GENERATIVE AUTO ENCODERS VIA PICARD ITERATED WASSERSTEIN GRADIENT FLOWS

Anonymous authors
Paper under double-blind review

ABSTRACT

Variational Auto-Encoders are trained with the goal of reconstructing the inputs from the latents accurately, and they also attempt to match the latent distribution to a desired prior (typically standard Gaussian). Due to the tradeoff between reconstruction and prior regularization in ELBO-based training, the latent distribution is far from the desired prior, which results in poor quality of VAEs when used as generative models. Motivated by the problem of shaping a latent distribution to the Gaussian, we introduce the smoothed KL divergence, a geodesically non-convex functional on the space of probability distributions and consider its Wasserstein Gradient Flow (WGF). This WGF requires only the score function of the smoothed distribution and is easily learned via the popular denoising score matching. We establish theoretically that it converges to the target distribution (Gaussian) via a Lyapunov decay argument, bypassing Polyak-Łojaciewicz type conditions usually used in this context. We apply the smoothed KL WGF to derive a novel generative modeling algorithm, based on autoencoders. Our algorithm introduces a time-parameterized neural network in the latent space to represent trajectories, and during training we iteratively update the entire family of trajectories using Picard's iteration, with the goal of eventually aligning the trajectories to the WGF. Simultaneously, we adapt the decoder to reconstruct from the endpoints of the trajectories using autoencoder reconstruction losses. We establish theoretical guarantees for convergence of the Picard iterates to the chosen WGF, as well as convergence of the WGF to Gaussian, under mild assumptions on the initial latent distribution. We empirically establish the efficacy of our method on the unconditional datasets CelebA-HQ, LSUN Church and LSUN Bedroom.

1 INTRODUCTION

Variational Auto-Encoders (VAE) constitute a class of latent generative models (Kingma & Welling, 2014). While a regular auto-encoder is only trained to reconstruct accurately (with little to no control over the encoded latent distribution), VAEs are trained to additionally match the latent distribution to a desired prior (e.g., standard Gaussian). This is done by maximizing the Evidence Lower Bound (ELBO) objective, which consists of two terms: a reconstruction term, and a KL regularizer. Once trained, VAEs can be used as generative models by first sampling from the desired latent prior, and then passing it through the decoder.

VAEs, however, face a fundamental issue when maximizing the ELBO objective, which is a lower bound to the actual log-likelihood. The reconstruction loss tends to favour the latent representations being pushed farther apart (to retain as much information about the input as possible), while the KL regularizer tries to bring them closer to match the desired prior. This tension leads to a mismatch between the encoded latent distribution and the desired prior (Zhao et al., 2019), which can result in sub-optimal generation quality. Several approaches have been explored in literature to address this problem. Some prominent approaches include using different regularizers and different weights for the loss terms (Tolstikhin et al., 2018; Kolouri et al., 2019; Makhzani et al., 2015; Higgins et al., 2017), using hierarchical VAEs to increase the expressivity of the desired prior and the approximate posterior (Vahdat & Kautz, 2020; Child, 2021), or using vector-quantized autoencoders which learn a discrete set of code vectors in the latent space, followed by fitting an autoregressive prior on the latents (van den Oord et al., 2017; Esser et al., 2021; Razavi et al., 2019). Despite these successes,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

the fundamental issue of mismatch between the trained latent distribution and the desired prior still remains. We instead take an alternate approach by asking the following question:

> Starting from a pre-trained VAE, can we develop a flow-based corrector that transforms the latent distribution into the desired prior (i.e., standard Gaussian)?

We address this question by presenting a new approach for learning flows to transform the initial distribution of the VAE into the standard Gaussian, based on Wasserstein gradient flows (WGF). We first theoretically construct a flow that provably transforms the initial distribution to the desired prior (specifically, the WGF of a smoothed KL functional). Our technical novelty lies in the mechanism used to learn this flow:

a) Our chosen Wasserstein gradient flow has the advantage that it requires only the score function of the smoothed distribution, instead of the exact score function. The former can be directly learned via the standard Denoising Score Matching (DSM) (Vincent (2011)) which underlies standard diffusion-model training (Song et al. (2021)). The latter requires more involved algorithms such as Implicit Score Matching (ISM) or its variants (Hyvärinen (2005); Song et al. (2020)), which are not used due to their complexity.

b) Instead of viewing the flow as a time-dynamic (i.e., iterating over time steps along the flow trajectory to learn the flow), we shift perspectives and view the iteration in *trajectory space*. We update the entire trajectory based on the previous trajectory using Picard's iteration (where trajectories are represented by a time-dependent neural network), and after sufficiently many iterations the trajectory provably converges to the constructed WGF flow. Furthermore, standard Picard iteration can result in unstable trajectories over long time horizons, which motivates us to instead develop a more stable *semi-implicit* iteration in which the linear term of the velocity field is evaluated at the updated trajectory rather than the previous iterate. Importantly, our approach comes with theoretical guarantees for both convergence of the WGF flow to the desired prior, and convergence of the Picard iterates to the WGF flow, making it a principled VAE corrector algorithm.

While the latent distribution of the VAE is transported toward a Gaussian, we simultaneously adapt the decoder to the end-points of the learned trajectories. This enables us to use the resulting auto-encoder as a one-step generative model, where inference simply consists of applying the adapted decoder on a sampled Gaussian noise.

Our specific contributions are as follows.

- We introduce our VAE corrector algorithm, called PC-VAE (Picard Corrected VAE), which learns the WGF of a smoothed KL functional by iteratively refining trajectories using semi-implicit Picard iteration (Section 4).
- We prove that the smoothed KL functional decays as $O(1/t)$ along its WGF (thereby establishing convergence of the WGF to the standard Gaussian in W_2), assuming that the initial distribution has finite KL divergence with respect to standard Gaussian (Theorem 1). Our approach involves using the unsmoothed KL divergence as a Lyapunov functional, avoiding the need to obtain a Polyak-Łojaciewicz (PL) inequality for the smoothed KL functional.
- We prove that the semi-implicit Picard iterates converge uniformly in L^2 to the WGF solution on every finite time interval (Theorem 2). Our method has strong guarantees: we show exponential decay in error for a large class of initial distributions requiring only bounded second moments (importantly, we do not need an LSI or Poincaré-type assumption on the data distribution).
- We evaluate our method on CelebA-HQ, LSUN Church and LSUN Bedroom datasets, and we demonstrate improved generation quality over the pretrained VAE and competitive performance relative to established baselines, while retaining reconstruction capability.

2 Related Works

Generative Models. Variational autoencoders (VAE) (Kingma & Welling, 2014) and Generative Adversarial Networks (GAN) (Goodfellow et al., 2014) are among the earliest milestones in deep generative modeling frameworks, which demonstrated the potential for neural networks to generate data. Since then, diffusion models (Ho et al., 2020; Song et al., 2021) and flow match-

ing (Lipman et al., 2023) established themselves as the dominant paradigms; these methods learn a multi-step mapping from noise to data using the framework of stochastic/ordinary differential equations. More recently, few/single-step generative models have garnered interest, including consistency models (Song et al., 2023), and drifting models (Deng et al., 2026). Our work revisits the VAE: rather than introducing a new model family, we aim to correct the mismatch between the VAE latent distribution and the desired prior, thereby improving the generative capabilities of the VAE.

Wasserstein gradient flows. Wasserstein gradient flows (WGF) provide a principled framework for constructing ODEs which converge to a target distribution, by performing gradient descent on an energy functional in probability space (Ambrosio et al., 2005). Recent work has connected drifting models to this framework and explored WGF-based one-step generative modeling (Gretton et al., 2026; Han et al., 2026). Our method also uses WGF to transport a pre-trained VAE's latent distribution toward a standard Gaussian, however the main difference is that we learn this flow by iterating in *trajectory space* (i.e., we iteratively refine the entire trajectories using Picard iteration), whereas the aforementioned approaches learn the flow by iterating over time steps.

For a detailed comparison with related works, we refer to Appendix A.

3 Background and Motivation

In this section, we give a brief background on Wasserstein gradient flows (Ambrosio et al., 2005) and Picard's iteration. We then motivate our algorithmic approach based on semi-implicit Picard iteration. A more detailed exposition is present in Appendix B and the references contained therein.

Wasserstein gradient flows (WGF) are our primary tool for constructing ODEs which converge to a desired distribution. Let $\mathcal{P}_2(\mathbb{R}^d)$ denote the space of probability distributions on $\mathbb{R}^d$ with finite second moments, and let $\mathcal{F} : \mathcal{P}_2(\mathbb{R}^d) \to \mathbb{R}$ be a functional on this space. Then, the Wasserstein gradient flow (WGF) with respect to $\mathcal{F}$ corresponds to the ODE

$$\frac{d\mathbf{x}_t}{dt} = -\nabla_x \frac{\delta \mathcal{F}}{\delta p_t}(\mathbf{x}_t, p_t), \qquad p_t := p_{\mathbf{x}_t} = \mathrm{Law}(\mathbf{x}_t). \tag{1}$$

We note that the velocity field depends on the current *density* p_t (in addition to the *state* $\mathbf{x}_t$); such an ODE is called a mean field ODE. Under mild regularity conditions, $\mathcal{F}(p_t)$ is non-increasing along the WGF. Thus, one recipe to construct a flow whose target is some desired prior q (e.g., standard Gaussian) is to consider the WGF with respect to an appropriate functional $\mathcal{F}$ which is minimized at q, and showing that $\mathcal{F}(p_t) \to \mathcal{F}(q)$ and thus $p_t \to q$.

Once we have an appropriate ODE as defined above, the next step is to solve it on an interval of interest. Picard's iteration solves the ODE by starting from an initial trajectory and then recursively updating the entire trajectory until convergence. For the mean field ODE $\dot{\mathbf{x}}_t = b(t, \mathbf{x}_t, p_{\mathbf{x}_t})$ with initial condition $\mathbf{x}_0 \sim p_0$, Picard's iteration is defined by the recursion $\frac{d\mathbf{x}_t^{n+1}}{dt} = b(t, \mathbf{x}_t^n, p_{\mathbf{x}_t^n})$, or equivalently in integral form

$$\mathbf{x}_t^{n+1} = \mathbf{x}_0 + \int_0^t b(s, \mathbf{x}_s^n, p_{\mathbf{x}_s^n}) ds. \tag{2}$$

Picard's iteration can thus be viewed as a fixed point iteration in trajectory space, where the fixed point trajectory is the solution to the ODE. Note that unlike regular Picard iteration for a standard ODE, here both the state and density arguments on the RHS come from the previous iterate $\mathbf{x}_t^n$; this can be viewed as Picard iteration in probability space. Under suitable Lipschitz and regularity conditions, the iterates $\mathbf{x}^n$ converge to a fixed point trajectory satisfying the ODE.

If the velocity field has a contractive linear component (i.e., $b(t, x, \mu) = -kx - v(t, x, \mu)$ where $k > 0$), we can also define *semi-implicit* Picard iteration by the recursion $\frac{d\mathbf{x}_t^{n+1}}{dt} = -k\mathbf{x}_t^{n+1} - v(t, \mathbf{x}_t^n, p_{\mathbf{x}_t^n})$, i.e., the linear component is evaluated on the current trajectory $\mathbf{x}_t^{n+1}$ instead of $\mathbf{x}_t^n$. In integral form (using integrating factor), the iteration is given by

$$\mathbf{x}_t^{n+1} = e^{-kt}\mathbf{x}_0 - \int_0^t e^{k(s-t)} v(s, \mathbf{x}_s^n, p_{\mathbf{x}_s^n}) ds. \tag{3}$$

Table 1: Conventional score-based approaches vs Picard-iterated Wasserstein Gradient Flow

Conventional score-based approaches	**Our approach: Picard-iterated WGF**
Learned output. A time-dependent score field $s_\phi(y,t) \in \mathbb{R}^d$, affinely related to the velocity field of the flow.	**Learned output.** A trajectory map $\Psi_\theta : \mathbb{R}^d \times [0,T] \to \mathbb{R}^d$ representing the entire family of WGF trajectories, together with an adapted decoder D_ω.
Training procedure. Learn the score using denoising score matching: $$\min_\phi \mathbb{E}\left[\lambda(t)\left\|s_\phi(\mathbf{x}_t,t)+\frac{\mathbf{z}}{\sigma_t}\right\|^2\right],$$ where $\mathbf{x}_t = \alpha_t\mathbf{x}_0 + \sigma_t\mathbf{z}$, $\mathbf{x}_0 \sim p_{\text{data}}$, $\mathbf{z} \sim \mathcal{N}(0,I)$ is independent of $\mathbf{x}_0$, $t \sim \text{Unif}[0,T]$, and $\lambda(t) > 0$. The noise schedule (α_t, σ_t) is prescribed.	**Training procedure.** Alternating three phase procedure (details in Algorithm 1): **1. Vector field estimation.** Freeze current trajectory map Ψ_θ and learn the vector field required for the trajectory update phase. **2. Trajectory update.** Update Ψ_θ using the semi-implicit Picard update equation based on the learned vector field. **3. Decoder Update.** Adapt the decoder D_ω to reconstruct from terminal latents $\Psi_\theta(\mathbf{x}_0, T)$.
Inference procedure. Discretize the time interval into many time steps; sample Gaussian noise and update the flow along the time steps using the learned velocity field.	**Inference procedure.** Sample $\mathbf{z} \sim \mathcal{N}(0,I)$ and generate image $\widehat{\mathbf{u}} = D_\omega(\mathbf{z})$. Generation requires only one decoder evaluation. Neither the learned vector field nor the trajectory map is evaluated (i.e., a distillation-like procedure).

Stability of semi-implicit Picard iteration: When analyzing error propagation using semi-implicit Picard iteration (see Appendix B.3 for details), we find that the $e^{k(s-t)}$ term exponentially down-weights the errors arising from portions of the previous trajectory that are not very close to the current time t, which leads to better finite-iteration stability compared to standard Picard iteration. Furthermore, in Appendix B.3 we also illustrate an example using the WGF of the unsmoothed KL functional, where standard Picard iterates diverge while the semi-implicit Picard iterates converge uniformly on finite time intervals. Since the WGF flow that we are interested in (defined in the next section) contains a linear contractive term, we will henceforth only use semi-implicit Picard iteration for the rest of the paper.

4 PICARD-CORRECTED VAE (PC-VAE)

In this section we provide an outline of the PC-VAE algorithm, followed by the mathematical description required to discuss the algorithm in detail.

4.1 ALGORITHM OUTLINE

Suppose we have a pre-trained autoencoder with encoder E and decoder D, and let p_0 denote the distribution of the encoded latent $\mathbf{x}_0 = E(\mathbf{u}_0), \mathbf{u}_0 \sim p_{\text{data}}$. If we wish to use the autoencoder as a generative model by applying the decoder to a latent $\mathbf{z} \sim \mathcal{N}(0,I)$, then we face a fundamental issue: the decoder is only adapted to the latent distribution p_0, which may not be exactly Gaussian. Due to this mismatch, Gaussian samples $\mathbf{z}$ can fall in regions of the latent space where the decoder is poorly adapted, resulting in poor generation quality. Therefore, our objective is to adapt the original decoder D into a new decoder D_ω that can reconstruct data from Gaussian latents.

To achieve this, our algorithm learns a pair (Ψ, D_ω), where $\Psi : \mathbb{R}^d \times [0,T] \to \mathbb{R}^d$ is a trajectory map whose starting point is the initial latent $\Psi(\mathbf{x}_0, 0) = \mathbf{x}_0$ and whose end-point $\Psi(\mathbf{x}_0, T) = \widetilde{\mathbf{z}}$, where $\widetilde{\mathbf{z}}$ is close to Gaussian (formal guarantee of closeness as a function of T is present in Theorem 1), and D_ω is the adapted decoder which is trained to reconstruct from the end-point $\widetilde{\mathbf{z}}$ instead of $\mathbf{x}_0$. The trajectory map Ψ that we aim to learn is chosen to be an appropriate ODE flow. However, unlike diffusion/score-based approaches which learn the velocity field of the flow, we learn the entire flow trajectories using Picard iteration. Table 1 summarizes the key differences between our approach and conventional score-based approaches. In the next section, we develop the mathematical formulation required to describe our algorithm, following which we discuss the algorithm in detail.

4.2 Mathematical Formulation

We now mathematically formulate the ODE whose solution supplies the trajectory map Ψ to be learned by our algorithm. The ODE solution must satisfy the property that its target distribution is the standard Gaussian $q = \mathcal{N}(0, I)$. Following the WGF framework introduced in Section 3, we thus seek an appropriate functional whose minimizer is q, and we design the ODE to be the WGF of that functional. A natural choice is the KL divergence functional $\mathcal{G}(p) = D_{\mathrm{KL}}(p\|q)$, whose WGF is given by (see Appendix B for derivation)

$$\frac{d\mathbf{x}_t}{dt} = -\nabla \log \frac{p_t(\mathbf{x}_t)}{q(\mathbf{x}_t)} = -\mathbf{x}_t - \nabla \log p_t(\mathbf{x}_t), \qquad p_t := p_{\mathbf{x}_t}. \tag{4}$$

The marginals p_t evolved by the above WGF coincide with that of the Ornstein-Uhlenbeck (OU) process $d\mathbf{x}_t = -\mathbf{x}_t dt + \sqrt{2} \cdot d\mathbf{w}_t$, where $\mathbf{w}_t$ denotes the standard Brownian motion (see Appendix B for details), for which convergence is well understood using the Gaussian Log-Sobolev inequality (specifically, $p_t \to q$ exponentially fast in KL). Despite its nice convergence properties, learning the above WGF using Picard's iteration requires the unsmoothed score $\nabla \log p_{\mathbf{x}_t^n}$ (where $\mathbf{x}_t^n$ refers to the current trajectory) during each iteration, which may be difficult to learn.

