Training Latent Diffusion Models with Interacting Particle Algorithms

Tim Y. J. Wang[†] **Juan Kuntz** **O. Deniz Akyildiz**[†]
[†]Department of Mathematics, Imperial College London

Abstract

We introduce a novel particle-based algorithm for end-to-end training of latent diffusion models. We reformulate the training task as minimizing a free energy functional and obtain a gradient flow that does so. By approximating the latter with a system of interacting particles, we obtain the algorithm, which we underpin theoretically by providing error guarantees. The novel algorithm compares favorably in experiments with previous particle-based methods and variational inference analogues.

1 INTRODUCTION

Diffusion Models (DMs) introduced in Sohl-Dickstein et al. (2015), and further developed in Ho et al. (2020); Song et al. (2021c), excel at numerous generative modeling tasks. Examples include image synthesis (Dhariwal and Nichol, 2021), protein design (Watson et al., 2023), and language modeling (Nie et al., 2025). They work by progressively adding noise to data to transform the data distribution into an easy-to-sample reference distribution, and then learn to revert this noising process.

However, these steps take place in the data's *ambient* space, which is typically high dimensional. For this reason, DM training and inference prove computationally expensive. To alleviate this issue, Rombach et al. (2022); Vahdat et al. (2021); Wehenkel and Louppe (2021) and others proposed using a Variational Autoencoder (VAE) (Kingma and Welling, 2013) to map back-and-forth between the high-dimensional ambient space and a low-dimensional *latent* space, and carrying out the noising/de-noising steps in the latter. To date, the world's most popular DMs (e.g., Stable Diffusion (Podell et al., 2024)) fall into this Latent Diffusion Model (LDM) category.

Proceedings of the 29[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

Recently, Kuntz et al. (2023); Lim et al. (2024); Lim and Johansen (2024) have reported performance gains for parameter estimation in simple latent variable models and generator networks by replacing variational approximations with particle-based ones. Here, we investigate whether this is also the case for LDMs and introduce, to the best of our knowledge, the first particle-based method for LDM training.

Contributions.

(C1) We recast the problem of LDM training as the minimization of a free energy functional, characterize the functional's minima (Proposition 2.1), identify a gradient flow that minimizes it, and establish its exponential convergence under standard assumptions (Theorem 3.1).

(C2) Approximating the flow in (C1), we obtain Interacting Particle Latent Diffusions (IPLDs)—a simple, particle-based, and encoder-free algorithm for LDM training well-adapted to modern compute environments—and we derive non-asymptotic bounds on its error (Theorem 3.2).

(C3) Through several practical improvements, we obtain an efficient and scalable version of the algorithm (Section 3.3) and demonstrate its effectiveness in numerical experiments (Section 5).

(C4) Lastly, by approaching LDMs from the above unexplored angle, we open the door to other novel LDM training methods. In particular, our approach connects latent diffusion models to the rich body of work on gradient flows and interacting particle systems stemming from the optimal transport literature (Villani et al., 2008; Chaintron and Diez, 2022)–a connection that can spur the design of new algorithms as we demonstrate here.

Paper structure. The paper is organized as follows. First, we review the necessary background on LDMs and identify relevant loss functions (Section 2). Next, we obtain IPLD: an algorithm for minimizing this loss (Section 3). We do so by identifying a gradient flow

that minimizes the loss (Section 3.1), approximating it (Section 3.2), and incorporating a series of practical improvements (Section 3.3). We then survey the related literature (Section 4) and experimentally compare IPLD with relevant baselines (Section 5). We conclude with a discussion of our results, IPLD's limitations, and future research directions (Section 6).

Notation. We use $\mathsf{X} = \mathbb{R}^{d_x}$ and $\mathsf{Z} = \mathbb{R}^{d_z}$ to denote the ambient and latent spaces, $\Theta = \mathbb{R}^{d_\theta}$ and $\Phi = \mathbb{R}^{d_\phi}$ the decoder's and DM's parameter spaces (c.f. Section 2), $\{x^1, \dots, x^M\}$ the training set, $[M] := \{1, \dots, M\}$ the set of indices, and $\mathcal{P}_2(\mathbb{R}^d)$ the space of probability distributions on $\mathbb{R}^d$ with finite second moment. To denote product measures, we write $q^{1:M} = (q^1, \dots, q^M) \in \mathcal{P}_2(\mathsf{Z})^M$ for M-tuples of distributions $q^1, \dots, q^M$ over Z.

2 PRELIMINARIES

We consider latent-space versions of Denoising Diffusion Probabilistic Models (DDPMs) (Ho et al., 2020) similar to those in Rombach et al. (2022); Vahdat et al. (2021); Wehenkel and Louppe (2021). To be specific, for a fixed data point $x \in \mathsf{X}$, we consider the following latent variable model:

$$p_{\theta,\phi}(x, z_{0:K}) = p_\phi(x|z_0)p_\theta(z_{0:K}) \tag{1}$$

where $p_\phi(x|z_0) = \mathcal{N}(x; g_\phi(z_0), \sigma^2 I)$ is an isotropic Gaussian decoder, with $g_\phi : \mathsf{Z} \to \mathsf{X}$ denoting a neural network parameterized by ϕ, that maps from the latent space to the ambient space, and the prior

$$p_\theta(z_{0:K}) := p(z_K) \prod_{k=1}^{K} p_{\theta,k}(z_{k-1}|z_k),$$

is a DDPM parameterized by θ (Ho et al., 2020). The DDPM end point is defined with a standard Gaussian distribution $p_K(z_K) := \mathcal{N}(z_K; 0, I)$ at time K, and its backward kernels are also Gaussian:

$$p_{\theta,k}(z_{k-1}|z_k) := \mathcal{N}(z_{k-1}; \mu_{\theta,k}(z_k), \beta_k^2 I),$$

where $(k, z) \mapsto \mu_{\theta,k}(z)$ denotes a neural network parameterized by θ; and $\{\beta_k\}_{k=1}^K$ a fixed noise schedule.

Suppose we are given a dataset $\{x^1, \dots, x^M\}$ where $x^m \sim p_{\text{data}}$ for $m \in [M]$. To fit the generative model in (1), we aim to find parameters $(\theta_\star, \phi_\star)$ that maximize the *expected log-likelihood* $\mathbb{E}_{p_{\text{data}}}[\log p_{\theta,\phi}(X)]$. Given that we only have access to the empirical measure $p_{\text{data}}^M = (1/M)\sum_{m=1}^M \delta_{x^m}$, we approximate the expected log-likelihood with the empirical average:

$$\ell(\theta, \phi) := \frac{1}{M} \sum_{m=1}^{M} \log p_{\theta,\phi}(x^m), \tag{2}$$

where

$$p_{\theta,\phi}(x) := \int p_{\theta,\phi}(x, z_{0:K}) \mathrm{d}z_{0:K}$$

denotes the probability density the model assigns to a given datapoint x. This quantity is also called the *marginal likelihood*.

2.1 Minimizing Free Energy

The direct computation of $\ell(\theta, \phi)$ (and consequently of its gradients) is intractable, as $p_{\theta,\phi}(x)$ involves marginalising over the latent variables $z_{0:K}$. To circumvent this issue, we instead look at utilising lower bounds on $\ell(\theta, \phi)$. Using the standard lower bound of marginal likelihood, we obtain using Jensen's inequality that $\log p_{\theta,\phi}(x) \geq \mathbb{E}_q[\log p_{\theta,\phi}(x, \cdot)/q(\cdot)]$ for any x and distribution q over the latent space. We aim at generalising this bound for M data points where q is the prior distribution defined by the DDPM in the latent space.

To this end, consider the product measure $q^{1:M} \in \mathcal{P}_2(\mathsf{Z})^M$ and note that

$$\ell(\theta, \phi) \geq \frac{1}{M} \sum_{m=1}^{M} \mathbb{E}_{q^m} \left[\log \frac{p_{\theta,\phi}(x^m, z_0)}{q^m(z_0)} \right] \tag{3}$$

where we have for all $(\theta, \phi) \in \Theta \times \Phi$:

$$p_{\theta,\phi}(\cdot, z_0) := \int p_{\theta,\phi}(\cdot, z_{0:K}) \mathrm{d}z_{1:K}. \tag{4}$$

The negative of the quantity in the r.h.s. of (3) is termed *the free energy*, denoted $F(\theta, \phi, q^{1:M})$ (Bishop, 2006):

$$F(\theta, \phi, q^{1:M}) := \frac{1}{M} \sum_{m=1}^{M} \mathbb{E}_{q^m} \left[\log \frac{q^m(z_0)}{p_{\theta,\phi}(x^m, z_0)} \right]. \tag{5}$$

Noting that $-\ell(\theta, \phi) \leq F(\theta, \phi, q^{1:M})$ for all $(\theta, \phi, q^{1:M}) \in \Theta \times \Phi \times \mathcal{P}_2(\mathsf{Z})^M$, we see that minimizing (5) above over all parameters θ, ϕ and M-tuples $q^{1:M} = (q^1, \dots, q^M)$ is equivalent to maximizing $\ell(\theta, \phi)$ over all θ, ϕ; see, e.g., Neal and Hinton (1998).

However, the joint density $p_{\theta,\phi}(x, z_0)$ in (5) is still computationally prohibitive to evaluate due to marginalizing over the entire diffusion trajectory $z_{1:K}$. We therefore resort to one more upper bound for the negative log-likelihood $-\ell(\theta, \phi)$, leading to the *tilted free energy* (see Appendix A.1 for the full derivation):

$$\tilde{F}(\theta, \phi, q^{1:M}) := \frac{1}{M} \sum_{m=1}^{M} \mathbb{E} \left[\log \frac{q^m(z_{0:K})}{p_{\theta,\phi}(x^m, z_{0:K})} \right], \tag{6}$$

where the expectation is taken with respect to $q^m(z_{0:K})$ with $q^m(z_{0:K}) := q(z_{1:K}|z_0)q^m(z_0)$ for all $m \in [M]$ and

$$q(z_{1:K}|z_0) = \prod_{k=1}^{K} q(z_k|z_{k-1})$$

is the *forward process*, a product of Gaussian kernels as in DDPM.

Examining the second lower bound in (6), we note that $\tilde{F}$ can be decomposed as (cf. Appendix A.1):

$$\tilde{F}(\theta, \phi, q^{1:M}) = F(\theta, \phi, q^{1:M}) + \frac{1}{M}\sum_{m=1}^{M} \mathbb{E}_{q^m}[\mathcal{R}(\theta, z_0)],$$

where

$$\mathcal{R}(\theta, z_0) = D_{\mathrm{KL}}\big(q(z_{1:K}|z_0)\|p_\theta(z_{1:K}|z_0)\big) \geq 0. \quad (7)$$

So $\tilde{F}$ amounts to a regularized version of F that penalizes deviations from the forward process. Therefore, we can alternatively view $\tilde{F}$ as the free energy obtained replacing $p_{\theta,\phi}(x^m, z_0)$ in (5) with the *tilted* version:

$$\tilde{p}_{\theta,\phi}(x^m, z_0) := p_{\theta,\phi}(x^m, z_0)\exp(-\mathcal{R}(\theta, z_0)), \quad (8)$$

for $m \in [M]$. For this reason, similar arguments to those behind (Neal and Hinton, 1998, Theorem 2) yield the following result.

Proposition 2.1. *$(\theta_\star, \phi_\star)$ maximize $\tilde{\ell}(\theta, \phi) := M^{-1}\sum_{m=1}^{M}\log\tilde{p}_{\theta,\phi}(x^m)$ iff $(\theta_\star, \phi_\star, q_\star^{1:M})$ minimize $\tilde{F}$ for some $q_\star^{1:M}$, where $\tilde{p}_{\theta,\phi}(x^m) := \int \tilde{p}_{\theta,\phi}(x^m, z_{0:K})\mathrm{d}z_{0:K}$ for all $m \in [M]$.*

In an idealized setting where the backward process parameterized by θ is expressive enough to match the chosen forward process, the penalty $\mathcal{R}(\theta, z_0)$ vanishes, and our tilted log-likelihood $\tilde{\ell}(\theta, \phi)$ shares the same maxima and maximizers with the true $\ell(\theta, \phi)$ in (2). Hence, for optimal parameters $\theta_\star$ such that

$$\theta_\star \in \Theta_0 := \{\theta \in \Theta : \mathcal{R}(\theta, \cdot) \equiv 0\},$$

Proposition 2.1 implies that the minimizers $(\theta_\star, \phi_\star)$-components of $\tilde{F}(\theta, \phi, q^{1:M})$, our tractable proxy for optimization, also maximize the marginal log-likelihood ℓ, our objective of interest. While this is never perfectly achieved in practice, we believe a close approximation is possible when the forward and reverse processes are chosen with care (generally, the more expressive the latter is, the better).

Previous works such as Wehenkel and Louppe (2021) restrict q^m to mean-field Gaussian distributions of the sort $\mathcal{N}(z_0; \mu_\psi(x^m), \Sigma_\psi(x^m))$, where the mean and (diagonal) covariance matrix are parameterized by an encoder with parameters ψ, and they optimize $\tilde{F}$ over (θ, ϕ, ψ). In the following, we take a different approach and replace these parametric variational approximations with particle-based ones.

3 INTERACTING PARTICLE LATENT DIFFUSION

To obtain an algorithm for minimizing $\tilde{F}$ in (6), we follow similar steps to those taken in Kuntz et al. (2023) to obtain particle-based algorithms for minimizing F in (5) for simpler latent variable models (in particular, ones for which $p_{\theta,\phi}(x, z_0)$ have a standard Gaussian prior and do not have complex dependencies on latent variables z_0).

3.1 The gradient flow

To derive our algorithm, we search for an analogue of gradient descent (GD) applicable to $\tilde{F}(\theta, \phi, q^{1:M})$. A single update step of GD in the Euclidean space $\mathbb{R}^d$ for minimizing a function $f : \mathbb{R}^d \to \mathbb{R}$ is given by $x_{k+1} = x_k - h\nabla_x f(x_k)$. This update step is exactly the Euler discretization with step size $h > 0$ of the continuous-time *gradient flow* $\dot{x}_t = -\nabla_x f(x_t)$. The analogue of the continuous time gradient flow we now identify resides in the joint space of parameters $\Theta \times \Phi \ni (\theta, \phi)$ and distributions $\mathcal{P}_2(\mathsf{Z})^M \ni q^{1:M}$. Under this geometry, $\nabla\tilde{F} = (\nabla_\theta\tilde{F}, \nabla_\phi\tilde{F}, \nabla_{q^1}\tilde{F}, \ldots, \nabla_{q^M}\tilde{F})$ (see Appendix A.2 for details):

$$\nabla_\theta\tilde{F}(\theta, \phi, q^{1:M}) = \frac{1}{M}\sum_{m=1}^{M}\mathbb{E}_{q^m}\left[\nabla_\theta\mathcal{L}_D(\theta, z_0)\right], \quad (9)$$

$$\nabla_\phi\tilde{F}(\theta, \phi, q^{1:M}) = -\frac{1}{M}\sum_{m=1}^{M}\mathbb{E}_{q^m}\left[\nabla_\phi\log p_\phi(x^m|z_0)\right], \quad (10)$$

$$\nabla_{q^m}\tilde{F}(\theta, \phi, q^{1:M}) = \left[\nabla_{z_0}\cdot\left[q^m(z_0)\nabla_{z_0}\mathcal{L}^m(\theta, \phi, z_0)\right] - \Delta_{z_0}q^m(z_0)\right], \quad (11)$$

where the last equation holds for all $m \in [M]$, and we have defined:

$$\mathcal{L}^m(\theta, \phi, z_0) := \log(p_\phi(x^m|z_0)) - \mathcal{L}_D(\theta, z_0), \quad (12)$$

with $\mathcal{L}_D(\theta, z_0)$ being the *diffusion loss* for the latent space DDPM prior[1]:

$$\mathcal{L}_D(\theta, z_0) := \mathbb{E}_{q(z_{1:K}|z_0)}\left[\log\frac{q(z_{1:K}|z_0)}{p_\theta(z_{0:K})}\right]. \quad (13)$$

The gradient flow then reads

$$(\dot{\theta}_t, \dot{\phi}_t, \ldots \dot{q}_t^{1:M}) = -\nabla\tilde{F}(\theta_t, \phi_t, q_t^{1:M}). \quad (14)$$

Using the results of Caprio et al. (2025), it is straightforward to obtain sufficient conditions under which the flow converges exponentially fast to $\tilde{F}$'s minimizers. We state a concise version of the result and associated assumptions; we defer the full statement and proof to Appendix C.1. Let $\tilde{\rho}_{\theta,\phi}(\cdot) := \tilde{p}_{\theta,\phi}(x, \cdot)$ be the unnormalized tilted posterior, and $\tilde{\pi}_{\theta,\phi}(\cdot) = \tilde{\rho}_{\theta,\phi}(\cdot)/A_{\theta,\phi}$ be its normalized version with $A_{\theta,\phi}$ being the normalizing constant.

[1]We point out that the loss is dependent on the number of diffusion time steps K.

A1 (Model regularity). *We assume that*

1. *For all $z \in \mathsf{Z}$, $(\theta, \phi) \mapsto \tilde{\pi}_{\theta,\phi}(z)$ and $(\theta, \phi) \mapsto A_{\theta,\phi}$ are differentiable;*
2. *for all $(\theta, \phi) \in \Theta \times \Phi$, $\tilde{\pi}_{\theta,\phi}$ is twice continuously differentiable;*
3. *$\tilde{\rho}_{\theta,\phi}(z) > 0$ for all $z \in \mathsf{Z}$ and $(\theta, \phi) \in \Theta \times \Phi$.*

A2 (Regularity of solutions). *For any initial conditions $(\theta, \phi, q^{1:M}) \in \mathcal{M}^{1:M}$, the gradient flow has a classical solution $(\theta_t, \phi_t, q_t^{1:M})_{t\geq 0}$ with $(\theta_0, \phi_0, q_0^{1:M}) = (\theta, \phi, q^{1:M})$. Furthermore, for all $m \in [M]$ and $t \geq 0$, q_t^m has a Lebesgue density in $\mathcal{C}^{1,2}([0, \infty) \times \mathsf{Z}, \mathbb{R}^+)$ and $(\theta_t, \phi_t) \in \mathcal{C}^1([0, \infty), \Theta \times \Phi)$.*

A3 (Strong log-concavity). *For all $x \in \mathsf{X}$, the tilted joint density $\tilde{p}_{\theta,\phi}(x, z)$ is λ-strongly log-concave in (θ, ϕ, z) for some $\lambda > 0$.*

Theorem 3.1. *Suppose Assumptions **A** 1-**A** 3 hold, then $\tilde{\ell}$ has a unique maximizer $(\theta_\star, \phi_\star)$ and the flow converges exponentially fast to it: for λ independent of M and all $t \geq 0$,*

$$||(\theta_t, \phi_t) - (\theta_\star, \phi_\star)|| \leq \sqrt{\frac{2[\tilde{F}(\theta_0, \phi_0, q_0^{1:M}) - \tilde{F}_\star]}{\lambda}} e^{-\lambda t},$$

where we denote $\tilde{F}_\star := \inf_{(\theta,\phi,q^{1:M})} \tilde{F}(\theta, \phi, q^{1:M})$ and $||\cdot||$ denotes the Euclidean norm.

Proof (Sketch). Given the fact that $(\theta, \phi, z_0) \mapsto \log(\tilde{p}_{\theta,\phi}(x^m, z_0))$ is λ-strongly concave for each m in $[M]$, it follows that

$$\tilde{\ell}(\theta, \phi, z_0^{1:M}) := \frac{1}{M} \sum_{m=1}^{M} \log \tilde{p}_{\theta,\phi}(x^m, z_0^m), \tag{15}$$

is also λ-strongly concave. The result is then an application of the extended log-Sobolev inequality under strong log-concavity (cf. Definition C.1 and Caprio et al. (2025)), whence the exponential convergence follows from Grönwall's inequality. □

3.2 Approximating the flow and a simple algorithm

In almost all cases, (9–14) defines an intractable set of equations and we must approximate it. To do so, we exploit the fact (Kuntz et al., 2023, Section 2) that they form the Fokker-Planck equation satisfied by the law of the following McKean-Vlasov SDE (Chaintron and Diez, 2022):

$$\mathrm{d}\theta_t = -\frac{1}{M} \sum_{m=1}^{M} \mathbb{E}_{q_t^m} \left[\nabla_\theta \mathcal{L}_D(\theta_t, Z_{0,t}^m)\right] \mathrm{d}t, \tag{16}$$

$$\mathrm{d}\phi_t = \frac{1}{M} \sum_{m=1}^{M} \mathbb{E}_{q_t^m} \left[\nabla_\phi \log p_{\phi_t}(x^m | Z_{0,t}^m)\right] \mathrm{d}t, \tag{17}$$

$$\mathrm{d}Z_{0,t}^m = \nabla_{z_0} \mathcal{L}^m(\theta_t, \phi_t, Z_{0,t}^m) dt + \sqrt{2} \mathrm{d}W_t^m, \tag{18}$$

where we define the last equation for all $m \in [M]$, q_t^m denotes the law of $Z_{0,t}^m$, and $(W_t^{1:M})_{t\geq 0}$ denotes a $d_z \times M$-dimensional Brownian motion.

