_t) = 0 \quad \text{in } \mathbb{R}^d \times (0,1).$$

Theorem A.1 (Ambrosio et al. (2008) Theorem 8.2.1)**.** *Let $\mu_t : [0,1] \to \mathcal{P}(\mathbb{R}^d)$ be a narrowly continuous solution of the continuity equation for a suitable Borel vector field $v_t(\boldsymbol{x})$ such that for some $p > 1$,*

$$\int_0^1 \int_{\mathbb{R}^d} |v_t(x)|^p d\mu_t(\boldsymbol{x}) dt < +\infty.$$

i Then (a) there exists a probability measure η in $\mathbb{R}^d \times \Gamma$ such that that concentrates on the set of pairs $(\boldsymbol{x}, \gamma)$ such that $\gamma \in AC^p(0,1;\mathbb{R}^d)$ is a solution of the ODE $\dot{\gamma}(t) = v_t(\gamma(t))$ for L^1-a.e. $t \in [0,1]$ with $\gamma(0) = \boldsymbol{x}$.
and (b) $\mu_t = \mu_t^\eta \ \forall t \in [0,1]$.

ii Conversely, any η satisfies (a) and $\int_0^1 \int_{\mathbb{R}^d \times \Gamma} |v_t(\gamma(t))| d\eta(\boldsymbol{x}, \gamma) dt < +\infty$ induces a solution of the continuity equation via $\mu_t^\eta = e_t \sharp \eta$, with $\mu_0 = e_0 \sharp \eta$.

The converse argument can be readily extended to the conditional vector field $v(\cdot, \cdot, \boldsymbol{z}) : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ for any ν-a.e. $\boldsymbol{z}$. Then the solution curve to the characteristic ODE would be indexed by $\boldsymbol{z}$, denoted as $\gamma_{\boldsymbol{z}}$.

A.1 PROOFS IN § 3

In the following, we use the same notation in the main paper for the proof details of Proposition 3.1 in § 3. We restate the theorem in the following for completeness

Proposition (Proposition 3.1)**.** *Given conditional vector field $v_t(\boldsymbol{x}|\boldsymbol{z})$ that generates the path $\{p_t(\boldsymbol{x}|\boldsymbol{z})\}$ of probability kernel for $p(\boldsymbol{z})$ a.e. $\boldsymbol{z}$. v_t is the marginal vector field that generates the marginal probability path $p_t(\boldsymbol{x})$ over $t \in [0,1]$ under regularity conditions.*

Proof. If the vector field $v_t(\cdot|\boldsymbol{z})$ is measurable w.r.t z, then the flow ψ_t solving the characteristic ODE is measurable w.r.t $\boldsymbol{z}$. Consequently, the probability $p_t(\cdot|z) = \psi_t \sharp p_0(\cdot)$ is a regular conditional probability. If we further impose regularity on the conditional vector field $v_t(\boldsymbol{x}|\boldsymbol{z})$, then there exists a unique solution to the continuity equation (see Lemma 8.1.4 Ambrosio et al. (2008)).

Assume continuity and boundedness of $v_t(\boldsymbol{x}|\boldsymbol{z})p_t(\boldsymbol{x}|\boldsymbol{z})$ and its divergence; continuity of $p_t(\boldsymbol{x}|\boldsymbol{z})$ and $\frac{d}{dt}p_t(\boldsymbol{x}|\boldsymbol{z})$ in both t and $\boldsymbol{z}$, as well as uniformly bounded $\frac{d}{dt}p_t(\boldsymbol{x}|\boldsymbol{z})$ for all $t \in [0,1]$ and almost all $\boldsymbol{z}$. These regularity conditions ensures Leibniz integral rule is satisfied, so that the exchange of derivative and divergence with integral is valid.

It is sufficient to show that $\mathbb{E}_{p_t(\boldsymbol{z}|\boldsymbol{x})}[v_t(\boldsymbol{x}|\boldsymbol{z})]$ and $p_t(\boldsymbol{x})$ satisfies the continuity equation. Firstly, it is given that

$$\frac{d}{dt} p_t(\boldsymbol{x}|\boldsymbol{z}) = -\nabla \cdot (v_t(\boldsymbol{x}|\boldsymbol{z}) p_t(\boldsymbol{x}|\boldsymbol{z})).$$

for $p(z)$ a.e. z. Now, using Bayes rule

$$\begin{aligned}\frac{d}{dt}p_t(\boldsymbol{x}) &= \int \frac{d}{dt}p_t(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z} \\ &= \int -\nabla\cdot(v_t(\boldsymbol{x}|\boldsymbol{z})p_t(\boldsymbol{x}|\boldsymbol{z}))p(\boldsymbol{z})d\boldsymbol{z} \\ &= -\nabla\cdot\int(v_t(\boldsymbol{x}|\boldsymbol{z})\frac{p_t(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})}{p_t(\boldsymbol{x})}p_t(\boldsymbol{x})d\boldsymbol{z} \\ &= -\nabla\cdot(\mathbb{E}_{p_t(\boldsymbol{z}|\boldsymbol{x})}[v_t(\boldsymbol{x}|\boldsymbol{z})]p_t(\boldsymbol{x}))\end{aligned}$$

which concludes the proof. □

A.2 Proofs in § 3.2

Recall the posterior arising from the latent dynamic model is

$$\boldsymbol{z}_i^{[S]}|\boldsymbol{x}_i^{[S]} \sim p(\boldsymbol{z}^{[S]}|\boldsymbol{x}^{[S]}) = \prod_{s\in[S]} p(\boldsymbol{z}^s|\boldsymbol{z}^{[s-1]},\boldsymbol{x}^{[S]}). \tag{10}$$

We present the extension of Proposition 3.1 to the latent dynamic model in the following theorem. The proof idea relies on verifying the joint continuity equation over the trajectory is satisfied and the corresponding structured conditional flow matching objective is well defined.