We instead consider the *smoothed* KL functional, $\mathcal{F}(p) = D_{\mathrm{KL}}(p_\epsilon\|q)$, where $\epsilon \in (0, 1)$ is the smoothing parameter, and p_ϵ denotes the law of $a\mathbf{x} + \sigma\mathbf{z}$ where $\mathbf{x} \sim p, \mathbf{z} \sim q$ are independent, $a := 1 - \epsilon$ and $a^2 + \sigma^2 = 1$. Since $q_\epsilon = q$ and Gaussian smoothing is injective, $\mathcal{F}(p)$ is indeed minimized at $p = q$. The corresponding WGF (derived in Appendix D) is given by

$$\frac{d\mathbf{x}_t}{dt} = -\nabla \frac{\delta \mathcal{F}}{\delta p_t}(\mathbf{x}_t) = -a^2\mathbf{x}_t - a \cdot \mathbb{E}_{\mathbf{z}\sim q}[\nabla \log p_{t,\epsilon}(a\mathbf{x}_t + \sigma\mathbf{z})], \qquad p_t := p_{\mathbf{x}_t}, \tag{5}$$

and its corresponding semi-implicit Picard update is given by

$$\frac{d\mathbf{x}_t^{n+1}}{dt} = -a^2\mathbf{x}_t^{n+1} - a\mathbb{E}_{\mathbf{z}\sim q}[\nabla \log p_{\mathbf{x}_t^n,\epsilon}(a\mathbf{x}_t^n + \sigma\mathbf{z})], \qquad \mathbf{x}_0^{n+1} = \mathbf{x}_0. \tag{6}$$

We note that the velocity field in equation 6 requires only the Gaussian smoothed score of the current distribution (i.e., $\nabla \log p_{\mathbf{x}_t^n,\epsilon}$), which is the exact population minimizer of the Denoising Score Matching (DSM) objective (Vincent, 2011), making it efficiently learnable using samples from the current trajectory distribution $p_{\mathbf{x}_t^n}$.

4.3 Algorithm Details

Using the mathematical formalism of the WGF and its semi-implicit Picard update defined above, we now discuss the complete training and inference procedure of our algorithm. At a high level, we parameterize the solution Ψ of the chosen ODE using a time-dependent neural network $\Psi_\theta(\mathbf{x}_0, t)$, and training involves iteratively refining Ψ_θ based on semi-implicit Picard iteration. We explain each component in detail below.

Trajectory Map Parameterization. We parameterize trajectories by $\Psi_\theta(\mathbf{x}_0, t) = \mathbf{x}_0 + t \cdot F_\theta(\mathbf{x}_0, t)$ (where F_θ is a neural network), to ensure that the initial condition is always satisfied as $\Psi(\mathbf{x}_0, 0) = \mathbf{x}_0$, and we initialize $F_\theta \equiv 0$ to represent the initial constant trajectory ($x_t^0 = \mathbf{x}_0$).

Phase 1: Vector Field estimation. We freeze the current trajectory map Ψ_θ, and train a score network $s_\phi(y, t)$ on the current trajectory $\mathbf{x}_t^n = \Psi_\theta(\mathbf{x}_0, t)$, using the denoising score matching (DSM) objective

$$\mathcal{L}_{\mathrm{score}}(\phi) = \mathbb{E}\left[\left\| s_\phi\big(a\Psi_\theta(\mathbf{x}_0, t) + \sigma\mathbf{z}, t\big) + \frac{\mathbf{z}}{\sigma}\right\|^2\right], \tag{7}$$

where $\mathbf{x}_0 \sim p_0$, $t \sim \mathrm{Unif}[0, T]$, and $\mathbf{z} \sim q$ independent of $\mathbf{x}_0$. The exact population minimizer of the above loss is $s_\phi(y, t) = \nabla \log p_{\mathbf{x}_t^n,\epsilon}(y)$, which is precisely the quantity we need for the semi-implicit Picard update (equation 6).

Phase 2: Trajectory Map Update. We first create a frozen copy $\widehat{\theta} \leftarrow \theta$, so that $\Psi_{\widehat{\theta}}(\mathbf{x}_0, t)$ represents the current trajectory $\mathbf{x}_t^n$ throughout, and we freeze the score network s_ϕ. Following equation 6, we then update θ by minimizing

$$\mathcal{L}_{\mathrm{flow}}(\theta) = \mathbb{E}_{\mathbf{x}_0,t,\mathbf{z}}\left[\left\|\partial_t\Psi_\theta(\mathbf{x}_0, t) - \big(-a^2\Psi_\theta(\mathbf{x}_0, t) - as_\phi(a\Psi_{\widehat{\theta}}(\mathbf{x}_0, t) + \sigma\mathbf{z}, t)\big)\right\|^2\right]. \tag{8}$$

The partial derivative $\partial_t \Psi_\theta(\mathbf{x}_0, t)$ is computed efficiently using forward mode auto-differentiation. Since $s_\phi(a\Psi_{\widehat{\theta}}(\mathbf{x}_0, t) + \sigma\mathbf{z}, t)$ is an unbiased estimate of its expectation with respect to $\mathbf{z}$, minimizing the above loss will result in the correct semi-implicit Picard update.

Phase 3: Decoder Update. We update the decoder D_ω to reconstruct from the terminal time latents $\Psi_\theta(\mathbf{x}_0, T)$; this allows the decoder to adapt to the changing terminal latent distribution, while the original encoder remains fixed.

The three alternating phases are repeated until convergence. The algorithm outputs (Ψ_θ, D_ω), where Ψ_θ represents the solution to the ODE, and D_ω is the adapted decoder which reconstructs from the end points $\Psi_\theta(\mathbf{x}_0, T)$. Algorithm 1 summarizes the training procedure. For additional implementation details, we refer to Appendix H.1.

Inference. Inference involves evaluating the decoder D_ω on a sampled noise $\mathbf{z} \sim \mathcal{N}(0, I)$, i.e., it is a one-step generative process. In this sense, our algorithm has similarities with consistency distillation (Song et al., 2023), where a pre-trained velocity field is used to supply numerical ODE steps between neighboring time points, and a consistency network learns to produce matching outputs at the resulting pair, thereby yielding a one-step generator. However a crucial difference is that our algorithm is *on-policy*: during each iteration we learn the vector field using samples from the current trajectory map, and this learned vector field is subsequently "distilled" into the updated trajectory. We require neither a pre-trained flow teacher nor samples from the actual WGF. In fact, in Appendix E we explain why distilling a pre-trained flow model using Picard's iteration may practically lead to unreliable training.

Algorithm 1 Picard-Corrected Variational AutoEncoder (PC-VAE) Training Procedure

Require: Pre-trained VAE with encoder E (frozen) and decoder D with trainable parameters ω; $T > 0$; $\epsilon \in (0, 1)$; iteration counts $N_{\text{picard}}, N_1, N_2, N_3$.

Initialize $\Psi_\theta(x, t) = x + tF_\theta(x, t)$ with $F_\theta \equiv 0$, and score network $s_\phi(x, t)$.
for $n = 0, \ldots, N_{\text{Picard}} - 1$ **do**
  Create a frozen copy $\widehat{\theta} \leftarrow \theta$
  for $k = 1, \ldots, N_1$ **do** ▷ **Phase 1: update the score network**
    Sample a minibatch of images $\mathbf{u}_0$, times $t \in [0, T]$, and $\mathbf{z} \sim \mathcal{N}(0, I)$; set $\mathbf{x}_0 = E(\mathbf{u}_0)$.
    Update ϕ using $\mathcal{L}_{\text{score}}(\phi)$ in equation 7.
  end for
  for $k = 1, \ldots, N_2$ **do** ▷ **Phase 2: update the trajectory**
    Sample a minibatch of images $\mathbf{u}_0$, times $t \in [0, T]$, and $\mathbf{z} \sim \mathcal{N}(0, I)$; set $\mathbf{x}_0 = E(\mathbf{u}_0)$.
    Update θ using $\mathcal{L}_{\text{flow}}(\theta)$ in equation 8.
  end for
  for $k = 1, \ldots, N_3$ **do** ▷ **Phase 3: adapt the decoder**
    Sample a minibatch of images $\mathbf{u}_0$; set $\mathbf{x}_0 = E(\mathbf{u}_0)$.
    Update ω by minimizing reconstruction losses on the terminal latents $\Psi_\theta(\mathbf{x}_0, T)$.
  end for
end for
return (Ψ_θ, D_ω).

5 Main Theoretical Results

In this section, we establish theoretical guarantees for the dynamics underlying the PC-VAE algorithm. Our first result (stated below and proved in Appendix F) shows that not only does $\mathcal{F}(p_t)$ decrease along the WGF equation 5, but that it converges to zero.

Theorem 1. *Let $\mathbf{x}_0 \sim p_0$ and consider the Wasserstein gradient flow with respect to $\mathcal{F}$ (whose ODE is given in equation 5). We denote $p_t = p_{\mathbf{x}_t}$ to be the law of $\mathbf{x}_t$. Under the assumption $D_{\mathrm{KL}}(p_0 \| q) < \infty$, we have that $\mathcal{F}(p_t) \leq \frac{D_{\mathrm{KL}}(p_0 \| q)}{2t}$, and hence $\mathcal{F}(p_t) \to 0$ as $t \to \infty$. As a corollary, we have that $W_2(p_t, q) \to 0$ as $t \to \infty$.*

Proof Sketch of Theorem 1: The standard approach to prove that $\mathcal{F}(p_t) \to 0$ along the WGF is to obtain a PL inequality for the functional $\mathcal{F}$ (i.e., $\mathbb{E}_{\mathbf{x} \sim p}\left[\left\|\nabla_x \frac{\delta \mathcal{F}}{\delta p}(\mathbf{x})\right\|^2\right] \geq c \cdot \mathcal{F}(p)$ for all p, for some

constant $c > 0$), as this would immediately yield exponential convergence of $\mathcal{F}(p_t) \leq e^{-ct}\mathcal{F}(p_0)$. However, in Appendix F we show that such a uniform constant $c > 0$ cannot exist for our smoothed KL functional $\mathcal{F}$.

Instead, we use the unsmoothed KL divergence $\mathcal{G}(p_t) := D_{\mathrm{KL}}(p_t\|q)$ as the Lyapunov functional to show that $\mathcal{F}(p_t) \to 0$ along the WGF. Since the first variation of the KL functional is $\frac{\delta\mathcal{G}}{\delta p}(x) = \log\frac{p(x)}{q(x)}$, we have $\frac{d}{dt}\mathcal{G}(p_t) = \int \log\frac{p_t(x)}{q(x)}\partial_t p_t(x)dx$, and $\partial_t p_t(x) = \mathrm{div}(p_t(x)\cdot\nabla\frac{\delta\mathcal{F}}{\delta p_t}(x))$ from the continuity equation. Upon further simplification using the divergence theorem and properties of the Ornstein–Uhlenbeck (OU) semigroup, we obtain $\frac{d}{dt}\mathcal{G}(p_t) = -I(p_{t,\epsilon}\|q)$, where $I(p\|q)$ denotes the relative Fisher information between p and q. Using the Gaussian log-Sobolev inequality, we then have $\frac{d}{dt}\mathcal{G}(p_t) \leq -2D_{\mathrm{KL}}(p_{t,\epsilon}\|q) = -2\mathcal{F}(p_t)$, and hence $\mathcal{G}(p_t) \leq \mathcal{G}(p_0) - 2\int_0^t \mathcal{F}(p_s)ds \leq \mathcal{G}(p_0) - 2t\mathcal{F}(p_t)$, as $\mathcal{F}(p_t)$ is non-increasing along the WGF. Since $\mathcal{G}(p_t) \geq 0$, we conclude that $\mathcal{F}(p_t) \leq \frac{\mathcal{G}(p_0)}{2t}$.

Since $p_{t,\epsilon} \to q$ in KL, using Talagrand's T_2 inequality for the Gaussian measure, we have $p_{t,\epsilon} \to q$ in W_2, which implies that p_t converges weakly to q, and the second moments of p_t converge to that of q. Hence, $p_t \to q$ in W_2. □

Theorem 1 shows that the WGF ODE equation 5 flows any distribution (under the stated assumption) to a standard Gaussian, making it a principled flow target for our VAE corrector algorithm.

Our next result regarding the convergence of semi-implicit Picard's iteration is stated below (and proved in Appendix G).

Theorem 2. *Assume the initial condition $\mathbf{x}_0 \sim p_0$ has finite second moment (i.e., $\mathbb{E}[\|\mathbf{x}_0\|^2] < \infty$). Let $\mathbf{x}_t^n$ denote the semi-implicit Picard iterates, and let $\mathbf{x}_t$ denote the fixed point trajectory (i.e., the solution to the Wasserstein gradient flow ODE equation 5). Then, for any $T > 0$ we have*

$$\mathbb{E}[\sup_{0\leq t\leq T}\|\mathbf{x}_t^n - \mathbf{x}_t\|^2]^{1/2} \leq \frac{2\sigma}{(1-\epsilon)}(0.5)^n \text{ for all } n \geq 4eC^2T^2 \tag{9}$$

where $C := \frac{2(1-\epsilon)^2}{\sigma^2}\sqrt{2de}$, and consequently,

$$\mathbb{E}[\sup_{t\in[0,T]}\|\mathbf{x}_t^n - \mathbf{x}_t\|^2] \to 0 \quad \text{as } n \to \infty, \tag{10}$$

i.e., the semi-implicit Picard iterates converge in L^2 to the fixed point trajectory.

Proof Sketch of Theorem 2: Denoting $g(x,p) := \mathbb{E}_{\mathbf{z}\sim q}[\nabla \log p_\epsilon(ax + \sigma\mathbf{z})]$, the velocity field for the WGF flow is $b(t,x,p) = -a^2x - ag(x,p)$. The first key step in the proof is to obtain a Lipschitz estimate for g, of the form

$$\mathbb{E}\left[\|g(\mathbf{x},p_{\mathbf{x}}) - g(\widetilde{\mathbf{x}},p_{\widetilde{\mathbf{x}}})\|^r\right]^{1/r} \leq L_{r,u}\cdot\mathbb{E}\left[\|\mathbf{x}-\widetilde{\mathbf{x}}\|^u\right]^{1/u}, \tag{11}$$

where $u > r \geq 2$; this is established in Lemma G.6.

The next step is to show that the semi-implicit Picard iterates $\mathbf{x}_t^n$ converge uniformly in L^2 to a trajectory $\mathbf{x}_t$ on $[0,T]$, which we establish in Lemma G.7 as follows: the successive difference between the semi-implicit Picard iterates can be written (using the integral form) as

$$\mathbf{x}_t^{n+1} - \mathbf{x}_t^n = -a\int_0^t e^{a^2(s-t)}\left(g(\mathbf{x}_s^n, p_{\mathbf{x}_s^n}) - g(\mathbf{x}_s^{n-1}, p_{\mathbf{x}_s^{n-1}})\right)ds, \tag{12}$$

and hence by applying Minkowski's inequality for the L^2 norm, followed by applying the Lipschitz estimate (Lemma G.6) through a carefully constructed sequence of increasing moment exponents, we obtain $\mathbb{E}[\sup_{0\leq t\leq T}\left\|\mathbf{x}_t^{n+1} - \mathbf{x}_t^n\right\|^2]^{1/2} \leq O((CT)^n/\sqrt{n!})$, which is summable in n, yielding that the iterates $\mathbf{x}_t^n$ are Cauchy in L^2, and hence converge to some trajectory $\mathbf{x}_t$. Finally, in Lemma G.8 we show that the limiting trajectory $\mathbf{x}_t$ satisfies the WGF integral equation. □

Theorem 2 is our central result: if we are able to update the trajectories perfectly using Picard's iteration (equation 6), then the trajectories will converge to the desired WGF. This leads to a provably correct VAE corrector algorithm based on Picard iteration. Theorem 2 also gives us an explicit rate of convergence, which is exponential decay once n is greater than a certain threshold. Concretely, for a fixed ϵ, if we allow for a uniform L^2 error of up to $\delta > 0$, then we would require $O(dT^2 + \log(1/\delta) + 1)$ semi-implicit Picard iterations.

6 EXPERIMENTS

6.1 EXPERIMENTAL SETUP

A more detailed description of the experimental setup can be found in Appendix H.2.

Datasets and evaluation. We evaluate our algorithm on CelebA-HQ (Karras et al., 2018), LSUN Bedroom, and LSUN Church (Yu et al., 2015) at 256×256 resolution. For our generation experiments, we compute the Fréchet inception distance (FID) using 50K generated images. We also report reconstruction FID to assess quality of reconstructed images.

Baselines. We compare against the pretrained VAE, flow matching in latent space (Dao et al., 2023), Shortcut Models (Frans et al., 2025), and Consistency Distillation (Song et al., 2023). The pretrained VAE is used as a reference for assessing the effect of latent correction and decoder adaptation. We retrain and evaluate Shortcut Models using the official implementation on all three datasets. For Consistency Distillation, we report published results on LSUN Bedroom, since that is the only dataset shared with our evaluation.

Architecture. For VAE, we use the standard VAE architecture from Stable Diffusion (Rombach et al., 2022) with an additional DiT (Peebles & Xie, 2023) bottleneck added to the encoder and decoder. We use VAE latent dimensions $4\times4\times8$ for CelebA-HQ and LSUN Church, and $8\times8\times4$ for the LSUN Bedroom. The score and trajectory networks are both DiTs with timestep conditioning.

Training. The VAE is trained using a combination of L_1 reconstruction loss, LPIPS perceptual loss, KL loss and adversarial loss (using a PatchGAN discriminator). Once the VAE is trained, the encoder is frozen, and we apply the PC-VAE algorithm. The decoder is trained to reconstruct from the terminal latents $\Psi(\mathbf{x}_0, T)$, using the L_1 reconstruction loss, perceptual loss and adversarial loss. We use a batch size of 128 throughout.

Generation and Reconstruction. Once trained, unconditional generation is done by sampling $\mathbf{z} \sim \mathcal{N}(0, I)$ and then applying the decoder to obtain an image $D_\omega(\mathbf{z})$. For generation, we use an exponential moving average (EMA) of the decoder weights, with decay rate $\beta = 0.9999$. Reconstruction is done by passing an image $\mathbf{u}_0$ to the effective encoder $\Psi_\theta(\cdot, T) \circ E$ followed by applying the decoder D_ω, to obtain the reconstructed image $\widehat{\mathbf{u}}_0 = D_\omega\big(\Psi_\theta(E(\mathbf{u}_0), T)\big)$.

6.2 MAIN RESULTS

Qualitative examples of images generated using our algorithm are shown in Figure 1; more generated images can be found in Appendix I. The main quantitative results are given in Table 3. We report further ablations in Appendix H.3.

Comparison with pre-trained VAE. We observe that naive generation using the pre-trained VAE (i.e., sampling $\mathbf{z} \sim \mathcal{N}(0, I)$ in the latent space and applying the decoder) is ineffective and results in poor FID across all datasets. This is consistent with a mismatch between the VAE latent distribution and the Gaussian prior. PC-VAE substantially improves generation quality over all datasets, suggesting that the transported latent distribution is better aligned with the standard Gaussian; additional diagnostics supporting this can be found in Appendix H.4. Further, our method retains reconstruction capability, as shown in Table 2 where reconstruction FID remains close to that of the pre-trained VAE.

Table 2: Comparison of VAE and PC-VAE reconstruction FID across benchmark datasets.