However, the laws $(q_t^{1:M})_{t\geq 0}$ are unknown and the continuous time axis intractable, so we must approximate the former and discretize the latter. In particular, for each m we approximate the laws using an empirical distribution:

$$q_t^m(\mathrm{d}z_0) \approx \frac{1}{N} \sum_{n=1}^{N} \delta_{Z_{0,t}^{m,n}}(\mathrm{d}z_0),$$

which is formed with N weakly-dependent particles $Z_{0,t}^{m,1}, \ldots, Z_{0,t}^{m,N}$ all (approximately) distributed according to q_t^m; and we discretize the time axis using the Euler-Maruyama scheme:

$$\theta_{t+1} = \theta_t - (h/MN) \sum_{m=1}^{M} \sum_{n=1}^{N} \nabla_\theta \mathcal{L}_D(\theta_t, Z_{0,t}^{m,n}), \tag{19}$$

$$\phi_{t+1} = \phi_t + (h/MN) \sum_{m=1}^{M} \sum_{n=1}^{N} \nabla_\phi \log p_{\phi_t}(x^m | Z_{0,t}^{m,n}), \tag{20}$$

$$Z_{0,t+1}^{m,n} = Z_{0,t}^{m,n} + h \nabla_{z_0} [\mathcal{L}^m(\theta_t, \phi_t, Z_{0,t}^{m,n})] + \sqrt{2h} W_t^{m,n}, \quad \forall (m, n) \in [M] \times [N], \tag{21}$$

where $h > 0$ denotes the discretization step size, $W_{1:T}^{1:M,1:N}$ a $T \times M \times N$-dimensional standard Gaussian random variable, and T denotes the total number of steps. Under the conditions in Theorem 3.1's premise, and a further Lipschitz gradients assumption, it is straightforward to obtain error bounds for (19–21) using the results in Caprio et al. (2025).

A 4 (Lipschitz gradient). *The log-likelihood $\tilde{\ell}^m(\theta, \phi, z_0) := \log \tilde{p}_{\theta,\phi}(x^m, z_0)$ is differentiable and its gradient is L-Lipschitz for some $L > 0$,* i.e. *for all $(\theta, \phi, z_0), (\theta', \phi', z_0') \in \Theta \times \Phi \times \mathsf{Z}$,*

$$\|\nabla \tilde{\ell}^m(\theta, \phi, z_0) - \nabla \tilde{\ell}^m(\theta', \phi', z_0')\| \leq L \|(\theta, \phi, z_0) - (\theta', \phi', z_0')\|.$$

Theorem 3.2. *Suppose that the premise of Theorem 3.1 and **A** 4 hold, and that $\mathcal{R}$ has Lipschitz gradients. For all sufficiently small $h > 0$, there exists a constant $C_{h,N}$ of order $\mathcal{O}(h^{1/2} + N^{-1/2})$ independent of*

T, a $\rho \in (0,1)$, and a $C > 0$ independent of (h, N, T), such that

$$\mathbb{E}\left[||(\theta_T, \phi_T) - (\theta_\star, \phi_\star)||^2\right]^{1/2} \leq C_{h,N} + C\rho^T \quad \forall T \in \mathbb{N},$$

where $(\theta_\star, \phi_\star)$ denote $\tilde{\ell}$'s unique maximizer.

Proof (Sketch). Given the extra Lipschitz gradients assumption on $\log p_{\theta,\phi}(x, z_0)$, the result follows from bounding the spatial and temporal discretization errors separately, which are then combined with the exponential convergence result in Theorem 3.1. We defer the full proof to Appendix C.2. □

In particular, Theorem 3.2 shows that the error can be made arbitrarily small by picking N, T sufficiently large and h sufficiently small.

3.3 A practical algorithm: IPLD

While amenable to theoretical analysis, the algorithm defined by (19–21) performs poorly in practice when applied to models of the sort we are interested in for several reasons. We deal with these one at a time and obtain a practical discretization, Interacting Particle Latent Diffusion (IPLD), which is well-suited to modern computing environments.

Subsampling. The updates in (19–21) incur a $\mathcal{O}(NMK)$ computational [2], which proves prohibitively expensive for all but the smallest of training sets. We overcome this issue by subsampling similarly as in Welling and Teh (2011); Ho et al. (2020). First, we rewrite (19–21) more concisely as

$$\begin{aligned}(\theta_{t+1}, \phi_{t+1}) &= (\theta_t, \phi_t) - h\nabla_{(\theta,\phi)}\mathcal{L}_t,\\ Z_{0,t+1}^{1:M,1:N} &= Z_{0,t}^{1:M,1:N} - (MNh)\nabla_{z_0^{1:M,1:N}}\mathcal{L}_t\\ &\quad + \sqrt{2h}W_t^{1:M,1:N},\end{aligned}$$

where $\mathcal{L}_t := (MN)^{-1}\sum_{m=1}^{M}\sum_{n=1}^{N}\left[\mathcal{L}^m(\theta, \phi, z_{0,t}^{m,n})\right]$ with $\mathcal{L}^m$ defined in (12). Then, we replace the loss $\mathcal{L}_t$ with the following unbiased estimate:

$$\begin{aligned}\hat{\mathcal{L}}_t &:= \hat{\mathcal{L}}(\theta_t, \phi_t, z_{0,t}^{1:M,1:N}) \qquad (22)\\ &:= \frac{1}{N|\mathcal{B}|}\sum_{(m,n)\in\mathcal{B}\times[N]}\left[\hat{\mathcal{L}}_D(\theta_t, z_{0,t}^{m,n}) - \log p_\phi(x^m|z_{0,t}^{m,n})\right]\end{aligned}$$

where $\mathcal{B}$ denotes a subset of $[M]$ of size $|\mathcal{B}|$ drawn uniformly at random (and independently of all other random variables), and $\hat{\mathcal{L}}_\mathcal{D}$ denotes the $\mathcal{O}(1)$-cost unbiased estimate of the diffusion loss $\mathcal{L}_\mathcal{D}$ specified in Appendix A.1; so bringing down the cost to $\mathcal{O}(N|\mathcal{B}|)$. We add noise to the particles' updates scaled by $\sqrt{|\mathcal{B}|/M}$ to match the variance of the noise in the full-batch updates in (21). See Algorithm 1 below for pseudocode.

[2] We use the convention in optimization (Nesterov, 2004) and only include the dominant gradient evaluation cost.

Algorithm 1 Interacting Particle Latent Diffusion (IPLD)

Inputs: Dataset $\{x^m\}_{m\in[M]}$, stepsize h,
particles $z_{0,0}^{1:M,1:N}$, parameters ϕ, θ
while not converged **do**
  Sample a mini-batch of indices $\mathcal{B} \subset [M]$
  Compute $\hat{\mathcal{L}}_t = \hat{\mathcal{L}}(\theta_t, \phi_t, z_{0,t}^{1:M,1:N})$ in (22)
  for $(m, n) \in [M] \times [N]$ **do**
    if $m \in \mathcal{B}$ **then**
      $z_{0,t}^{m,n} \leftarrow z_{0,t}^{m,n} - (MNh)\nabla_{z_0^{m,n}}\hat{\mathcal{L}}_t$
    end if
    $z_{0,t+1}^{m,n} \leftarrow z_{0,t}^{m,n} + \sqrt{\frac{2h|\mathcal{B}|}{M}}W_t^{m,n}$
  end for
  $\theta_{t+1} \leftarrow \theta_t - h\nabla_\theta\hat{\mathcal{L}}_t$
  $\phi_{t+1} \leftarrow \phi_t - h\nabla_\phi\hat{\mathcal{L}}_t$
end while
Outputs: $(\theta_t, \phi_t, z_{0,t}^{1:M,1:N})$

Distributed Training. (19–21) require $\mathcal{O}(MN)$ memory. However, IPLD is well-suited for distributed training: $Z_{0,t}^{m,n}$'s update for a given pair (m, n) is independent of that for all other pairs, and requires only access to the mth datapoint. Thus, in distributed setups, we allocate each accelerator a disjoint subset of the training set and it handles the corresponding updates for all N particles, reducing per-device memory costs and communication costs. This contrasts with autoencoder-based LDMs that necessitate synchronizing gradients for encoders (typically, deep networks) when trained end-to-end.

Reweighting and annealing. We re-weight the diffusion loss $\mathcal{L}_\mathcal{D}$ similarly as in Ho et al. (2020), as this re-weighting is known to improve sample quality (Song et al., 2021b; Kingma and Gao, 2023); see Appendix A.1 for details. Furthermore, we anneal the KL term in the free energy, replacing the summand in (6) with: $-\mathbb{E}_{q^m}\left[\log p_\phi(x^m|z_0)\right] + \gamma_t D_{\mathrm{KL}}(q^m(z_0)||p_\theta(z_0))$, where $\gamma_t : [0, \infty) \to (0, \infty)$ denotes the (non-decreasing) annealing schedule. This is a practice commonly used when training deep Latent Variable Models (LVMs) to encourage accurate reconstruction during the early stages of training (Sønderby et al., 2016; Fu et al., 2019; Vahdat and Kautz, 2020; Vahdat et al., 2021). Adjusting $\tilde{F}$ correspondingly results in γ_t premultiplying $\mathcal{L}_\mathcal{D}$ in (11); which in turn corresponds to adjusting the noise levels in (18,21) by multiplying a factor of $\sqrt{\gamma_t}$

(cf. Appendix A.4.1 for details).

Preconditioning and momentum. When training simpler latent variable models with algorithms similar to ours, Kuntz et al. (2023); Lim et al. (2024) observed that preconditioning the models' parameters similarly as in RMSProp (Tieleman and Hinton, 2012) mitigated ill-conditioning and stabilized the training. Lim et al. (2024) further noted gains in both training speeds and test-time performance by incorporating momentum into the parameter and particle updates. Here we precondition and incorporate momentum for both the parameters and particles. We use optimizers like AdamW (Loshchilov and Hutter, 2019) for the former and adaptive Langevin algorithms (Li et al., 2016; Kim et al., 2020) for the latter. Lastly, we choose different step sizes h_θ, h_ϕ, and h_z respectively for θ, ϕ, and the particles to account for the multi-scale nature of the interacting particle system (Akyildiz et al., 2024; Pavliotis and Stuart, 2008). We provide the pseudocode for the full training algorithm in Appendix A.4.

4 RELATED WORK

Interacting Particle Algorithms. Interacting particle systems have long been the foundation of much statistical and optimization methodology, e.g. (Del Moral, 2004; Kennedy and Eberhart, 1995). Their use in algorithms that fit latent variable models by jointly updating model's parameters and a latent-space particle cloud to maximize the likelihood is more recent: Kuntz et al. (2023) proposed several such algorithms and Lim et al. (2024) improved their performance by incorporating momentum into the algorithms' updates (see also Encinar et al. (2024); Sharrock et al. (2024); Oliva and Akyildiz (2024); Marion et al. (2025); Akyildiz et al. (2025); Marks et al. (2025)). The theoretical guarantees of our algorithm are derived from the results in Caprio et al. (2025). Similar error bounds have been established in Akyildiz et al. (2025) for an alternative approximation to the gradient flow featuring noise in the parameter updates. Also related to our method are numerous works approximating gradient flows with particles to obtain alternatives to conventional Variational Inference (VI); e.g., Liu and Wang (2016); Lambert et al. (2022); Duncan et al. (2023); Diao et al. (2023); Lim and Johansen (2024). Lastly, there is a growing body of work exploring training methodologies for generative models in the ambient space using particle-based methods (Arbel et al., 2019; Yi et al., 2023; Franceschi et al., 2023; Galashov et al., 2024; Zhou et al., 2025).

Latent Diffusion Models. There have been various attempts at incorporating DMs into LVMs as the prior $p_\theta(z)$. Vahdat et al. (2021) proposed to jointly train a continuous-time score-based DM (Song et al., 2021c) in the latent space of a deep hierarchical VAE (Vahdat and Kautz, 2020). Similarly, Wehenkel and Louppe (2021) considered the joint training of a discrete-time DM with a conventional VAE. Cohen et al. (2022) instead applied a discrete-time diffusion prior to a Vector-Quantized Variational Autoencoder (VQ-VAE) (van den Oord et al., 2017). The seminal work by Rombach et al. (2022) can also be viewed as learning a diffusion prior post training of the VAE. Additionally, several studies have proposed a trainable forward process for diffusion models (Kim et al., 2022; Bartosh et al., 2024a,b; Nielsen et al., 2024), which can be viewed as generalizations of latent diffusion models. More recently, combinations of diffusion-based priors with other types of probabilistic models in hierarchical VAEs have been explored, such as Energy-based Model (EBM) (Cui and Han, 2024) and Variational Mixture of Posteriors prior (Kuzina and Tomczak, 2024). Additionally, the work by Silvestri et al. (2025) can also be viewed as an VAE with a consistency model (Song et al., 2023) learning the aggregated posterior. Concurrent to our work, Leng et al. (2025) also considers the joint training of a latent diffusion model with the VAE, but their objective is instead based on the REPresentation Alignment (REPA) loss (Yu et al., 2025), which require forward passes through pretrained vision foundational models like DINO (Caron et al., 2021; Oquab et al., 2024).

Decoder-only Models. Connected to our work are other methods that, similarly to us, optimize latent variables rather than relying on an encoder network. Several studies (Han et al., 2017; Nijkamp et al., 2019, 2020; Pang et al., 2020; Nijkamp et al., 2022; Yu et al., 2023) have considered short-run and persistent Langevin dynamics within an Expectation-Maximization (EM) framework to train latent variable models featuring a top-down generator network. Also related to our work are approaches like Bojanowski et al. (2018); Luise et al. (2020) that optimize latent distributions as an alternative or enhancement to Generative Adversarial Networks (GANs) (Goodfellow et al., 2014).

5 EXPERIMENTS

We evaluate IPLD's performance on two synthetic datasets (Section 5.1) and three image datasets (Section 5.2) [3]. In the synthetic case, we benchmark against the closest VI analogue to IPLD we have been able to locate in the literature: DiffusionVAE (Wehenkel and Louppe, 2021); see the end of Section 2 for more details. For the image datasets, we additionally compare with several other decoder-only LVMs (cf. Section 4).

[3] Code available at IPLD-release

Models	FID(↓)
Decoder-only LVMs	
PGD (Kuntz et al., 2023)	101.4
MPGD (Lim et al., 2024)	91.7
DAMC (Yu et al., 2023)	57.72
LP-EBM (Pang et al., 2020)	70.15
EBM-SR (Nijkamp et al., 2019)	44.50
VAE	
DiffusionVAE (Wehenkel and Louppe, 2021)	153.1
DiffusionVAE*	62.07
Ours	
IPLD (1 particle)	51.60
IPLD (5 particles)	48.30
IPLD (10 particles)	46.95

(a) **FID(↓) of models trained on CIFAR-10**. We report both the original results from Wehenkel and Louppe (2021) and our re-implementation (denoted by "*") of the VAE with a Diffusion Prior (DiffusionVAE) using the same architecture (see Appendix B).

Models	FID(↓)
DAMC (Yu et al., 2023)	30.83
LP-EBM (Pang et al., 2020)	37.87
EBM-SR (Nijkamp et al., 2019)	23.03
DiffusionVAE (Wehenkel and Louppe, 2021)	67.95
DiffusionVAE*	29.24
Ours	
IPLD (1 particle)	22.86
IPLD (5 particles)	21.55
IPLD (10 particles)	21.43

(b) **FID(↓) of models trained on CelebA64**.

Models	FID(↓)
DAMC (Yu et al., 2023)	18.76
LP-EBM (Pang et al., 2020)	29.44
DiffusionVAE* (Wehenkel and Louppe, 2021)	20.89
Ours	
IPLD (1 particle)	17.55
IPLD (5 particles)	14.02
IPLD (10 particles)	13.51

(c) **FID(↓) of models trained on SVHN**.

Table 1: FID scores for CIFAR-10, SVHN, and CelebA64 estimated using 50,000 samples.

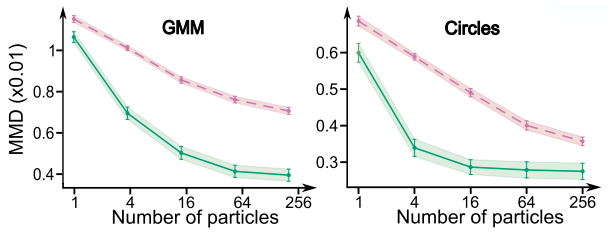


Figure 1: Estimated MMD between the ground truth and the distribution learned with IPLD (solid line) and DiffusionVAE (dashed line) for the GMM (left) and concentric circles (right) datasets. Shaded regions indicate ±1 standard error.

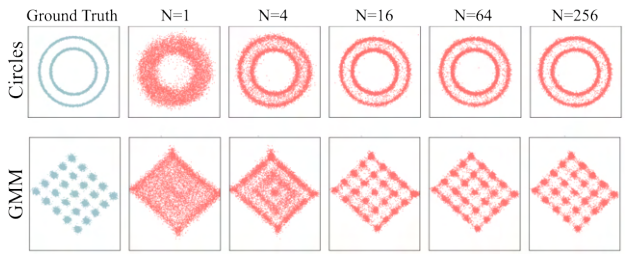


Figure 2: Samples generated by IPLD trained with varying numbers of particles for 1,000 steps (showing the first two out of $d_x = 64$ dimensions).

5.1 Synthetic Datasets

We first validate the effectiveness of our method on two toy datasets. We generate these by first sampling from a distribution on a 2-dimensional latent space, and then mapping the samples into a 64-dimensional ambient space using a matrix $A \in \mathbb{R}^{2\times 64}$ with orthogonal rows. We consider 1) a Gaussian Mixture Model (GMM) with 25 components as in Boys et al. (2024); Cardoso et al. (2024), and 2) a distribution concentrated on concentric circles similar to that[4] in `scikit-learn`. We train both DiffusionVAE and IPLD for 1,000 steps and varying numbers of particles $N = 1, 4, 16, 64, 256$ (in the case of DiffusionVAE, N refers to the number of samples drawn from the encoder at each step). For both datasets, increasing N increases the quality of the samples generated by IPLD (Figure 2). To quantitatively compare IPLD and DiffusionVAE, we estimate the Maximum Mean Discrepancy (MMD) (Gretton et al., 2012) between the distribution learned by each and the ground truth. The MMD values are averaged over 50 training runs and the standard error is reported. For all particle numbers, IPLD outperformed DiffusionVAE (Figure 1). See Appendix B.1 for more details.

5.2 Image Modeling

Next, we test our model on three image datasets: CIFAR-10 (Krizhevsky and Hinton, 2009), SVHN (Net-

[4] https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_circles.html

Figure 3: Samples generated with IPLD trained on CIFAR-10, CelebA64, and SVHN. The CelebA64 samples have been curated for better visualization. See Appendix D.6 for additional samples.

zer et al., 2011), and CelebA64 (Liu et al., 2015). As before, we benchmark against DiffusionVAE using the same decoder, diffusion model, and $4\times8\times8$-dimensional latent space as for IPLD. We train both models for the same number of epochs (400 for CIFAR10 and SVHN, 200 for CelebA64); see Appendix B.2 for details. For IPLD, we vary the particle number $N = 1, 5, 10$ (we use a single particle for DiffusionVAE because back-propagating the corresponding gradients through the model's encoder with more particles proved too memory-consuming for our hardware). We warm-start the multi-particle IPLD runs using a single particle as in Kuntz et al. (2023); see Appendix B.3 for details. In both cases, we compute Fréchet Inception Distance (FID) scores for the trained model (Table 1). IPLD outperforms DiffusionVAE, and its performance increases with the number of particles used for training. IPLD also proves competitive with previous decoder-only models discussed in Section 4 (see also Table 1).

6 DISCUSSION

Like many before us, we recast fitting an LDM via maximum likelihood as minimizing a free energy functional (Section 2). Unlike others, we then identify a gradient flow that minimizes the functional and approximate it using systems of interacting particles (Sections 3.1, 3.2), and we theoretically characterize both the flow and the approximations (Theorems 3.1, 3.2) under standard assumptions. Following these steps, we obtain IPLD (Section 3.3): a theoretically-principled algorithm for end-to-end LDM training. Because it entails updating a cloud of particles and each particle's update is independent of the others', IPLD is well-suited for modern distributed compute environments and is easy to scale.