Theorem A.2. *With conditional flow defined by Eq. 7, and posterior defined by Eq. 10, the FM objective is derived to be Eq. 9, which has the same gradient as the flow matching objective that matches v_t to the marginal vector field u_t.*

$$\mathcal{L}_{SCFM} = \mathbb{E}_{\substack{\boldsymbol{x}_1\sim p_{data},\boldsymbol{x}_t\sim p_t(\boldsymbol{x}_t|\boldsymbol{x}_1)\\ t\sim Unif[0,1]}}\left\|\sum_{s\in[S]}\mathbb{E}_{p_t(\boldsymbol{z}_t^{[S]}|\boldsymbol{x}_t^{[S]})}\left[v_t(\boldsymbol{x}_t^s|\boldsymbol{z}_t^s;\theta)\right] - u_t\left(\boldsymbol{x}_t^s \mid \boldsymbol{x}_1^s\right)\right\|^2,$$

Proof. Assume regularity conditions that gaurantees the exchange of integration and divergence, differentiation w.r.t. t.

We first show that $\int\sum_{s\in[S]}u_t(\boldsymbol{x}_s|\boldsymbol{x}_s^1)p_t(\boldsymbol{x}_{[S]}^1|\boldsymbol{x}_{[S]})d\boldsymbol{x}_s^1$ is the marginalized vector field that generates $\{p_t(\boldsymbol{x}_{[S]})\}$. In the following $p(\boldsymbol{x}_1^1|\boldsymbol{x}_0) = p(\boldsymbol{x}_1^1)$ for simplicity of indexing,

$$\begin{aligned}\frac{d}{dt}p_t(\boldsymbol{x}_{[S]}) &= \int\frac{d}{dt}p_t(\boldsymbol{x}_{[S]}|\boldsymbol{x}_{[S]}^1)p(\boldsymbol{x}_{[S]}^1)d\boldsymbol{x}_{[S]} \\ &= \int\sum_{s\in[S]}\frac{d}{dt}p_t(\boldsymbol{x}_s|\boldsymbol{x}_s^1)\cdot\prod_{j\neq s}p_t(\boldsymbol{x}_j|\boldsymbol{x}_j^1)p(\boldsymbol{x}_{[S]}^1)d\boldsymbol{x}_{[S]} \\ &= \int\sum_{s\in[S]}-\nabla\cdot(p_t(\boldsymbol{x}_s|\boldsymbol{x}_s^1)u_t(\boldsymbol{x}_s|\boldsymbol{x}_s^1))\prod_{j\neq s}p_t(\boldsymbol{x}_j|\boldsymbol{x}_j^1)p(\boldsymbol{x}_{[S]})d\boldsymbol{x}_{[S]}^1 \\ &= -\nabla\cdot\left(\sum_{s\in[S]}\int u_t(\boldsymbol{x}_s|\boldsymbol{x}_s^1)p_t(\boldsymbol{x}_{[S]}^1|\boldsymbol{x}_{[S]})d\boldsymbol{x}_{[S]}^1p_t(\boldsymbol{x}_{[S]})\right) \\ &= -\nabla\cdot\left(\mathbb{E}_{p_t(\boldsymbol{x}_s^1|\boldsymbol{x}_{[S]})}\left[\sum_{s\in[S]}u_t(\boldsymbol{x}_s|\boldsymbol{x}_s^1)\right]p_t(\boldsymbol{x}_{[S]})\right).\end{aligned} \tag{11}$$

The second equality is by conditional independence of the transported samples for each $s\in[S]$ and applying chain rule on the product $p_t(x_s|x_s^1)\prod_{s\in[S]}p_t(\boldsymbol{x}_s|\boldsymbol{x}_s^1)$. This shows the marginal vector field is additive in the time index s following the marginal vector field defined for each $s\in[S]$.

Now, we'd like to derive the conditional flow matching objective from the marginal flow matching, and show the two has the same gradient with respect to the NN parameterized marginal vector field v_t. The marginal VF for LDS takes the form

$$\mathbb{E}_{p_t(\boldsymbol{x}_{[S]})}\|v_t(\boldsymbol{x}_{[S]}) - u_t(\boldsymbol{x}_{[S]})\|^2. \tag{12}$$