	Reconstruction FID ↓	
Dataset	**VAE**	**PC-VAE**
CelebA-HQ	0.89	2.04
LSUN Church	1.68	2.34
LSUN Bedroom	3.88	4.17

Comparison with benchmarks. With a single decoder evaluation, PC-VAE achieves lower FID than both one-step and 4-step Shortcut Models on all three datasets. It also achieves lower FID than one-step Consistency Distillation on LSUN Bedroom. Further, we achieve comparable generation quality against latent flow matching (which uses an adaptive-step ODE solver) on LSUN Bedroom and Church datasets.



Figure 1: Generated images for CelebA-HQ, LSUN Bedroom, and LSUN Church.

Table 3: Unconditional generation on CelebA-HQ, LSUN Bedroom, and LSUN Church at 256×256 resolution. Lower FID is better. Bold indicates best score with 1 NFE. Results marked with * are from our own runs of the official baseline implementation. †NFE is reported for CelebA-HQ only; NFE for LSUN Bedroom and Church was not reported.

		FID ↓		
Method	**NFE ↓**	**CelebA-HQ**	**LSUN Bedroom**	**LSUN Church**
FM in Latent Space (DiT-L/2) (Dao et al., 2023)	$89^\dagger$	5.26	4.92	5.54
Shortcut Models (Frans et al., 2025)	32	8.40*	6.36*	6.65*
Shortcut Models	4	13.67*	9.08*	9.17*
Consistency Distillation (Song et al., 2023)	2	–	5.22	–
Pretrained VAE	1	136.50	139.13	69.77
Shortcut Models	1	22.08*	21.78*	24.94*
Consistency Distillation	1	–	7.80	–
PC-VAE	1	**9.95**	**6.04**	**6.49**

7 Conclusions

We introduced PC-VAE, a method for correcting the mismatch between a pre-trained VAE's latent distribution and the desired latent prior (standard Gaussian). We formulate the transport from the initial distribution to the Gaussian as the Wasserstein gradient flow of a smoothed KL functional, and we learn this flow in trajectory space using semi-implicit Picard iteration, while adapting the decoder to the transported latents. We show theoretical guarantees for both the convergence of the underlying flow to Gaussian, as well as convergence of the semi-implicit Picard iterates to the limiting flow. We empirically demonstrate improved generation quality over the pretrained VAE and competitive performance against the evaluated generative baselines, while retaining reconstruction capability.

Limitations and Future Work. Our emphasis has been more on the algorithmic and theoretical developments, with relatively small-scale experiments performed on unconditional image datasets. It would be of interest to scale PC-VAE to larger architectures and datasets.

AI USE STATEMENT

In this work, we used generative AI tools for providing critical ingredients for proving mathematical claims, assisting in the writing of proofs (verifying derivations, identifying and correcting errors), and for implementing methods used in our experiments. We have not used generative AI tools to formulate mathematical claims, to help develop theoretical models and conceptual frameworks, to propose or refine hypotheses,to design or provide feedback on research methodology or experiments, or to interpret results. Generating synthetic data sets, assisting with translation, cleaning and reformatting dataset, and qualitative or thematic data analysis are not applicable to this work. Additionally, we used generative AI tools to edit the paper for readability, to draft parts of the text, to create and edit software code, to identify relevant literature, and to check and format references. We have reviewed all AI-assisted work: every line of the paper has been examined and revised by the authors, and AI generated codes have been checked for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient flows: in metric spaces and in the space of probability measures*. Springer, 2005.

Dominique Bakry, Ivan Gentil, and Michel Ledoux. *Analysis and Geometry of Markov Diffusion Operators*, volume 348 of *Grundlehren der mathematischen Wissenschaften*. Springer, Cham, 2014. ISBN 978-3-319-00226-2. doi: 10.1007/978-3-319-00227-9.

Nicholas Matthew Boffi, Michael Samuel Albergo, and Eric Vanden-Eijnden. Flow map matching with stochastic interpolants: A mathematical framework for consistency models. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://arxiv.org/abs/2406.07507.

Haoxuan Chen, Yinuo Ren, Lexing Ying, and Grant M. Rotskoff. Accelerating diffusion models with parallel sampling: inference at sub-linear time complexity. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

Rewon Child. Very deep vaes generalize autoregressive models and can outperform them on images. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=RLRXCV6DbEJ.

Quan Dao, Hao Phung, Binh Nguyen, and Anh Tran. Flow matching in latent space, 2023. URL https://arxiv.org/abs/2307.08698.

Mingyang Deng, He Li, Tianhong Li, Yilun Du, and Kaiming He. Generative modeling via drifting, 2026. URL https://arxiv.org/abs/2602.04770.

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming Transformers for High-Resolution Image Synthesis . In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12868–12878, Los Alamitos, CA, USA, June 2021. IEEE Computer Society. doi: 10.1109/CVPR46437.2021.01268. URL https://doi.ieeecomputersociety.org/10.1109/CVPR46437.2021.01268.

Jiaojiao Fan, Qinsheng Zhang, Amirhossein Taghvaei, and Yongxin Chen. Variational Wasserstein gradient flow. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 6185–6215. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/fan22d.html.

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.12557.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2505.13447.

Ivan Gentil. Logarithmic sobolev inequality for diffusion semigroups, 2010. URL https://arxiv.org/abs/1009.3421.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 27, 2014.

Arthur Gretton, Li Kevin Wenliang, Alexandre Galashov, James Thornton, Valentin De Bortoli, and Arnaud Doucet. On the wasserstein gradient flow interpretation of drifting models, 2026. URL https://arxiv.org/abs/2605.05118.

Jiaqi Han, Puheng Li, Qiushan Guo, Renyuan Xu, Stefano Ermon, and Emmanuel J. Candès. One-step generative modeling via wasserstein gradient flows, 2026. URL https://arxiv.org/abs/2605.11755.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Sy2fzU9gl.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24):695–709, 2005. URL http://jmlr.org/papers/v6/hyvarinen05a.html.

Richard Jordan, David Kinderlehrer, and Felix Otto. The variational formulation of the fokker–planck equation. *SIAM Journal on Mathematical Analysis*, 29(1):1–17, 1998. doi: 10.1137/S0036141096303359. URL https://doi.org/10.1137/S0036141096303359.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Hk99zCeAb.

Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ODE trajectory of diffusion. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.02279.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014*, 2014. URL http://arxiv.org/abs/1312.6114.

Soheil Kolouri, Phillip E. Pope, Charles E. Martin, and Gustavo K. Rohde. Sliced wasserstein auto-encoders. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1804.01947.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.02747.

Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

Petr Mokrov, Alexander Korotin, Lingxiao Li, Aude Genevay, Justin Solomon, and Evgeny Burnaev. Large-scale wasserstein gradient flows. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://arxiv.org/abs/2106.00736.

F. Otto and C. Villani. Generalization of an inequality by talagrand and links with the logarithmic sobolev inequality. *Journal of Functional Analysis*, 173(2):361–400, 2000. ISSN 0022-1236. doi: https://doi.org/10.1006/jfan.1999.3557. URL https://www.sciencedirect.com/science/article/pii/S0022123699935577.

William Peebles and Saining Xie. Scalable Diffusion Models with Transformers . In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182, Los Alamitos, CA, USA, October 2023. IEEE Computer Society. doi: 10.1109/ICCV51070.2023.00387. URL https://doi.ieeecomputersociety.org/10.1109/ICCV51070.2023.00387.

Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/5f8e2fa1718d1bbcadf1cd9c7a54fb8c-Paper.pdf.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1530–1538, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/rezende15.html.

Herbert Robbins. An empirical bayes approach to statistics. In *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pp. 157–163. University of California Press, 1956.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bjorn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models . In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10674–10685, Los Alamitos, CA, USA, June 2022. IEEE Computer Society. doi: 10.1109/CVPR52688.2022.01042. URL https://doi.ieeecomputersociety.org/10.1109/CVPR52688.2022.01042.

Filippo Santambrogio. {Euclidean, metric, and Wasserstein} gradient flows: an overview. *Bulletin of Mathematical Sciences*, 7(1):87–154, April 2017. ISSN 1664-3615. doi: 10.1007/s13373-017-0101-1. URL https://doi.org/10.1007/s13373-017-0101-1.

Andy Shih, Suneel Belkhale, Stefano Ermon, Dorsa Sadigh, and Nima Anari. Parallel sampling of diffusion models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2305.16317.

Junhyuk So, Jiwoong Shin, Chaeyeon Jang, and Eunhyeok Park. Pcm : Picard consistency model for fast parallel sampling of diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pp. 23313–23322, June 2025.

Yang Song, Sahaj Garg, Jiaxin Shi, and Stefano Ermon. Sliced score matching: A scalable approach to density and score estimation. In *Uncertainty in artificial intelligence*, pp. 574–584. PMLR, 2020.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2011.13456.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Charles M. Stein. Estimation of the Mean of a Multivariate Normal Distribution. *The Annals of Statistics*, 9(6):1135 – 1151, 1981. doi: 10.1214/aos/1176345632. URL https://doi.org/10.1214/aos/1176345632.

Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. In *International Conference on Learning Representations*, 2018. URL https://arxiv.org/abs/1711.01558.

Erkan Turan, Nicolas Dufour, and Maks Ovsjanikov. Generative drifting is secretly score matching: a spectral and variational perspective, 2026. URL https://arxiv.org/abs/2603.09936.

Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 19667–19679. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/e3b21256183cf7c2c7a66be163579d37-Paper.pdf.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 6309–6318, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Cédric Villani. *Topics in Optimal Transportation*. Graduate Studies in Mathematics. American Mathematical Society, Providence, UNITED STATES, 2003.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

Zhisheng Xiao, Qing Yan, and Yali Amit. Generative latent flow, 2019. URL https://arxiv.org/abs/1905.10485.

Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *ArXiv*, abs/1506.03365, 2015. URL https://api.semanticscholar.org/CorpusID:8317437.

Shuangfei Zhai, Ruixiang ZHANG, Preetum Nakkiran, David Berthelot, Jiatao Gu, Huangjie Zheng, Tianrong Chen, Miguel Ángel Bautista, Navdeep Jaitly, and Joshua M. Susskind. Normalizing flows are capable generative models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://arxiv.org/abs/2412.06329.

Shengjia Zhao, Jiaming Song, and Stefano Ermon. Infovae: balancing learning and inference in variational autoencoders. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019. ISBN 978-1-57735-809-1. doi: 10.1609/aaai.v33i01.33015885. URL https://doi.org/10.1609/aaai.v33i01.33015885.

A Extended Related Work

Variational autoencoders. Variational autoencoders (VAEs) were originally introduced as a method for performing variational inference (Kingma & Welling, 2014). Classical VAE training minimizes the sum of a reconstruction loss and a KL regularizer, corresponding to the negative evidence lower bound, with the aim of encouraging the latent distribution to match a desired prior (typically a standard Gaussian). Since then, numerous methods have been proposed to encourage the latent distribution of an autoencoder to match a desired prior. Notable examples include the use of the Wasserstein distance for latent regularization (Tolstikhin et al., 2018; Kolouri et al., 2019), adversarial losses in the latent space (Makhzani et al., 2015), and an increased weight on the KL term (Higgins et al., 2017). Our method differs from these approaches in that we do not directly regularize the latent distribution. Instead, we freeze a pretrained encoder and learn an additional transport network in the latent space that maps the encoder's latent distribution to the desired distribution (standard Gaussian).

Normalizing Flows. Normalizing Flows (NF) learn an invertible mapping between the data distribution and a noise distribution (Rezende & Mohamed, 2015; Zhai et al., 2025). The mapping is learnt by maximizing the log likelihood of the samples, which is evaluated using the change of variables formula. Once learnt, inference is done by sampling the noise and then applying the inverse of the learnt mapping. NF can also be applied in the latent space (Xiao et al., 2019). NF necessarily requires the use of an invertible architecture.

Drifting Models. Drifting models (Deng et al., 2026) is a recent paradigm for one-step generative modeling. The main principle is to evolve the generator's distribution during training by moving the samples along a suitably constructed drift field, and updating the generator by matching it to the drifted samples. In Deng et al. (2026), the drift field is constructed as the difference between an attraction field and a repulsive field. The attraction field is computed using a kernel-weighted average over the target distribution, while the repulsive field is computed analogously for the current distribution. Under a Gaussian kernel, it has been established that the drift field is precisely the (smoothed) score difference between the target and current distributions (Turan et al., 2026). By noting that the score difference is equal to the Wasserstein gradient of the KL divergence functional (Gretton et al., 2026), drifting models can be interpreted as simulating a Wasserstein Gradient flow from an initial distribution (such as Gaussian) toward the data distribution during training. Different drift fields can be constructed by computing the Wasserstein gradient of different functionals (Han et al., 2026). Our work is related to this line of research, but differs in two ways. First, drifting models typically learn the flow by viewing it as a time-dynamic (i.e., by iterating over many time steps along the trajectory), whereas we interpret the dynamic in the trajectory-space, and learn the flow by updating the entire trajectory using Picard iteration. Second, we flow from the data distribution toward noise, whereas drifting models flow in the opposite direction. Flowing toward a Gaussian allows us to obtain theoretical convergence guarantees, whereas analogous guarantees cannot generally be obtained for an arbitrary target distribution. Moreover, flowing toward noise allows us to use powerful reconstruction losses through the decoder.

Wasserstein Gradient Flows. Wasserstein gradient flows describe gradient descent with respect to a functional on the space of probability distributions, and WGF based generative models attempt to solve these flows directly in the distribution space (Ambrosio et al., 2005). A classical discretization scheme is the Jordan–Kinderlehrer–Otto (JKO) scheme (Jordan et al., 1998), which constructs a sequence of distributions through implicit updates. Each JKO step involves solving an optimization problem over the distribution space (Mokrov et al., 2021; Fan et al., 2022). In contrast, we formulate the flow as a mean field ODE and approximate its entire continuous-time trajectory using Picard iteration, rather than through successive JKO steps.

Picard Iteration. Explicit Picard iteration has recently been used in generative modeling to parallelize sampling in diffusion models (Shih et al., 2023; So et al., 2025; Chen et al., 2024). While usual inference in diffusion/flow models involves flowing along the learnt trajectory by starting from noise and iteratively denoising over many time steps, inference can also be done by using Picard's iteration to iteratively update the trajectory until convergence. Each trajectory update step involves evaluation of the velocity field on discretized time-steps along the trajectory, whose computation can be parallelized. To our knowledge, our work is the first to use Picard iteration itself as a training procedure for learning the target flow.

Score Matching. Score matching refers to estimating the gradient of the log-density of a distribution from its samples. It was introduced in Hyvärinen (2005) with an algorithm commonly called Implicit Score Matching, which does not require knowing the data density or its score. Denoising Score Matching (DSM) learns the score of a Gaussian-perturbed distribution using an L_2 regression objective based on Tweedie's formula (Vincent, 2011), and is widely used to train score-based diffusion models (Song et al., 2021). Other score matching approaches include Sliced Score Matching (Song et al., 2020) which uses projections along random vectors to reduce computational cost compared to Implicit Score Matching. Our method uses DSM to estimate the smoothed scores required by the WGF (elaborated in Section 4).

Consistency models and Flow Maps. Consistency models (Song et al., 2023) are trained to map points on the same probability-flow ODE trajectory to a common endpoint, enabling one-step generation. They can either be trained from scratch or by distilling a pre-trained diffusion model. This idea can be extended to mappings between any two arbitrary times (Kim et al., 2024; Boffi et al., 2025), enabling both single and multi-step generation. Related approaches include shortcut models (Frans et al., 2025) and MeanFlow (Geng et al., 2025), with shortcut models trained on a self-consistency objective that matches one large time step with two successive time steps, while MeanFlow parameterizes the average velocity between two times and learns it through an identity relating the average and instantaneous velocities. Our method also shares a similar goal of learning the trajectories of an appropriate ODE, however our training dynamics is different: rather than using consistency-based losses, we iteratively refine trajectories by first learning a vector field using samples from the current trajectory, followed by updating trajectories using this vector field (based on Picard iteration). Additionally, our method is *on-policy*: we do not require samples from the actual ODE trajectories.

B Extended Background

B.1 Wasserstein Gradient Flows

The following is an informal discussion on the key ideas of Wasserstein Gradient Flows which are relevant for us. For a rigorous treatment on Wasserstein Gradient Flows, we refer the reader to Ambrosio et al. (2005) and Santambrogio (2017).

Let $\mathcal{P}_2(\mathbb{R}^d)$ denote the space of probability distributions on $\mathbb{R}^d$ with finite second moments, and let $\mathcal{F} : \mathcal{P}_2(\mathbb{R}^d) \to \mathbb{R}$ be a functional on this space.

For a given $p \in \mathcal{P}_2(\mathbb{R}^d)$, the first variation of $\mathcal{F}$ at p, if it exists, is a function $\frac{\delta \mathcal{F}}{\delta p} : \mathbb{R}^d \to \mathbb{R}$ which is determined by the following relation (up to an additive constant)

$$\frac{d}{dh}\mathcal{F}(p + h\chi)\Big|_{h=0} = \int_{\mathbb{R}^d} \frac{\delta \mathcal{F}}{\delta p}(x) \cdot \chi(x) dx, \tag{13}$$

for any mass-preserving perturbation χ (i.e., $\int_{\mathbb{R}^d} \chi(x)dx = 0$). Intuitively, the first variation is a notion of derivative of $\mathcal{F}$ at p: if p is slightly perturbed by Δp, then $\mathcal{F}(p + \Delta p) \approx \mathcal{F}(p) + \int \frac{\delta \mathcal{F}}{\delta p}(x) \cdot \Delta p(x) dx$.

Now, the Wasserstein Gradient Flow with respect to the functional $\mathcal{F}$ (with initial condition p_0) is given by the continuity equation

$$\partial_t p_t(x) = \operatorname{div}\left(p_t(x) \cdot \nabla \frac{\delta \mathcal{F}}{\delta p_t}(x)\right), \tag{14}$$

which corresponds to the particle ODE

$$\frac{d\mathbf{x}_t}{dt} = -\nabla \frac{\delta \mathcal{F}}{\delta p_t}(\mathbf{x}_t), \qquad p_t := p_{\mathbf{x}_t} = \operatorname{Law}(\mathbf{x}_t) \tag{15}$$

with initial condition $\mathbf{x}_0 \sim p_0$. The velocity field depends on both the particle's current state ($\mathbf{x}_t$) and its current law ($p_{\mathbf{x}_t}$), making this a mean field ODE. Intuitively, this can be interpreted as the dynamics of a given particle being affected by the entire ensemble of particles.