In numerical experiments involving both synthetic and image data, IPLD compares favorably with relevant benchmarks (Section 5). However, our results on the image datasets fall short of today's state-of-the-art. We believe this may be due to the relatively small latent spaces, decoder, and DM architectures we use (e.g., compare Appendix B.2 with Vahdat et al. (2021, Appendices G.1,2)) and our limited computational budget (e.g., compare Appendix B.4 with Rombach et al. (2022, Appendix E)), rather than a fundamental limitation of the approach. Our work may be limited by the fact that, to date, LDMs trained in a two-stage manner (e.g., Rombach et al. (2022)) have achieved results that those trained end-to-end have not, and our approach is fundamentally an end-to-end one. However, recent works such as Leng et al. (2025) have demonstrated promising breakthroughs in end-to-end LDM training, and their innovations are relatively straightforward to incorporate in our algorithm. Further improvements may be possible through more careful subsampling schemes than that in Section 3.3 and the use of variance reduction techniques (Zou et al., 2018). Indeed, we hope our work paves the way to other more effective particle-based algorithms for LDM training.

Lastly, our theoretical analysis is limited to the simplified algorithm in Section 3.2. However, we believe that it may be possible to extend the analysis to more practical versions using techniques along the lines of those used to study adaptive optimizers (Malladi et al., 2022) and momentum-enriched interacting particle systems (Oliva and Akyildiz, 2024; Lim et al., 2024).

Acknowledgements

We thank the anonymous reviewers for their constructive comments. TW is supported by the Roth Scholarship from the Department of Mathematics, Imperial College London. We acknowledge computational resources and support provided by the Department of Mathematics and the Imperial College Research Computing Service, DOI: 10.14469/hpc/2232.

References

Akyildiz, Ö. D., Crucinio, F. R., Girolami, M., Johnston, T., and Sabanis, S. (2025). Interacting particle Langevin algorithm for maximum marginal likelihood estimation. *ESAIM: Probability and Statistics*.

Akyildiz, O. D., Ottobre, M., and Souttar, I. (2024). A multiscale perspective on maximum marginal likelihood estimation. *arXiv preprint arXiv:2406.04187*.

Arbel, M., Korba, A., SALIM, A., and Gretton, A. (2019). Maximum mean discrepancy gradient flow. In *Advances in Neural Information Processing Systems*, volume 32.

Bartosh, G., Vetrov, D., and Naesseth, C. A. (2024a). Neural diffusion models. In *Forty-first International Conference on Machine Learning*.

Bartosh, G., Vetrov, D., and Naesseth, C. A. (2024b). Neural flow diffusion models: Learnable forward pro-

cess for improved diffusion modelling. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Bishop, C. M. (2006). *Pattern recognition and machine learning*, volume 4. Springer.

Bojanowski, P., Joulin, A., Lopez-Pas, D., and Szlam, A. (2018). Optimizing the latent space of generative networks. In *International Conference on Machine Learning*. PMLR.

Boys, B., Girolami, M., Pidstrigach, J., Reich, S., Mosca, A., and Akyildiz, O. D. (2024). Tweedie moment projected diffusions for inverse problems. *Transactions on Machine Learning Research*. Featured Certification.

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., and Varoquaux, G. (2013). API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*.

Caprio, R., Kuntz, J., Power, S., and Johansen, A. M. (2025). Error bounds for particle gradient descent, and extensions of the log-sobolev and talagrand inequalities. *Journal of Machine Learning Research*, 26(103):1–38.

Cardoso, G., el idrissi, Y. J., Corff, S. L., and Moulines, E. (2024). Monte carlo guided denoising diffusion models for bayesian linear inverse problems. In *The Twelfth International Conference on Learning Representations*.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. (2021). Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

Chaintron, L.-P. and Diez, A. (2022). Propagation of chaos: A review of models, methods and applications. i. models and methods. *Kinetic and Related Models*, 15(6):895.

Cohen, M., Quispe, G., Corff, S. L., Ollion, C., and Moulines, E. (2022). Diffusion bridges vector quantized variational autoencoders. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*.

Cui, J. and Han, T. (2024). Learning latent space hierarchical EBM diffusion models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*.

Del Moral, P. (2004). *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*. Springer.

Dhariwal, P. and Nichol, A. Q. (2021). Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*.

Diao, M. Z., Balasubramanian, K., Chewi, S., and Salim, A. (2023). Forward-backward gaussian variational inference via jko in the bures-wasserstein space. In *International Conference on Machine Learning*, pages 7960–7991. PMLR.

Dieleman, S. (2025). Generative modelling in latent space.

Duncan, A., Nuesken, N., and Szpruch, L. (2023). On the geometry of stein variational gradient descent. *Journal of Machine Learning Research*, 24(56):1–39.

Efron, B. (2011). Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614.

Encinar, P. C., Crucinio, F. R., and Akyildiz, O. D. (2024). Proximal interacting particle langevin algorithms. *arXiv preprint arXiv:2406.14292*.

Figalli, A. and Glaudo, F. (2021). *An invitation to optimal transport, Wasserstein distances, and gradient flows*.

Franceschi, J.-Y., Gartrell, M., Santos, L. D., Issenhuth, T., de Bezenac, E., Chen, M., and Rakotomamonjy, A. (2023). Unifying GANs and score-based diffusion as generative particle models. In *Thirty-Seventh Conference on Neural Information Processing Systems*.

Fu, H., Li, C., Liu, X., Gao, J., Celikyilmaz, A., and Carin, L. (2019). Cyclical annealing schedule: A simple approach to mitigating kl vanishing. *North American Chapter of the Association for Computational Linguistics*.

Galashov, A., de Bortoli, V., and Gretton, A. (2024). Deep MMD gradient flow without adversarial training. *arXiv preprint arXiv:2405.06780*.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 27.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. (2012). A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773.

Han, T., Lu, Y., Zhu, S.-C., and Wu, Y. N. (2017). Alternating back-propagation for generator network. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17.

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33.

Huang, Y.-J. and Zhang, Y. (2023). GANs as gradient flows that converge. *Journal of Machine Learning Research*, 24(217):1–40.

Ivchenko, D., Van Der Staay, D., Taylor, C., Liu, X., Feng, W., Kindi, R., Sudarshan, A., and Sefati, S. (2022). Torchrec: a pytorch domain library for recommendation systems. In *Proceedings of the 16th ACM Conference on Recommender Systems*, RecSys '22, page 482–483, New York, NY, USA. Association for Computing Machinery.

Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4.

Kim, D., Na, B., Kwon, S. J., Lee, D., Kang, W., and chul Moon, I. (2022). Maximum likelihood training of implicit nonlinear diffusion model. In *Advances in Neural Information Processing Systems*.

Kim, S., Song, Q., and Liang, F. (2020). Stochastic gradient langevin dynamics algorithms with adaptive drifts. *arXiv preprint arXiv:2009.09535*.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

Kingma, D. P. and Gao, R. (2023). Understanding diffusion objectives as the ELBO with simple data augmentation. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *CoRR*, abs/1312.6114.

Krizhevsky, A. and Hinton, G. (2009). Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario.

Kuntz, J., Lim, J. N., and Johansen, A. M. (2023). Particle algorithms for maximum likelihood training of latent variable models. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*.

Kuzina, A. and Tomczak, J. M. (2024). Hierarchical VAE with a diffusion-based vampprior. *Transactions on Machine Learning Research*.

Lambert, M., Chewi, S., Bach, F., Bonnabel, S., and Rigollet, P. (2022). Variational inference via wasserstein gradient flows. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K., editors, *Advances in Neural Information Processing Systems*.

Lee, J. M. (2018). *Introduction to Riemannian Manifolds*, volume 176 of *Graduate Texts in Mathematics*. Springer.

Leng, X., Singh, J., Hou, Y., Xing, Z., Xie, S., and Zheng, L. (2025). Repa-e: Unlocking vae for end-to-end tuning of latent diffusion transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Li, C., Chen, C., Carlson, D., and Carin, L. (2016). Preconditioned stochastic gradient langevin dynamics for deep neural networks. In *AAAI*.

Lim, J. N. and Johansen, A. M. (2024). Particle semi-implicit variational inference. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Lim, J. N., Kuntz, J., Power, S., and Johansen, A. M. (2024). Momentum particle maximum likelihood. In *Forty-First International Conference on Machine Learning*.

Liu, Q. and Wang, D. (2016). Stein variational gradient descent: A general purpose bayesian inference algorithm. In *Advances in Neural Information Processing Systems*, volume 29.

Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.

Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Luise, G., Pontil, M., and Ciliberto, C. (2020). Generalization properties of optimal transport GANs with latent distribution learning. *arXiv preprint arXiv:2007.14641*.

Malladi, S., Lyu, K., Panigrahi, A., and Arora, S. (2022). On the SDEs and scaling rules for adaptive gradient algorithms. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K., editors, *Advances in Neural Information Processing Systems*.

Marion, P., Korba, A., Bartlett, P., Blondel, M., Bortoli, V. D., Doucet, A., Llinares-López, F., Paquette, C., and Berthet, Q. (2025). Implicit diffusion: Efficient optimization through stochastic sampling. *arXiv preprint arXiv:2402.05468*.

Marks, J., Wang, T. Y. J., and Akyildiz, O. D. (2025). Learning latent energy-based models via interacting particle langevin dynamics.

Neal, R. M. and Hinton, G. E. (1998). *A View of the Em Algorithm that Justifies Incremental, Sparse, and other Variants*, pages 355–368. Springer Netherlands, Dordrecht.

Nesterov, Y. (2004). *Nonlinear Optimization*, pages 1–50. Springer US, Boston, MA.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. (2011). Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*.

Nie, S., Zhu, F., You, Z., Zhang, X., Ou, J., Hu, J., Zhou, J., Lin, Y., Wen, J.-R., and Li, C. (2025). Large language diffusion models. *arXiv preprint arXiv:2502.09992*.

Nielsen, B. M. G., Christensen, A., Dittadi, A., and Winther, O. (2024). Diffenc: Variational diffusion with a learned encoder. In *The Twelfth International Conference on Learning Representations*.

Nijkamp, E., Gao, R., Sountsov, P., Vasudevan, S., Pang, B., Zhu, S.-C., and Wu, Y. N. (2022). MCMC should mix: Learning energy-based model with neural transport latent space MCMC. In *International Conference on Learning Representations*.

Nijkamp, E., Hill, M., Zhu, S.-C., and Wu, Y. N. (2019). Learning non-convergent non-persistent short-run MCMC toward energy-based model. In *Advances in Neural Information Processing Systems*, volume 32.

Nijkamp, E., Pang, B., Han, T., Zhu, S.-C., and Wu, Y. N. (2020). Learning multi-layer latent variable model via variational optimization of short run mcmc for approximate inference. *ECCV*.

Oliva, P. F. V. and Akyildiz, O. D. (2024). Kinetic interacting particle langevin monte carlo. *arXiv preprint arXiv:2407.05790*.

Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.-Y., Li, S.-W., Misra, I., Rabbat, M., Sharma, V., Synnaeve, G., Xu, H., Jegou, H., Mairal, J., Labatut, P., Joulin, A., and Bojanowski, P. (2024). Dinov2: Learning robust visual features without supervision.

Otto, F. (2001). The geometry of dissipative evolution equations: The porous medium equation. *Communications in Partial Differential Equations*, 26(1-2):101–174.

Pang, B., Han, T., Nijkamp, E., Zhu, S.-C., and Wu, Y. N. (2020). Learning latent space energy-based prior model. In *Advances in Neural Information Processing Systems*, volume 33.

Pavliotis, G. A. and Stuart, A. M. (2008). *Averaging for ODEs and SDEs*, pages 145–156. Springer New York, New York, NY.

Peebles, W. and Xie, S. (2023). Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., and Rombach, R. (2024). SDXL: Improving latent diffusion models for high-resolution image synthesis. In *The Twelfth International Conference on Learning Representations*.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Santambrogio, F. (2015). *Optimal transport for applied mathematicians*. Springer.

Saxe, A. M., McClelland, J. L., and Ganguli, S. (2014). Exact solutions to the nonlinear dynamics of learning in deep linear neural networks.

Sharrock, L., Dodd, D., and Nemeth, C. (2024). Tuning-free maximum likelihood training of latent variable models via coin betting. In *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*.

Silvestri, G., Ambrogioni, L., Lai, C.-H., Takida, Y., and Mitsufuji, Y. (2025). Training consistency models with variational noise coupling. *arXiv preprint arXiv:2502.18197*.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*.

Sønderby, C. K., Raiko, T., Maaløe, L., Sønderby, S. K., and Winther, O. (2016). Ladder variational autoencoders. In *Advances in Neural Information Processing Systems*, volume 29.

Song, J., Meng, C., and Ermon, S. (2021a). Denoising diffusion implicit models. In *International Conference on Learning Representations*.

Song, Y., Dhariwal, P., Chen, M., and Sutskever, I. (2023). Consistency models. *International Conference on Machine Learning*.

Song, Y., Durkan, C., Murray, I., and Ermon, S. (2021b). Maximum likelihood training of score-based diffusion models. *arXiv preprint arXiv:2101.09258*, pages 1415–1428.

Song, Y. and Ermon, S. (2020). Improved techniques for training score-based generative models. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2021c). Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*.

Tieleman, T. and Hinton, G. (2012). Lecture 6.5-rmsprop, coursera: Neural networks for machine learning. *University of Toronto, Technical Report*, 6.

Vahdat, A. and Kautz, J. (2020). Nvae: A deep hierarchical variational autoencoder. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33.

Vahdat, A., Kreis, K., and Kautz, J. (2021). Score-based generative modeling in latent space. In *Advances in Neural Information Processing Systems*, volume 34.

van den Oord, A., Vinyals, O., and Kavukcuoglu, K. (2017). Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, volume 30.

Villani, C. et al. (2008). *Optimal transport: old and new*, volume 338. Springer.

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., Wicky, B. I. M., Hanikel, N., Pellock, S. J., Courbet, A., Sheffler, W., Wang, J., Venkatesh, P., Sappington, I., Torres, S. V., Lauko, A., De Bortoli, V., Mathieu, E., Ovchinnikov, S., Barzilay, R., Jaakkola, T. S., DiMaio, F., Baek, M., and Baker, D. (2023). De novo design of protein structure and function with RFdiffusion. *Nature*, 620(7976):1089–1100.

Wehenkel, A. and Louppe, G. (2021). Diffusion priors in variational autoencoders. In *ICML Workshop on Invertible Neural Networks, Normalizing Flows, and Explicit Likelihood Models*.

Welling, M. and Teh, Y. W. (2011). Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11.

Yi, M., Zhu, Z., and Liu, S. (2023). Monoflow: Rethinking divergence gans via the perspective of wasserstein gradient flows. *International Conference on Machine Learning*.

Yu, P., Zhu, Y., Xie, S., Ma, X., Gao, R., Zhu, S.-C., and Wu, Y. N. (2023). Learning energy-based prior model with diffusion-amortized MCMC. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Yu, S., Kwak, S., Jang, H., Jeong, J., Huang, J., Shin, J., and Xie, S. (2025). Representation alignment for generation: Training diffusion transformers is easier than you think. In *The Thirteenth International Conference on Learning Representations*.

Zhou, L., Ermon, S., and Song, J. (2025). Inductive moment matching. *arXiv preprint arXiv:2503.07565*.

Zou, D., Xu, P., and Gu, Q. (2018). Subsampled stochastic variance-reduced gradient langevin dynamics. In *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018*.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   We have provided these in Appendix A.4 and Appendix C where the algorithm, setting, and assumptions are described in detail.
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   This is provided in the main text, Appendix C, and Appendix D.5.
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [No]
   Our code will be provided upon publication.

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   See the main text and Appendix A.4 for full set of assumptions and analysis.
   (b) Complete proofs of all theoretical results. [Yes]
   Complete proofs are included in Appendix C.
   (c) Clear explanations of any assumptions. [Yes]
   We have provided the meaning of the assumptions in the main text and Appendix C.

3. For all figures and tables that present empirical results, check if you include:

(a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [No]
Our code will be provided upon publication. Our data consists of publicly available benchmark datasets.

(b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
This is provided in Supplementary Material.

(c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]

(d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

(a) Citations of the creator If your work uses existing assets. [Yes]

(b) The license information of the assets, if applicable. [Not Applicable]

(c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]

(d) Information about consent from data providers/curators. [Not Applicable]

(e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

(a) The full text of instructions given to participants and screenshots. [Not Applicable]

(b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]

(c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Supplementary Material for Training Latent Diffusion Models with Interacting Particle Algorithms

A Derivations

A.1 Derivation of the Training Objective

We derive the training objective, the tilted free energy $\tilde{F}(\theta, \phi, q^{1:M})$ in (6) and include a derivation of the standard reparametrized diffusion objective $\mathcal{L}_D(\theta, z_0)$ analogous to Ho et al. (2020); Sohl-Dickstein et al. (2015). For convenience, we re-state the objective for a single data point here and omit the superscript m:

$$\tilde{F}(\theta, \phi, q) := \mathbb{E}_{q(z_{0:K})}\left[\log \frac{q(z_{0:K})}{p_{\theta,\phi}(x, z_{0:K})}\right]. \tag{23}$$

Negative Lower Bound. To see that $-\tilde{F}(\theta, \phi, q)$ is a lower bound of the log-likelihood $\log p_{\theta,\phi}(x)$, we first note that the tilted free energy in (23) can be re-written as:

$$\begin{aligned}
\tilde{F}(\theta, \phi, q) &= \mathbb{E}_{q(z_0)q(z_{1:K}|z_0)}\left[\log \frac{q(z_0)q(z_{1:K}|z_0)}{p_{\theta,\phi}(x|z_0)p_\theta(z_{1:K}|z_0)p_\theta(z_0)}\right] \\
&= \mathbb{E}_{q(z_0)}\left[\mathbb{E}_{q(z_{1:K}|z_0)}\left[\log \frac{q(z_0)}{p_{\theta,\phi}(x, z_0)} + \log \frac{q(z_{1:K}|z_0)}{p_\theta(z_{1:K}|z_0)}\right]\right] \\
&= \mathbb{E}_{q(z_0)}\left[\log \frac{q(z_0)}{p_{\theta,\phi}(x, z_0)}\right] + \mathbb{E}_{q(z_0)}[D_{\mathrm{KL}}(q(z_{1:K}|z_0)\|p_\theta(z_{1:K}|z_0))],
\end{aligned}$$

where we have assumed the independence structure of the decoder $p_\phi(x|z_{0:K}) = p_\phi(x|z_0)$; the first term is the usual free energy $F(\theta, \phi, q) = \mathbb{E}_{q(z_0)}\left[\log q(z_0) - \log p_{\theta,\phi}(x, z_0)\right]$ in (5) and the second term is non-negative. It is thus easy to see that the tilted free energy $\tilde{F}(\theta, \phi, q)$ is obtained by replacing $p_{\theta,\phi}(x^m, z_0)$ in (5) with the tilted $\tilde{p}_{\theta,\phi}(x^m, z_0) := p_{\theta,\phi}(x^m, z_0)\exp(-\mathcal{R}(\theta, z_0))$, where $\mathcal{R}(\theta, z_0) = D_{\mathrm{KL}}\big(q(z_{1:K}|z_0)\|p_\theta(z_{1:K}|z_0)\big)$.