It is then sufficient to look at the cross term and the squared term on v_t. Firstly,

$$\begin{aligned}
&\mathbb{E}_{p_t(\boldsymbol{x}_{[S]})}\langle v_t(\boldsymbol{x}), u_t(\boldsymbol{x})\rangle \\
=&\int \left\langle v_t(\boldsymbol{x}), \sum_{s\in[S]} \int u_t(\boldsymbol{x}_s|\boldsymbol{x}_s^1) p_t(\boldsymbol{x}_s^1|\boldsymbol{x}_{[S]}) d\boldsymbol{x}_s^1 \right\rangle p_t(\boldsymbol{x}_{[S]}) d\boldsymbol{x}_{[S]} \\
=&\int \sum_{s\in[S]} \int \langle v_t(\boldsymbol{x}), u_t(\boldsymbol{x}_s|\boldsymbol{x}_s^1)\rangle \int p_t(\boldsymbol{x}_{[S]}^1|\boldsymbol{x}_{[S]}) d\boldsymbol{x}_{-s}^1 d\boldsymbol{x}_s^1 p_t(\boldsymbol{x}_{[S]}) d\boldsymbol{x}_{[S]} \\
=&\int \sum_{s\in[S]} \langle v_t(\boldsymbol{x}), u_t(\boldsymbol{x}_s|\boldsymbol{x}_s^1)\rangle p_t(\boldsymbol{x}_{[S]}|\boldsymbol{x}_{[S]}^1) p_t(\boldsymbol{x}_{[S]}^1) d\boldsymbol{x}_{[S]}^1 d\boldsymbol{x}_{[S]} \\
=&\mathbb{E}_{p_t(\boldsymbol{x}_{[S]}|\boldsymbol{x}_{[S]}^1)p(\boldsymbol{x}_{[S]}^1)} \left\langle v_t(\boldsymbol{x}), \sum_{s\in[S]} u_t(\boldsymbol{x}_s|\boldsymbol{x}_s^1) \right\rangle
\end{aligned}$$

for the quadratic term, it directly follows from iterated expectations

$$\mathbb{E}_{p_t(\boldsymbol{x}_{[S]})}\|v_t(\boldsymbol{x})\|^2 = \mathbb{E}_{p_t(\boldsymbol{x}_{[S]}|\boldsymbol{x}_{[S]}^1)p(\boldsymbol{x}_{[S]}^1)}\|v_t(\boldsymbol{x})\|^2.$$

Therefore optimizing v_t with Eq. 12 is equivalent to optimizing the marginal flow matching objective

$$\inf_{v_t} \mathbb{E}_{p_t(\boldsymbol{x}_{[S]}|\boldsymbol{x}_{[S]}^1)p(\boldsymbol{x}_{[S]}^1)} \left\| v_t(\boldsymbol{x}) - \sum_{s\in[S]} u_t(\boldsymbol{x}_s|\boldsymbol{x}_s^1) \right\|^2.$$

Finally, to introduce the structured FM objective with latent dynamical system, we verify the marginal vector field v_t arisen from marginalizing $v_t(\boldsymbol{x}_s|\boldsymbol{z}_s)$ generates the probability path $\{p_t(\boldsymbol{x}_{[S]})\}$. The proof is similar to the previous ones, where we verify that the continuity equation is satisfied.

$$\begin{aligned}
\frac{d}{dt}p_t(\boldsymbol{x}_{[S]}) &= \int \frac{d}{dt} p_t(\boldsymbol{x}_{[S]}|\boldsymbol{z}_{[S]}) p(\boldsymbol{z}_{[S]}) d\boldsymbol{z}_{[S]} \\
&= \int \sum_{s\in[S]} \frac{d}{dt} p_t(\boldsymbol{x}_s|\boldsymbol{z}_s) \prod_{j\neq s} p_t(\boldsymbol{x}_j|\boldsymbol{z}_j) p(\boldsymbol{z}_{[S]}) d\boldsymbol{z}_{[S]} \\
&= \int \sum_{s\in[S]} -\nabla\cdot(v_t(\boldsymbol{x}_s|\boldsymbol{z}_s) p_t(\boldsymbol{x}_s|\boldsymbol{z}_s)) \prod_{j\neq s} p_t(\boldsymbol{x}_j|\boldsymbol{z}_j) p(\boldsymbol{z}_{[S]}) d\boldsymbol{z}_{[S]} \\
&= -\nabla\cdot \sum_{s\in[S]} \int v_t(\boldsymbol{x}_s|\boldsymbol{z}_s) p_t(\boldsymbol{z}_{[S]}|\boldsymbol{x}_{[S]}) d\boldsymbol{z}_{[S]} \cdot p(\boldsymbol{x}_{[S]}) \\
&= -\nabla\cdot \left(\mathbb{E}_{p_t(\boldsymbol{z}_{[S]}|\boldsymbol{x}_{[S]})} \left[\sum_{s\in[S]} v_t(\boldsymbol{x}_s|\boldsymbol{z}_s) \right] p(\boldsymbol{x}_{[S]}) \right).
\end{aligned}$$

It is notable that the conditional independence and Markov assumption gives rise to the filtering probability $p(\boldsymbol{z}_s|\boldsymbol{x}_{[S]})$ and $p(\boldsymbol{x}_s^1|\boldsymbol{x}_{[S]})$.

In particular, the objective Eq. 9 depends on the entire sequence through the sum over $[S]$, due to the conditional independence structure of the likelihood. As it requires access to the full observed sequence at every step $s \in [S]$, the training procedure is entirely offline.

□

B Experiment Details

All experiments are conducted on a MacBook Pro equipped with an Apple M2 Pro chip and 16 GB of memory.

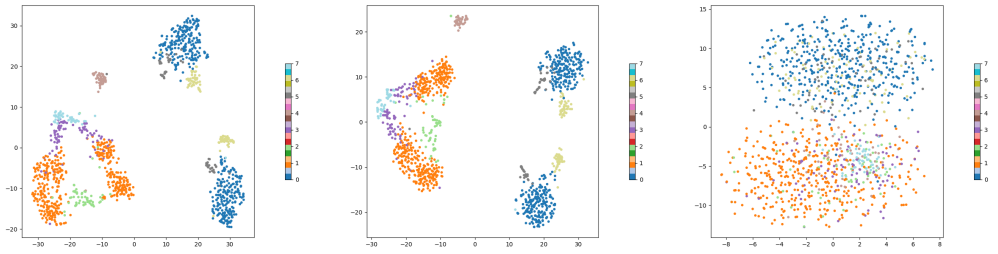
The Pinwheel dataset is a classic benchmark for density estimation task. SFA is demonstrated to be able to meaningfully capture latent distribution as well as observed distribution.