We illustrate this with a simple but important example. Consider the KL divergence functional $\mathcal{F}(p) = D_{\mathrm{KL}}(p\|q) = \int \log \frac{p(x)}{q(x)} dp(x)$, where q is a reference distribution. Then, for any perturba-

tion χ we have (ignoring technicalities)

$$\frac{d}{dh}\mathcal{F}(p+h\chi)\Big|_{h=0} = \frac{d}{dh}\left(\int \log\frac{p(x)+h\chi(x)}{q(x)} d(p+h\chi)(x)\right)\Bigg|_{h=0} \tag{16}$$

$$= \lim_{h\to 0}\frac{1}{h}\left(\int \log\frac{p(x)+h\chi(x)}{q(x)} d(p+h\chi)(x) - \int \log\frac{p(x)}{q(x)} dp(x)\right) \tag{17}$$

$$= \lim_{h\to 0}\frac{1}{h}\left(\int \log\frac{p(x)+h\chi(x)}{p(x)} dp(x) + h\cdot\int \log\frac{p(x)+h\chi(x)}{q(x)}\chi(x)dx\right) \tag{18}$$

$$= \int \lim_{h\to 0}\left(\frac{1}{h}\log\frac{p(x)+h\chi(x)}{p(x)}\right) dp(x) + \int \log\frac{p(x)}{q(x)}\chi(x)dx \tag{19}$$

$$= \int \chi(x)dx + \int \log\frac{p(x)}{q(x)}\chi(x)dx = \int\left(\log\frac{p(x)}{q(x)}+1\right)\chi(x)dx, \tag{20}$$

and hence $\frac{\delta\mathcal{F}}{\delta p} = \log\frac{p(x)}{q(x)} + 1$. Thus, its WGF is given by

$$\frac{d\mathbf{x}_t}{dt} = -\nabla\frac{\delta\mathcal{F}}{\delta p_t}(\mathbf{x}_t) = -\nabla\log\frac{p_t(\mathbf{x}_t)}{q(\mathbf{x}_t)}, \qquad p_t := p_{\mathbf{x}_t}. \tag{21}$$

We now consider the density evolution of the above WGF:

$$\partial_t p_t(x) = \operatorname{div}\left(p_t(x)\cdot\nabla\frac{\delta\mathcal{F}}{\delta p_t}(x)\right) \tag{22}$$

$$= \operatorname{div}\left(p_t(x)\cdot(-\nabla\log q(x) + \nabla\log p_t(x))\right) \tag{23}$$

$$= -\operatorname{div}\left(p_t(x)\cdot\nabla\log q(x)\right) + \operatorname{div}\left(p_t(x)\cdot\nabla\log p_t(x)\right) \tag{24}$$

$$= -\operatorname{div}\left(p_t(x)\cdot\nabla\log q(x)\right) + \operatorname{div}\left(\nabla p_t(x)\right) \tag{25}$$

$$= -\operatorname{div}\left(p_t(x)\cdot\nabla\log q(x)\right) + \Delta p_t(x), \tag{26}$$

which is precisely the Fokker-Planck equation for the density evolution of the Langevin dynamics $d\mathbf{x}_t = \nabla\log q(\mathbf{x}_t)dt + \sqrt{2}\cdot d\mathbf{w}_t$ (where $\mathbf{w}_t$ is the standard Brownian motion).

An important fact about the WGF with respect to any functional $\mathcal{F}$ is that $\mathcal{F}(p_t)$ is non-increasing in t. This can be shown using the continuity equation and divergence theorem as follows:

$$\frac{d}{dt}\mathcal{F}(p_t) = \int \frac{\delta\mathcal{F}}{\delta p_t}(x)\cdot\partial_t p_t(x)dx \tag{27}$$

$$= \int \frac{\delta\mathcal{F}}{\delta p_t}(x)\cdot\operatorname{div}\left(p_t(x)\cdot\nabla\frac{\delta\mathcal{F}}{\delta p_t}(x)\right)dx \tag{28}$$

$$= -\int\langle\nabla\frac{\delta\mathcal{F}}{\delta p_t}(x), \nabla\frac{\delta\mathcal{F}}{\delta p_t}(x)\rangle\cdot p_t(x)dx = -\mathbb{E}\left[\left\|\nabla\frac{\delta\mathcal{F}}{\delta p_t}(\mathbf{x}_t)\right\|^2\right] \le 0. \tag{29}$$

B.2 Picard's Iteration

Picard's iteration solves an ODE by recursively updating the entire trajectory using the current trajectory. Consider an ODE

$$\frac{dx_t}{dt} = f(t, x_t), \qquad x_0 = \bar{x}. \tag{30}$$

If x_t is a solution to the above ODE in a specified time interval $[0, T]$, then it is equivalently a fixed point of the trajectory-valued operator

$$(\mathcal{T}x)_t := \bar{x} + \int_0^t f(s, x_s)ds. \tag{31}$$

Starting from the constant trajectory $x_t^0 = \bar{x}$, Picard iteration successively defines $x_t^{n+1} = (\mathcal{T}x^n)_t$, or equivalently,

$$\frac{dx_t^{n+1}}{dt} = f(t, x_t^n), \qquad x_0^{n+1} = \bar{x}. \tag{32}$$

Each Picard step thus updates the entire trajectory $t \mapsto x_t^n$ on the interval of interest using the previous trajectory. Under certain Lipschitz and regularity conditions, the Picard iterates converge in the L^∞ norm to the fixed point trajectory, and this trajectory is the unique solution to the ODE.

The same fixed-point construction can be extended to the mean field ODE

$$\frac{d\mathbf{x}_t}{dt} = b(t, \mathbf{x}_t, p_t), \qquad p_t := p_{\mathbf{x}_t}, \tag{33}$$

with initial condition $\mathbf{x}_0 \sim p_0$.

Let $\mathbf{x}_t^0 = \mathbf{x}_0$ and write $p_t^n := p_{\mathbf{x}_t^n} = \mathrm{Law}(\mathbf{x}_t^n)$. Then, Picard's iteration is

$$\frac{d\mathbf{x}_t^{n+1}}{dt} = b(t, \mathbf{x}_t^n, p_t^n), \tag{34}$$

where each iterate shares the same initial starting state ($\mathbf{x}_0^n = \mathbf{x}_0$ for all n). Equivalently in integral form,

$$\mathbf{x}_t^{n+1} = \mathbf{x}_0 + \int_0^t b(s, \mathbf{x}_s^n, p_s^n) ds. \tag{35}$$

We note that both the state and measure arguments on the right hand side are evaluated on the previous trajectory x_t^n. Under suitable Lipschitz assumptions in both the state and measure arguments, the iterates converge to the fixed point solution.

B.3 Semi-Implicit Picard Iteration

In this section we introduce *semi-implicit* Picard iteration, where the contractive linear component of the velocity field, if it exists, is evaluated on the current (updated) trajectory rather than the previous trajectory, unlike standard Picard iteration where the entire velocity field is evaluated on the previous trajectory. We then explain our motivation through an illustrative error-propagation analysis, followed by an example where a standard Picard iterate blows up in finite time while semi-implicit Picard iterates converge to the solution uniformly on all finite intervals.

Consider an ODE

$$\frac{dx_t}{dt} = f(t, x_t) = -kx_t - v(t, x_t), \qquad x_0 = \bar{x}, k > 0, \tag{36}$$

i.e., the velocity field consists of a contractive linear component. Let x_t denote the solution to the above ODE in a specified time interval (say, $[0, T]$). Now, standard Picard iteration as described in the previous section is given by the recursion

$$\frac{dx_t^{n+1}}{dt} = -kx_t^n - v(t, x_t^n), \qquad x_0^{n+1} = \bar{x} \tag{37}$$

$$\iff \quad x_t^{n+1} = \bar{x} + \int_0^t (-kx_s^n - v(s, x_s^n)) ds, \tag{38}$$

whereas semi-implicit Picard iteration is given by the recursion

$$\frac{dx_t^{n+1}}{dt} = -kx_t^{n+1} - v(t, x_t^n), \qquad x_0^{n+1} = \bar{x}, \tag{39}$$

i.e., the linear component of the velocity field is evaluated on the current trajectory (x_t^{n+1}) rather than the previous trajectory (x_t^n). Using integration factor, we can rewrite this recursion as $\frac{d}{dt}(e^{kt} x_t^{n+1}) = -e^{kt} v(t, x_t^n)$, or equivalently in integral form,

$$x_t^{n+1} = e^{-kt}\bar{x} - \int_0^t e^{k(s-t)} v(s, x_s^n) ds. \tag{40}$$

Illustrative Error Analysis. For illustration, assume that the velocity field $v(t, x)$ is globally $L-$Lipschitz in x, for all $t \in [0, T]$. We denote the L_∞ norm of a trajectory $\{y_t\}_{t\in[0,T]}$ by $\|y\|_\infty := \sup_{0\le t\le T} \|y_t\|$.

For standard Picard iteration, we analyze the error in the updated trajectory as

$$\|x_t^{n+1} - x_t\| = \left\| \left(\bar{x} - \int_0^t (kx_s^n + v(s, x_s^n))ds \right) - \left(\bar{x} - \int_0^t (kx_s + v(s, x_s))ds \right) \right\| \tag{41}$$

$$\leq \int_0^t k\|x_s^n - x_s\|ds + \int_0^t \|v(s, x_s^n) - v(s, x_s)\|ds \tag{42}$$

$$\leq \int_0^t (k + L) \cdot \|x_s^n - x_s\| \, ds \tag{43}$$

$$\leq (k + L)t \cdot \sup_{0 \leq s \leq t} \|x_s^n - x_s\| \tag{44}$$

$$\implies \left\|x^{n+1} - x\right\|_\infty \leq (k + L)T \cdot \|x^n - x\|_\infty \,, \tag{45}$$

whereas for semi-implicit Picard iteration, we have

$$\|x_t^{n+1} - x_t\| = \left\| \left(e^{-kt}\bar{x} - \int_0^t e^{k(s-t)}v(s, x_s^n)ds \right) - \left(e^{-kt}\bar{x} - \int_0^t e^{k(s-t)}v(s, x_s)ds \right) \right\| \tag{46}$$

$$\leq \int_0^t e^{k(s-t)} \cdot \|v(s, x_s^n) - v(s, x_s)\|ds \tag{47}$$

$$\leq \int_0^t e^{k(s-t)} \cdot L \, \|x_s^n - x_s\| \, ds \tag{48}$$

$$\leq \frac{L}{k} \left(1 - e^{-kt}\right) \cdot \sup_{0 \leq s \leq t} \|x_s^n - x_s\| \tag{49}$$

$$\implies \left\|x^{n+1} - x\right\|_\infty \leq \frac{L}{k} \left(1 - e^{-kT}\right) \cdot \|x^n - x\|_\infty \,. \tag{50}$$

We thus note that the semi-implicit Picard iteration yields a smaller error propagation factor compared to standard Picard iteration (since $\frac{L}{k}(1 - e^{-kT}) \leq \frac{L}{k}(kT) < (L + k)T$), and moreover, using the inequality $1 - e^{-kT} \leq 1$, we observe that the multiplicative factor in semi-implicit Picard iteration does not depend on T, unlike in standard Picard iteration where it scales linearly in T. Intuitively, this happens because the factor $e^{k(s-t)}$ exponentially down-weights the errors arising from portions of the previous trajectory that are not very close to the current time t; such a down-weighting term is absent in equation 45. This exponential damping results in a more favorable mechanism for propagating errors compared to standard Picard iteration, thereby leading to improved finite-iteration stability. We illustrate this stability advantage numerically with a scalar ODE in Figure 2.

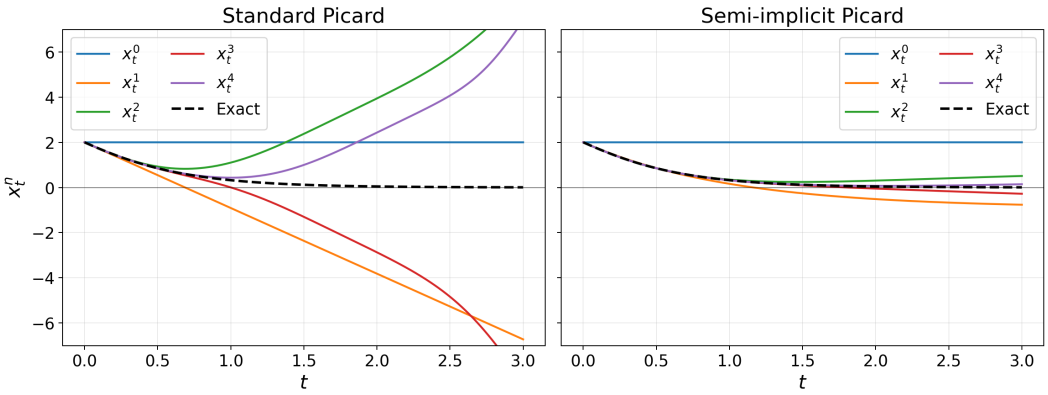

Figure 2: Standard versus semi-implicit Picard iteration for a scalar ODE $\frac{dx_t}{dt} = -x_t - \sin(x_t)$ with initial condition $x_0 = 2$. The dashed curve denotes the exact solution.

Example of divergence of Standard Picard Iteration. We highlight an example where standard Picard iteration in fact diverges (i.e., it blows up in finite time) whereas semi-implicit Picard iteration converges.

We consider the exact (unsmoothed) KL functional $\mathcal{G}(p) = D_{\mathrm{KL}}(p\|q)$ where $q = \mathcal{N}(0, I)$ is the standard Gaussian. Its WGF is given by

$$\frac{d\mathbf{x}_t}{dt} = -\nabla \frac{\delta \mathcal{G}}{\delta p_t}(\mathbf{x}_t) = -\mathbf{x}_t - \nabla \log p_t(\mathbf{x}_t), \qquad p_t := p_{\mathbf{x}_t}. \tag{51}$$

We consider a Gaussian initial condition $\mathbf{x}_0 \sim p_0$ where $p_0 = \mathcal{N}(0, s^2 I)$, $s > 1$. Since the score of the Gaussian $\mathcal{N}(0, vI)$ evaluated at x is $-x/v$ (linear in x), it is easy to see that the Gaussian structure is preserved along this WGF, i.e., we may write $\mathbf{x}_t = c_t \cdot \mathbf{x}_0$ for a scalar function c_t. Then, the equivalent ODE for c_t is given by

$$\frac{dc_t}{dt} = -c_t + \frac{c_t}{c_t^2 s^2} = -c_t + \frac{1}{c_t s^2}, \qquad c_0 = 1. \tag{52}$$

The solution to the above ODE can be explicitly computed as $c_t = \sqrt{\frac{1}{s^2} + e^{-2t}\left(1 - \frac{1}{s^2}\right)}$ for all $t \geq 0$. As a sanity check, $c_t \to \frac{1}{s}$ as $t \to \infty$, which means $\mathbf{x}_t \to \frac{1}{s}\mathbf{x}_0 \sim \mathcal{N}(0, I)$.

We now show that the standard Picard iteration for the above ODE will diverge. Starting with the initial trajectory $c_t^0 = 1$ for all t (where the superscript denotes the iteration index), we have

$$\frac{dc_t^1}{dt} = -c_t^0 + \frac{1}{c_t^0 s^2} = -1 + \frac{1}{s^2}, \qquad c_0^1 = 1 \tag{53}$$

$$\implies c_t^1 = 1 + t \cdot \left(-1 + \frac{1}{s^2}\right). \tag{54}$$

In particular, we note that $c_t^1 = 0$ when $t = \frac{1}{1 - \frac{1}{s^2}}$. Now, when solving for the next iterate $dc_t^2/dt = -c_t^1 + \frac{1}{c_t^1 s^2}$, we obtain

$$c_t^2 = 1 - \left(t - \frac{t^2}{2}A\right) - \frac{1}{As^2}\log(1 - At), \qquad A := 1 - \frac{1}{s^2}, \tag{55}$$

and hence the second iterate c_t^2 blows up to $+\infty$ as $t \uparrow A^{-1}$, even though the actual solution $c_t = \sqrt{\frac{1}{s^2} + e^{-2t}\left(1 - \frac{1}{s^2}\right)}$ is well defined and bounded for all $t \geq 0$. Consequently, standard Picard iteration cannot be carried out beyond the second iterate on an interval $[0, T]$ where $T > A^{-1}$.

We contrast this with semi-implicit Picard iteration for the same ODE, which is given by

$$\frac{dc_t^{n+1}}{dt} = -c_t^{n+1} + \frac{1}{c_t^n s^2}, \qquad c_0^{n+1} = 1 \tag{56}$$

$$\iff c_t^{n+1} = e^{-t} + \int_0^t e^{\tau - t} \cdot \frac{1}{c_\tau^n s^2} d\tau. \tag{57}$$

We will show that the semi-implicit Picard iterates (with initial trajectory $c_t^0 = 1$ for all $t \geq 0$) will converge uniformly to the solution c_t on any interval $[0, T]$.

Firstly, we can show by induction that $c_t^n \in [s^{-2}, 1]$ for all $t \geq 0$ and for all $n \geq 0$. The base case $n = 0$ is trivial. For the induction step, assume $c_t^n \in [s^{-2}, 1]$ for all $t \geq 0$. Then, $\frac{1}{c_t^n s^2} \in [s^{-2}, 1]$ for all $t \geq 0$, and hence

$$e^{-t} + \frac{1}{s^2}\int_0^t e^{\tau - t} d\tau \leq c_t^{n+1} \leq e^{-t} + \int_0^t e^{\tau - t} d\tau \tag{58}$$

$$\implies e^{-t} + \frac{1}{s^2}(1 - e^{-t}) \leq c_t^{n+1} \leq e^{-t} + (1 - e^{-t}) \tag{59}$$

$$\implies \frac{1}{s^2} \leq c_t^{n+1} \leq 1. \tag{60}$$

Hence $c_t^n \in [s^{-2}, 1]$ for all n, and similarly the solution $c_t \in [s^{-2}, 1]$. By noting that the map $u \mapsto 1/u$ is s^4-Lipschitz on $[s^{-2}, 1]$, we have

$$|c_t^{n+1} - c_t| \leq \int_0^t e^{\tau - t} \left| \frac{1}{c_\tau^n s^2} - \frac{1}{c_\tau s^2} \right| d\tau \tag{61}$$

$$\leq s^2 \int_0^t e^{\tau - t} |c_\tau^n - c_\tau| d\tau \tag{62}$$

$$\leq s^2 \int_0^t |c_\tau^n - c_\tau| d\tau \tag{63}$$

$$\leq s^2 \int_0^t \sup_{0 \leq t' \leq \tau} |c_{t'}^n - c_{t'}| d\tau \tag{64}$$

$$\implies \sup_{0 \leq t' \leq t} |c_{t'}^{n+1} - c_{t'}| \leq s^2 \int_0^t \sup_{0 \leq t' \leq \tau} |c_{t'}^n - c_{t'}| d\tau. \tag{65}$$

By repeated integration and noting that $|c_t^0 - c_t| \leq 1$ for all $t \geq 0$ we obtain

$$\sup_{0 \leq t' \leq t} |c_{t'}^n - c_{t'}| \leq \frac{(s^2 t)^n}{n!} \to 0 \qquad \text{as } n \to \infty, \tag{66}$$

and since t is arbitrary, we thus have that $c_t^n \to c_t$ uniformly on any finite time interval $[0, T]$.