Using Jensen's inequality, we see the negative of the first term is an upper bound of $-\log p_{\theta,\phi}(x)$:

$$\mathbb{E}_{q(z_0)}\left[-\log \frac{p_{\theta,\phi}(x, z_0)}{q(z_0)}\right] \geq -\log \mathbb{E}_{q(z_0)}\left[\frac{p_{\theta,\phi}(x, z_0)}{q(z_0)}\right] = -\log p_{\theta,\phi}(x),$$

which leads to the sequence of inequalities:

$$\tilde{F}(\theta, \phi, q) \geq F(\theta, \phi, q) \geq -\log p_{\theta,\phi}(x). \tag{24}$$

An alternative form of the objective $\tilde{F}(\theta, \phi, q)$ amenable to computation can be obtained by decomposing it as follows:

$$\tilde{F}(\theta, \phi, q) = \int \underbrace{\int \log \frac{q(z_{1:K}|z_0)}{p_\theta(z_{0:K})} q(z_{1:K}|z_0)\mathrm{d}z_{1:K}}_{\text{Diffusion Objective } \mathcal{L}_D(\theta, z_0)} q(z_0)\mathrm{d}z_0 - \int \log \frac{p_\phi(x|z_0)}{q(z_0)} q(z_0)\, \mathrm{d}z_0. \tag{25}$$

A.1.1 Derivation of the Diffusion Objective

The diffusion objective $\mathcal{L}_D(\theta, z_0)$ can be re-written as:

$$\mathcal{L}_D(\theta, z_0) = \int \left(-\log p_K(z_K) + \log \frac{q_1(z_1|z_0)}{p_{\theta,1}(z_0|z_1)} + \sum_{k=2}^{K} \log \frac{q_k(z_k|z_{k-1})}{p_{\theta,k}(z_{k-1}|z_k)}\right) q(z_{1:K}|z_0)\mathrm{d}z_{1:K}. \tag{26}$$

We can re-write $q_k(z_k|z_{k-1})$ using Bayes' rule as:

$$q_k(z_k|z_{k-1}) = q_k(z_{k-1}|z_k, z_0)\frac{q(z_k|z_0)}{q(z_{k-1}|z_0)}.$$

Combined with the Markov assumption $q(z_{1:K}|z_0) = q(z_K|z_0)\prod_{k=2}^K q_k(z_{k-1}|z_k, z_0)$, we can rewrite (26) above as:

$$\mathcal{L}_D(\theta, z_0) = \mathbb{E}_{q(z_{1:K}|z_0)}\left[D_{\mathrm{KL}}(q(z_K|z_0)\|p(z_K)) - \log p_{\theta,1}(z_0|z_1)\right] \tag{27}$$

$$+ \mathbb{E}_{q(z_{1:K}|z_0)}\left[\sum_{k=2}^K D_{\mathrm{KL}}(q_k(z_{k-1}|z_k, z_0)\|p_{\theta,k}(z_{k-1}|z_k))\right] \tag{28}$$

Gaussian Transition Kernel. For the forward process, we take

$$q_k(z_k|z_{k-1}) = \mathcal{N}(z_k; \sqrt{1-\beta_k}z_{k-1}, \beta_k I), \tag{29}$$

where $\{\beta_k\}_{k=1}^K$ is a linear noise schedule with $\beta_k = (1-k/K)\beta_0 + (k/K)\beta_K$. Using the notation $\alpha_k := 1-\beta_k$ and $\bar{\alpha}_k := \prod_{j=1}^k \alpha_j$, we can derive the k-step transition kernel:

$$q(z_k|z_0) = \mathcal{N}(z_k; \sqrt{\bar{\alpha}_k}z_0, (1-\bar{\alpha}_k)I), \quad \forall k = 1, \dots, K. \tag{30}$$

By Bayes' rule, we can compute:

$$q_k(z_{k-1}|z_k, z_0) = \mathcal{N}(z_{k-1}; \tilde{\mu}_k(z_k, z_0), \beta_k I), \tag{31}$$

$$\tilde{\mu}_k(z_k, z_0) = \frac{\sqrt{\bar{\alpha}_{k-1}}\beta_k}{1-\bar{\alpha}_k}z_0 + \frac{\sqrt{\alpha_k}(1-\bar{\alpha}_{k-1})}{1-\bar{\alpha}_k}z_k. \tag{32}$$

For the backward process, we set:

$$p_K(z_K) = \mathcal{N}(0, I) \tag{33}$$

$$p_{\theta,k}(z_{k-1}|z_k) = \mathcal{N}(z_{k-1}; \mu_{\theta,k}(z_k), \beta_k I) \quad \forall k = 2, \dots, K \tag{34}$$

$$p_{\theta,1}(z_0|z_1) = \mathcal{N}(z_{k-1}; \mu_{\theta,k}(z_k), I) \tag{35}$$

Using the formula for the Kullback-Leibler divergence between isotropic Gaussian distributions, we obtain:

$$D_{\mathrm{KL}}(q_k(z_{k-1}|z_k, z_0)\|p_{\theta,k}(z_{k-1}|z_k)) = \frac{1}{2}\log\left(\frac{1-\bar{\alpha}_k}{\beta_k}\right) - \frac{d_z}{2} + \frac{d_z\beta_k + \|\tilde{\mu}_k(z_k, z_0) - \mu_{\theta,k}(z_k)\|^2}{2(1-\bar{\alpha}_k)} \tag{36}$$

$$D_{\mathrm{KL}}(q(z_K|z_0)\|p_\theta(z_K)) = -\log(1-\bar{\alpha}_K) + \frac{d_z(1-\bar{\alpha}_K)^2 - d_z + \bar{\alpha}_K\|z_0\|^2}{2} \tag{37}$$

$$-\log(p_\theta(z_0|z_1)) = \frac{d_z}{2}\log(2\pi(1-\bar{\alpha}_1)) + \frac{\|z_0 - \mu_{\theta,1}(z_1)\|^2}{2(1-\bar{\alpha}_1)}. \tag{38}$$

Note that in contrast to the standard pixel-space DDPM, we have the extra term in (37) depending on the latent z_0; we also use a Gaussian distribution with identity variance for $p_\theta(z_0|z_1)$ in (38) instead of the independent discrete decoder from Ho et al. (2020).

Reparametrization. We adopt the same ϵ-prediction reparameterization as in Ho et al. (2020) to write:

$$\mu_{\theta,k}(z_k) = \tilde{\mu}_k\left(z_k, \frac{1}{\sqrt{\alpha_k}}\left(z_k - \sqrt{1-\bar{\alpha}_k}\,\epsilon_{\theta,k}(z_k)\right)\right) = \frac{1}{\sqrt{\alpha_k}}\left(z_k - \frac{\beta_k}{\sqrt{1-\bar{\alpha}_k}}\epsilon_{\theta,k}(z_k)\right),$$

which gives:

$$D_{\mathrm{KL}}(q_k(z_{k-1}|z_k, z_0)\|p_{\theta,k}(z_{k-1}|z_k)) = \frac{1}{2}\log\left(\frac{1-\bar{\alpha}_k}{\beta_k}\right) - \frac{d_z}{2} + \frac{d_z\beta_k + \beta_k\|\epsilon - \epsilon_{\theta,k}(z_k)\|^2}{2\alpha_k(1-\bar{\alpha}_k)}, \tag{39}$$

where $\epsilon \sim \mathcal{N}(0, I)$ is a random vector in $\mathbb{R}^{d_z}$ sampled independently from a standard Gaussian. Additionally, using the forward k-step transition kernel, we can write $z_k = z_k(z_0, \epsilon) = \sqrt{\bar{\alpha}_k}z_0 + \sqrt{1-\bar{\alpha}_k}\epsilon$. Up to a constant

term independent of θ, we have the loss function $\mathcal{L}_D(\theta, z_0; \epsilon) \approx \hat{\mathcal{L}}(\theta, z_0; \epsilon, k)$, where we sample $k \sim \text{Unif}(1, \dots, K)$ and $\epsilon \sim \mathcal{N}(0, I)$. We thus have the following loss:

$$\hat{\mathcal{L}}_D(\theta, z_0; \epsilon, k) := \begin{cases} \frac{1}{2}\|z_0 - \mu_{\theta,1}(z_1(z_0, \epsilon))\|^2 + \frac{\bar{\alpha}_K \|z_0\|^2}{2} & \text{if } k = 1 \\ \frac{\beta_k}{2\alpha_k(1-\bar{\alpha}_k)}\|\epsilon - \epsilon_{\theta,k}(z_k(z_0, \epsilon))\|^2 + \frac{\bar{\alpha}_K \|z_0\|^2}{2} & \text{if } 2 \le k \le K \end{cases} \tag{40}$$

In practice, we follow the approach in Ho et al. (2020) to use a simplified version of the diffusion objective:

$$\hat{\mathcal{L}}_{\text{simple}}(\theta, z_0; \epsilon) \approx \hat{\mathcal{L}}_{\text{simple}}(\theta, z_0; \epsilon, k) := \begin{cases} \frac{1}{2}\|z_0 - \mu_{\theta,1}(z_1(z_0, \epsilon))\|^2 + \frac{\bar{\alpha}_K \|z_0\|^2}{2} & \text{if } k = 1 \\ \|\epsilon - \epsilon_{\theta,k}(z_k(z_0, \epsilon))\|^2 + \frac{\bar{\alpha}_K \|z_0\|^2}{2} & \text{if } 2 \le k \le K \end{cases} \tag{41}$$

More specifically, we draw a minibatch of indices $\mathcal{B} \subseteq [M]$, standard Gaussian random variables $\epsilon^{m,n} \sim \mathcal{N}(0, I)$, and $t^{m,n} \sim \text{Unif}(1, \dots, K)$ for $m \in \mathcal{B}, n \in [N]$ to approximate the loss by:

$$\hat{\mathcal{L}}_{\text{simple}}(\theta, z_0^{\mathcal{B},1:N}) = \frac{1}{|\mathcal{B}|N} \sum_{(m,n)\in\mathcal{B}\times[N]} \hat{\mathcal{L}}_{\text{simple}}(\theta, z_0^{m,n}; \epsilon^{m,n}, t^{m,n}). \tag{42}$$

To simplify the notation, we will denote the one-sample approximation $\hat{\mathcal{L}}_{\text{simple}}(\theta, z_0^{m,n}; \epsilon^{m,n}, t^{m,n})$ with $\hat{\mathcal{L}}_{\text{simple}}(\theta, z_0^{m,n})$ as in Section 3.3 and we use the same simplified notation in the rest of the appendix.

A.2 The Euclidean-Wasserstein Geometry on $\mathbb{R}^{d_\theta} \times \mathbb{R}^{d_\phi} \times \mathcal{P}_2(\mathbb{R}^{d_z})^M$

To obtain the gradient flow in Section 3.1, we view the product space of parameters and probability distributions $\mathbb{R}^{d_\theta} \times \mathbb{R}^{d_\phi} \times \mathcal{P}_2(\mathbb{R}^{d_z})^M$ as a Riemannian manifold. We will equip this product space with suitable tangent spaces and Riemannian metrics, which enable us to define the gradients and perform optimization. We omit the subscript on z_0 and denote it by z for simplicity throughout this section.

Tangent spaces. Using the same set-up as in Kuntz et al. (2023), we concatenate the parameters as $\vartheta := (\theta, \phi) \in \mathbb{R}^D$, where $D = d_\theta + d_\phi$, and assume the approximate posterior is a distribution with strictly positive density w.r.t. the Lebesgue measure and support $\mathsf{Z} = \mathbb{R}^{d_z}$. We let the product manifold be $\mathcal{M}^{1:M} := \mathbb{R}^D \times \mathcal{P}_2(\mathsf{Z})^M$. For each point $(\vartheta, q^{1:M}) \in \mathcal{M}^{1:M}$, we can define the tangent space $T\mathcal{M}^{1:M}$ and its dual $T^*\mathcal{M}^{1:M}$ as:

$$T_{(\vartheta,q^{1:M})}\mathcal{M}^{1:M} = T_\vartheta\mathbb{R}^D \times \prod_{m\in[M]} T_{q^m}\mathcal{P}_2(\mathsf{Z})$$
$$T^*_{(\vartheta,q^{1:M})}\mathcal{M}^{1:M} = T^*_\vartheta\mathbb{R}^D \times \prod_{m\in[M]} T^*_{q^m}\mathcal{P}_2(\mathsf{Z})$$

where we note that $T_\vartheta\mathbb{R}^D \cong T^*_\vartheta\mathbb{R}^D \cong \mathbb{R}^D$. For each $q^m \in \mathcal{P}_2(\mathsf{Z})$, we also define the tangent and cotangent space of $\mathcal{P}_2(\mathsf{Z})$ at q^m as in Otto (2001):

$$T_{q^m}\mathcal{P}_2(\mathsf{Z}) := \left\{ r : \mathsf{Z} \to \mathbb{R} : \int r(z)\mathrm{d}z = 0 \right\}$$
$$T^*_{q^m}\mathcal{P}_2(\mathsf{Z}) := \{f : \mathsf{Z} \to \mathbb{R}\}/\mathbb{R}$$

where the cotangent space $T^*_{q^m}\mathcal{P}_2(\mathsf{Z})$ is identified with the space of equivalence classes of functions that differ by an additive constant.

Furthermore, we define the *duality pairing*, which is a triplet $(T_{(\vartheta,q^{1:M})}, T^*_{(\vartheta,q^{1:M})}, \langle\cdot,\cdot\rangle)$ with

$$\langle\cdot,\cdot\rangle : T_{(\vartheta,q^{1:M})}\mathcal{M}^{1:M} \times T^*_{(\vartheta,q^{1:M})}\mathcal{M}^{1:M} \to \mathbb{R}$$

being a bilinear map that is the sum of the Euclidean inner product on the parameter space and the duality pairing on the Wasserstein-2 space:

$$\langle(\tau, r^{1:M}), (v, f^{1:M})\rangle := \langle\tau, v\rangle + \sum_{m\in[M]} \langle r^m, f^m\rangle,$$

where the Wasserstein-2 duality pairing is $(T_{q^m}\mathcal{P}_2(\mathsf{Z}), T^*_{q^m}\mathcal{P}_2(\mathsf{Z}), \langle\cdot,\cdot\rangle)$ with $\langle\cdot,\cdot\rangle : T_{q^m}\mathcal{P}_2(\mathsf{Z}) \times T^*_{q^m}\mathcal{P}_2(\mathsf{Z}) \to \mathbb{R}$ given by:

$$\langle r^m, f^m\rangle := \int f^m(z) r^m(z)\mathrm{d}z, \quad \forall r^m \in T_{q^m}\mathcal{P}_2(\mathsf{Z}), f^m \in T^*_{q^m}\mathcal{P}_2(\mathsf{Z}). \tag{43}$$

The metric. We can thus equip the manifold $\mathcal{M}^{1:M}$ with Riemannian metric $g = (g_{(\vartheta,q^{1:M})})_{(\vartheta,q^{1:M})\in\mathcal{M}^{1:M}}$ defined as:

$$g_{(\vartheta,q^{1:M})}((\tau,r^{1:M}),(\tau',r'^{1:M})) := \langle(\tau,r^{1:M}), G_{(\vartheta,q^{1:M})}(\tau',r'^{1:M})\rangle, \quad \forall(\vartheta,q^{1:M})\in\mathcal{M}^{1:M},$$

where $G_{(\vartheta,q^{1:M})} : T\mathcal{M}^{1:M} \to T^*\mathcal{M}^{1:M}$ is an invertible, self-adjoint, and positive-definite linear map. We will only consider tensors in a block-diagonal form (cf. Lee (2018, Chapter 2)), which defines the metric via:

$$\langle(\tau,r^{1:M}), G_{(\vartheta,q^{1:M})}(\tau',r'^{1:M})\rangle = \langle\tau,\mathrm{G}_\vartheta\tau'\rangle + \frac{1}{M}\sum_{m\in[M]}\langle r^m, \mathbf{G}^W_{q^m} r'^m\rangle \quad \forall(\tau,r^{1:M})\in T\mathcal{M}^{1:M},\ (\vartheta,q^{1:M})\in\mathcal{M}^{1:M}. \tag{44}$$

Here we take $\mathrm{G}_\vartheta : T\mathbb{R}^D \to T^*\mathbb{R}^D$ as the identity map (which corresponds to the usual Euclidean metric) and $\mathbf{G}^W_q : T\mathcal{P}_2(\mathsf{Z}) \to T^*\mathcal{P}_2(\mathsf{Z})$ to be the tensor for Wasserstein-2 distance on $\mathcal{P}_2(\mathsf{Z})$ defined through its inverse:

$$(\mathbf{G}^W_q)^{-1} f := -\nabla_z\cdot(q\nabla_z f), \quad \forall f\in C^\infty(\mathcal{P}_2(\mathsf{Z})). \tag{45}$$

When the context is clear, we will also write $\langle\tau,\tau'\rangle_\vartheta = \langle\tau,\mathrm{G}_\vartheta\tau'\rangle$ and $\langle r^m, r'^m\rangle_{q^m} = \langle r^m, \mathbf{G}^W_{q^m} r'^m\rangle$.

The gradient. To perform gradient descent on manifolds, we further need an analogue of the gradient for a smooth function $F : \mathcal{M}^{1:M} \to \mathbb{R}$ as in the Euclidean space. This is a vector field $\nabla F : \mathcal{M}^{1:M} \to T\mathcal{M}^{1:M}$ satisfying:

$$g_{(\vartheta,q^{1:M})}(\nabla F(\vartheta,q^{1:M}),(\tau,r^{1:M})) = \lim_{t\to0}\frac{F(\vartheta+t\tau,q^{1:M}+tr^{1:M})-F(\vartheta,q^{1:M})}{t} \quad \forall(\tau,r^{1:M})\in T\mathcal{M}^{1:M},\ (\vartheta,q^{1:M})\in\mathcal{M}^{1:M}. \tag{46}$$

By expressing in local coordinates, we can compute the gradient as:

$$\nabla F(\vartheta,q^{1:M}) = G^{-1}_{(\vartheta,q^{1:M})}\delta F(\vartheta,q^{1:M}), \quad \forall(\vartheta,q^{1:M})\in\mathcal{M}^{1:M}, \tag{47}$$

where $\delta F : \mathcal{M}^{1:M} \to T^*\mathcal{M}^{1:M}$ is the first variation of F defined as the unique cotangent vector field satisfying:

$$\langle(\tau,r^{1:M}),\delta F(\vartheta,q^{1:M})\rangle = \lim_{t\to0}\frac{F(\vartheta+t\tau,q^{1:M}+tr^{1:M})-F(\vartheta,q^{1:M})}{t} \quad \forall(\tau,r^{1:M})\in T\mathcal{M}^{1:M},\ (\vartheta,q^{1:M})\in\mathcal{M}^{1:M}. \tag{48}$$

The distance function. To define the distance on the manifold $\mathcal{M}^{1:M}$, we recall that the Riemannian distance function is defined as the length of the minimizing geodesics $\gamma : [0,1] \to \mathcal{M}^{1:M}$ between $(\vartheta,q^{1:M})$ and $(\vartheta',q'^{1:M})$:

$$\mathsf{d}_{\mathcal{M}^{1:M}}((\vartheta,q^{1:M}),(\vartheta',q'^{1:M}))^2 := \inf_\gamma\int_0^1 g_{(\vartheta_t,q_t^{1:M})}(\dot\gamma(t),\dot\gamma(t))\mathrm{d}t = \|\vartheta-\vartheta'\|_2^2 + \frac{1}{M}\sum_{m=1}^M \mathsf{d}_{W_2}(q^m,(q')^m)^2, \tag{49}$$

where the second equality follows from Otto (2001, Section 4.3) and W_2 is the Wasserstein-2 distance on $\mathcal{P}_2(\mathsf{Z})$.

A.3 Derivation of the Gradient Flow

Equipped with the tools above, we now derive the gradients of the averaged free energy $\tilde F(\theta,\phi,q^{1:M}) = M^{-1}\sum_{m=1}^M \tilde F^m(\theta,\phi,q^m)$, where $\tilde F^m$ is the single-datapoint free energy for x^m. We first compute the single-datapoint first variation (48), and then lift it componentwise to the averaged objective:

Lemma A.1. *Given the free energy of the form in (25):*

$$\tilde F(\theta,\phi,q) = \int\mathcal{L}_D(\theta,z)q(z)\mathrm{d}z - \int\log\left(\frac{p_\phi(x|z)}{q(z)}\right)q(z)\mathrm{d}z. \tag{50}$$

The first variation $\delta\tilde F(\theta,\phi,q) = (\delta_q\tilde F(\theta,\phi,q),\delta_\theta\tilde F(\theta,\phi,q),\delta_\phi\tilde F(\theta,\phi,q))$ is given by:

$$\delta_q\tilde F(\theta,\phi,q) = \log\left(\frac{q(z)}{p_\phi(x|z)}\right) + \mathcal{L}_D(\theta,z) \tag{51}$$

$$\delta_\theta\tilde F(\theta,\phi,q) = \int\nabla_\theta\mathcal{L}_D(\theta,z)q(z)\mathrm{d}z \tag{52}$$

$$\delta_\phi\tilde F(\theta,\phi,q) = -\int\nabla_\phi\log p_\phi(x|z)q(z)\mathrm{d}z \tag{53}$$

Proof. For eqs. (52) and (53), it suffices to note that by grouping the parameters into $\tilde{\theta} := (\theta, \phi) \in \Theta = \mathbb{R}^D$, we can compute by Taylor expansion:

$$\begin{aligned}\tilde{F}(\tilde{\theta} + t\tau, q) &= \tilde{F}(\tilde{\theta}, q) + \int f(\tilde{\theta} + t\tau, z) q(z) \mathrm{d}z - \int f(\tilde{\theta}, z) q(z) \mathrm{d}z \\ &= \tilde{F}(\tilde{\theta}, q) + \int [t\langle \tau, \nabla_{\tilde{\theta}} f(\tilde{\theta}, z)\rangle + o(t)] q(z) \mathrm{d}z \\ &= \tilde{F}(\tilde{\theta}, q) + t \left\langle \tau, \int \nabla_{\tilde{\theta}} f(\tilde{\theta}, z) q(z) \mathrm{d}z \right\rangle + o(t)\end{aligned}$$

where we have set $f(\tilde{\theta}, z) := \mathcal{L}_D(\theta, z) - \log p_\phi(x|z)$ and used $f(\tilde{\theta} + t\tau, z) = f(\tilde{\theta}, z) + t\langle \tau, \nabla_{\tilde{\theta}} f(\tilde{\theta}, z)\rangle + o(t)$.