For VAE, we use Gaussian distribution with diagonal covariance for posterior and likelihood family. The parameters of Gaussian are parameterized by Multilayer Perceptrons (MLPs) and mapped from context vector. For SFA, we use MLP to parameterize the conditional vector fields v_t, and use Gaussian parametric family indexed by $t, \boldsymbol{x}$ for the posterior flow.

With latent mixture, we use constant time Gumbel-Softmax network to model the latent class probability for both GMVAE and Mixture-SFA. We use tanh activation function for all models applied to this dataset.

To compare, the samples and latent representations are generated based on the estimated conditional probabilities,

$$\boldsymbol{z}_i \overset{i.i.d.}{\sim} p(\boldsymbol{z}),\ \boldsymbol{x}_i|\boldsymbol{z}_i \sim p_1(\boldsymbol{x}|\boldsymbol{z}_i),\ \tilde{\boldsymbol{z}}_i|\boldsymbol{x}_i \overset{i.i.d.}{\sim} q_1(\boldsymbol{z}|\boldsymbol{x}_i).$$



(a) SFA (b) Latent FM (c) VAE

Figure 4: RNAseq dataset: Latent space visualization in 2D, projected with TSNE (perplexity=30).

B.1 Single Cell RNA-seq

The Single Cell RNA-seq dataset Kang et al. (2018) consists of transformed count vector of size 5000, therefore presents challenges in modelling with CNF and likelihood based optimization. SFA directly tackles this complexity by learning a meaningful latent representation while not requiring computation of log-likelihood during training.

We parameterize the CNF v_t with MLP, and uses a 32-dim Gaussian approximation family for the posterior that varies across time t and observation x. The VAE model uses Gaussian encoder and decoder with NN parameterized parameters that is time independent. For the latent FM, we use MLP encoder and decoder, 32 dim latent space and CNF to learn the latent distribution $p(\boldsymbol{z})$.

B.2 MNIST Data

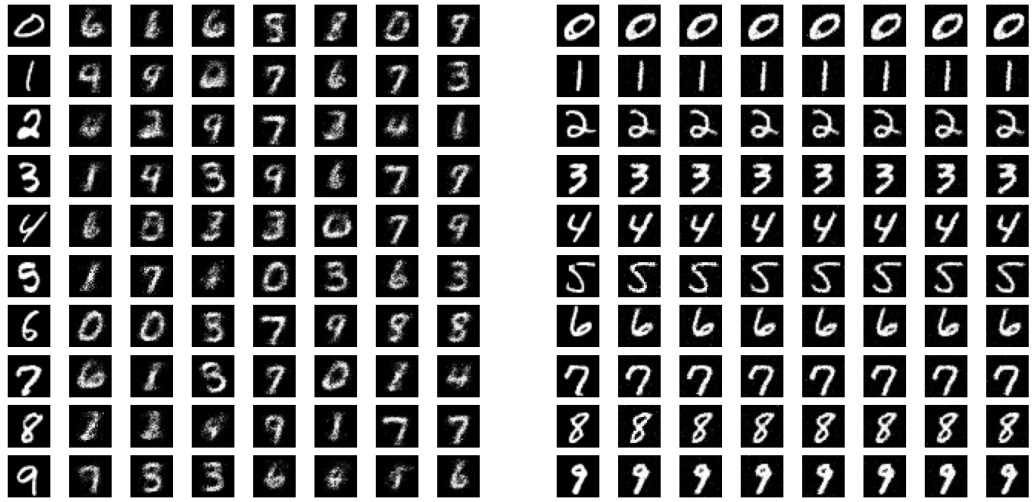
For GMVAE, we use Gaussian distribution with diagonal covariance for posterior and likelihood family. MLPs are used to map context vectors to the means and covariances of the Gaussians. The latent class probability is via a Gumbel-Softmax network, which uses MLP to map from context vector to logits, then apply Gumbel-Softmax trick for sampling.

For Mixture-SFA, we also use MLP to parameterize the conditional vector fields v_t and Gaussian posterior with parameter indexed by $t, \boldsymbol{x}$. It is notable that with larger differences in the dimensionality and scale, a linear map is used to firstly map the context vectors to vectors of the same size. Then apply concatenation and feed to the main network. We also use Gumbel-Softmax network to model the latent class probability. Alternatively, we can use a 10 dimensional vector field to model the distribution of logits.

For both model, we use softplus activation function and train until convergence. We observe that a smaller network is usually sufficed for modelling the latent, which also increases training speed.

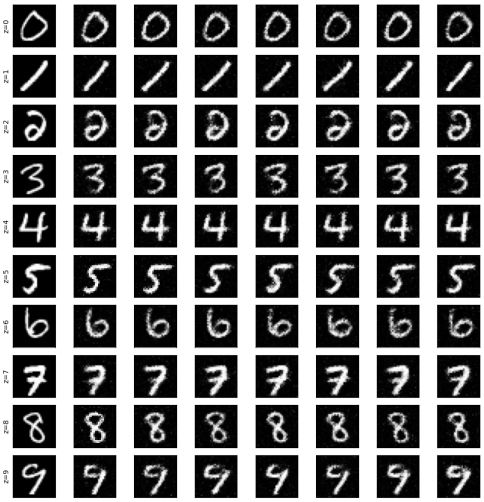
We compare the performance of the two methods from 2 perspectives.