C Auxiliary Results

Here, we collect some results which will be used in later proofs.

Lemma C.1 (Tweedie's formula, Robbins (1956))**.** *Let $\mathbf{x}$ be a random vector, $\mathbf{z} \sim \mathcal{N}(0, I)$ be independent of $\mathbf{x}$, and let $\mathbf{y} = a\mathbf{x} + \sigma\mathbf{z}$, where $a, \sigma \in \mathbb{R}$ and $\sigma > 0$. Let $p_{\mathbf{y}}$ denote the law of $\mathbf{y}$. Then, for any $y \in \mathbb{R}^d$,*

$$\nabla \log p_{\mathbf{y}}(y) = -\frac{y}{\sigma^2} + \frac{a}{\sigma^2} \mathbb{E}[\mathbf{x} | \mathbf{y} = y]. \tag{67}$$

Lemma C.2 (Stein's Lemma/Gaussian integration by parts, Stein (1981))**.** *Let $f : \mathbb{R} \to \mathbb{R}$ be a differentiable function, and let $\mathrm{z} \sim \mathcal{N}(0, 1)$. Then, assuming $\mathbb{E}[|f'(\mathrm{z})|] < \infty$ we have*

$$\mathbb{E}[f'(\mathrm{z})] = \mathbb{E}[\mathrm{z} f(\mathrm{z})]. \tag{68}$$

In the multi-dimensional setting, if $f : \mathbb{R}^d \to \mathbb{R}$ is differentiable and $\mathbb{E}[\|\nabla f(\mathbf{z})\|] < \infty$ where $\mathbf{z} \sim \mathcal{N}(0, I)$ then

$$\mathbb{E}[\nabla f(\mathbf{z})] = \mathbb{E}[\mathbf{z} f(\mathbf{z})]. \tag{69}$$

Lemma C.3 (Gaussian Log-Sobolev Inequality (LSI), Gentil (2010))**.** *Let q denote the standard normal distribution $\mathcal{N}(0, I)$ in $\mathbb{R}^d$, and let p be a distribution on $\mathbb{R}^d$ that is absolutely continuous with respect to q. Then,*

$$I(p \| q) \geq 2 \cdot D_{\mathrm{KL}}(p \| q), \tag{70}$$

where $I(p\|q) := \mathbb{E}_{X \sim p}\left[\left\| \nabla \log \frac{p(X)}{q(X)} \right\|^2 \right]$ denotes the Fisher divergence between p and q.

Lemma C.4 (Talagrand's T_2 inequality, Otto & Villani (2000))**.** *Let q denote the standard normal distribution $\mathcal{N}(0, I)$ in $\mathbb{R}^d$, and let p be an arbitrary distribution on $\mathbb{R}^d$ with finite second moment. Then,*

$$W_2^2(p, q) \leq 2 \cdot D_{\mathrm{KL}}(p \| q). \tag{71}$$

Lemma C.5 (Properties of Ornstein-Uhlenbeck Semigroup, Gentil (2010))**.** *Let $(P_t)_{t \in [0, \infty)}$ denote a family of operators which are defined by their action on a function $f : \mathbb{R}^d \to \mathbb{R}$ as*

$$(P_t f)(x) = \mathbb{E}_{\mathbf{z} \sim \mathcal{N}(0, I)} \left[f\left(e^{-t} x + \sqrt{1 - e^{-2t}} \mathbf{z} \right) \right] \tag{72}$$

The family $(P_t)_{t \in [0, \infty)}$ is called the Ornstein-Uhlenbeck (OU) semigroup, and its infinitesimal generator is given by the operator $L := \Delta - \langle x, \nabla \rangle$. Denoting $q = \mathcal{N}(0, I)$, the following are true:

1. *For any two functions f, g and for any $t \in [0, \infty)$ we have*

$$\int f \cdot (P_t g) dq = \int (P_t f) \cdot g dq \tag{73}$$

2. *P_t commutes with L for any $t \in [0, \infty)$, i.e.,*

$$L(P_t f) = P_t(Lf). \tag{74}$$

3. *For any two functions f, g, we have*

$$\int f \cdot (Lg) dq = \int (Lf) \cdot g dq = -\int \langle \nabla f, \nabla g \rangle dq. \tag{75}$$

We refer the reader to Bakry et al. (2014) for a detailed exposition on Markov semigroups.

D WGF OF $\mathcal{F}(p) = D_{\mathrm{KL}}(p_\epsilon \| q)$

Let $a := 1 - \epsilon$, and p_ϵ denote the law of $(1-\epsilon)\mathbf{x} + \sigma\mathbf{z}$, where $\mathbf{x} \sim p$ and $\mathbf{z} \sim q = \mathcal{N}(0, I)$ is independent of $\mathbf{x}$. For convenience, let $\bar{p}$ denote the density of $(1-\epsilon)\mathbf{x}$. We then have

$$\bar{p}(x) = \frac{1}{a^d} p\left(\frac{x}{a}\right). \tag{76}$$

Also, let ϕ_σ denote the density of $\mathcal{N}(0, \sigma^2 I)$. We have

$$p_\epsilon(x) = \int \bar{p}(y) \phi_\sigma(x - y) dy \tag{77}$$

$$= \int \frac{1}{a^d} p\left(\frac{y}{a}\right) \phi_\sigma(x - y) dy \tag{78}$$

$$= \int p(u) \cdot \phi_\sigma(x - au) du. \tag{79}$$

Now suppose we perturb p in a mass-preserving direction δp, so that $\int \delta p(u) du = 0$. The corresponding perturbation of p_ϵ is

$$\delta p_\epsilon(x) = \int \delta p(u) \cdot \phi_\sigma(x - au) du. \tag{80}$$

Let $p^\tau := p + \tau \delta p$. Then,

$$\left.\frac{d}{d\tau} \mathcal{F}(p^\tau)\right|_{\tau=0} = \int \left(\log \frac{p_\epsilon(x)}{q(x)} + 1\right) \delta p_\epsilon(x) dx \tag{81}$$

$$= \int \log \frac{p_\epsilon(x)}{q(x)} \delta p_\epsilon(x) dx \tag{82}$$

$$= \int \int \log \frac{p_\epsilon(x)}{q(x)} \delta p(u) \phi_\sigma(x - au) du\, dx \tag{83}$$

$$= \int du\, \delta p(u) \left(\int \log \frac{p_\epsilon(x)}{q(x)} \phi_\sigma(x - au) dx \right) \tag{84}$$

$$= \int du\, \delta p(u) \left(\int \log \frac{p_\epsilon(y + au)}{q(y + au)} \phi_\sigma(y) dy \right) \tag{85}$$

$$= \int du\, \delta p(u)\, \mathbb{E}_{\mathbf{y} \sim \mathcal{N}(0, \sigma^2 I)} \left[\log \frac{p_\epsilon(\mathbf{y} + au)}{q(\mathbf{y} + au)} \right]. \tag{86}$$

In the second line, we used $\int \delta p_\epsilon(x) dx = 0$. Hence, up to an additive constant, the first variation is

$$\frac{\delta \mathcal{F}}{\delta p}(x) = \mathbb{E}_{\mathbf{y} \sim \mathcal{N}(0, \sigma^2 I)} \left[\log \frac{p_\epsilon(\mathbf{y} + ax)}{q(\mathbf{y} + ax)} \right] \tag{87}$$

$$= \mathbb{E}_{\mathbf{z} \sim \mathcal{N}(0, I)} \left[\log \frac{p_\epsilon(ax + \sigma\mathbf{z})}{q(ax + \sigma\mathbf{z})} \right]. \tag{88}$$

Taking the gradient with respect to x and noting that $q(y) = (2\pi)^{-d/2} \cdot \exp(-\|y\|^2/2)$ gives

$$\nabla \frac{\delta \mathcal{F}}{\delta p}(x) = -a \cdot \mathbb{E}[\nabla \log q(ax + \sigma \mathbf{z})] + a \cdot \mathbb{E}[\nabla \log p_\epsilon(ax + \sigma \mathbf{z})] \tag{89}$$

$$= -a \cdot \mathbb{E}[-(ax + \sigma \mathbf{z})] + a \cdot \mathbb{E}[\nabla \log p_\epsilon(ax + \sigma \mathbf{z})] \tag{90}$$

$$= a^2 x + a \cdot \mathbb{E}[\nabla \log p_\epsilon(ax + \sigma \mathbf{z})]. \tag{91}$$

E Incompatibility between Picard iteration and Distillation

The goal of distillation here is to learn a trajectory map $\Psi(\mathbf{x}_0, t)$ which approximates the state $\mathbf{x}_t$ obtained by integrating the velocity field of an ODE till time t (where the velocity field is supplied by a pre-trained model), starting from $\mathbf{x}_0$. Once trained, $\mathbf{x}_t$ can be generated from $\mathbf{x}_0$ using a single neural network call, rather than having to integrate the velocity field which requires multiple evaluations of the pre-trained model. Here, we explain briefly why distilling a pre-trained flow or diffusion model using Picard iteration can lead to a distribution mismatch which may lead to unreliable training in practice.

We illustrate this with the WGF of exact KL functional $\mathcal{G}(p) = D_{\mathrm{KL}}(p\|q)$, which is given by

$$\frac{d\mathbf{x}_t}{dt} = -\mathbf{x}_t - \nabla \log p_t(\mathbf{x}_t), \qquad p_t := p_{\mathbf{x}_t}. \tag{92}$$

As shown in Appendix B.1, the marginals of this WGF coincide with those of the OU process, $d\mathbf{x}_t = -\mathbf{x}_t dt + \sqrt{2} \cdot d\mathbf{w}_t$ (where $\mathbf{w}_t$ is the standard Brownian motion), and the OU marginals can be described explicitly as

$$p_t = \mathrm{Law}(e^{-t}\mathbf{x}_0 + \sqrt{1 - e^{-2t}}\mathbf{z}), \tag{93}$$

where $\mathbf{x}_0 \sim p_0, \mathbf{z} \sim q = \mathcal{N}(0, I)$ are independent. That is, $\mathbf{x}_t$ is marginally equivalent to a Gaussian smoothed version of the initial random vector $\mathbf{x}_0$. Thus, as is typically done in diffusion model training, we learn $\nabla \log p_t$ using the DSM loss, based on Tweedie's formula

$$\mathcal{L}_{\mathrm{DSM}}(\phi) = \mathbb{E}_{\mathbf{x}_0, t, \mathbf{z}}\left[\left\| s_\phi(\widetilde{\mathbf{x}}_t, t) + \frac{\mathbf{z}}{\sqrt{1 - e^{-2t}}} \right\|^2\right], \qquad \widetilde{\mathbf{x}}_t = e^{-t}\mathbf{x}_0 + \sqrt{1 - e^{-2t}}\mathbf{z}. \tag{94}$$

The exact population minimizer of $\mathcal{L}_{\mathrm{DSM}}(\phi)$ will yield $s_\phi(x, t) = \nabla \log p_t(x)$ for all x, t. In practice, the empirical version of the above loss is minimized using finitely many samples. Thus, for a given t, $s_\phi(x, t)$ is a good approximation to $\nabla \log p_t(x)$ when x belongs to high-probability regions under p_t, and the approximation may not be good in low probability regions. More generally, a small L^2 error need not imply that the point-wise errors are small.

Now, if we replace $\nabla \log p_t$ with $s_\phi(\cdot, t)$, then the ODE becomes

$$\frac{d\mathbf{x}_t}{dt} = -\mathbf{x}_t - s_\phi(\mathbf{x}_t, t), \tag{95}$$

i.e., this is no longer a mean field ODE, since the velocity field has no explicit dependence on the distribution of $\mathbf{x}_t$. Therefore, semi-implicit Picard iteration is now given by

$$\frac{d\mathbf{x}_t^{n+1}}{dt} = -\mathbf{x}_t^{n+1} - s_\phi(\mathbf{x}_t^n, t), \tag{96}$$

and hence, when implementing semi-implicit Picard iteration using the trajectory map Ψ_θ, we minimize the loss

$$\mathcal{L}_{\mathrm{flow}}(\theta) = \mathbb{E}\left[\left\| \partial_t \Psi_\theta(\mathbf{x}_0, t) - \left(-\Psi_\theta(\mathbf{x}_0, t) - s_\phi(\Psi_{\widehat{\theta}}(\mathbf{x}_0, t), t)\right)\right\|^2\right]. \tag{97}$$

In particular, we note that for a given t, s_ϕ is evaluated on $\Psi_{\widehat{\theta}}(\mathbf{x}_0, t)$ in the above loss. During initial stages of training, the marginal distribution of $\Psi_{\widehat{\theta}}(\mathbf{x}_0, t)$ will typically have significant mismatch with p_t, and as we mentioned earlier, s_ϕ may not approximate well when $\Psi_{\widehat{\theta}}(\mathbf{x}_0, t)$ has low probability under p_t. Due to this mismatch, the initial training dynamics may have significant errors compared to the correct dynamics, and this error can propagate through later iterations as well.

F Proof of Theorem 1

We recall that $a = 1 - \epsilon$, $a^2 + \sigma^2 = 1$ and $p_\epsilon = \mathrm{Law}(a\mathbf{x} + \sigma\mathbf{z})$ where $\mathbf{x} \sim p, \mathbf{z} \sim q = \mathcal{N}(0, I)$ are independent. The WGF corresponding to the functional $\mathcal{F}(p) = D_{\mathrm{KL}}(p_\epsilon \| q)$ is given by

$$\frac{d\mathbf{x}_t}{dt} = -\nabla \frac{\delta \mathcal{F}}{\delta p_t}(\mathbf{x}_t) = -a^2 \mathbf{x}_t - a\mathbb{E}_{\mathbf{z}\sim q}[\nabla \log p_{t,\epsilon}(a\mathbf{x}_t + \sigma\mathbf{z})], \qquad p_t := p_{\mathbf{x}_t}. \tag{98}$$

We split the proof into two parts. In Lemma F.1 we show that $\mathcal{F}(p_t) \to 0$ as $O(1/t)$, along the WGF. Then, in Lemma F.3 we show that $W_2(p_t, q) \to 0$ as $t \to \infty$.

Lemma F.1. *Assume that $\mathbf{x}_0 \sim p_0$ has finite KL divergence with respect to the standard Gaussian (i.e., $D_{\mathrm{KL}}(p_0 \| q) < \infty$). Then, denoting $p_t := p_{\mathbf{x}_t}$, we have $\mathcal{F}(p_t) \le \frac{D_{\mathrm{KL}}(p_0\|q)}{2t}$ along the WGF with respect to $\mathcal{F}$ (equation 98), and in particular $\mathcal{F}(p_t) \to 0$ as $t \to \infty$.*

Proof. Denote K_a to be the OU Markov operator which acts on a function f as

$$(K_a f)(x) = \mathbb{E}_{\mathbf{z}\sim q}[f(ax + \sigma\mathbf{z})]. \tag{99}$$

Using the notation of Lemma C.5 we have $K_a = P_{-\log a}$, and we denote $L := \Delta - \langle x, \nabla \rangle$ to be the OU generator.

Now, the action of K_a on the density ratio p/q (where p is an arbitrary probability density) is given by

$$\begin{aligned}
\left(K_a \frac{p}{q}\right)(x) &= \mathbb{E}_{\mathbf{z}\sim q}\left[\frac{p(ax + \sigma\mathbf{z})}{q(ax + \sigma\mathbf{z})}\right] && (100)\\
&= \int \frac{p(ax + \sigma z)}{q(ax + \sigma z)} q(z) dz && (101)\\
&= \int p(ax + \sigma z) \exp\left(-\frac{(\|z\|^2 - \|ax + \sigma z\|^2)}{2}\right) dz && (102)\\
&= \int p(ax + \sigma z) \exp\left(-\frac{(a^2\|z\|^2 - a^2\|x\|^2 - 2a\sigma\langle x, z\rangle)}{2}\right) dz && (103)\\
&= \int p(ax + \sigma z) \exp\left(\frac{\|x\|^2}{2}\right) \exp\left(-\frac{(a^2\|z\|^2 + \sigma^2\|x\|^2 - 2a\sigma\langle x, z\rangle)}{2}\right) dz && (104)\\
&= \int p(ax + \sigma z) \exp\left(\frac{\|x\|^2}{2}\right) \exp\left(-\frac{\|\sigma x - az\|^2}{2}\right) dz && (105)\\
&= \int \frac{p(ax + \sigma z)}{q(x)} q(\sigma x - az) dz && (106)\\
&= \frac{1}{q(x)} \int \frac{1}{a^d} p\left(\frac{x - \sigma u}{a}\right) q(u) du \quad \text{(substituted } u = \sigma x - az\text{)} && (107)\\
&= \frac{p_\epsilon(x)}{q(x)}, && (108)
\end{aligned}$$

i.e., $K_a \frac{p}{q} = \frac{p_\epsilon}{q}$.

We denote $\mathcal{G}(p) := D_{\mathrm{KL}}(p\|q)$. Its first variation is given by

$$\frac{\delta \mathcal{G}}{\delta p}(x) = \log \frac{p(x)}{q(x)} \tag{109}$$

We will now track $\mathcal{G}(p_t)$ along the WGF. Recall that for the WGF with respect to $\mathcal{F}$, the following continuity equation holds

$$\partial_t p_t(x) = \mathrm{div}(p_t(x) \nabla \frac{\delta \mathcal{F}}{\delta p_t}(x)). \tag{110}$$

Hence,

$$\partial_t \mathcal{G}(p_t) = \int \frac{\delta \mathcal{G}}{\delta p}(x) \partial_t p_t(x) dx \tag{111}$$

$$= \int \log \frac{p_t(x)}{q(x)} \cdot \operatorname{div}(p_t(x) \nabla \frac{\delta \mathcal{F}}{\delta p_t}(x)) dx \tag{112}$$

$$= -\int p_t(x) \langle \nabla \log \frac{p_t(x)}{q(x)}, \nabla \frac{\delta \mathcal{F}}{\delta p_t}(x) \rangle dx \tag{113}$$

$$= -\int \langle \nabla \frac{p_t(x)}{q(x)}, \nabla \frac{\delta \mathcal{F}}{\delta p_t}(x) \rangle \cdot q(x) dx \tag{114}$$

$$= \int \frac{p_t(x)}{q(x)} \cdot L \left(\frac{\delta \mathcal{F}}{\delta p_t} \right)(x) \cdot q(x) dx, \tag{115}$$

where in the first step we used chain rule, in the third step we used the divergence theorem, and in the last step we used Lemma C.5 (specifically, $\int \langle \nabla f, \nabla g \rangle dq = -\int f \cdot L g dq$).