For (51), we recall the first variation is linear, thus it suffices to compute $\delta_q \int \mathcal{L}_D(\theta, z) q(z) \mathrm{d}z$ and $\delta_q \int (\log q(z) - \log p_\phi(x|z)) q(z) \mathrm{d}z$ separately. Now note that:

$$\int \mathcal{L}_D(\theta, z)[q(z) + tr(z)] \mathrm{d}z = \int \mathcal{L}_D(\theta, z) q(z) \mathrm{d}z + t \int \mathcal{L}_D(\theta, z) r(z) \mathrm{d}z$$

and for the latter we note that $\log(z + t)(z + t) = \log(z)z + [\log(z) + 1]t + o(t)$, from which it follows:

$$\begin{aligned}\int \log\left(\frac{q(z) + tr(z)}{p_\phi(x|z)}\right)[q(z) + tr(z)] \mathrm{d}z &= \int (\log q(z)) q(z) + [\log q(z) + 1] tr(z) + o(t) \mathrm{d}z \\ &\quad - \int \log p_\phi(x|z) q(z) \mathrm{d}z - t \int \log p_\phi(x|z) r(z) \mathrm{d}z \\ &= \int \log\left(\frac{q(z)}{p_\phi(x|z)}\right) q(z) \mathrm{d}z \\ &\quad + t \int \log\left(\frac{q(z)}{p_\phi(x|z)}\right) r(z) \mathrm{d}z + o(t)\end{aligned}$$

where we have used that $\int r(z) \mathrm{d}z = 0$ for all $r \in T\mathcal{P}_2(\mathsf{Z})$. □

Proposition A.2. *Under the normalized product geometry of Appendix A.2, the gradients of the averaged free energy $\tilde{F}(\theta, \phi, q^{1:M})$ with respect to $(\theta, \phi, q^{1:M})$ are given by:*

$$\nabla_\theta \tilde{F}(\theta, \phi, q^{1:M}) = \frac{1}{M} \sum_{m=1}^{M} \int [\nabla_\theta \mathcal{L}_D(\theta, z)]\, q^m(z) \mathrm{d}z \tag{54}$$

$$\nabla_\phi \tilde{F}(\theta, \phi, q^{1:M}) = -\frac{1}{M} \sum_{m=1}^{M} \int [\nabla_\phi \log p_\phi(x^m|z)]\, q^m(z) \mathrm{d}z \tag{55}$$

$$\nabla_{q^m} \tilde{F}(\theta, \phi, q^{1:M}) = \nabla_z \cdot \left[q^m(z) \nabla_z \left[\log\left(\frac{p_\phi(x^m|z)}{q^m(z)}\right) - \mathcal{L}_D(\theta, z) \right] \right], \quad \forall m \in [M]. \tag{56}$$

Proof. Recall that on the manifold $\mathcal{M}^{1:M}$, the gradient can be computed from the first variation via the metric tensor (47):

$$\nabla_{(\tilde{\theta}, q^{1:M})} \tilde{F}(\tilde{\theta}, q^{1:M}) = G^{-1}_{(\tilde{\theta}, q^{1:M})} \delta \tilde{F}(\tilde{\theta}, q^{1:M}),$$

where $\tilde{\theta} = (\theta, \phi)$. For the parameter block, the metric is Euclidean, so the θ- and ϕ-gradients are the corresponding averaged first variations. For each distribution q^m, we have

$$\delta_{q^m} \tilde{F}(\theta, \phi, q^{1:M}) = \frac{1}{M} \delta_q \tilde{F}^m(\theta, \phi, q^m),$$

where $\delta_q \tilde{F}^m$ is given by Lemma A.1 with x replaced by x^m and q by q^m. Since the inverse metric on the mth Wasserstein block is $M(\mathbf{G}^W_{q^m})^{-1}$, the factor M^{-1} in $\delta_{q^m} \tilde{F}$ is cancelled, yielding

$$\nabla_{q^m} \tilde{F}(\theta, \phi, q^{1:M}) = (\mathbf{G}^W_{q^m})^{-1} \delta_q \tilde{F}^m(\theta, \phi, q^m).$$

Recalling that $(\mathbf{G}^W_q)^{-1} f = -\nabla_z \cdot (q \nabla_z f)$ for all $f \in C^\infty(\mathcal{P}_2(\mathsf{Z}))$ gives the desired result. □

A.4 The Full Training Algorithm

We detail the algorithmic considerations in Section 3.3 and provide the pseudocode for practical training in Algorithm 2.

A.4.1 KL Divergence Annealing

We note that annealing the KL divergence with γ_t is equivalent to applying the same weighting to the gradient of the entropy and the prior terms. In our case, we modify the gradient flow in (9–14) as follows:

$$\nabla_\theta \tilde{F}(\theta_t, \phi_t, q_t^{1:M}) = \frac{1}{M}\sum_{m=1}^{M} \mathbb{E}_{q_t^m(z_{0,t})}\left[\nabla_\theta \gamma_t \mathcal{L}_D(\theta_t, z_0)\right], \tag{57}$$

$$\nabla_\phi \tilde{F}(\theta_t, \phi_t, q_t^{1:M}) = -\frac{1}{M}\sum_{m=1}^{M} \mathbb{E}_{q_t^m(z_{0,t})}\left[\nabla_\phi \log p_{\phi_t}(x^m|z_{0,t})\right], \tag{58}$$

$$\begin{aligned}\nabla_{q^m} \tilde{F}(\theta_t, \phi_t, q_t^{1:M}) &= \nabla_{z_0}\cdot[q^m(z_{0,t})\nabla_{z_0}\log p_{\phi_t}(x^m|z_{0,t})] &(59)\\ &\quad - \gamma_t \nabla_{z_0}\cdot[q^m(z_{0,t})\nabla_{z_0}[\log q^m(z_{0,t}) - \mathcal{L}_D(\theta_t, z_{0,t})]], \quad \forall m \in [M] &(60)\end{aligned}$$

which correspond to the Fokker-Planck equation of the following system of SDEs:

$$\mathrm{d}\theta_t = -\frac{1}{M}\sum_{m=1}^{M} \mathbb{E}_{q_t^m(z_0)}\left[\nabla_\theta \gamma_t \mathcal{L}_D(\theta_t, Z_{0,t}^m)\right]\mathrm{d}t, \tag{61}$$

$$\mathrm{d}\phi_t = \frac{1}{M}\sum_{m=1}^{M} \mathbb{E}_{q_t^m(z_0)}\left[\nabla_\phi \log p_{\phi_t}(x^m|Z_{0,t}^m)\right]\mathrm{d}t, \tag{62}$$

$$\mathrm{d}Z_{0,t}^m = \nabla_{z_0}[\log(p_{\phi_t}(x^m|Z_{0,t}^m)) - \gamma_t \mathcal{L}_D(\theta_t, Z_{0,t}^m)]dt + \sqrt{2\gamma_t}\mathrm{d}W_t^m, \quad \forall m \in [M] \tag{63}$$

Thus, we only need to adjust the amount of noise injected to the in (21) by setting $\sqrt{2h_z}$ to $\sqrt{2h_z\gamma_t}$ and weigh the diffusion loss $\mathcal{L}_D(\theta, z_0)$ by γ_t. Using the KL annealing scheme and re-weighted diffusion loss (41) discussed in Section 3.3, we thus modify the loss in (22) with:

$$\hat{\mathcal{L}}(\theta_t, \phi_t, z_{0,t}^{1:M,1:N}) := \frac{1}{N|\mathcal{B}|}\sum_{(m,n)\in\mathcal{B}\times[N]}\left[\gamma_t \hat{\mathcal{L}}_{\text{simple}}(\theta_t, z_{0,t}^{m,n}) - \log p_\phi(x^m|z_{0,t}^{m,n})\right], \tag{64}$$

where $\hat{\mathcal{L}}_{\text{simple}}$ is the simplified diffusion loss in Appendix A.1 and we take $\gamma_t = c_{KL}\min(1, t/T)$ for T the number of KL warm-up steps and c_{KL} a constant coefficient.

A.4.2 Preconditioning and Momentum

To avoid flat minima and ill-conditioning, we use an adaptive version of Langevin dynamics similar to Li et al. (2016); Kim et al. (2020) based on Adam (Kingma and Ba, 2014). In particular, we update each particle by running:

$$z_{0,t+1}^{m,n} \leftarrow z_{0,t}^{m,n} + h_z G_t^{m,n} M_t^{m,n} + \sqrt{\frac{2h_z}{M}}(G_t^{m,n})^{1/2} W_t^{m,n}, \quad \forall (m,n) \in [M]\times[N] \tag{65}$$

where h_z is the stepsize of particle updates, and we compute the moment $M_t^{m,n}$ and the preconditioner $G_t^{m,n}$ by:

$$M_t^{m,n} \leftarrow aM_{t-1}^{m,n} + MN(1-a)\nabla_{z_0^{m,n}}\hat{\mathcal{L}}(\theta_t, \phi_t, z_{0,t}^{1:M,1:N}) \tag{66a}$$

$$V_t^{m,n} \leftarrow bV_{t-1}^{m,n} + (1-b)\mathrm{diag}\left(\left[MN\nabla_{z_0^{m,n}}\hat{\mathcal{L}}(\theta_t, \phi_t, z_{0,t}^{1:M,1:N})\right]^{\otimes 2}\right) \tag{66b}$$

$$G_t^{m,n} \leftarrow (V_t^{m,n} + \epsilon I)^{-1/2} \tag{66c}$$

Here $\hat{\mathcal{L}}$ is defined as in (64), $\epsilon > 0$ is a positive constant to avoid numerical instabilities, $0 < a, b < 1$ are hyperparameters like in Adam (Kingma and Ba, 2014) (we take $a = 0.9, b = 0.999$ following the default), and we

used the notation $\mathrm{diag}(v^{\otimes 2})$ to denote the diagonal matrix with the $(i,j)^{th}$ entry being $v_i^2\delta_{ij}$. In practice, we set the preconditioner on the noise in (65) to $G_t^{m,n} = I$ during the first epoch for numerical stability. We also scale the noise by $|\mathcal{B}|^{-1}$ in implementation to prevent the noise from dominating the gradient. We now present the full algorithm for training.

Algorithm 2 IPLD with adaptive Langevin dynamics

Inputs: Training data points $\{x^m\}_{m\in[M]}$, optimizers $\texttt{Opt}_\theta, \texttt{Opt}_\phi$, Step sizes h_θ, h_ϕ, h_z, KL weights γ_t, Initial particles $\{z_{0,0}^{m,n}\}_{(m,n)\in[M]\times[N]}$ sampled from $\mathcal{N}(0,I)$, Initial parameters ϕ, θ.
while not converged **do**
  Sample a mini-batch of indices $\mathcal{B} \subset [M]$
  Compute the loss $\hat{\mathcal{L}}(\theta_t, \phi_t, z_{0,t}^{1:M,1:N})$ as in (64)
  for $(m,n) \in \mathcal{B} \times [M]$ **do** ▷ Update the particle cloud
    Compute momentum $M_t^{m,n}$ and preconditioner $G_t^{m,n}$ as in (66)
    Sample independent $W_{t,m}^i$ for $(i,m) \in [N] \times [M]$
    if $m \in \mathcal{B}$ **then**
      $z_{0,t+1}^{m,n} \leftarrow z_{0,t}^{m,n} + h_z G_t^{m,n} M_t^{m,n}$
    end if
    $z_{0,t+1}^{m,n} \leftarrow z_{0,t}^{m,n} + \sqrt{\frac{2h_z\gamma_t}{M}}(G_t^{m,n})^{1/2} W_t^{m,n}$
  end for ▷ Update model parameters
  $\theta_{t+1} \leftarrow \texttt{Opt}_\theta(h_\theta, \theta_t, \hat{\mathcal{L}})$
  $\phi_{t+1} \leftarrow \texttt{Opt}_\phi(h_\phi, \phi_t, \hat{\mathcal{L}})$
  $t \leftarrow t+1$
end while
return $\theta_t, \phi_t, z_{0,t}^{1:M,1:N}$

B Experimental Details

For simplicity, we use a Gaussian decoder with identity covariance $p_\phi(x|z_0) = \mathcal{N}(x; g_\phi(z_0), I)$ throughout the experiments, where g_ϕ is the decoder network parametrized by ϕ. We also fix the noise schedule $\{\beta_k\}_{k=1}^K$ as the linear schedule used in Ho et al. (2020) with $\beta_0 = 1\times10^{-4}$ and $\beta_K = 0.02$ with $K = 1000$.

B.1 Details on the synthetic experiments

Data. We create the training data by first drawing 10,000 samples from: 1) a GMM with 25 components of dimension $d_z = 2$ as in Boys et al. (2024); Cardoso et al. (2024), and 2) a concentric circle distribution also of dimension $d_z = 2$ similar to that in `scikit-learn`[5](Buitinck et al., 2013), then we project the data into a higher-dimensional ambient space with dimension $d_x = 64$ using matrices $A \in \mathbb{R}^{d_z \times d_x}$ with orthogonal rows. We generate the matrices using the default implementation in PyTorch of orthogonal weight initialization (Saxe et al., 2014). For better visualization, we set the first 2×2 block to the identity for the concentric circle dataset.

Architecture. For the diffusion backbones, we use a multi-layer perceptron (MLP) with 3 hidden layers, each having 128 hidden units and ReLU activation. The decoder is parametrized by a single linear layer which is equivalent to a matrix with dimension $d_z \times d_x$. For the DiffusionVAE, we implement the encoder also using a single linear layer equivalent to a matrix of size $d_x \times 2d_z$, where it outputs the mean and diagonal of the log-covariance matrix.

Training. We train all models for 50 epochs with a batch size of 500. We use the AdamW optimizer (Loshchilov and Hutter, 2019) with a learning rate of 1×10^{-3} for all models. For IPLD, we use a version of adaptive Langevin dynamics (Kim et al., 2020) (cf. Algorithm 2) to optimize the particles; we set the step size to 1×10^{-1} for faster convergence. The KL annealing constant c_{KL} is set to 0.01 and number of warm up steps is set to 1000 (cf. Appendix A.4).

[5] https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_circles.html

Evaluation. We report the MMD (Gretton et al., 2012) between the generated samples and the ground truth on the GMM dataset. For samples $\{x_i\}_{i=1}^m \sim P$ and $\{y_i\}_{j=1}^m \sim Q$, the unbiased Monte-Carlo estimate of MMD is defined as:

$$\mathrm{MMD}(P,Q) \approx \frac{1}{m(m-1)} \sum_{i=1}^{m} \sum_{\substack{j=1 \\ j\neq i}}^{m} k(x_i, x_j) + \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j\neq i}}^{n} k(y_i, y_j) - \frac{2}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} k(x_i, y_j),$$

where we use the Radial Basis Function (RBF) kernel $k : \mathbb{R}^{d_x} \times \mathbb{R}^{d_x} \to \mathbb{R}_{\geq 0}$ with a bandwidth $\gamma = 0.1$ defined as $k(x,y) = \exp\left(-\gamma ||x-y||^2\right)$. A total of $m = n = 10,000$ samples were generated via the reverse process to compute the approximation of the MMD. We repeat the training runs with 50 different random seeds (thus different initializations) and report the mean and standard error. The error bars shown in Figure 1 are the 1-standard error $\hat{\sigma}/\sqrt{50}$, where the standard deviation $\hat{\sigma}$ is estimated using `numpy`'s default implementation.

B.2 Details on the image experiments

Architecture for IPLD. We adopt the Diffusion Transformer architecture `DiT-S`, the smallest configuration from Peebles and Xie (2023), as the backbone for our latent diffusion model. We use a patch size of 1×1, as our latent space of dimension $4 \times 8 \times 8$ is relatively small. For the Gaussian decoder $p_\phi(x|z_0)$, we use a simplified version of VAE's decoder without attention from Rombach et al. (2022). We comment that in line

	SVHN	CIFAR-10	CelebA64
z-shape	8 x 8 x 4	8 x 8 x 4	8 x 8 x 4
Base channels	128	128	128
Number residual blocks per resolution	2	2	2
Channel Multiplier	1,2,3	1,2,3	1,2,2,3
Batch Size	128	128	16
Number of Epochs	400	400	200
Diffusion Learning Rate	1e-4	1e-4	1e-4
Decoder Learning Rate	2e-4	2e-4	2e-4
Particle Step Size	5e-2	5e-2	5e-2
KL Warm-up Steps	40000	40000	40000
KL Constant Coefficient	0.001	0.01	0.01
EMA Decay Rate	0.999	0.999	0.9999
EMA Start Step	40000	40000	40000

Table 2: An overview of IPLD's settings for the experiments. Here z-shape refers to the shape of the latent vector, which in the case of IPLD, is the shape of a single particle. We refer the readers to LDM (Rombach et al., 2022) and the implementation thereof for more details on the architecture.

with recent discussions on latent diffusions (Dieleman, 2025), the spatial structure of the latent space is crucial for the successful training of a latent diffusion model. Therefore, we choose to use a decoder that can induce explicit spatial structures in the latents z instead of the ones used in the EBM literature (Yu et al., 2023), which compress the images to a flattened vector. However, we note our decoders have similar or fewer parameters than the implementation of Yu et al. (2023) (both around 18 million parameters).

Architecture for DiffusionVAE. We use the exact same diffusion backbone and decoder for our re-implementation of VAE with diffusion prior (Wehenkel and Louppe, 2021). For the encoder $q_\phi(z_0|x)$ of the VAE, we parametrize it with a Gaussian $\mathcal{N}(z_0; \mu_\phi(x), \Sigma_\phi)$, where Σ_ϕ is a diagonal matrix. The encoder's architecture also follows from Rombach et al. (2022), which is an inverted version of the decoder.

Training and Evaluation. As in the synthetic experiments, we use the AdamW optimizer (Loshchilov and Hutter, 2019) for with a learning rate of 2×10^{-4} for decoders (and encoder for DiffusionVAE) and 1×10^{-4} for diffusion backbones; an exponential learning rate schedule with rate $\gamma = 0.999$ was used for the decoder. For IPLD, we use adaptive Langevin dynamics (Kim et al., 2020) with step size 5×10^{-2} and exponential learning rate decay with $\gamma = 0.995$ across all configurations following Kuntz et al. (2023). Similar to Song and Ermon

(2020), we maintain an Exponential Moving Average (EMA) of the weights of the diffusion model for evaluation; the hyperparameters for EMA are reported in Table 2. To evaluate the generative performance, we calculate the FID (Heusel et al., 2017) between the true data and 50,000 generated samples using the DDIM sampler (Song et al., 2021a) with 100 network function evaluations (NFEs).

B.3 Warming-up with one particle

For the image experiments, we use a warm-started approach (Kuntz et al., 2023) for faster training. Namely, we initiate IPLD with a single particle $\{z_m\}_{m\in[M]}$ and run Algorithm 2 for 200 epochs on SVHN and CIFAR-10 (100 on CelebA64) before switching to $N > 1$ particles by replicating the each particle $\{z_m\}_{m\in[M]}$ for N times. We point out that due to the noise added in Langevin dynamics, the particles do not collapse to a single point.

B.4 Computational Resources

For all experiments, we use NVIDIA GPUs. We use a single RTX 3090 for all synthetic experiments. For image experiments, the 10-particle version of IPLD training runs on CIFAR-10 and SVHN datasets were performed on two RTX A6000 GPUs or a single A100 GPU. The remaining image experiments were performed on a single L40S GPU. Our longest experiment takes about 27 hours on a single A100 GPU, which is around 2.5 GPU days measured in V100 using the conversion rule in Rombach et al. (2022).