1. **Posterior Predictive**: for every test sample $\boldsymbol{x}_i$, we first sample from the posterior $\boldsymbol{z}_i|\boldsymbol{x}_i \sim q(\boldsymbol{z}|\boldsymbol{x})$, then sample from the likelihood $\boldsymbol{x}_{i,new}|\boldsymbol{z}_i \sim p(\boldsymbol{x}|\boldsymbol{z}_i)$.
2. **Latent Space Representation**: for every test sample $\boldsymbol{x}_i$, we sample from the posterior $\xi_i|\boldsymbol{x}_i \sim q(\xi|\boldsymbol{x}_i)$, then from $\boldsymbol{z}_i|\boldsymbol{x}_i \sim q(\boldsymbol{z}|\boldsymbol{x})$ to obtain a latent representation of the observed

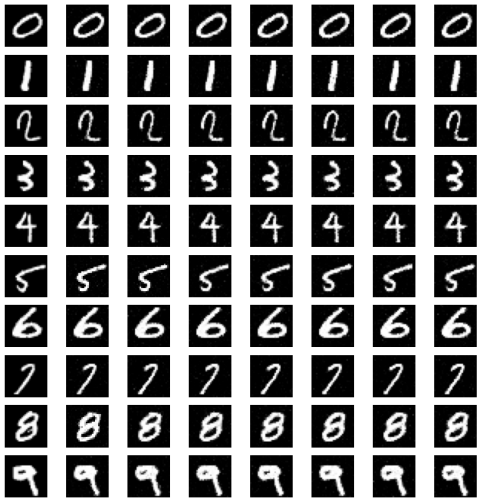


(a) VAE Posterior Predictive

(b) SFA Posterior Predictive



(c) GMVAE Posterior Predictive



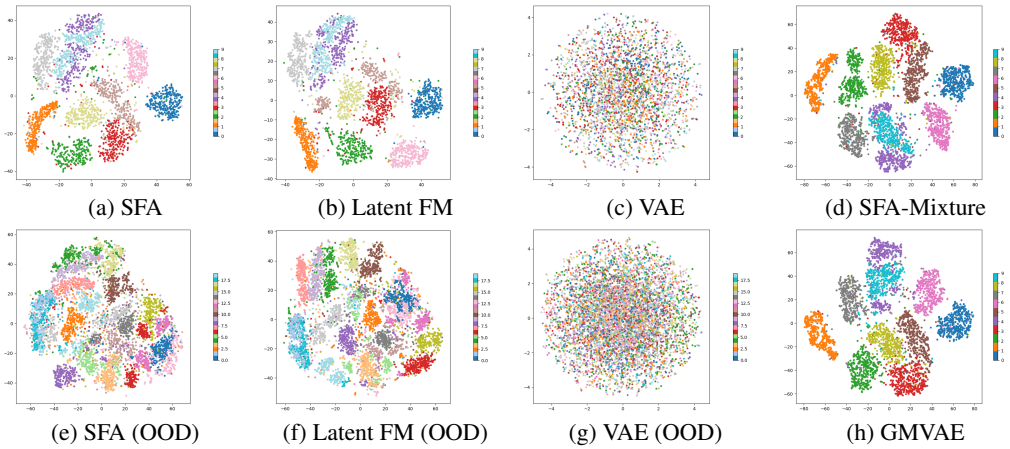
(d) Mixture-SFA Posterior Predictive

Figure 5: Comparison of Posterior Predictive Results: (a) GMVAE and (b) Mixture-SFA.

data point. We visualize the class probability samples $\{\xi_i\}$ with histogram, and the sample $\{\boldsymbol{z}_i\}$ with TSNE (complexity = 30) projected from 64 dimensional space onto 3 dimensional space.

Table 4: Comparison of metrics for MNIST dataset between VAE, latent FM and SFA. Evaluated on a held-out set of size 1000. The OOD dataset consists of first 10 classes of letters and the first 10 classes of digits in EMNIST. The clustering is done in the latent space via k-means with k given.