Now, from Appendix D we have

$$\frac{\delta \mathcal{F}}{\delta p}(x) = \left(K_a \log \frac{p_\epsilon}{q} \right)(x) \tag{116}$$

and hence

$$\partial_t \mathcal{G}(p_t) = \int \frac{p_t(x)}{q(x)} \cdot L \left(K_a \log \frac{p_{t,\epsilon}}{q} \right)(x) \cdot q(x) dx \tag{117}$$

$$= \int \frac{p_t(x)}{q(x)} \cdot K_a \left(L \log \frac{p_{t,\epsilon}}{q} \right)(x) \cdot q(x) dx \tag{118}$$

$$= \int (K_a \frac{p_t}{q})(x) \cdot \left(L \log \frac{p_{t,\epsilon}}{q} \right)(x) \cdot q(x) dx \tag{119}$$

$$= \int \frac{p_{t,\epsilon}(x)}{q(x)} \cdot \left(L \log \frac{p_{t,\epsilon}}{q} \right)(x) \cdot q(x) dx \tag{120}$$

$$= -\int \langle \nabla \frac{p_{t,\epsilon}}{q}(x), \nabla \log \frac{p_{t,\epsilon}}{q} \rangle q(x) dx \tag{121}$$

$$= -\int \| \nabla \log \frac{p_{t,\epsilon}}{q}(x) \|^2 p_{t,\epsilon}(x) dx \tag{122}$$

$$= -I(p_{t,\epsilon} \| q), \tag{123}$$

where in the second step we used commutativity of K_a and L (Lemma C.5), in the third step we used self-adjointness of K_a (Lemma C.5), in the fourth step we used the earlier established fact that $K_a \frac{p}{q} = \frac{p_\epsilon}{q}$, and in the fifth step we used $\int f \cdot L g dq = -\int \langle \nabla f, \nabla g \rangle dq$ (Lemma C.5). Here, $I(p\|q)$ denotes the Fisher divergence between p and q.

Now, from Gaussian LSI (Lemma C.3) we have $I(p\|q) \geq 2 D_{\mathrm{KL}}(p\|q)$. Hence,

$$\partial_t \mathcal{G}(p_t) \leq -2 \cdot D_{\mathrm{KL}}(p_{t,\epsilon} \| q) = -2\mathcal{F}(p_t) \tag{124}$$

$$\implies \mathcal{G}(p_t) \leq \mathcal{G}(p_0) - 2 \int_0^t \mathcal{F}(p_s) ds \leq \mathcal{G}(p_0) - 2t \cdot \mathcal{F}(p_t) \tag{125}$$

where we used the fact that $\mathcal{F}(p_t)$ is non-increasing in t (property of Wasserstein gradient flow). But we also note that $\mathcal{G}(p_t) \geq 0$ for all t. Thus, we get

$$\mathcal{F}(p_t) \leq \frac{\mathcal{G}(p_0)}{2t} = \frac{D_{\mathrm{KL}}(p_0 \| q)}{2t}, \tag{126}$$

and hence $\mathcal{F}(p_t) \to 0$ as $t \to \infty$, with rate $O(1/t)$. □

We now need to show that $W_2(p_t, q) \to 0$ as $t \to \infty$. We will require the following lemma.

Lemma F.2. *For a distribution μ, denote μ_ϵ to be the law of $a\mathbf{x} + \sigma\mathbf{z}$, where $\mathbf{x} \sim \mu$, $\mathbf{z} \sim q$ independent of $\mathbf{x}$. If $\{\mu_n\}_{n\geq 1} \subseteq \mathcal{P}_2(\mathbb{R}^d)$ is a sequence of distributions that satisfies*

$$W_2(\mu_{n,\epsilon}, q) \longrightarrow 0, \tag{127}$$

then

$$W_2(\mu_n, q) \longrightarrow 0. \tag{128}$$

Proof. In order to show $W_2(\mu_n, q) \to 0$, it is sufficient to show that μ_n converges weakly to q, and that the second moments of μ_n converge to that of q (Villani (2003)).

We first prove weak convergence. For a probability measure μ, its characteristic function is given by

$$\widehat{\mu}(\xi) := \mathbb{E}_{\mathbf{x}\sim\mu}[e^{i\langle\xi,\mathbf{x}\rangle}] = \int_{\mathbb{R}^d} e^{i\langle\xi,x\rangle}\, d\mu(x) \tag{129}$$

Now, we have

$$\widehat{\mu_{n,\epsilon}}(\xi) = \mathbb{E}_{\mathbf{x}_n\sim\mu_n,\mathbf{z}\sim q}[e^{i\langle\xi,a\mathbf{x}_n+\sigma\mathbf{z}\rangle}] \tag{130}$$

$$= \widehat{\mu_n}(a\xi)\exp\left(-\frac{\sigma^2\|\xi\|^2}{2}\right) \tag{131}$$

$$\implies \widehat{\mu_n}(\eta) = \exp\left(\frac{\sigma^2\|\eta\|^2}{2a^2}\right)\widehat{\mu_{n,\epsilon}}\left(\frac{\eta}{a}\right). \tag{132}$$

Since $\mu_{n,\epsilon} \to q$ in W_2 and convergence in W_2 implies weak convergence, we have

$$\widehat{\mu_{n,\epsilon}}(\xi) \longrightarrow \widehat{q}(\xi) = \exp\left(-\frac{\|\xi\|^2}{2}\right) \tag{133}$$

for every $\xi \in \mathbb{R}^d$. Hence, for every $\eta \in \mathbb{R}^d$, equation 132 gives

$$\widehat{\mu_n}(\eta) \longrightarrow \exp\left(\frac{\sigma^2\|\eta\|^2}{2a^2} - \frac{\|\eta\|^2}{2a^2}\right) = \exp\left(-\frac{\|\eta\|^2}{2}\right) = \widehat{q}(\eta), \tag{134}$$

where we used $1 - \sigma^2 = a^2$. Lévy's continuity theorem therefore implies μ_n converges weakly to q.

It remains to prove convergence of second moments. If $\mathbf{x}_n \sim \mu_n$ and $\mathbf{z} \sim q$ are independent, then

$$\mathbb{E}\|a\mathbf{x}_n + \sigma\mathbf{z}\|^2 = a^2\mathbb{E}\|\mathbf{x}_n\|^2 + \sigma^2 d. \tag{135}$$

Because $\mu_{n,\epsilon} \to q$ in W_2, their second moments converge to that of q, giving

$$\mathbb{E}\|a\mathbf{x}_n + \sigma\mathbf{z}\|^2 \longrightarrow d. \tag{136}$$

Substituting this into equation 135 and using $1 - \sigma^2 = a^2$ gives

$$\mathbb{E}\|\mathbf{x}_n\|^2 \longrightarrow d. \tag{137}$$

Thus, μ_n converges weakly to q and its second moments converge to the second moment of q. This proves that $W_2(\mu_n, q) \to 0$. □

Lemma F.3. *For the WGF with respect to $\mathcal{F}$, $W_2(p_t, q) \to 0$ as $t \to \infty$.*

Proof. Using Talagrand T_2 inequality (Lemma C.4), we have for every $t > 0$,

$$W_2^2(p_{t,\epsilon}, q) \leq 2D_{\mathrm{KL}}(p_{t,\epsilon}\|q) \tag{138}$$

$$= 2\mathcal{F}(p_t) \tag{139}$$

$$\leq \frac{D_{\mathrm{KL}}(p_0\|q)}{t}. \tag{140}$$

In particular,

$$W_2(p_{t,\epsilon}, q) \longrightarrow 0 \qquad \text{as } t \longrightarrow \infty. \tag{141}$$

Now let $\{t_n\}_{n\geq 1}$ be any sequence such that $t_n \to \infty$. Applying Lemma F.2 with $\mu_n = p_{t_n}$ gives

$$W_2(p_{t_n}, q) \longrightarrow 0. \tag{142}$$

Since this holds for every sequence $t_n \to \infty$, we conclude that

$$W_2(p_t, q) \longrightarrow 0 \qquad \text{as } t \longrightarrow \infty. \tag{143}$$

□

F.1 Failure of a uniform PL Inequality

Here, we discuss why we cannot obtain a uniform PL type inequality for the functional $\mathcal{F}$, i.e., there does not exist $c > 0$ for which

$$\mathbb{E}_{\mathbf{x}\sim p}\left[\left\|\nabla\frac{\delta\mathcal{F}}{\delta p}(\mathbf{x})\right\|^2\right] \geq c\cdot\mathcal{F}(p), \quad \text{for all } p. \tag{144}$$

Consider a family of distributions $\{p_v\}_{v>0}$ on $\mathbb{R}^d$, defined by $p_v = \mathcal{N}(0, vI)$, i.e. p_v is a $d-$dimensional zero-mean Gaussian with covariance vI. Then, we have $p_{v,\epsilon} = \mathcal{N}(0, (a^2v+\sigma^2)I)$, whose score is given by

$$\nabla \log p_{v,\epsilon}(y) = -\frac{y}{a^2v+\sigma^2}, \tag{145}$$

and hence

$$\nabla\frac{\delta\mathcal{F}}{\delta p_v}(x) = a^2x + a\cdot\mathbb{E}_{\mathbf{z}\sim\mathcal{N}(0,I)}[\nabla\log p_{v,\epsilon}(ax+\sigma\mathbf{z})] \tag{146}$$

$$= a^2x + a\cdot\left(-\frac{ax}{a^2v+\sigma^2}\right) \tag{147}$$

$$= a^2x\left(-\frac{1}{a^2v+\sigma^2}+1\right). \tag{148}$$

$$\implies \mathbb{E}_{\mathbf{x}\sim p_v}\left[\left\|\nabla\frac{\delta\mathcal{F}}{\delta p_v}(\mathbf{x})\right\|^2\right] = a^4\cdot\left(\frac{1}{a^2v+\sigma^2}-1\right)^2\cdot\mathbb{E}_{\mathbf{x}\sim p_v}[\|\mathbf{x}\|^2] \tag{149}$$

$$= a^4\cdot\left(\frac{1}{a^2v+\sigma^2}-1\right)^2\cdot dv, \tag{150}$$

and in particular, we have $\mathbb{E}_{\mathbf{x}\sim p_v}\left[\left\|\nabla\frac{\delta\mathcal{F}}{\delta p_v}(\mathbf{x})\right\|^2\right] \to 0$ as $v\to 0$.

Now, we have

$$\mathcal{F}(p_v) = D_{\mathrm{KL}}(p_{v,\epsilon}\|q) \tag{151}$$

$$= d\cdot D_{\mathrm{KL}}\big(\mathcal{N}(0,a^2v+\sigma^2)\|\mathcal{N}(0,1)\big) \tag{152}$$

$$= \frac{d}{2}\cdot\left(-\log(a^2v+\sigma^2)+a^2v+\sigma^2-1\right), \tag{153}$$

and in particular, $\mathcal{F}(p_v)\to\frac{d}{2}(-2\log\sigma+\sigma^2-1) > 0$ as $v\to 0$ (since $\epsilon\in(0,1)$, we strictly have $\sigma\in(0,1)$).

Hence, we observe that the ratio $\frac{\mathbb{E}_{\mathbf{x}\sim p_v}\left[\left\|\nabla\frac{\delta\mathcal{F}}{\delta p_v}(\mathbf{x})\right\|^2\right]}{\mathcal{F}(p_v)} \to 0$ as $v\to 0$, implying that there is no $c>0$ for which $\frac{\mathbb{E}_{\mathbf{x}\sim p_v}\left[\left\|\nabla\frac{\delta\mathcal{F}}{\delta p_v}(\mathbf{x})\right\|^2\right]}{\mathcal{F}(p_v)} \geq c$ for all $v\in(0,1)$. This shows that we cannot have a uniform PL type inequality for $\mathcal{F}$.

G Proof of Theorem 2

Notation. For a random element $\mathbf{w}$ taking values in a normed vector space $(X, \|\cdot\|_X)$, we denote its L^r norm ($1\leq r<\infty$) by

$$\|\mathbf{w}\|_{L^r} := \mathbb{E}[\|\mathbf{w}\|_X^r]^{1/r}. \tag{154}$$

In this section, X is either $\mathbb{R}$, $\mathbb{R}^d$ (with Euclidean 2-norm), or the space of continuous functions from $[0,T]$ to $\mathbb{R}^d$ (denoted by $C([0,T];\mathbb{R}^d)$), equipped with the sup-norm $\|\mathbf{w}\|_X := \sup_{0\leq t\leq T}\|\mathbf{w}_t\|$. The underlying space is clear from the context.

As before, we denote $a := 1 - \epsilon$, $a^2 + \sigma^2 = 1$, and for a probability distribution p on $\mathbb{R}^d$, p_ϵ denotes the law of $a\mathbf{x} + \sigma\mathbf{z}$, where $\mathbf{x} \sim p$, $\mathbf{z} \sim q = \mathcal{N}(0, I)$, and $\mathbf{x}$ and $\mathbf{z}$ are independent. In particular, for this section, $\mathbf{z} \sim q$ will represent an auxiliary Gaussian random vector that is independent of the underlying random vectors under consideration, and for an arbitrary function f we write $\mathbb{E}_{\mathbf{z}}[f(a\mathbf{x} + \sigma\mathbf{z})]$ to mean that the expectation is only with respect to $\mathbf{z} \sim \mathcal{N}(0, I)$, keeping $\mathbf{x}$ fixed.

We denote the scalar and vector standard Gaussian moments respectively by

$$c_r := \mathbb{E}_{\mathrm{z}\sim\mathcal{N}(0,1)}[|\mathrm{z}|^r]^{1/r}, \tag{155}$$

$$m_{d,r} := \mathbb{E}_{\mathbf{z}\sim\mathcal{N}(0,I)}[\|\mathbf{z}\|^r]^{1/r}. \tag{156}$$

We define

$$s_p(y) := \nabla \log p_\epsilon(y), \qquad g(x, p) := \mathbb{E}_{\mathbf{z}\sim q}\left[s_p(ax + \sigma\mathbf{z})\right], \tag{157}$$

and the individual components of $s_p(y)$ are denoted $s_{p,i}(y), i = 1, \dots d$.

The WGF with respect to $\mathcal{F}(p) = D_{\mathrm{KL}}(p_\epsilon \| q)$ can then be written as

$$\frac{d\mathbf{x}_t}{dt} = -\nabla \frac{\delta \mathcal{F}}{\delta p_t}(\mathbf{x}_t) = -a^2\mathbf{x}_t - a g(\mathbf{x}_t, p_t), \qquad p_t := p_{\mathbf{x}_t}, \mathbf{x}_0 \sim p_0 \tag{158}$$

and denoting $p_t^n := p_{\mathbf{x}_t^n}$, the semi-implicit Picard iteration is

$$\frac{d\mathbf{x}_t^{n+1}}{dt} = -a^2\mathbf{x}_t^{n+1} - a\, g(\mathbf{x}_t^n, p_t^n), \qquad \mathbf{x}_0^{n+1} = \mathbf{x}_0, \tag{159}$$

with initial trajectory $\mathbf{x}_t^0 = \mathbf{x}_0$. Equivalently, using the integrating factor we have

$$\mathbf{x}_t^{n+1} = e^{-a^2 t}\mathbf{x}_0 - a \int_0^t e^{-a^2(t-s)} g(\mathbf{x}_s^n, p_s^n)\, ds. \tag{160}$$

Proof Outline. We first establish a Lipschitz estimate in Lemma G.6 which controls $\|g(\mathbf{x}, p_{\mathbf{x}}) - g(\widetilde{\mathbf{x}}, p_{\widetilde{\mathbf{x}}})\|_{L^r}$ by $\|\mathbf{x} - \widetilde{\mathbf{x}}\|_{L^u}$, where $u > r \geq 2$. Specifically, we show

$$\|g(\mathbf{x}, p_{\mathbf{x}}) - g(\widetilde{\mathbf{x}}, p_{\widetilde{\mathbf{x}}})\|_{L^r} \leq O\left(\sqrt{ds} + \sqrt{r}\right) \cdot \|\mathbf{x} - \widetilde{\mathbf{x}}\|_{L^u}, \qquad \frac{1}{r} = \frac{1}{s} + \frac{1}{u}. \tag{161}$$

To obtain the above bound, we define $\mathbf{x}_\theta := \mathbf{x} + \theta(\widetilde{\mathbf{x}} - \mathbf{x})$ and $p_\theta := p_{\mathbf{x}_\theta}$; then $g(\widetilde{\mathbf{x}}, p_{\widetilde{\mathbf{x}}}) - g(\mathbf{x}, p_{\mathbf{x}}) = \int_0^1 \frac{d}{d\theta} g(\mathbf{x}_\theta, p_\theta) \cdot d\theta$, and hence by Minkowski's inequality we reduce the problem to upper bounding $\left\| \frac{d}{d\theta} g(\mathbf{x}_\theta, p_\theta) \right\|_{L^r}$. Now, $\frac{d}{d\theta} g(\mathbf{x}_\theta, p_\theta)$ consists of two terms (derived in detail in Lemma G.5):

- The first term, due to variation of g in x, is $\nabla_x g(\mathbf{x}_\theta, p_\theta)(\widetilde{\mathbf{x}} - \mathbf{x})$; we upper bound its L_r norm using Hölder's inequality as $\left\| \|\nabla_x g(\mathbf{x}_\theta, p_\theta)\|_{\mathrm{op}} \right\|_{L_s} \cdot \|\widetilde{\mathbf{x}} - \mathbf{x}\|_{L_u}$ (where $\|\cdot\|_{\mathrm{op}}$ denotes the operator norm), and Lemma G.4 provides the necessary upper bound on the L_s norm of $\|\nabla_x g(\mathbf{x}_\theta, p_\theta)\|_{\mathrm{op}}$.
- The second term, due to the variation of $g(x, p)$ in p, is given by $\mathbb{E}_{\mathbf{z}}[(\partial_\theta s_{p_\theta})(a\mathbf{x}_\theta + \sigma\mathbf{z})]$. Using the identity $p_{\theta,\epsilon}(y) = \mathbb{E}[\phi_\sigma(y - a\mathbf{x}_\theta)]$ where ϕ_σ is the density of $\mathcal{N}(0, \sigma^2 I)$, we obtain

  $$(\partial_\theta s_{p_\theta})(y) = \frac{a}{\sigma} \nabla_y f_\theta(y), \qquad f_\theta(y) := \mathbb{E}[\langle \mathbf{z}, \widetilde{\mathbf{x}} - \mathbf{x} \rangle | a\mathbf{x}_\theta + \sigma\mathbf{z} = y], \tag{162}$$

  and hence $\mathbb{E}_{\mathbf{z}}[(\partial_\theta s_{p_\theta})(a\mathbf{x}_\theta + \sigma\mathbf{z})] = \frac{a}{\sigma}\mathbb{E}_{\mathbf{z}}[\nabla f_\theta(a\mathbf{x}_\theta + \sigma\mathbf{z})] = \frac{a}{\sigma^2}\mathbb{E}_{\mathbf{z}}[\mathbf{z} f_\theta(a\mathbf{x}_\theta + \sigma\mathbf{z})]$ using Gaussian integration by parts. Lemma G.3 then allows us to upper bound the L_r norm of $\mathbb{E}_{\mathbf{z}}[\mathbf{z} f_\theta(a\mathbf{x}_\theta + \sigma\mathbf{z})]$ by the L^r norm of $f_\theta(a\mathbf{x}_\theta + \sigma\mathbf{z})$, and using conditional Jensen's inequality and tower property we upper bound further by $\|\widetilde{\mathbf{x}} - \mathbf{x}\|_{L^r}$ (which is trivially upper bounded by $\|\widetilde{\mathbf{x}} - \mathbf{x}\|_{L^u}$ as $u \geq r$).