C Proofs of Theoretical Results

Notation and assumptions. We denote $\rho_{\theta,\phi}(\cdot)$ as the unnormalized density for:

$$p_{\theta,\phi}(x,\cdot) = p_\phi(x|\cdot)p_\theta(\cdot). \tag{67}$$

Similarly, we define $\tilde{\rho}_{\theta,\phi}(\cdot)$ as the unnormalized density for:

$$\tilde{p}_{\theta,\phi}(x,\cdot) := p_{\theta,\phi}(x,\cdot)\exp(-\mathcal{R}(\theta,\cdot)), \tag{68}$$

where $p_{\theta,\phi}(x,z_0) = p_\theta(z_0)p_\phi(x|z_0)$ and $\mathcal{R}(\theta,z_0) := D_{\mathrm{KL}}(q(z_{1:K}|z_0)||p_\theta(z_{1:K}|z_0))$. In multi-datapoint settings, we add a superscript to denote the dependence on x^m i.e.

$$\tilde{\rho}^m_{\theta,\phi}(\cdot) := \tilde{p}_{\theta,\phi}(x^m,\cdot). \tag{69}$$

The M-datapoint version is defined as:

$$\tilde{\rho}_{\theta,\phi}(z^{1:M}) := \prod_{m=1}^M \rho^m_{\theta,\phi}(z^m) = \prod_{m=1}^M \tilde{p}_{\theta,\phi}(x^m,z^m). \tag{70}$$

And the joint log density is:

$$\ell^m(\theta,\phi,z) := \log \rho^m_{\theta,\phi}(z) = \log p_\phi(x^m|z) + \log p_\theta(z). \tag{71}$$

We also denote $\pi^m_{\theta,\phi}(\cdot) := p_{\theta,\phi}(\cdot|x^m)$ as the normalized density and define similarly $\tilde{\pi}^m_{\theta,\phi}(\cdot) := \tilde{p}_{\theta,\phi}(\cdot|x^m)$ for the tilted model. We further denote the normalizing constant of $p_{\theta,\phi}(x,\cdot)$ as $A_{\theta,\phi}$ and that of $\tilde{p}_{\theta,\phi}(x,\cdot)$ as $\tilde{A}_{\theta,\phi}$. We recall the definition of the free energy $F(\theta,\phi,q)$ as:

$$F(\theta,\phi,q) := \mathbb{E}_{q(z_0)}\left[\log \frac{q(z_0)}{p_{\theta,\phi}(x,z_0)}\right]. \tag{72}$$

We set the modified free energy as:

$$\tilde{F}(\theta,\phi,q) := \mathbb{E}_{q(z_{0:K})}\left[\log \frac{q(z_{0:K})}{p_{\theta,\phi}(x,z_{0:K})}\right] = F(\theta,\phi,q) + \mathbb{E}_{q(z_0)}\left[\mathcal{R}(\theta,z_0)\right]. \tag{73}$$

The tilted free energy aggregated over multiple data points $\{x^m\}_{m=1}^M$ is defined as:

$$\tilde{F}(\theta,\phi,q^{1:M}) := \frac{1}{M}\sum_{m=1}^M \tilde{F}(\theta,\phi,q^m). \tag{74}$$

For clarity, we drop the subscript on z_0 and instead write z for the latent variable in subsequent sections and we use the notations $\Theta := \mathbb{R}^{d_\theta}$, $\Phi := \mathbb{R}^{d_\phi}$ for the parameter spaces and $\mathcal{P}_2(\mathsf{Z})$ for the space of probability measures over the latent space $\mathsf{Z} = \mathbb{R}^{d_z}$, which we assume to have densities with respect to the Lebesgue measure. We define the product manifold and the distance $\mathsf{d}_{\mathcal{M}}$ as in Appendix A.2:

$$\mathcal{M} := \Theta \times \Phi \times \mathcal{P}_2(\mathsf{Z}), \quad \mathsf{d}_{\mathcal{M}}((\theta,\phi,q),(\theta',\phi',q')) := \sqrt{\|(\theta,\phi)-(\theta',\phi')\|^2 + \mathsf{d}_{W_2}(q,q')^2}. \tag{75}$$

When (θ, ϕ, q) are random variables, we overload the notation d to denote:

$$\mathsf{d}((\theta,\phi,q),(\theta',\phi',q')) := \sqrt{\mathbb{E}[\|(\theta,\phi)-(\theta',\phi')\|^2] + \mathbb{E}[\mathsf{d}_{W_2}(q,q')^2]}. \tag{76}$$

To extend the above to multiple datapoints, we set:

$$\mathcal{M}^{1:M} := \Theta \times \Phi \times \mathcal{P}_2(\mathsf{Z})^M, \tag{77}$$

$$\mathsf{d}_{\mathcal{M}^{1:M}}((\theta,\phi,q^{1:M}),(\theta',\phi',q'^{1:M})) := \sqrt{\|(\theta,\phi)-(\theta',\phi')\|^2 + \frac{1}{M}\sum_{m=1}^{M}\mathsf{d}_{W_2}(q^m,q'^m)^2}. \tag{78}$$

We additionally denote $\mathcal{P}_2^1(\mathsf{Z})$ as the subset of $\mathcal{P}_2(\mathsf{Z})$ with densities differentiable almost everywhere w.r.t. the Lebesgue measure. We use a superscript 1 for related product spaces (*e.g.*, $\mathcal{M}^1$ and $\mathcal{M}^{1:M,1}$) to indicate the restriction to $\mathcal{P}_2^1(\mathsf{Z})$.

We now restate the full assumptions for the theoretical results in the main text.

A1 (Model regularity)**.** *We assume that*

1. *For all $z \in \mathsf{Z}$, $(\theta,\phi) \mapsto \tilde{\pi}_{\theta,\phi}(z)$ and $(\theta,\phi) \mapsto A_{\theta,\phi}$ are differentiable;*
2. *for all $(\theta,\phi) \in \Theta \times \Phi$, $\tilde{\pi}_{\theta,\phi}$ is twice continuously differentiable;*
3. *$\tilde{\rho}_{\theta,\phi}(z) > 0$ for all $z \in \mathsf{Z}$ and $(\theta,\phi) \in \Theta \times \Phi$.*

A2 (Regularity of solutions)**.** *For any initial conditions $(\theta,\phi,q^{1:M}) \in \mathcal{M}^{1:M}$, the gradient flow has a classical solution $(\theta_t,\phi_t,q_t^{1:M})_{t\geq 0}$ with $(\theta_0,\phi_0,q_0^{1:M}) = (\theta,\phi,q^{1:M})$. Furthermore, for all $m \in [M]$ and $t \geq 0$, q_t^m has a Lebesgue density in $\mathcal{C}^{1,2}([0,\infty)\times\mathsf{Z},\mathbb{R}^+)$ and $(\theta_t,\phi_t) \in \mathcal{C}^1([0,\infty),\Theta\times\Phi)$.*

A3 (Strong log-concavity)**.** *For all $x \in \mathsf{X}$, the tilted joint density $\tilde{p}_{\theta,\phi}(x,z)$ is λ-strongly log-concave in (θ,ϕ,z) for some $\lambda > 0$.*

C.1 Full Statement and Proof of Theorem 3.1

We now provide the full statement and proof of Theorem 3.1. The strategy will be the same as Caprio et al. (2025), where we first establish the extended log-Sobolev inequality under strong log-concavity and subsequently the extended Talagrand inequality. Our main difference from Caprio et al. (2025) is that 1) we work with the tilted model $\tilde{p}_{\theta,\phi}(x,z)$, which has an additional component arising from the diffusion loss (cf. Lemma C.9), and 2) we consider the multi-datapoint setting, which leads to a different definition of the manifold and the distance thereon.

C.1.1 Extended Log-Sobolev Inequality

In this section, we show that strong log-concavity implies extended log-Sobolev inequality. We first define the M-datapoint version of the extended log-Sobolev inequality in Caprio et al. (2025):

Definition C.1 (Extended Log-Sobolev Inequality (xLSI))**.** Denote $\tilde{F}_\star := \inf_{(\theta,\phi,q^{1:M})\in\mathcal{M}^{1:M,1}} \tilde{F}(\theta,\phi,q^{1:M})$ as the optimum of $\tilde{F}(\theta,\phi,q^{1:M})$. We say the measure $(\tilde{\rho}_{\theta,\phi}(\mathrm{d}z^{1:M}))_{(\theta,\phi)\in\Theta\times\Phi}$ satisfies the extended log-Sobolev inequality with constant $\lambda > 0$ if for all $(\theta,\phi,q^{1:M}) \in \mathcal{M}^{1:M,1}$ we have:

$$2\lambda[\tilde{F}(\theta,\phi,q^{1:M}) - \tilde{F}_\star] \leq I(\theta,\phi,q^{1:M}), \tag{79}$$

where we define $I(\theta, \phi, q^{1:M})$ as:

$$\begin{aligned} I(\theta, \phi, q^{1:M}) &:= \|\nabla_{(\theta,\phi)} \tilde{F}(\theta, \phi, q^{1:M})\|^2 + \frac{1}{M} \sum_{m=1}^{M} \int \left\| \nabla_z \log \left(\frac{q^m(z)}{\tilde{p}_{\theta,\phi}(x^m, z)} \right) \right\|^2 q^m(\mathrm{d}z) \\ &= \left\| \frac{1}{M} \sum_{m=1}^{M} \int \nabla_{(\theta,\phi)} \log \tilde{p}_{\theta,\phi}(x^m, z)\, q^m(\mathrm{d}z) \right\|^2 + \frac{1}{M} \sum_{m=1}^{M} \int \left\| \nabla_z \log \left(\frac{q^m(z)}{\tilde{p}_{\theta,\phi}(x^m, z)} \right) \right\|^2 q^m(\mathrm{d}z). \end{aligned} \tag{80}$$

Using the functional $I(\theta, \phi, q^{1:M})$ defined in (80), we can state the following extension of de Bruijn's identity:

Proposition C.2 (de Bruijn's Identity)**.** *Under Assumption **A**2, we have:*

$$\frac{\mathrm{d}}{\mathrm{d}t} \tilde{F}(\theta_t, \phi_t, q_t^{1:M}) = -I(\theta_t, \phi_t, q_t^{1:M}), \qquad \forall t > 0. \tag{81}$$

Proof. By definition,

$$\tilde{F}(\theta_t, \phi_t, q_t^{1:M}) = \frac{1}{M} \sum_{m=1}^{M} \left[\int \log \left(\frac{q_t^m(z)}{\tilde{p}_{\theta_t,\phi_t}(x^m, z)} \right) q_t^m(\mathrm{d}z) \right].$$

Under Assumption **A**2, we can differentiate under the integral sign and the flow satisfies

$$\begin{aligned} \frac{\partial_t q_t^m(z)}{\partial t} &= \nabla_z \cdot \left[q_t^m(z) \nabla_z \log \left(\frac{q_t^m(z)}{\tilde{p}_{\theta_t,\phi_t}(x^m, z)} \right) \right], \\ (\dot{\theta}_t, \dot{\phi}_t) &= \frac{1}{M} \sum_{m=1}^{M} \int \nabla_{(\theta,\phi)} \log \tilde{p}_{\theta_t,\phi_t}(x^m, z)\, q_t^m(\mathrm{d}z). \end{aligned}$$

Using integration by parts, for each m we obtain

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} \int \log(q_t^m(z)) q_t^m(\mathrm{d}z) &= \int (\log(q_t^m(z)) + 1) \frac{\partial_t q_t^m(z)}{\partial t} \mathrm{d}z \\ &= -\int \left\langle \nabla_z \log(q_t^m(z)), \nabla_z \log \left(\frac{q_t^m(z)}{\tilde{p}_{\theta_t,\phi_t}(x^m, z)} \right) \right\rangle q_t^m(\mathrm{d}z) \end{aligned}$$

and

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} \int \log(\tilde{p}_{\theta_t,\phi_t}(x^m, z)) q_t^m(\mathrm{d}z) &= \int \left\langle \nabla_{(\theta,\phi)} \log \tilde{p}_{\theta_t,\phi_t}(x^m, z), (\dot{\theta}_t, \dot{\phi}_t) \right\rangle q_t^m(\mathrm{d}z) \\ &\quad - \int \left\langle \nabla_z \log(\tilde{p}_{\theta_t,\phi_t}(x^m, z)), \nabla_z \log \left(\frac{q_t^m(z)}{\tilde{p}_{\theta_t,\phi_t}(x^m, z)} \right) \right\rangle q_t^m(\mathrm{d}z). \end{aligned}$$

Therefore,

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} \tilde{F}(\theta_t, \phi_t, q_t^{1:M}) &= -\|(\dot{\theta}_t, \dot{\phi}_t)\|^2 - \frac{1}{M} \sum_{m=1}^{M} \int \left\| \nabla_z \log \left(\frac{q_t^m(z)}{\tilde{p}_{\theta_t,\phi_t}(x^m, z)} \right) \right\|^2 q_t^m(\mathrm{d}z) \\ &= -I(\theta_t, \phi_t, q_t^{1:M}), \end{aligned}$$

where the last equality follows from the definition of I in (80). □

We need a few auxiliary results that are extensions of those in Caprio et al. (2025).

Lemma C.3 (Geodesics on $\mathcal{M}^{1:M}$)**.** *A curve $\gamma(t) : t \in [0,1] \mapsto \mathcal{M}^{1:M}$ is a geodesic if and only if $\gamma(t) = (\gamma_\theta(t), \gamma_\phi(t), \gamma_{q^1}(t), \ldots, \gamma_{q^M}(t))$, where γ_θ and γ_ϕ are geodesics in the Euclidean parameter space and each γ_{q^m} is a geodesic in $(\mathcal{P}_2(\mathsf{Z}), \mathrm{d}_{W_2})$. In particular, if $\gamma(t)$ is a geodesic in $\mathcal{M}^{1:M}$ connecting $(\theta, \phi, q^{1:M})$ and $(\theta', \phi', (q')^{1:M})$ then*

$$\begin{aligned} \gamma_\theta(t) &= (1-t)\theta + t\theta', \\ \gamma_\phi(t) &= (1-t)\phi + t\phi', \\ \gamma_{q^m}(t) &= (h_t)_{\#} \varrho^m, \quad m = 1, \ldots, M, \end{aligned}$$

where ϱ^m's are Wasserstein-2 optimal transport plans for $(q^m, (q')^m)$ and $h_t(z, z') = (1-t)z + tz'$. Furthermore, if q^m's have densities with respect to the Lebesgue measure, then we can also write:

$$\gamma_{q^m}(t) = ((1-t)\mathrm{id} + t\nabla_z f^m)_{\#} q^m, \quad \forall m \in [M],$$

for some convex function f^m.

Proof. The first claim follows from the definition of the product metric (78). The second claim is a result of the characterization of geodesics (cf. Santambrogio (2015, Theorem 5.27)). The last claim follows from Brenier's theorem (cf. Santambrogio (2015, Theorem 1.17)). □

Lemma C.4 (Geodesic convexity of $\tilde{F}$)**.** *Under Assumption **A**3, the tilted free energy $\tilde{F}(\theta, \phi, q^{1:M})$ is λ-geodesically convex on $\mathcal{M}^{1:M}$, that is, for any pair $(\theta, \phi, q^{1:M})$ and $(\theta', \phi', (q')^{1:M})$ in $\mathcal{M}^{1:M}$ and any geodesic $\gamma(t)$ connecting them, we have:*

$$\tilde{F}(\gamma(t)) \leq (1-t)\tilde{F}(\theta, \phi, q^{1:M}) + t\tilde{F}(\theta', \phi', (q')^{1:M}) - \frac{\lambda t(1-t)}{2} \mathsf{d}_{\mathcal{M}^{1:M}}((\theta, \phi, q^{1:M}), (\theta', \phi', (q')^{1:M}))^2.$$

Proof. First recall that $\tilde{F}(\theta, \phi, q^{1:M}) = M^{-1} \sum_{m=1}^M \mathbb{E}_{q^m(z)}[\log(q^m(z)) - \tilde{\ell}^m(\theta, \phi, z)]$. We note that the negative entropy $q^m \mapsto \int \log(q^m(z)) q^m(\mathrm{d}z)$ is geodesically convex on $\mathcal{P}_2(\mathsf{Z})$ (cf. Santambrogio (2015, Theorem 7.28)) and thus the average $q^{1:M} \mapsto M^{-1} \sum_{m=1}^M \int \log(q^m(z)) q^m(\mathrm{d}z)$ is geodesically convex on $\mathcal{P}_2(\mathsf{Z})^M$. By an argument similar to Caprio et al. (2025, Lemma 21), we can show the map $V : (\theta, \phi, z) \mapsto -M^{-1} \sum_{m=1}^M \int \tilde{\ell}^m(\theta, \phi, z) q^m(\mathrm{d}z)$ is λ-strongly convex along the geodesic $\gamma(t)$:

$$\begin{aligned}
V(\gamma(t)) &= -\int \frac{1}{M} \sum_{m=1}^M \tilde{\ell}^m(\gamma_\theta(t), \gamma_\phi(t), z) \gamma_{q^m}(t)(\mathrm{d}z) \\
&= -\int \frac{1}{M} \sum_{m=1}^M \tilde{\ell}^m((1-t)\theta + t\theta', (1-t)\phi + t\phi', (1-t)z + tz') \varrho^m(\mathrm{d}z, \mathrm{d}z') \\
&\leq -\int \frac{1}{M} \sum_{m=1}^M (1-t)\tilde{\ell}^m(\theta, \phi, z) + t\tilde{\ell}^m(\theta', \phi', z') \varrho^m(\mathrm{d}z, \mathrm{d}z') \\
&\quad - \frac{\lambda t(1-t)}{2} \int \frac{1}{M} \sum_{m=1}^M \|(\theta, \phi, z) - (\theta', \phi', z')\|^2 \varrho^m(\mathrm{d}z, \mathrm{d}z') \\
&= (1-t)V(\theta, \phi, q^{1:M}) + tV(\theta', \phi', (q')^{1:M}) - \frac{\lambda t(1-t)}{2} \mathsf{d}_{\mathcal{M}^{1:M}}((\theta, \phi, q^{1:M}), (\theta', \phi', (q')^{1:M}))^2,
\end{aligned}$$

where we have used Lemma C.3 in the second equality and the strong log-concavity in the inequality. The conclusion follows from the definition of the product metric (78). □

We similarly provide an extension to Caprio et al. (2025, Lemma 22):

Lemma C.5. *Let $\gamma(t)$ be a geodesic in $\mathcal{M}^{1:M,1}$ connecting $(\theta, \phi, q^{1:M})$ and $(\theta', \phi', (q')^{1:M})$. Then we have:*

$$\liminf_{t \to 0^+} \frac{\tilde{F}(\gamma(t)) - \tilde{F}(\gamma(0))}{t} \geq \frac{1}{M} \sum_{m=1}^M \left\langle \dot{\gamma}_{q^m}(0), \nabla_{q^m} \tilde{F}(\theta, \phi, q^m) \right\rangle_{q^m} + \langle (\theta', \phi') - (\theta, \phi), \nabla_{(\theta,\phi)} \tilde{F}(\theta, \phi, q^{1:M}) \rangle$$

Proof. The proof is an adaptation of Caprio et al. (2025, Lemma 22) by noting that the computation can be done separately for each m. □

Theorem C.6 (Strong log-concavity $\implies$ xLSI)**.** *Suppose Assumptions **A**1 and **A**3 hold, then the family of measures $(\tilde{\rho}_{\theta,\phi}(\mathrm{d}z^{1:M}))_{(\theta,\phi) \in \Theta \times \Phi}$ satisfies the extended log-Sobolev inequality with constant $\lambda > 0$.*

Proof. The proof extends that of Caprio et al. (2025, Theorem 6) to the product manifold $\Theta \times \Phi \times \mathcal{P}_2(\mathsf{Z})^M$. Let $\gamma(t)$ be a geodesic in $\mathcal{M}^{1:M}$ connecting $(\theta, \phi, q^{1:M})$ and $(\theta', \phi', (q')^{1:M})$. By Lemma C.4, we have:

$$\liminf_{t \to 0^+} \frac{\tilde{F}(\gamma(t)) - \tilde{F}(\gamma(0))}{t} \leq \tilde{F}(\theta', \phi', (q')^{1:M}) - \tilde{F}(\theta, \phi, q^{1:M}) - \frac{\lambda}{2} \mathsf{d}_{\mathcal{M}^{1:M}}((\theta, \phi, q^{1:M}), (\theta', \phi', (q')^{1:M}))^2.$$

Setting $(\theta', \phi', (q')^{1:M}) = (\theta_\star, \phi_\star, (q'_\star)^{1:M})$ to be a minimizer of $\tilde{F}$ and using the previous Lemma C.5, we obtain:

$$\begin{aligned}
\tilde{F}(\theta, \phi, q^{1:M}) - \tilde{F}_\star &\leq -\frac{1}{M} \sum_{m=1}^M \left\langle \dot{\gamma}_{q^m}(0), \nabla_q \tilde{F}(\theta, \phi, q^m) \right\rangle_{q^m} - \langle (\theta_\star, \phi_\star) - (\theta, \phi), \nabla_{(\theta,\phi)} \tilde{F}(\theta, \phi, q^{1:M}) \rangle \\
&\quad - \frac{\lambda}{2} \mathsf{d}_{\mathcal{M}^{1:M}}((\theta, \phi, q^{1:M}), (\theta_\star, \phi_\star, q_\star^{1:M}))^2 \\
&\leq \frac{1}{M} \sum_{m=1}^M \mathsf{d}_{W_2}(q^m, q_\star^m) \left\| \nabla_q \tilde{F}(\theta, \phi, q^m) \right\|_{q^m} + \|(\theta_\star, \phi_\star) - (\theta, \phi)\| \|\nabla_{(\theta,\phi)} \tilde{F}(\theta, \phi, q^{1:M})\| \\
&\quad - \frac{\lambda}{2} \mathsf{d}_{\mathcal{M}^{1:M}}((\theta, \phi, q^{1:M}), (\theta_\star, \phi_\star, q_\star^{1:M}))^2,
\end{aligned}$$

where we have used the definition of the inner product on the tangent space $T_q \mathcal{P}_2^1(\mathsf{Z})$ and Lemma C.3:

$$\|\dot{\gamma}_{q^m}(0)\|_{q^m} = \|\nabla_z f^m - \mathrm{id}\|_{L^2(q^m)} = \mathsf{d}_{W_2}(q^m, q_\star^m),$$

and the Cauchy-Schwarz inequality. One more application of the Cauchy-Schwarz inequality yields:

$$\begin{aligned}
\tilde{F}(\theta, \phi, q^{1:M}) - \tilde{F}_\star &\leq \left[\frac{1}{M} \sum_{m=1}^M \mathsf{d}_{W_2}(q^m, q_\star^m)^2 + \|(\theta, \phi) - (\theta_\star, \phi_\star)\|^2 \right]^{1/2} \\
&\quad \times \left[\frac{1}{M} \sum_{m=1}^M \left\| \nabla_q \tilde{F}(\theta, \phi, q^m) \right\|_{q^m}^2 + \|\nabla_{(\theta,\phi)} \tilde{F}(\theta, \phi, q^{1:M})\|^2 \right]^{1/2} \\
&\quad - \frac{\lambda}{2} \mathsf{d}_{\mathcal{M}^{1:M}}((\theta, \phi, q^{1:M}), (\theta_\star, \phi_\star, q_\star^{1:M}))^2 \\
&= \mathsf{d}_{\mathcal{M}^{1:M}}((\theta, \phi, q^{1:M}), (\theta_\star, \phi_\star, q_\star^{1:M})) \sqrt{I(\theta, \phi, q^{1:M})} - \frac{\lambda}{2} \mathsf{d}_{\mathcal{M}^{1:M}}((\theta, \phi, q^{1:M}), (\theta_\star, \phi_\star, q_\star^{1:M}))^2 \\
&\leq \frac{1}{2\lambda} I(\theta, \phi, q^{1:M}),
\end{aligned}$$

where we have upper bounded the first term using Young's inequality $ab \leq a^2/(2\lambda) + \lambda b^2/2$ in the last step. □

C.1.2 Extended Talagrand-type Inequality

We now show that xLSI implies an extended Talagrand-type inequality. Throughout this section, we assume Assumptions **A**1 and **A**2 hold. We first provide the M-datapoint version of extended Talagrand in Caprio et al. (2025).