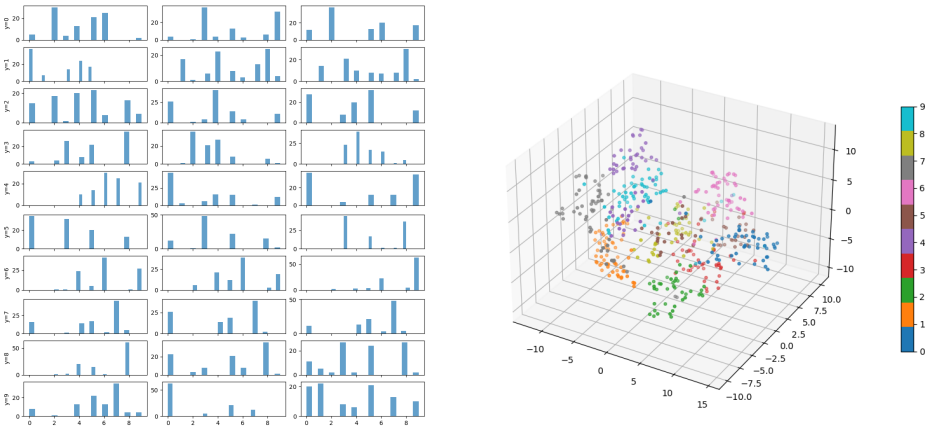
	$\log p(x\|z)\uparrow$	$\log p(z\|x)\uparrow$	Vendi $\uparrow$	SSIM $\uparrow$	NMI (OOD) $\uparrow$	ARI(OOD) $\uparrow$
VAE	$\mathbf{-453.648}$	-85.448	63.286	0.419	0.039(0.033)	0.017(0.012)
VampVAE	-584.845	155.820	1.140	0.866	0.006(0.006)	0.000(0.000)
Latent FM	-	-	8.380	**0.980**	0.488(0.392)	**0.381**(0.194)
Latent FM (VAE)	-910925	-11.192	19.631	0.697	0.309(0.152)	0.205(0.073)
SFA	-916.901	**793.262**	25.589	0.716	0.490(**0.394**)	0.356(**0.208**)
w/Deterministic Latent	-858.385	-	10.189	0.679	**0.501**(0.333)	0.379(0.155)
w/CNF Posterior	-907.998	356.141	23.166	0.654	0.485(0.325)	0.355(0.118)



(a) SFA (b) Latent FM (c) VAE (d) SFA-Mixture

(e) SFA (OOD) (f) Latent FM (OOD) (g) VAE (OOD) (h) GMVAE

Figure 6: MNIST dataset: Latent space visualization in 2D projected with TSNE (perplexity=50). (a)-(c),(e)-(g) follows from the continuous latent graphical model, (d) and (b) employs latent mixture model. The OOD dataset consists of first 10 classes of letters and the first 10 classes of digits in EMNIST.



(a) Posterior Samples (ξ) (b) Posterior Samples ($\boldsymbol{z}$)

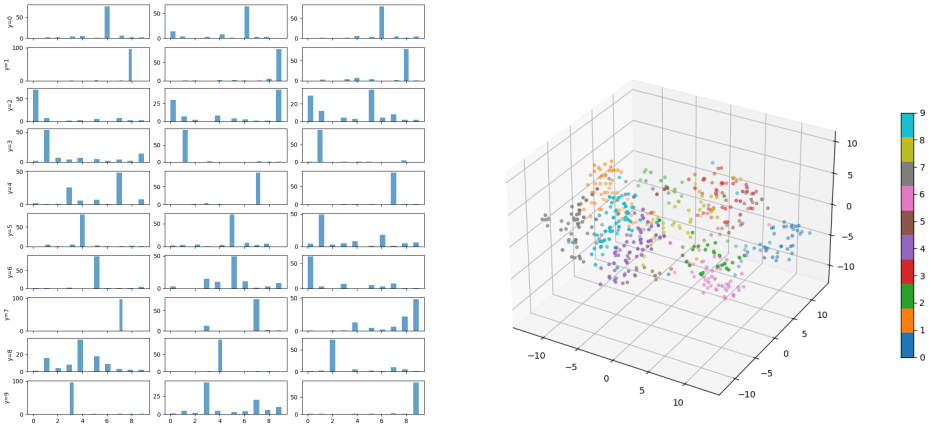
Figure 7: GMVAE Posterior Samples: (a) samples from latent variable ξ: each column corresponds to different images, each row corresponds to different class label; and (b) latent variable $\boldsymbol{z}$, with TSNE projection from 64 dimensional to 3 dimensional space.

SoftNMI To assess the quality of latent probabilistic cluster assignment for Mixture-SFA and GMVAE, we use a soft Normalized Mutual Information (softNMI), which computes the discrepancy between a one-hot label vector and a probability vector based on entropy,

$$\text{softNMI}(\text{p}, \text{q}) = \frac{H(p) + H(q) - H(p, q)}{H(p) + H(q)} \in [0, 1], \tag{13}$$

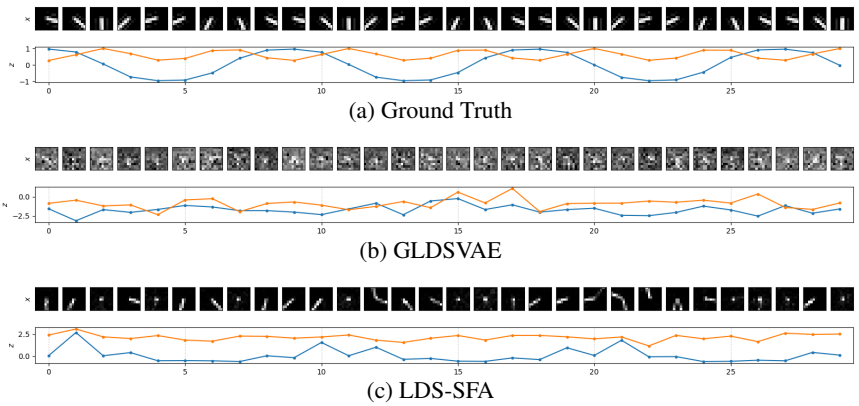
where $H(p)$ is the entropy function on the marginal, $H(p, q)$ is the entropy on the joint. Higher score suggests higher correlation between the posterior class assignment probability and true class label.