Combining the bounds on the two terms gives us the Lipschitz bound in Lemma G.6.

We then establish in Lemma G.7 that the semi-implicit Picard iterates $\mathbf{x}_t^n$ form a Cauchy sequence in $L^2(\Omega; C([0, T]; \mathbb{R}^d))$ (and hence converge to a trajectory $\mathbf{x}_t$) by upper bounding the L^2 norm

of successive Picard difference as $\mathbb{E}\left[\sup_{0\le t\le T}\left\|\mathbf{x}_t^{n+1}-\mathbf{x}_t^n\right\|^2\right]^{1/2} \le O((kT)^n/\sqrt{n!})$ for some constant k, and noting that $(kT)^n/\sqrt{n!}$ is summable in n.

To obtain this bound, we use the integral form of the semi-implicit Picard update followed by Minkowski's inequality for the L^r norm to obtain

$$\mathbb{E}[\sup_{0\le t'\le t}\left\|\mathbf{x}_{t'}^{n+1}-\mathbf{x}_{t'}^n\right\|^r]^{1/r} \le a\cdot\int_0^t \left\|g(\mathbf{x}_s^n,p_s^n)-g(\mathbf{x}_s^{n-1},p_s^{n-1})\right\|_{L^r} ds. \tag{163}$$

The idea now is to recursively apply Lemma G.6 to the RHS; this involves raising the moment exponents at each step. Specifically, define the sequence $r_j = 2n/j, \quad j = 1\dots n$ (in particular, $r_n = 2$ and $r_1 = 2n$). At step j, we apply Lemma G.6 with $r = r_{n-j+1}, u = r_{n-j}$ and $s = 2n$ (which gives us a uniform Lipschitz constant over all steps, that behaves like $O(\sqrt{dn})$). Solving the resulting recursive inequality gives us the desired bound.

Finally, we show that the limiting trajectory $\mathbf{x}_t$ satisfies the WGF equation 5 in Lemma G.8.

We now give the detailed proof.

Lemma G.1. *We have $c_r \le \sqrt{r}$ and $m_{d,r} \le \sqrt{dr}$ for all $r \ge 2$. Additionally, $c_r \le 1$ for all $1 \le r < 2$.*

Proof. For $r = 2$ we have $c_r = 1 \le \sqrt{2}$. Now suppose $r > 2$. Define $\psi : \mathbb{R} \to \mathbb{R}$ as $\psi(z) = z\cdot|z|^{r-2}$. Then, we have $\psi'(z) = (r-1)|z|^{r-2}$. Let $\mathrm{z} \sim \mathcal{N}(0,1)$ be the standard scalar Gaussian. We have

$$\begin{aligned}\mathbb{E}[|\mathrm{z}|^r] &= \mathbb{E}[\mathrm{z}\cdot\psi(\mathrm{z})] &(164)\\ &= \mathbb{E}[\psi'(\mathrm{z})] &(165)\\ &= \mathbb{E}[(r-1)|z|^{r-2}] \le r\cdot\mathbb{E}[|\mathrm{z}|^{r-2}], &(166)\end{aligned}$$

where in the second step we used the Gaussian integration by parts (which is justified because $\psi(z)$ grows polynomially in z). By propagating the above recurrence k times (where $k = \lceil r/2\rceil - 1$ is the unique non-negative integer satisfying $r - 2k \in (0,2]$), we have

$$\begin{aligned}\mathbb{E}[|\mathrm{z}|^r] &\le r^k\cdot\|\mathrm{z}\|_{L^{r-2k}}^{r-2k} &(167)\\ &\le r^k\cdot\|\mathrm{z}\|_{L^2}^{r-2k} = r^k \le r^{r/2}, &(168)\end{aligned}$$

and taking the rth root on both sides gives $c_r \le \sqrt{r}$.

Now let $\mathbf{z} \sim \mathcal{N}(0, I)$ be the d-dimensional standard Gaussian. We have

$$\begin{aligned}m_{d,r}^2 &= \mathbb{E}[\|\mathbf{z}\|^r]^{2/r} &(169)\\ &= \mathbb{E}[(\|\mathbf{z}\|^2)^{r/2}]^{2/r} &(170)\\ &= \mathbb{E}\left[\left(\sum_{i=1}^d |\mathrm{z}_i|^2\right)^{r/2}\right]^{2/r} &(171)\\ &\le \sum_{i=1}^d \mathbb{E}[|\mathrm{z}_i|^{2\cdot r/2}]^{2/r} = dc_r^2, &(172)\end{aligned}$$

where in the last step we used Minkowski's inequality for the $L^{r/2}$ norm (which is valid since $r/2 \ge 1$). Thus, taking square root on both sides gives $m_{d,r} \le \sqrt{d}c_r \le \sqrt{dr}$.

Finally, for $1 \le r < 2$, we have $c_r \le c_2 = 1$ using monotonicity of norms.

□

Lemma G.2. *Let $\mathbf{x} \sim p$ be any random vector, and let $\mathbf{y} = a\mathbf{x} + \sigma\mathbf{z}$ where $\mathbf{z} \sim q = \mathcal{N}(0, I)$ is independent of $\mathbf{x}$. Then, for any $r \geq 1$ we have*

$$\mathbb{E}[\|s_p(\mathbf{y})\|^r]^{1/r} \leq \frac{m_{d,r}}{\sigma}, \tag{173}$$

$$\mathbb{E}[|s_{p,j}(\mathbf{y})|^r]^{1/r} \leq \frac{c_r}{\sigma}, \qquad 1 \leq j \leq d, \tag{174}$$

$$\|g(\mathbf{x}, p)\|_{L^r} \leq \frac{m_{d,r}}{\sigma}, \tag{175}$$

where $s_{p,j}(y)$ denotes the jth component of the vector $s_p(y)$.

Proof. From Tweedie's formula (Lemma C.1), we have

$$s_p(\mathbf{y}) = -\frac{1}{\sigma}\mathbb{E}[\mathbf{z} \mid \mathbf{y}]. \tag{176}$$

Using conditional Jensen's inequality and the tower property, we have

$$\mathbb{E}[\|s_p(\mathbf{y})\|^r] \leq \sigma^{-r}\mathbb{E}[\mathbb{E}[\|\mathbf{z}\|^r \mid \mathbf{y}]] = \sigma^{-r} m_{d,r}^r, \tag{177}$$

$$\mathbb{E}[|s_{p,j}(\mathbf{y})|^r] \leq \sigma^{-r}\mathbb{E}[\mathbb{E}[|\mathbf{z}_j|^r \mid \mathbf{y}]] = \sigma^{-r} c_r^r. \tag{178}$$

Moreover, independence of $\mathbf{z}$ and $\mathbf{x}$ means that

$$g(\mathbf{x}, p) = \mathbb{E}[s_p(\mathbf{y}) \mid \mathbf{x}], \tag{179}$$

$$\implies \mathbb{E}[\|g(\mathbf{x}, p)\|^r] \leq \mathbb{E}[\mathbb{E}[\|s_p(\mathbf{y})\|^r \mid \mathbf{x}]] \leq \sigma^{-r} m_{d,r}^r. \tag{180}$$

Taking rth roots gives us the desired bounds. □

Lemma G.3. *Let $\mathbf{x}$ be a random vector, and let $\mathbf{y} = a\mathbf{x} + \sigma\mathbf{z}$ where $\mathbf{z} \sim q = \mathcal{N}(0, I)$ is independent of $\mathbf{x}$. Let $r \geq 2$ and $r' = r/(r-1)$. Then, for any function $f : \mathbb{R}^d \to \mathbb{R}$ with $\mathbb{E}[|f(\mathbf{y})|^r] < \infty$, we have*

$$\|\mathbb{E}_{\mathbf{z}}[\mathbf{z} f(a\mathbf{x} + \sigma\mathbf{z})]\|_{L^r} \leq c_{r'} \cdot \mathbb{E}[|f(\mathbf{y})|^r]^{1/r} \tag{181}$$

Proof. Let $g(\mathbf{x}) = \mathbb{E}_{\mathbf{z}}[\mathbf{z} f(a\mathbf{x} + \sigma\mathbf{z})] := \mathbb{E}[\mathbf{z} f(\mathbf{y}) \big| \mathbf{x}]$. Then, using the fact that $\|v\| = \sup_{w:\|w\|=1} \langle w, v\rangle$ for any vector $v \in \mathbb{R}^d$, and using Hölder's inequality we have

$$\|g(\mathbf{x})\| = \big\|\mathbb{E}[\mathbf{z} f(\mathbf{y}) \big| \mathbf{x}]\big\| \tag{182}$$

$$= \sup_{w:\|w\|=1} \mathbb{E}[\langle w, \mathbf{z}\rangle \cdot f(\mathbf{y}) \big| \mathbf{x}] \tag{183}$$

$$\leq \sup_{w:\|w\|=1} \Big(\mathbb{E}[|\langle w, \mathbf{z}\rangle|^{r'} \big| \mathbf{x}]^{1/r'} \cdot \mathbb{E}[|f(\mathbf{y})|^r \big| \mathbf{x}]^{1/r}\Big) \tag{184}$$

$$\leq \sup_{w:\|w\|=1} \Big(\mathbb{E}[|\langle w, \mathbf{z}\rangle|^{r'}]^{1/r'} \cdot \mathbb{E}[|f(\mathbf{y})|^r \big| \mathbf{x}]^{1/r}\Big) \tag{185}$$

$$= c_{r'} \cdot \mathbb{E}[|f(\mathbf{y})|^r \big| \mathbf{x}]^{1/r}, \tag{186}$$

where in the last line we used the fact that $\langle w, \mathbf{z}\rangle \sim \mathcal{N}(0, 1)$ for any $\|w\| = 1$. Hence,

$$\big\|\mathbb{E}[\mathbf{z} f(\mathbf{y}) \big| \mathbf{x}]\big\|_{L^r} = \|g(\mathbf{x})\|_{L^r} \tag{187}$$

$$= \mathbb{E}[\|g(\mathbf{x})\|^r]^{1/r} \tag{188}$$

$$\leq \mathbb{E}\left[c_{r'}^r \cdot \mathbb{E}[|f(\mathbf{y})|^r \big| \mathbf{x}]\right]^{1/r} \tag{189}$$

$$= c_{r'} \cdot \mathbb{E}[|f(\mathbf{y})|^r]^{1/r}. \tag{190}$$

□

Lemma G.4. *Define*

$$J_p(x) := \nabla_x g(x, p) = a \cdot \mathbb{E}_{\mathbf{z}\sim q}[\nabla^2 \log p_\epsilon(ax + \sigma\mathbf{z})]. \tag{191}$$

Then, for any $\mathbf{x} \sim p$ and $r \geq 2$, we have

$$\mathbb{E}[\|J_p(\mathbf{x})\|_{\mathrm{op}}^r]^{1/r} \leq \frac{a\sqrt{d}}{\sigma^2} c_r c_{r'} \leq \frac{a}{\sigma^2}\sqrt{dr} \qquad r' = \frac{r}{r-1}, \tag{192}$$

where $\|A\|_{\mathrm{op}}$ denotes the operator norm of a matrix, defined by $\|A\|_{\mathrm{op}} := \sup_{x\neq 0} \frac{\|Ax\|}{\|x\|}$.

Proof. Let $s_{p,i}(x), i = 1, \dots d$ denote the components of the smoothed score $s_p(x) = \nabla \log p_\epsilon(x)$. Then, since the Hessian is symmetric, we have

$$(J_p(x))_{ij} = a \cdot \mathbb{E}[\partial_i s_{p,j}(ax + \sigma \mathbf{z})] \tag{193}$$

$$= \frac{a}{\sigma} \cdot \mathbb{E}[\partial_{z_i} s'_{p,j}(\mathbf{z})] \qquad (s'_{p,i}(z) := s_{p,i}(ax + \sigma z)) \tag{194}$$

$$= \frac{a}{\sigma} \cdot \mathbb{E}[\mathbf{z}_i s'_{p,j}(\mathbf{z})] \tag{195}$$

$$= \frac{a}{\sigma} \cdot \mathbb{E}[\mathbf{z}_i s_{p,j}(ax + \sigma \mathbf{z})], \tag{196}$$

where in the third step we used the Gaussian integration by parts (Lemma C.2). Hence, if $e_i, i = 1, \dots d$ denote the canonical basis vectors in $\mathbb{R}^d$, we have for all j

$$J_p(x) e_j = \frac{a}{\sigma} \mathbb{E}[\mathbf{z} s_{p,j}(ax + \sigma \mathbf{z})]. \tag{197}$$

Now, by Tweedie's formula we have

$$s_p(y) = \frac{1}{\sigma} \mathbb{E}[-\mathbf{z} | \mathbf{y} = y], \tag{198}$$

and hence using conditional Jensen's inequality we obtain

$$\mathbb{E}[|s_{p,j}(\mathbf{y})|^r] \le \sigma^{-r} \mathbb{E}[|\mathbf{z}_j|^r] = \sigma^{-r} \cdot c_r^r. \tag{199}$$

Hence, using Lemma G.3 we have

$$\|J_p(\mathbf{x}) e_j\|_{L^r} \le \frac{a}{\sigma} c_{r'} \mathbb{E}[|s_{p,j}(\mathbf{y})|^r]^{1/r} \le \frac{a}{\sigma^2} c_{r'} c_r. \tag{200}$$

Finally, we note that for a matrix A, its squared operator norm is upper bounded by its squared Frobenius norm, which is given by the sum of the squared column norms (i.e., $\|A\|_{\text{op}}^2 \le \sum_i \|A e_i\|^2$). Using this, we obtain

$$\mathbb{E}[\|J_p(\mathbf{x})\|_{\text{op}}^r]^{2/r} \le \mathbb{E}\left[\left(\sum_{j=1}^d \|J_p(\mathbf{x}) e_j\|^2 \right)^{r/2} \right]^{2/r} \tag{201}$$

$$\le \sum_{j=1}^d \mathbb{E}[\|J_p(\mathbf{x}) e_j\|^r]^{2/r} \le d \left(\frac{a}{\sigma^2} c_r c_{r'} \right)^2, \tag{202}$$

where in the second step we used Minkowski's inequality for the $L^{r/2}$ norm (which is valid since $r/2 \ge 1$). Taking square root on both sides and noting that $c_r \le \sqrt{r}$ from Lemma G.1 and $c_{r'} \le c_2 = 1$ gives us the desired inequality. □

Lemma G.5. *Let $\mathbf{x}, \widetilde{\mathbf{x}}$ be two random vectors, such that $\mathbf{x} - \widetilde{\mathbf{x}} \in L^2$. Define*

$$\mathbf{h} := \widetilde{\mathbf{x}} - \mathbf{x}, \qquad \mathbf{x}_\theta := \mathbf{x} + \theta \mathbf{h}, \qquad p_\theta := p_{\mathbf{x}_\theta}, \qquad 0 \le \theta \le 1. \tag{203}$$

Let $\mathbf{z} \sim q$ be independent of $(\mathbf{x}, \widetilde{\mathbf{x}})$ (therefore making $\mathbf{z}$ independent of all $\mathbf{x}_\theta$), and write

$$\mathbf{y}_\theta := a \mathbf{x}_\theta + \sigma \mathbf{z}, \qquad f_\theta(y) := \mathbb{E}[\langle \mathbf{h}, \mathbf{z} \rangle \mid \mathbf{y}_\theta = y]. \tag{204}$$

Then

$$\frac{d}{d\theta} g(\mathbf{x}_\theta, p_\theta) = J_{p_\theta}(\mathbf{x}_\theta) \mathbf{h} + \frac{a}{\sigma^2} \mathbb{E}_{\mathbf{z}}[\mathbf{z} f_\theta(a \mathbf{x}_\theta + \sigma \mathbf{z})], \tag{205}$$

where $J_p(x)$ is defined in Lemma G.4.