Definition C.7 (Extended Talagrand-type Inequality)**.** The family of measures $(\tilde{\rho}_{\theta,\phi}(\mathrm{d}z^{1:M}))_{(\theta,\phi) \in \Theta \times \Phi}$ satisfy the extended Talagrand-type inequality with constant $\lambda > 0$ if for all $(\theta, \phi, q^{1:M}) \in \mathcal{M}^{1:M}$:

$$2[\tilde{F}(\theta, \phi, q^{1:M}) - \tilde{F}_\star] \geq \lambda \inf_{(\theta,\phi,q^{1:M}) \in \mathcal{M}^{1:M}} \mathsf{d}((\theta, \phi, q^{1:M}), \mathcal{M}_\star^{1:M})^2 \tag{82}$$

where $\mathcal{M}_\star^{1:M} := \arg\min_{(\theta,\phi,q^{1:M}) \in \mathcal{M}^{1:M}} \tilde{F}(\theta, \phi, q^{1:M})$ is the optimal set of $\tilde{F}$.

The proof hinges on a few auxiliary results adapted from Caprio et al. (2025).

Lemma C.8. *Under Assumption **A**2, we have:*

$$\frac{\mathrm{d}}{\mathrm{d}t} \mathsf{d}_{\mathcal{M}^{1:M}}((\theta_t, \phi_t, q_t^{1:M}), (\theta, \phi, q^{1:M})) \leq \sqrt{I(\theta_t, \phi_t, q_t^{1:M})}, \quad \forall t > 0,$$

where $I(\theta, \phi, q^{1:M})$ is defined in (80).

Proof. For $(\theta, \phi, q^{1:M}) \in \mathcal{M}^{1:M}$, we define the velocity field $v_t^{1:M} = (v_t^1, \dots, v_t^M)$ as:

$$v_t^m(z) = -\nabla_z \log\left(\frac{q_t^m(z)}{\tilde{\rho}^m_{\theta_t,\phi_t}(z)}\right), \quad \forall m \in [M].$$

Using the proof to Caprio et al. (2025, Lemma 16), for each m, we have:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathsf{d}_{W_2}(q_t^m, q^m)^2 \leq 2\mathsf{d}_{W_2}(q_t^m, q^m)\sqrt{\int \|v_t^m(z)\|^2 q_t^m(\mathrm{d}z)}.$$

By the Cauchy-Schwarz inequality, we have:

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{1}{M}\sum_{m=1}^M \mathsf{d}_{W_2}(q_t^m, q^m)^2 \leq \frac{2}{M}\sum_{m=1}^M \mathsf{d}_{W_2}(q_t^m, q^m)\sqrt{\int \|v_t^m(z)\|^2 q_t^m(\mathrm{d}z)}.$$

Combining this with the definition of the Euclidean counterpart:

$$\frac{\mathrm{d}}{\mathrm{d}t}\|(\theta_t, \phi_t) - (\theta, \phi)\|^2 = 2\left\langle \frac{\mathrm{d}}{\mathrm{d}t}(\theta_t, \phi_t), (\theta_t, \phi_t) - (\theta, \phi)\right\rangle \leq 2\left\|\nabla_{(\theta,\phi)}\tilde{F}(\theta_t, \phi_t, q_t^{1:M})\right\| \|(\theta_t, \phi_t) - (\theta, \phi)\|,$$

we obtain with an application of the Cauchy-Schwarz inequality similar to Theorem C.6:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\frac{1}{2}\mathsf{d}_{\mathcal{M}^{1:M}}((\theta_t, \phi_t, q_t^{1:M}), (\theta, \phi, q^{1:M}))^2 &\leq \left(\frac{1}{M}\sum_{m=1}^M \mathsf{d}_{W_2}(q_t^m, q^m)^2\right)^{1/2}\left(\frac{1}{M}\sum_{m=1}^M \int \|v_t^m(z)\|^2 q_t^m(\mathrm{d}z)\right)^{1/2} \\ &\quad + \|(\theta_t, \phi_t) - (\theta, \phi)\|\|\nabla_{(\theta,\phi)}\tilde{F}(\theta_t, \phi_t, q_t^{1:M})\| \\ &\leq \sqrt{I(\theta_t, \phi_t, q_t^{1:M})}\mathsf{d}_{\mathcal{M}^{1:M}}((\theta_t, \phi_t, q_t^{1:M}), (\theta, \phi, q^{1:M})),\end{aligned}$$

which implies the conclusion. □

Lemma C.9. *The tilted free energy $\tilde{F}(\theta, \phi, q^{1:M})$ is lower semi-continuous on $\mathcal{M}^{1:M,1}$.*

Proof. We can re-write $\tilde{F}(\theta, \phi, q^{1:M}) = M^{-1}\sum_{m=1}^M F(\theta, \phi, q^m) + \mathbb{E}_{q^m}[\mathcal{R}(\theta, z)]$, where we recall $\mathcal{R}(\theta, z_0) = D_{\mathrm{KL}}\big(q(z_{1:K}|z_0)\|p_\theta(z_{1:K}|z_0)\big) \geq 0$. We note that it suffices to show that $U : (\theta, q) \mapsto \mathbb{E}_q[\mathcal{R}(\theta, z)] = \int \mathcal{R}(\theta, z)q(\mathrm{d}z)$ is lower semi-continuous on $\Theta \times \mathcal{P}_2(\mathsf{Z})$ (hence $\Theta \times \mathcal{P}_2^1(\mathsf{Z})$), since Caprio et al. (2025, Lemma 18) shows that the untilted free energy $F(\theta, \phi, q)$ is lower semi-continuous on $\mathcal{M}^1$. The conclusion thus follows from the definition of the space $(\mathcal{M}^{1:M,1}, \mathsf{d}_{\mathcal{M}^{1:M}})$.

To show U is lower semi-continuous, let $\{(\theta_n, q_n)\}_{n\in\mathbb{N}}$ be a sequence in $\Theta \times \mathcal{P}_2(\mathsf{Z})$ converging to some $(\theta_\infty, q_\infty) \in \Theta \times \mathcal{P}_2(\mathsf{Z})$ in the topology induced by $\mathsf{d}_{\mathcal{M}}$. The convergence in d_{W_2} implies weak convergence of $q_n \rightharpoonup q_\infty$ (Figalli and Glaudo, 2021) and we have $\theta_n \to \theta_\infty$. Since $\mathcal{R}$ is non-negative and continuous in both arguments, by applying Portmanteau theorem on the product measures $\mu_n := \delta_{\theta_n}(\mathrm{d}\theta) \otimes q_n(\mathrm{d}z)$, we have:

$$\begin{aligned}\liminf_{n\to\infty} U(\theta_n, q_n) &= \liminf_{n\to\infty} \int \mathcal{R}(\theta_n, z)\mu_n(\mathrm{d}\theta, \mathrm{d}z) \\ &\geq \int \mathcal{R}(\theta, z)\mu_\infty(\mathrm{d}\theta, \mathrm{d}z) = U(\theta_\infty, q_\infty),\end{aligned}$$

which concludes the proof. □

Lemma C.10. *Denote $\mathcal{M}_\star := \arg\min_{(\theta,\phi,q^{1:M})\in\mathcal{M}^{1:M}} \tilde{F}(\theta, \phi, q^{1:M})$ as the set of minimizers of $\tilde{F}$ and $\tilde{F}_\star := \inf_{(\theta,\phi,q^{1:M})\in\mathcal{M}^{1:M}} \tilde{F}(\theta, \phi, q^{1:M})$ as the optimal value. Under the extended log-Sobolev inequality, for a Cauchy sequence $\{(\theta_n, \phi_n, q_n^{1:M})\}_{n\in\mathbb{N}}$ in $\mathcal{M}^{1:M,1}$, there exists an increasing sequence of $t_n \to +\infty$ such that:*

$$(\theta_{t_n}, \phi_{t_n}, q_{t_n}^{1:M}) \to (\theta_\star, \phi_\star, q_\star^{1:M}) \in \mathcal{M}_\star, \quad \text{as } n \to +\infty,$$

for some $(\theta_\star, \phi_\star, q_\star^{1:M}) \in \mathcal{M}_\star$ in the topology induced by $\mathsf{d}_{\mathcal{M}^{1:M}}$ on $\mathcal{M}^{1:M}$.

Proof. Since both the Euclidean space and $\mathcal{P}_2(\mathsf{Z})$ are complete metric spaces (Villani et al., 2008, Theorem 6.18), the product space $(\mathcal{M}^{1:M}, \mathsf{d}_{\mathcal{M}^{1:M}})$ is also complete. Therefore, the sequence $(\theta_\infty, \phi_\infty, q_\infty^{1:M})$ is also Cauchy in $\mathcal{M}^{1:M}$ and converges to some limit $(\theta_\infty, \phi_\infty, q_\infty^{1:M}) \in \mathcal{M}^{1:M}$. By the lower semi-continuity of $\tilde{F}$, we have:

$$\tilde{F}_\star \leq \tilde{F}(\theta_\infty, \phi_\infty, q_\infty^{1:M}) \leq \liminf_{n\to\infty} \tilde{F}(\theta_{t_n}, \phi_{t_n}, q_{t_n}^{1:M}) = \tilde{F}_\star,$$

where the first inequality is by the optimality of $\tilde{F}_\star$ and the equality follows from the fact the exponential convergence induced by xLSI (cf. the rightmost inequality of Theorem C.12 below). □

Theorem C.11 (xLSI $\implies$ Extended Talagrand-type Inequality ($\mathrm{xT_2I}$))**.** *Suppose the family of measures $(\tilde{\rho}_{\theta,\phi}(\mathrm{d}z^{1:M}))_{(\theta,\phi)\in\Theta\times\Phi}$ satisfies the extended log-Sobolev inequality with constant $\lambda > 0$. Then it also satisfies the extended Talagrand-type inequality with the same constant $\lambda > 0$.*

Proof. Using Lemma C.8 and the xLSI, an argument similar to that in Caprio et al. (2025, Theorem 4) yields the following sequence of inequalities for any $(\theta, \phi, q^{1:M}) \in \mathcal{M}^{1:M}$:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathsf{d}_{\mathcal{M}^{1:M}}((\theta_t, \phi_t, q_t^{1:M}), (\theta, \phi, q^{1:M})) \leq \sqrt{I(\theta_t, \phi_t, q_t^{1:M})} \leq \frac{I(\theta_t, \phi_t, q_t^{1:M})}{\sqrt{2\lambda[\tilde{F}(\theta_t, \phi_t, q_t^{1:M}) - \tilde{F}_\star]}}.$$

Using de Bruijn's identity (81), we have:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathsf{d}_{\mathcal{M}^{1:M}}((\theta_t, \phi_t, q_t^{1:M}), (\theta, \phi, q^{1:M})) \leq -\frac{\mathrm{d}}{\mathrm{d}t}\sqrt{\frac{2[\tilde{F}(\theta_t, \phi_t, q_t^{1:M}) - \tilde{F}_\star]}{\lambda}}.$$

For any interval (t, t'), integrating yields:

$$\begin{aligned}&\mathsf{d}_{\mathcal{M}^{1:M}}((\theta_{t'}, \phi_{t'}, q_{t'}^{1:M}), (\theta, \phi, q^{1:M})) - \mathsf{d}_{\mathcal{M}^{1:M}}((\theta_t, \phi_t, q_t^{1:M}), (\theta, \phi, q^{1:M})) \\ &\leq \sqrt{\frac{2[\tilde{F}(\theta_t, \phi_t, q_t^{1:M}) - \tilde{F}_\star]}{\lambda}} - \sqrt{\frac{2[\tilde{F}(\theta_{t'}, \phi_{t'}, q_{t'}^{1:M}) - \tilde{F}_\star]}{\lambda}}. \end{aligned} \tag{83}$$

We can thus construct a Cauchy sequence $\{(\theta_{t_n}, \phi_{t_n}, q_{t_n}^{1:M})\}_{n\in\mathbb{N}}$ in $\mathcal{M}^{1:M,1}$ from (83) for some increasing sequence $t_n \to +\infty$. By the previous Lemma C.10, we have the limit point $(\theta_\infty, \phi_\infty, q_\infty^{1:M}) \in \mathcal{M}_\star$. Setting $(t, t') = (0, t_n)$ in (83) and letting $n \to +\infty$, we have:

$$\mathsf{d}_{\mathcal{M}^{1:M}}((\theta_0, \phi_0, q_0^{1:M}), (\theta_\infty, \phi_\infty, q_\infty^{1:M})) \leq \sqrt{\frac{2[\tilde{F}(\theta_0, \phi_0, q_0^{1:M}) - \tilde{F}_\star]}{\lambda}},$$

whence the conclusion follows by noting the distance function is continuous and the infimum is attained. □

We now state the full-version of Theorem 3.1.

Theorem C.12. *Suppose Assumptions **A**1-**A**3 are satisfied. Then, $\tilde{\ell}$ has a unique maximizer $(\theta_\star, \phi_\star)$ and the flow converges exponentially fast to it: for some $\lambda > 0$ independent of M,*

$$\mathsf{d}_{\mathcal{M}^{1:M}}((\theta_t, \phi_t, q_t^{1:M}), (\theta_\star, \phi_\star, q_\star^{1:M})) \leq \sqrt{\frac{2[\tilde{F}(\theta_t, \phi_t, q_t^{1:M}) - \tilde{F}_\star]}{\lambda}} \leq \sqrt{\frac{2[\tilde{F}(\theta_0, \phi_0, q_0^{1:M}) - \tilde{F}_\star]}{\lambda}} e^{-\lambda t}, \quad \forall t > 0; \tag{84}$$

where $\tilde{F}_\star := \inf_{(\theta,\phi,q^{1:M})\in\mathcal{M}^{1:M}} \tilde{F}(\theta, \phi, q^{1:M})$ and $||\cdot||$ denotes the Euclidean norm.

Proof. Under Assumption **A**3, each map $(\theta, \phi, z) \mapsto \log \tilde{p}_{\theta,\phi}(x^m, z)$ is λ-strongly concave. Hence the product density

$$\bar{p}_{\theta,\phi}(x^{1:M}, z^{1:M}) := \prod_{m=1}^{M} \tilde{p}_{\theta,\phi}(x^m, z^m)$$

has joint log-density

$$\bar{\ell}(\theta, \phi, z^{1:M}) := \log \bar{p}_{\theta,\phi}(x^{1:M}, z^{1:M}) = \sum_{m=1}^{M} \log \tilde{p}_{\theta,\phi}(x^m, z^m),$$

which is strictly concave in $((\theta, \phi), z^{1:M})$. Applying Kuntz et al. (2023, Theorem 4) with parameter $u = (\theta, \phi)$ and latent variable $z^{1:M}$ yields that the marginal log-likelihood

$$\bar{\ell}(\theta, \phi) := \log \int \bar{p}_{\theta,\phi}(x^{1:M}, z^{1:M}) \mathrm{d}z^{1:M} = \sum_{m=1}^{M} \log \tilde{p}_{\theta,\phi}(x^m)$$

has a unique maximizer. Since $\tilde{\ell}(\theta, \phi) = M^{-1}\bar{\ell}(\theta, \phi)$, the same pair $(\theta_\star, \phi_\star)$ is the unique maximizer of $\tilde{\ell}$.

Under the log-concavity assumption, the family of distributions $(\tilde{\rho}_{\theta,\phi}(\mathrm{d}z^{1:M}))_{(\theta,\phi)\in\Theta\times\Phi}$ satisfies the extended log-Sobolev inequality (cf. Definition C.1) using Theorem C.6.

The rightmost inequality in (84) then follows from a combination of de Bruijn's identity (81), the extended log-Sobolev inequality, and Grönwall's lemma.

The leftmost inequality follows from a combination of the uniquness of minimizers and Theorem C.11. □

Remark C.13. Since $\tilde{\ell}$ has a unique maximizer $(\theta_\star, \phi_\star)$ and Proposition 2.1 identifies the corresponding minimizer of $\tilde{F}$, we can write $\tilde{F}_\star = \tilde{F}(\theta_\star, \phi_\star, \tilde{\pi}^{1:M}_{\theta_\star,\phi_\star})$ and $q^{1:M}_\star = \tilde{\pi}^{1:M}_{\theta_\star,\phi_\star}$, where we recall $\tilde{\pi}_{\theta,\phi}(\cdot) = \tilde{p}_{\theta,\phi}(\cdot|x)$ is the true posterior distribution.

C.2 Proof of Theorem 3.2

We now provide the proof to Theorem 3.2. The proof will be based on the spatial and temporal discretization error bounds established in Caprio et al. (2025), which are combined with the exponential convergence result in Theorem C.12. In what follows, we will use the notation

$$\tilde{\theta} := (\theta, \phi)$$

s to denote the concatenation of the diffusion prior and decoder parameters as in Appendix A.3.

First, we recall the additional assumption.

Assumption 4 (Lipschitz gradient). *The log-likelihood $\tilde{\ell}^m(\tilde{\theta}, z) := \log p_{\tilde{\theta}}(x^m, z)$ is differentiable and its gradient $\nabla\tilde{\ell}^m := (\nabla_{\tilde{\theta}}\tilde{\ell}^m, \nabla_{z_0}\tilde{\ell}^m)$ is L-Lipschitz for some $L > 0$, that is, for all $(\tilde{\theta}, z), (\tilde{\theta}', z') \in (\Theta \times \Phi) \times \mathsf{Z}$:*

$$\|\nabla\tilde{\ell}^m(\tilde{\theta}, z) - \nabla\tilde{\ell}^m(\tilde{\theta}', z')\| \le L\|(\tilde{\theta}, z) - (\tilde{\theta}', z')\|.$$

We now consider the continuous-time system of SDEs that correspond to the gradient flow:

$$\mathrm{d}\tilde{\theta}_t = \nabla_{\tilde{\theta}} \int \frac{1}{MN} \sum_{m=1}^{M} \sum_{n=1}^{N} \tilde{\ell}^m(\tilde{\theta}_t, z) q^{m,n}(\mathrm{d}z)\, \mathrm{d}t \tag{85}$$

$$\mathrm{d}Z^{m,n}_t = \nabla_z \tilde{\ell}^m(\tilde{\theta}_t, Z^{m,n}_t)\, \mathrm{d}t + \sqrt{2}\, \mathrm{d}W^{m,n}_t, \quad (m, n) \in [M] \times [N] \tag{86}$$

where $q^{m,n}_t := \mathrm{Law}(Z^{m,n}_t)$ is the law of the particles at time t. Since the $Z^{m,n}_t$ are i.i.d., we note that the first equation is equivalent to $\mathrm{d}\tilde{\theta} = \nabla_{\tilde{\theta}} \int M^{-1} \sum_{m=1}^{M} \tilde{\ell}^m(\tilde{\theta}_t, z) q^m_t(\mathrm{d}z)\, \mathrm{d}t$, where q^m_t is the law of $Z^{m,n}_t$ for any n.