From Table 4, we observe the stochastic latent helps to increase the diversity of the generation, while variants of VAE has poor generation quality, the FM based models have better performance in image generation quality. However, there's a notable trade-off in reconstruction quality and generation diversity, where stochastic latent have higher Vendi score while suffer a slight decrease in SSIM score, and vice versa for the deterministic latent. In addition, for downstream clustering task, SFA with deterministic latent has better NMI and ARI score for within distribution sample, yet SFA with stochastic latent has better NMI and ARI for out-of-distribution samples.



(a) Posterior Samples (ξ) (b) Posterior Samples ($\boldsymbol{z}$)

Figure 8: Mixture-SFA Results: (a) Posterior Samples (ξ) and (b) Posterior Samples ($\boldsymbol{z}$).



(a) Ground Truth

(b) GLDSVAE

(c) LDS-SFA

Figure 9: Comparison of Pendulum Dynamics: Ground Truth and GLDSVAE.

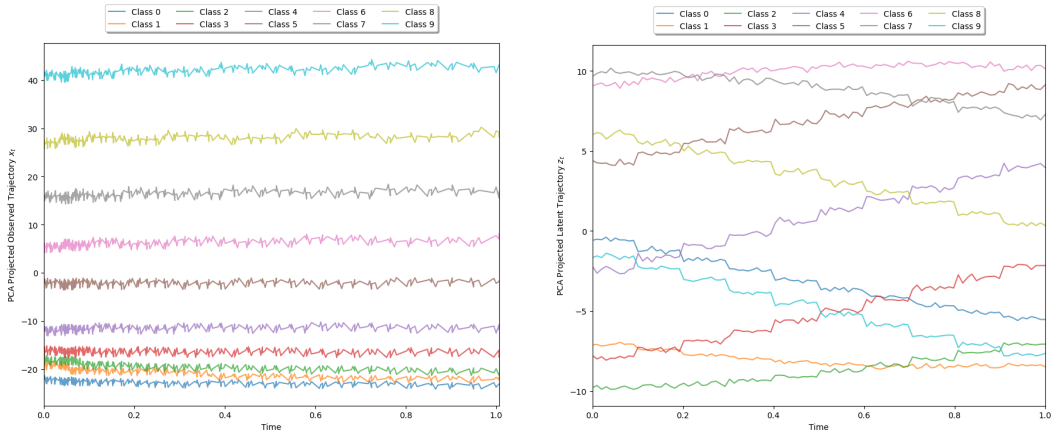
B.3 Latent Dynamical System

To model the full conditional posterior $q(\boldsymbol{z}_s|\boldsymbol{z}^{[s-1],\boldsymbol{x}^{[S]}})$, we use sequence model to separately encode the observation sequence $\boldsymbol{x}_{[S]}$ and full latent history $\boldsymbol{z}^{[s-1]}$ to a context vector. For the input sequence, as it is fixed in length, we use an attention pooling to reduce it to a fixed length vector. For the latent, we use GRU to iteratively encode the historic latent sequence into fixed length latent embeddings. We apply this to both GLDSVAE and LDS-SFA. Therefore, the full posterior of GLDSVAE is modeled by

$$\boldsymbol{z}^s|\boldsymbol{z}^{[s-1]}, \boldsymbol{x}^{[S]} \sim Gaussian\left(\mu\left(h_z(\boldsymbol{z}^{[s-1]}), h_x(\boldsymbol{x}^{[S]})\right), \Sigma\left(h_z(\boldsymbol{z}^{[s-1]}), h_x(\boldsymbol{x}^{[S]})\right)\right).$$

The likelihood, with conditional independence assumption is modeled by $Gaussian\left(\mu(\boldsymbol{z}_s), \Sigma(\boldsymbol{z}_s)\right)$.

The likelihood is modeled with CNF, where the vector field is indexed by MLP, the incorporation of time and latent representation $\boldsymbol{z}$ is done via concatenation.



(a) Sampled trajectory $\{x_t\}_{t\in[0,1]}$ from $p_t(x_t|z_1)$.

(b) Sampled trajectory $\{z_t\}_{t\in[0,1]}$ from $q_t(z_t|x_1)$.

Figure 10: 1D PCA projected sampled trajectories in observation space and latent spaces.

B.4 Trajectory in Time

We investigate the behavior of the conditional trajectory for both z_t and x_t with MNIST dataset. Fig. 10b visualizes the latent trajectory $\{z_t\}_{t\in[0,1]}$, sampled from $q_t(z_t|x_1)$. The position at $t = 1$ deviates from the random sample from Gaussian at $t = 0$. Upon conditioning on z_1, we sample the observed trajectory $\{x_t\}_{t\in[0,1]}$ from $p_t(x_t|z_1)$. From Fig. 10a, we observe that the paths of different integers does not cross across $t \in [0, 1]$. This suggests that the latent $q_t(z_t|x_1)$ is learned meaningfully to accounts for the structure that generates the observed data x_1. This observation is precisely why our proposed SFA does not suffer from posterior collapse.

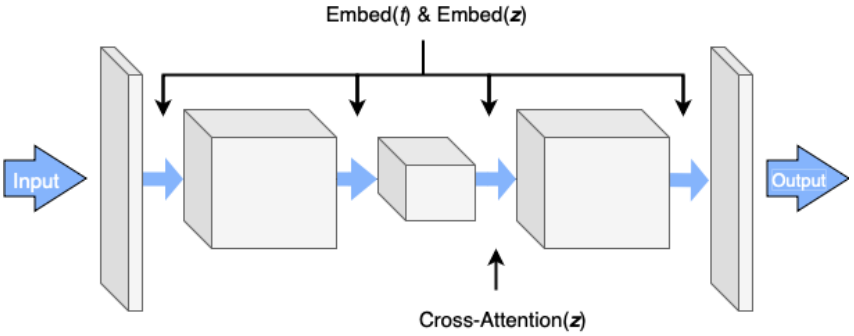
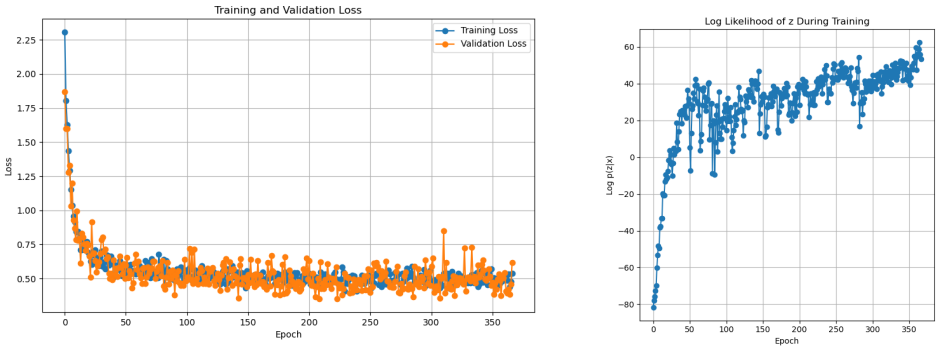


Figure 11: Simplified illustrative Diagram for UNet architecture: parameterizes likelihood conditional vector field, conditioning on time t, and latent variable $\boldsymbol{z}$. The three blocks represents the encoder, bottleneck and decoder. The input and output are both tensors of the same shape as the observed image.

B.5 CIFAR10

For more realistic image dataset, we adapt UNet architecture to accommodate the incorporation of additional context of latent representation vector $\boldsymbol{z}$. In particular, we modified the architecture so the information of $\boldsymbol{z}$ is used robustly even in the presence of strong UNet skip connections. We experimented with FiLM style conditioning (such as in (Wang et al., 2023)), which is not suitable for our paradigm and produce blurry images, which maybe owing to flexible stochastic latent distributions. Instead, we consider injecting latent information through (1) concatenation with time embedding, (2) cross-attention. (1) is applied across all blocks and layers replacing the original time embedding. (2) treats $\boldsymbol{z}$ as a set of tokens and lets each spatial location query the relevant part of $\boldsymbol{z}$. We experimented with applying cross-attention at (a) the end of encoder, bottleneck and decoder, and (b) bottleneck block only. (b) appears to be more robust in incorporating the latent information, and is stable in training. The specific architecture implemented is illustrated in Fig. 11.

The likelihood model's vector field is chosen to be parameterized by the UNet, whereas the posterior model is chosen to be less flexible, as we notice the more powerful encoder model does not provide meaningful directions to the generation model. We also notice that the role of time embedding is different from the $\boldsymbol{z}$ embedding, where larger dimension for t embedding helps to refine the details of the image; whereas the $\boldsymbol{z}$ embedding directs the coarser structure of the image.
We include preliminary result on Cifar10. Fig. 12 includes the train and validation loss, and the posterior log likelihood for the latent during training. Fig. 13 includes the generated samples (columns starting from the 2nd position) given a reference image x_1 that is in the 1st column.



(a) Train/Validation Loss (b) Log-likelihood

Figure 12: CIFAR-10: SFA metrics during training.

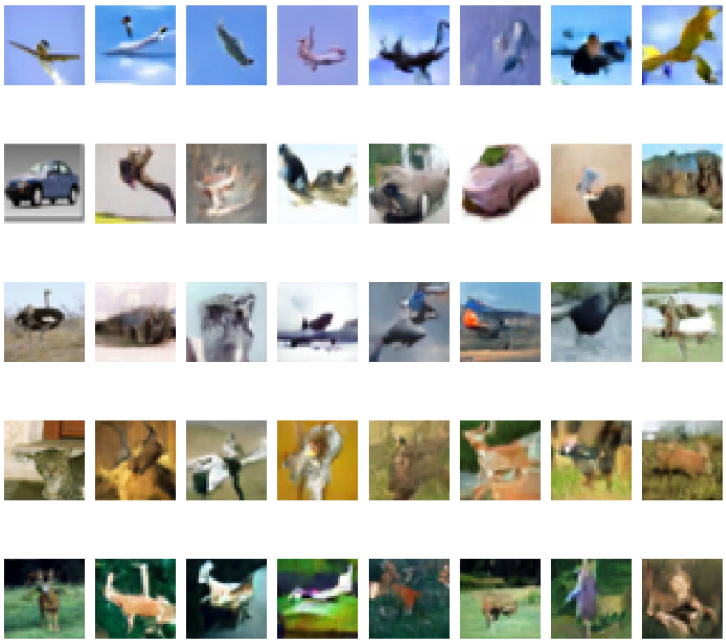


Figure 13: Reconstruction plot for Cifar10 dataset. The left most column represents the image x_1 from Cifar10 that we condition on, the other columns represents the generated images by sampling from $p(x|z_1)$ where the latent variable is sampled from the posterior approximation $z_1 \sim q(z|x_1)$.

C THE USE OF LARGE LANGUAGE MODELS (LLMS)

LLM is used to refine the code of model architecture and is used to polish the writing of the draft. LLM is not used in generate research idea or writing to the extent that they could be regarded as a contributor.