Lemma C.14. *Under Assumptions **A**1, **A**3, and **A**4, the system of SDEs in (85)-(86) admits a unique strong solution $(\tilde{\theta}_t, Z^{1:M,1:N}_t)_{t\ge 0}$ and the solution satisfies*

$$\mathsf{d}_{\mathcal{M}^{1:M}}((\tilde{\theta}_t, Q^{1:M,N}_t), (\tilde{\theta}_\star, Q^{1:M,N}_\star)) \le Ce^{-\lambda t}, \quad \forall t > 0,$$

where $C > 0$ is a constant independent of N and t, and we defined the empirical distributions to q^m_t and $q^m_\star$ as $Q^{m,N}_t := N^{-1} \sum_{n=1}^{N} \delta_{Z^{m,n}_t}$ and $Q^{m,N}_\star := N^{-1} \sum_{n=1}^{N} \delta_{Z^{m,n}_\star}$ respectively, where $Z^{m,n}_\star$ are i.i.d. samples from $\tilde{\pi}_{\tilde{\theta}_\star}$ (cf. Remark C.13).

Proof. The existence and uniqueness of a strong solution follows from Proposition 8 in Caprio et al. (2025) by replacing the system of SDEs therein with ours. To prove the inequality, we note that by a coupling argument for each pair $(Z_t^{m,n}, Z_\star^{m,n})$, we have:

$$\begin{aligned}\mathsf{d}_{\mathcal{M}^{1:M}}((\tilde{\theta}_t, Q_t^{1:M,N}), (\tilde{\theta}_\star, Q_\star^{1:M,N}))^2 &\leq \|\tilde{\theta}_t - \tilde{\theta}_\star\|^2 + \frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}\mathbb{E}[\|Z_t^{m,n} - Z_\star^{m,n}\|^2] \\ &= \|\tilde{\theta}_t - \tilde{\theta}_\star\|^2 + \frac{1}{M}\sum_{m=1}^{M}\mathbb{E}[\|Z_t^{m,1} - Z_\star^{m,1}\|^2] \\ &= \mathsf{d}_{\mathcal{M}^{1:M}}((\tilde{\theta}_t, q_t^{1:M}), (\tilde{\theta}_\star, q_\star^{1:M}))^2 \leq \frac{2e^{-2\lambda t}}{\lambda}[\tilde{F}(\tilde{\theta}_0, q_0) - \tilde{F}_\star],\end{aligned}$$

where we used Theorem C.12 in the last inequality. □

Lemma C.15 (Spatial Discretization Error)**.** *Suppose Assumptions **A**3 and **A**4 hold. Then the following system of SDEs:*

$$\mathrm{d}\tilde{\theta}_t^N = \frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}\nabla_{\tilde{\theta}}\tilde{\ell}^m(\tilde{\theta}_t^N, \bar{Z}_t^{m,n})\,\mathrm{d}t \tag{87}$$

$$\mathrm{d}\bar{Z}_t^{m,n} = \nabla_z\tilde{\ell}^m(\tilde{\theta}_t^N, \bar{Z}_t^{m,n})\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}W_t^{m,n}, \quad (m,n) \in [M]\times[N], \tag{88}$$

has a strong solution $(\tilde{\theta}_t^N, \bar{Z}_t^{1:M,1:N})_{t\geq 0}$. Furthermore, there exists a constant $C(N) > 0$ of order $\mathcal{O}(N^{-1/2})$ independent of t and M, such that:

$$\mathsf{d}_{\mathcal{M}^{1:M}}((\tilde{\theta}_t^N, \bar{Q}_t^{1:M,N}), (\tilde{\theta}_t, Q_t^{1:M,N})) \leq C(N), \quad \forall t > 0,$$

where $\bar{Q}_t^{m,N} := N^{-1}\sum_{n=1}^{N}\delta_{\bar{Z}_t^{m,n}}$ for each $m \in [M]$ and $Q_t^{m,N}$ is defined as in Lemma C.14.

Proof. The proof is a modification of Caprio et al. (2025, Lemma 13) by replacing the SDEs therein with (87)-(88). More specifically, we define the quantity ξ_t^N as:

$$\xi_t^N := \|\tilde{\theta}_t^N - \tilde{\theta}_t\|^2 + \frac{1}{M}\sum_{m=1}^{M}\frac{1}{N}\sum_{n=1}^{N}\|Z_t^{m,n} - \bar{Z}_t^{m,n}\|^2, \tag{89}$$

and we will show that $\mathbb{E}[\xi_t^N] \leq C(N)$ for some constant $C(N)$ which upper bounds the spatial discretization error. Using Itô's formula:

$$\mathrm{d}\|\tilde{\theta}_t^N - \tilde{\theta}_t\|^2 = 2\left\langle \tilde{\theta}_t^N - \tilde{\theta}_t, \frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}\nabla_{\tilde{\theta}}\tilde{\ell}^m(\tilde{\theta}_t^N, \bar{Z}_t^{m,n}) - \nabla_{\tilde{\theta}}\int\frac{1}{M}\sum_{m=1}^{M}\tilde{\ell}^m(\tilde{\theta}_t, z)q_t^m(\mathrm{d}z)\right\rangle \mathrm{d}t, \tag{90}$$

$$\mathrm{d}\|\bar{Z}_t^{m,n} - Z_t^{m,n}\|^2 = 2\left\langle \bar{Z}_t^{m,n} - Z_t^{m,n}, \nabla_z\tilde{\ell}^m(\tilde{\theta}_t^N, \bar{Z}_t^{m,n}) - \nabla_z\tilde{\ell}^m(\tilde{\theta}_t, Z_t^{m,n})\right\rangle \mathrm{d}t. \tag{91}$$

Adding and subtracting $(MN)^{-1}\sum_m\sum_n\nabla_{\tilde{\theta}}\tilde{\ell}^m(\tilde{\theta}_t, Z_t^{m,n})$ from (90) and summing with (91) scaled by $(MN)^{-1}$ yields:

$$\begin{aligned}\mathrm{d}\xi_t^N = \frac{2}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}2\big[&\left\langle \tilde{\theta}_t^N - \tilde{\theta}_t, \nabla_{\tilde{\theta}}\tilde{\ell}^m(\tilde{\theta}_t^N, \bar{Z}_t^{m,n}) - \nabla_{\tilde{\theta}}\tilde{\ell}^m(\tilde{\theta}_t, Z_t^{m,n})\right\rangle \mathrm{d}t \\ &+ \left\langle \bar{Z}_t^{m,n} - Z_t^{m,n}, \nabla_z\tilde{\ell}^m(\tilde{\theta}_t^N, \bar{Z}_t^{m,n}) - \nabla_z\tilde{\ell}^m(\tilde{\theta}_t, Z_t^{m,n})\right\rangle\big]\mathrm{d}t + 2G_t^N\mathrm{d}t,\end{aligned}$$

where we defined the term G_t^N as:

$$G_t^N := \frac{1}{M}\sum_{m=1}^{M}G_t^{m,N}, \quad \text{where } G_t^{m,N} := \left\langle \tilde{\theta}_t^N - \tilde{\theta}_t, \frac{1}{N}\sum_{n=1}^{N}\nabla_{\tilde{\theta}}\tilde{\ell}^m(\tilde{\theta}_t, Z_t^{m,n}) - \nabla_{\tilde{\theta}}\int\tilde{\ell}^m(\tilde{\theta}_t, z)q_t^m(\mathrm{d}z)\right\rangle.$$

Using the strong log-concavity **A**3, and taking the expectation yields

$$\mathrm{d}\mathbb{E}[\xi_t^N] \leq -2\lambda\mathbb{E}[\xi_t^N]\mathrm{d}t + 2\mathbb{E}[G_t^N]\mathrm{d}t.$$

Following the same argument as in Caprio et al. (2025, Lemma 13), together with the analogous second-moment bound from Caprio et al. (2025, Proposition 26) and the L-Lipschitz gradient assumption **A**4, we can upper bound $|\mathbb{E}[G_t^{m,N}]|$ for each $m \in [M]$:

$$\left|\mathbb{E}[G_t^{m,N}]\right| \leq L\sqrt{\frac{2\mathbb{E}\left[\|\tilde{\theta}_t^N - \tilde{\theta}_t\|^2\right]}{N}\left(\|\tilde{\theta}_0\|^2 + \mathbb{E}[\|Z_0\|^2] + \frac{d_z}{\lambda}\right)} \leq L\sqrt{\frac{2\mathbb{E}[\xi_t^N]}{N}\left(\|\tilde{\theta}_0\|^2 + \mathbb{E}[\|Z_0\|^2] + \frac{d_z}{\lambda}\right)}. \tag{92}$$

Now we have the following differential inequality for $\mathbb{E}[\xi_t^N]$:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}[\xi_t^N]^{1/2} \leq -\lambda\mathbb{E}[\xi_t^N]^{1/2} + L\sqrt{\frac{2}{N}\left(\|\tilde{\theta}_0\|^2 + \mathbb{E}[\|Z_0\|^2] + \frac{d_z}{\lambda}\right)}. \tag{93}$$

Applying Grönwall's lemma yields the result:

$$\mathbb{E}[\xi_t^N]^{1/2} \leq e^{-\lambda t}\mathbb{E}[\xi_0^N]^{1/2} + \frac{(1-e^{-\lambda t})L}{\lambda}\sqrt{\frac{2}{N}\left(\|\tilde{\theta}_0\|^2 + \mathbb{E}[\|Z_0\|^2] + \frac{d_z}{\lambda}\right)}, \tag{94}$$

where the first term is zero due to construction. □

Lemma C.16 (Temporal Discretization Error)**.** *Suppose Assumptions **A**3 and **A**4 hold. Then for the following Euler-Maruyama scheme for the SDEs in (87)-(88):*

$$\tilde{\theta}_{k+1}^{N,h} = \tilde{\theta}_k^{N,h} + \frac{h}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}\nabla_{\tilde{\theta}}\tilde{\ell}(\tilde{\theta}_k^{N,h}, \bar{Z}_k^{m,n,N,h}) \tag{95}$$

$$\bar{Z}_{k+1}^{m,n,N,h} = \bar{Z}_k^{m,n,N,h} + h\nabla_z\tilde{\ell}(\tilde{\theta}_k^{N,h}, \bar{Z}_k^{m,n,N,h}) + \sqrt{2h}W_k^n, \quad (m,n) \in [M]\times[N], \tag{96}$$

*we have for $h \leq 1/(\lambda + L)$ with λ being the strong concavity constant in Assumption **A**3 and L being the Lipschitz constant in **A**4, there exists a constant $C(h) > 0$ independent of k and N of order $\mathcal{O}(h^{1/2})$, such that:*

$$\mathsf{d}_{\mathcal{M}^{1:M}}((\tilde{\theta}_k^{N,h}, \bar{Q}_k^{1:M,N,h}), (\tilde{\theta}_{kh}^N, \bar{Q}_{kh}^{1:M,N})) \leq \sqrt{h}C(h), \quad \forall k \in \mathbb{N},$$

Proof. The proof is a direct adaptation of Caprio et al. (2025, Lemma 14) by replacing the discretization therein with eqs. (95) and (96) and using the same argument for the $M \times N$ equations for the particles and replacing the loss with $M^{-1}\sum_{m=1}^{M}\tilde{\ell}^m$ for the parameters. □

Theorem C.17. *Suppose that the premise of Theorem 3.1 and **A**4 hold, and that $\mathcal{R}$ has Lipschitz gradients. For all sufficiently small $h > 0$, there exists an $\mathcal{O}(h^{1/2} + N^{-1/2})$ constant $C_{h,N}$ independent of T, a $\rho \in (0,1)$, and a $C > 0$ independent of (h, N, T), such that*

$$\mathbb{E}\left[||\tilde{\theta}_T - \tilde{\theta}_\star||^2\right]^{1/2} \leq C_{h,N} + C\rho^T \quad \forall T \in \mathbb{N},$$

where $\tilde{\theta}_\star$ denotes $\tilde{\ell}$'s unique maximizer.

Proof. The proof follows from a combination of Lemma C.14, Lemma C.15, and Lemma C.16 by an application of the triangle inequality and the definition of the metric $\mathsf{d}_{\mathcal{M}^{1:M}}$. □

D Additional Results

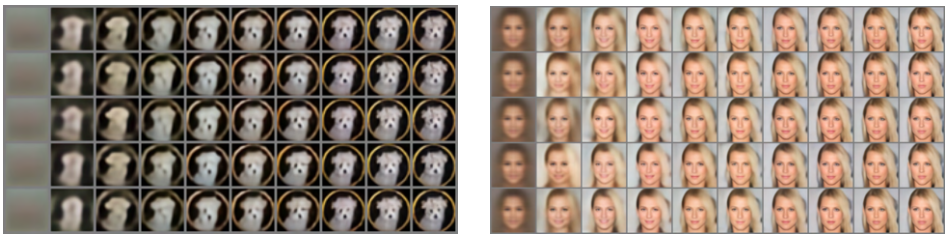
D.1 Interpolating the Latent Space

As a deep LVM, IPLD is able to learn a smooth latent space, enabling semantically meaningful interpolations. In Figure 4, we take two indices $m_1, m_2 \in [M]$ from the training set indices and extract the particles $z_0^{m_1,0}, z_0^{m_2,0}$ from the particle cloud. We compute the linear interpolation via $\texttt{lerp}(x, y, s) = sx + (1-s)y$ between those particles for $s \in [0,1]$ and decode back into the pixel space with the decoder g_ϕ.

Figure 4: Linear interpolation between the particles learned on the CelebA64 training set.

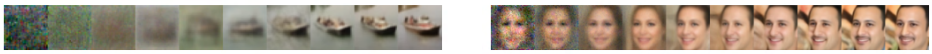
D.2 Evolution of the Particle Cloud

We also visualize how the particle cloud $z_0^{1:M,1:N}$ evolves through the gradient flow. To achieve this, we save the particle cloud at training epochs $\{0, 5, 10, 15, 20, 25, 30, 50, 75, 90\}$ and pass them through the decoder g_ϕ (Figures 5a and 5b). We note that there is an intriguing perceptual similarity between the particle evolution and the posterior mean $\mathbb{E}[x_0|x_t]$ of a diffusion model. For comparison, we show the evolution of the posterior mean predicted in Figures 6a and 6b. The posterior mean is computed by a pretrained score network taken from Song et al. (2021a) in the pixel space via Tweedie's formula (Efron, 2011): $\mathbb{E}[x_0|x_t] \approx (\sqrt{\bar{\alpha}_t})^{-1}[x_t + \sigma_t^2 \nabla_{x_t} \boldsymbol{s}_\theta(x_t, t)]$, where σ_t^2 is the variance of the reverse process and $\boldsymbol{s}_\theta(x_t, t)$ is the score estimating network. We remark that there have been several recent works attempting to delineate the connection between diffusion and gradient flow (Yi et al., 2023; Huang and Zhang, 2023; Franceschi et al., 2023). Our method can be thought as an attempt of learning the gradient flow via a diffusion model.



(a) CIFAR-10 (b) CelebA64

Figure 5: Evolution of the particle cloud of IPLD trained on the CIFAR-10 and CelebA64 dataset with 5 particles. Zoom in to view the details better.



(a) CIFAR-10 (b) CelebA64

Figure 6: Evolution of the predicted $\mathbb{E}[x_0|x_t]$ of DDPM trained on the CIFAR-10 and CelebA64 dataset.

D.3 Ablation on Diffusion Weighting

As reported in Ho et al. (2020); Kingma and Gao (2023), using the simplified diffusion objective can yield samples with better visual quality. To verify this effect within our framework, we compare two variants of our model: IPLD likelihood using the diffusion loss in (40) and IPLD usual using the loss in (42). We train all models with the same training hyperparameters specified in Appendix B.2. The results in Table 3 show that the simplified objective yields a consistently better FID score, which aligns with prior findings.

D.4 Ablation on Two Stage Training

Leng et al. (2025) hypothesized that backpropogating the diffusion loss to the VAE directly make the latent space simpler. They suggested this could inadvertently "hack" the denoising objective, leading the diffusion model to

	SVHN	CIFAR-10
IPLD usual	17.55	51.60
IPLD likelihood	18.62	53.22

Table 3: Ablation study on diffusion weighting. The final FID scores are reported.

simply predict noise from the VAE's Gaussian approximate posterior.

To investigate this, we conduct an ablation study comparing end-to-end training and two-stage training. We train both models with 1 particle using the same training hyperparameters; for two-stage training we detach the gradients on the particles before passing them to the diffusion model, hence preventing backpropogating the diffusion loss. We show in Table 4 that end-to-end training in IPLD does not suffer from the same issue observed in Leng et al. (2025). This aligns with the results in Vahdat et al. (2021), who also found benefits to end-to-end training. It is important to note, however, that both our experiments and those of Vahdat et al. (2021) were conducted on relatively lower-resolution datasets. Therefore, the conclusions drawn here may not directly extrapolate to the larger-scale experimental settings used in Leng et al. (2025).

	SVHN	CIFAR-10	CelebA64
IPLD usual	17.55	51.60	22.86
IPLD detached	19.07	52.96	23.19

Table 4: Ablation study on two-stage training. The final FID scores are reported.

D.5 Runtime and Memory Comparisons

In Table 5, we benchmark the peak memory usage and walltime for a single forward-backward pass with a batch size of 64 on a single A6000 GPU (48GB) for both DiffusionVAE and IPLD: Using `torchrec` (Ivchenko et al., 2022) and custom sharding strategies enabled by it, we implement a distributed version of IPLD by allocating each accelerator a different subset of the particles (cf. Section 3.3). The results in Table 6 confirms the scalability of our algorithm.

Table 5: Performance comparison on a single A6000 GPU. Lower values are better.

Method	**Peak Memory (GB) ↓**	**Walltime (s) ↓**
DiffusionVAE 1-particle	5.19	0.15
DiffusionVAE 5-particle	17.51	0.54
DiffusionVAE 10-particle	33.89	1.01
IPLD 1-particle	4.19	0.12
IPLD 5-particle	17.26	0.49
IPLD 10-particle	33.62	0.98

Table 6: Performance of our custom distributed IPLD implementation on two A6000 GPUs. Lower values are better.

Method	**Peak Memory (GB) ↓**	**Walltime (s) ↓**
IPLD 1-particle	2.81	0.09
IPLD 5-particle	9.62	0.29
IPLD 10-particle	17.95	0.56

D.6 Additional Uncurated Samples

We present additional uncurated samples of IPLD trained with 10 particles on CIFAR-10, SVHN, and CelebA64 dataset produced by the DDIM sampler (Song et al., 2021a) with 100 NFEs. We remark that only the CelebA64 samples in the main text have been curated for better visualization.

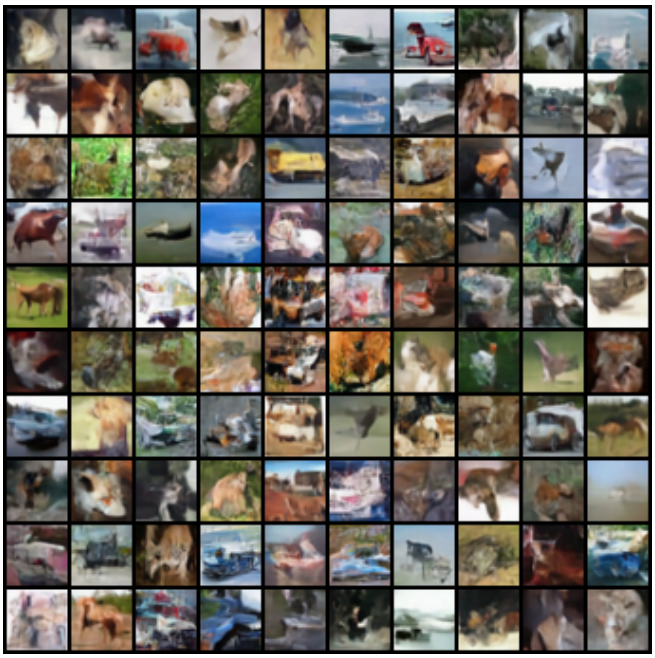


Figure 7: Uncurated samples on CIFAR-10.

Figure 8: Uncurated samples on SVHN.

Figure 9: Uncurated samples on CelebA64.