Proof. We have

$$g(\mathbf{x}_\theta, p_\theta) = \mathbb{E}_{\mathbf{z}}[\nabla \log p_{\theta,\epsilon}(a \mathbf{x}_\theta + \sigma \mathbf{z})] = \mathbb{E}_{\mathbf{z}}[s_{p_\theta}(a \mathbf{x}_\theta + \sigma \mathbf{z})], \tag{206}$$

and hence

$$\frac{d}{d\theta}g(\mathbf{x}_\theta, p_\theta) = \mathbb{E}_{\mathbf{z}}[(\partial_\theta s_{p_\theta})(a\mathbf{x}_\theta + \sigma\mathbf{z}) + \nabla s_{p_\theta}(a\mathbf{x}_\theta + \sigma\mathbf{z})(a\mathbf{h})] \tag{207}$$

$$= \mathbb{E}_{\mathbf{z}}[(\partial_\theta s_{p_\theta})(a\mathbf{x}_\theta + \sigma\mathbf{z})] + J_{p_\theta}(\mathbf{x}_\theta)\mathbf{h}. \tag{208}$$

Now, we have

$$p_{\theta,\epsilon}(y) = \mathbb{E}[\phi_\sigma(y - a\mathbf{x}_\theta)], \tag{209}$$

where $\phi_\sigma(u) = (2\pi\sigma^2)^{-d/2} \cdot \exp(-\|u\|^2/2\sigma^2)$ denotes the density of $\mathcal{N}(0, \sigma^2 I)$. Hence,

$$\partial_\theta \phi_\sigma(y - a\mathbf{x}_\theta) = \frac{a}{\sigma^2}\langle \mathbf{h}, y - a\mathbf{x}_\theta\rangle \phi_\sigma(y - a\mathbf{x}_\theta), \tag{210}$$

$$\implies \partial_\theta p_{\theta,\epsilon}(y) = \frac{a}{\sigma^2}\mathbb{E}\left[\langle \mathbf{h}, y - a\mathbf{x}_\theta\rangle\, \phi_\sigma(y - a\mathbf{x}_\theta)\right] \tag{211}$$

$$\implies \partial_\theta \log p_{\theta,\epsilon}(y) = \frac{\partial_\theta p_{\theta,\epsilon}(y)}{p_{\theta,\epsilon}(y)} = \frac{\frac{a}{\sigma^2}\mathbb{E}\left[\langle \mathbf{h}, y - a\mathbf{x}_\theta\rangle\, \phi_\sigma(y - a\mathbf{x}_\theta)\right]}{\mathbb{E}[\phi_\sigma(y - a\mathbf{x}_\theta)]} = \frac{a}{\sigma} f_\theta(y). \tag{212}$$

Hence,

$$\partial_\theta s_{p_\theta}(y) = \partial_\theta \nabla_y \log p_{\theta,\epsilon}(y) \tag{213}$$

$$= \nabla_y \partial_\theta \log p_{\theta,\epsilon}(y) \tag{214}$$

$$= \frac{a}{\sigma}\nabla f_\theta(y). \tag{215}$$

Plugging this in the formula for $\frac{d}{d\theta}g(\mathbf{x}_\theta, p_\theta)$ we have

$$\frac{d}{d\theta}g(\mathbf{x}_\theta, p_\theta) = \frac{a}{\sigma}\mathbb{E}_{\mathbf{z}}[\nabla f_\theta(a\mathbf{x}_\theta + \sigma\mathbf{z})] + J_{p_\theta}(\mathbf{x}_\theta)\mathbf{h} \tag{216}$$

$$= \frac{a}{\sigma^2}\mathbb{E}_{\mathbf{z}}[\mathbf{z} f_\theta(a\mathbf{x}_\theta + \sigma\mathbf{z})] + J_{p_\theta}(\mathbf{x}_\theta)\mathbf{h}, \tag{217}$$

where we used the Gaussian integration by parts (Lemma C.2) in the final step. □

Lemma G.6. *Let $r, s \geq 2$ and $u < \infty$ satisfy $1/r = 1/s + 1/u$. Let $\mathbf{x}, \widetilde{\mathbf{x}}$ be two random vectors with laws $p, \widetilde{p}$ respectively, and suppose $\mathbf{x} - \widetilde{\mathbf{x}} \in L^u$. Then, we have*

$$\|g(\mathbf{x}, p) - g(\widetilde{\mathbf{x}}, \widetilde{p})\|_{L^r} \leq \frac{a}{\sigma^2}(\sqrt{ds} + \sqrt{r})\|\mathbf{x} - \widetilde{\mathbf{x}}\|_{L^u}. \tag{218}$$

Proof. Using the same definitions from Lemma G.5, we have

$$g(\widetilde{\mathbf{x}}, \widetilde{p}) - g(\mathbf{x}, p) = \int_0^1 \frac{d}{d\theta} g(\mathbf{x}_\theta, p_\theta) d\theta \tag{219}$$

$$= \int_0^1 \left(J_{p_\theta}(\mathbf{x}_\theta)\mathbf{h} + \frac{a}{\sigma^2}\mathbb{E}_{\mathbf{z}}[\mathbf{z} f_\theta(a\mathbf{x}_\theta + \sigma\mathbf{z})]\right) d\theta, \tag{220}$$

and hence using Minkowski's inequality we have

$$\|g(\widetilde{\mathbf{x}}, \widetilde{p}) - g(\mathbf{x}, p)\|_{L^r} \leq \int_0^1 \|J_{p_\theta}(\mathbf{x}_\theta)\mathbf{h}\|_{L^r}\, d\theta + \frac{a}{\sigma^2}\int_0^1 \|\mathbb{E}_{\mathbf{z}}[\mathbf{z} f_\theta(a\mathbf{x}_\theta + \sigma\mathbf{z})]\|_{L^r}\, d\theta. \tag{221}$$

Now, using Hölder's inequality and Lemma G.4 we have

$$\|J_{p_\theta}(\mathbf{x}_\theta)\mathbf{h}\|_{L^r} \leq \left\| \|J_{p_\theta}(\mathbf{x}_\theta)\|_{\mathrm{op}} \cdot \|\mathbf{h}\| \right\|_{L^r} \tag{222}$$

$$\leq \left\| \|J_{p_\theta}(\mathbf{x}_\theta)\|_{\mathrm{op}} \right\|_{L^s} \cdot \|\mathbf{h}\|_{L^u} \tag{223}$$

$$\leq \frac{a}{\sigma^2}\sqrt{ds} \cdot \|\mathbf{h}\|_{L^u} \tag{224}$$

Next, using Lemma G.3 we have

$$\|\mathbb{E}_{\mathbf{z}}[\mathbf{z} f_\theta(a\mathbf{x}_\theta + \sigma\mathbf{z})]\|_{L^r} \leq c_{r'} \cdot \mathbb{E}[|f_\theta(\mathbf{y}_\theta)|^r]^{1/r}, \qquad r' = \frac{r}{r-1} \tag{225}$$

and noting that $f_\theta(y) = \mathbb{E}[\langle \mathbf{h}, \mathbf{z}\rangle | \mathbf{y}_\theta = y]$, using conditional Jensen's inequality and the tower property we have

$$\mathbb{E}[|f_\theta(\mathbf{y}_\theta)|^r]^{1/r} \le \mathbb{E}[|\langle \mathbf{h}, \mathbf{z}\rangle|^r]^{1/r} \tag{226}$$

$$= \mathbb{E}[\mathbb{E}[|\langle \mathbf{h}, \mathbf{z}\rangle|^r | \mathbf{h}]]^{1/r} \tag{227}$$

$$= \mathbb{E}[\|\mathbf{h}\|^r \cdot c_r^r]^{1/r} \tag{228}$$

$$= c_r \cdot \|\mathbf{h}\|_{L^r} \le \sqrt{r} \cdot \|\mathbf{h}\|_{L^r}, \tag{229}$$

where we used that $\mathbf{z}$ is independent of $\mathbf{h}$, and that conditioned on $\mathbf{h}$, $\langle \mathbf{h}, \mathbf{z}\rangle$ is distributed as $\mathcal{N}(0, \|\mathbf{h}\|^2)$. Since $u \ge r$, we have $\|\mathbf{h}\|_{L^r} \le \|\mathbf{h}\|_{L^u}$, and hence

$$\|g(\widetilde{\mathbf{x}}, \widetilde{p}) - g(\mathbf{x}, p)\|_{L^r} \le \frac{a}{\sigma^2}(\sqrt{ds} + \sqrt{r})\|\mathbf{h}\|_{L^u} \tag{230}$$

$$= \frac{a}{\sigma^2}(\sqrt{ds} + \sqrt{r})\|\mathbf{x} - \widetilde{\mathbf{x}}\|_{L^u}, \tag{231}$$

as desired. □

Lemma G.7. *Let $\mathbf{x}_t^n$ denote the semi-implicit Picard iterates defined in equation 160, and suppose that the initial condition $\mathbf{x}_0 \sim p_0$ has finite second moment (i.e., $\mathbb{E}[\|\mathbf{x}_0\|^2] < \infty$). Then, there exists a continuous trajectory $\mathbf{x}_t$ such that for any $T > 0$,*

$$\mathbb{E}[\sup_{0 \le t \le T} \|\mathbf{x}_t^n - \mathbf{x}_t\|^2]^{1/2} \le \frac{\sigma}{a}\left(\frac{1}{2}\right)^{n-1} \text{ for all } n \ge 4eC^2T^2 \tag{232}$$

where $C := \frac{2a^2}{\sigma^2}\sqrt{2de}$, and consequently,

$$\lim_{n\to\infty} \mathbb{E}[\sup_{0 \le t \le T} \|\mathbf{x}_t^n - \mathbf{x}_t\|^2]^{1/2} = 0. \tag{233}$$

Proof. We first show by induction that $\mathbf{x}_t^n \in L^2(\Omega; C([0,T]; \mathbb{R}^d))$ for all n. The initial trajectory is $\mathbf{x}_t^0 = \mathbf{x}_0$ and since $\mathbb{E}[\|\mathbf{x}_0\|^2] < \infty$, we have $\mathbf{x}_t^0 \in L^2(\Omega; C([0,T]; \mathbb{R}^d))$. Now suppose that $\mathbf{x}_t^n \in L^2(\Omega; C([0,T]; \mathbb{R}^d))$. Then, from equation 160 we have for all $t \in [0,T]$:

$$\left\|\mathbf{x}_t^{n+1}\right\| \le e^{-a^2 t}\|\mathbf{x}_0\| + a\int_0^t e^{a^2(s-t)}\|g(\mathbf{x}_s^n, p_s^n)\|\,ds \tag{234}$$

$$\le \|\mathbf{x}_0\| + a\int_0^T \|g(\mathbf{x}_s^n, p_s^n)\|\,ds \tag{235}$$

$$\implies \sup_{t\le T}\left\|\mathbf{x}_t^{n+1}\right\| \le \|\mathbf{x}_0\| + a\int_0^T \|g(\mathbf{x}_s^n, p_s^n)\|\,ds \tag{236}$$

and hence applying Minkowski's inequality for L^2 norm we obtain

$$\mathbb{E}\left[\sup_{t\le T}\left\|\mathbf{x}_t^{n+1}\right\|^2\right]^{1/2} \le \mathbb{E}[\|\mathbf{x}_0\|^2]^{1/2} + a\int_0^T \mathbb{E}[\|g(\mathbf{x}_s^n, p_s^n)\|^2]^{1/2} ds \tag{237}$$

$$\le \mathbb{E}[\|\mathbf{x}_0\|^2]^{1/2} + \frac{am_{d,2}}{\sigma}T < \infty, \tag{238}$$

where in the last step we used Lemma G.2. Hence, $\mathbf{x}_t^{n+1} \in L^2(\Omega; C([0,T]; \mathbb{R}^d))$, and by induction we have $\mathbf{x}_t^n \in L^2(\Omega; C([0,T]; \mathbb{R}^d))$ for all n.

Next, we show that $\sup_{t' \le t}\left\|\mathbf{x}_{t'}^{n+1} - \mathbf{x}_{t'}^n\right\| \in L^r$ for all $n \ge 1$ and for any $r \ge 1$. We have

$$\mathbf{x}_t^{n+1} - \mathbf{x}_t^n = -a\int_0^t e^{a^2(s-t)}(g(\mathbf{x}_s^n, p_s^n) - g(\mathbf{x}_s^{n-1}, p_s^{n-1}))ds \tag{239}$$

$$\implies \left\|\mathbf{x}_t^{n+1} - \mathbf{x}_t^n\right\| \le a\int_0^t \left\|g(\mathbf{x}_s^n, p_s^n) - g(\mathbf{x}_s^{n-1}, p_s^{n-1})\right\| ds \tag{240}$$

where we used $e^{a^2(s-t)} \leq 1$ inside the integral. In particular, we note that the right hand side is an upper bound for any $\left\|\mathbf{x}_{t'}^{n+1} - \mathbf{x}_{t'}^{n}\right\|$ where $t' \leq t$. Thus, we have

$$\sup_{0\leq t'\leq t} \left\|\mathbf{x}_{t'}^{n+1} - \mathbf{x}_{t'}^{n}\right\| \leq a \int_0^t \left\|g(\mathbf{x}_s^n, p_s^n) - g(\mathbf{x}_s^{n-1}, p_s^{n-1})\right\| ds, \tag{241}$$

and applying Minkowski's inequality for the L^r norm gives

$$\begin{aligned}
\mathbb{E}\left[\sup_{0\leq t'\leq t} \left\|\mathbf{x}_{t'}^{n+1} - \mathbf{x}_{t'}^{n}\right\|^r\right]^{1/r} &\leq a \int_0^t \mathbb{E}\left[\left\|g(\mathbf{x}_s^n, p_s^n) - g(\mathbf{x}_s^{n-1}, p_s^{n-1})\right\|^r\right]^{1/r} ds && (242)\\
&= a \int_0^t \left\|g(\mathbf{x}_s^n, p_s^n) - g(\mathbf{x}_s^{n-1}, p_s^{n-1})\right\|_{L^r} ds && (243)\\
&\leq a \int_0^t \left(\|g(\mathbf{x}_s^n, p_s^n)\|_{L^r} + \left\|g(\mathbf{x}_s^{n-1}, p_s^{n-1})\right\|_{L^r} \right) ds && (244)\\
&\leq \frac{2am_{d,r}}{\sigma} t < \infty, && (245)
\end{aligned}$$

where in the last line we used Lemma G.2. Hence, $\sup_{t'\leq t} \left\|\mathbf{x}_{t'}^{n+1} - \mathbf{x}_{t'}^{n}\right\| \in L^r$.

Denoting

$$\mathcal{D}_{n,r}(t) := \mathbb{E}\left[\sup_{0\leq t'\leq t} \left\|\mathbf{x}_{t'}^{n+1} - \mathbf{x}_{t'}^{n}\right\|^r\right]^{1/r}, \tag{246}$$

then we have for any $n \geq 1$,

$$\mathcal{D}_{n,r}(t) \leq \frac{2am_{d,r}}{\sigma} t. \tag{247}$$

Now, we define a sequence $r_j, j = 1, \dots n$ by the recurrence

$$\begin{aligned}
r_1 &= 2n, && (248)\\
\frac{1}{r_j} &= \frac{1}{2n} + \frac{1}{r_{j-1}}, \qquad j \geq 2 && (249)
\end{aligned}$$

or equivalently $r_j = \frac{2n}{j}$. In particular, we have $r_n = 2$. Now, for any $2 \leq j \leq n$ we have

$$\begin{aligned}
\mathcal{D}_{j,r_j}(t) &\leq a \int_0^t \left\|g(\mathbf{x}_s^j, p_s^j) - g(\mathbf{x}_s^{j-1}, p_s^{j-1})\right\|_{L^{r_j}} ds && (250)\\
&\leq \frac{a^2}{\sigma^2}(\sqrt{2nd} + \sqrt{r_j}) \int_0^t \left\|\mathbf{x}_s^j - \mathbf{x}_s^{j-1}\right\|_{L^{r_{j-1}}} ds && (251)\\
&\leq \frac{2a^2}{\sigma^2}\sqrt{2nd} \cdot \int_0^t \mathcal{D}_{j-1,r_{j-1}}(s) ds, && (252)
\end{aligned}$$

where in the second line we used Lemma G.6, and in the last line we used $r_j \leq 2n \leq 2nd$, and $\left\|\mathbf{x}_s^j - \mathbf{x}_s^{j-1}\right\| \leq \sup_{s'\leq s} \left\|\mathbf{x}_{s'}^j - \mathbf{x}_{s'}^{j-1}\right\| \implies \left\|\mathbf{x}_s^j - \mathbf{x}_s^{j-1}\right\|_{L^{r_{j-1}}} \leq \mathcal{D}_{j-1,r_{j-1}}(s)$.

Denoting $C_1 := \frac{2a^2}{\sigma^2}\sqrt{d}$ and $\delta_j(t) := \mathcal{D}_{j,r_j}(t)$ we have for all $2 \leq j \leq n$,

$$\delta_j(t) \leq C_1\sqrt{2n} \cdot \int_0^t \delta_{j-1}(s) ds, \tag{253}$$

and from equation 247 we have $\delta_1(t) \leq \frac{2am_{d,2n}}{\sigma} t \leq \frac{2a\sqrt{2nd}}{\sigma} t := C_2 t$. Thus, we can show by induction that

$$\delta_j(t) \leq \frac{C_2}{C_1\sqrt{2n}} \cdot \frac{(C_1\sqrt{2n} \cdot t)^j}{j!} = \frac{\sigma}{a} \frac{(C_1\sqrt{2n} \cdot t)^j}{j!}, \qquad j = 1, \dots n \tag{254}$$

and hence

$$\mathbb{E}\left[\sup_{0\le t'\le t}\left\|\mathbf{x}_{t'}^{n+1}-\mathbf{x}_{t'}^{n}\right\|^2\right]^{1/2}=\mathcal{D}_{n,2}(t)=\delta_n(t)\le\frac{\sigma}{a}\frac{(C_1\sqrt{2n}\cdot t)^n}{n!}. \tag{255}$$

We use the fact that $n!\ge(n/e)^n$ to write

$$\frac{(C_1\sqrt{2n}\cdot t)^n}{n!}=\frac{(C_1\sqrt{2n}\cdot t)^n}{\sqrt{n!}}\cdot\frac{1}{\sqrt{n!}} \tag{256}$$

$$\le\left(\frac{C_1\sqrt{e}\cdot\sqrt{2n}\cdot t}{\sqrt{n}}\right)^n\cdot\frac{1}{\sqrt{n!}} \tag{257}$$

$$=\frac{(C_3t)^n}{\sqrt{n!}},\qquad C_3:=C_1\sqrt{2e}, \tag{258}$$

and hence

$$\mathbb{E}\left[\sup_{0\le t'\le t}\left\|\mathbf{x}_{t'}^{n+1}-\mathbf{x}_{t'}^{n}\right\|^2\right]^{1/2}\le\frac{\sigma}{a}\cdot\frac{(C_3t)^n}{\sqrt{n!}} \tag{259}$$

Now, for any $m>n$, we have

$$\sup_{0\le t'\le T}\|\mathbf{x}_{t'}^{m}-\mathbf{x}_{t'}^{n}\|\le\sum_{k=n}^{m-1}\sup_{0\le t'\le T}\left\|\mathbf{x}_{t'}^{k+1}-\mathbf{x}_{t'}^{k}\right\| \tag{260}$$

$$\implies\mathbb{E}[\sup_{0\le t'\le T}\|\mathbf{x}_{t'}^{m}-\mathbf{x}_{t'}^{n}\|^2]^{1/2}\le\sum_{k=n}^{m-1}\mathbb{E}[\sup_{0\le t'\le T}\left\|\mathbf{x}_{t'}^{k+1}-\mathbf{x}_{t'}^{k}\right\|^2]^{1/2} \tag{261}$$

$$\le\sum_{k=n}^{\infty}\mathbb{E}[\sup_{0\le t'\le T}\left\|\mathbf{x}_{t'}^{k+1}-\mathbf{x}_{t'}^{k}\right\|^2]^{1/2} \tag{262}$$

$$\le\sum_{k=n}^{\infty}\frac{\sigma}{a}\frac{(C_3T)^k}{\sqrt{k!}}, \tag{263}$$

and we note that the tail sum of the above series converges to 0 (because by the ratio test, the ratio between the kth and $(k+1)$th terms is $\frac{C_3T}{\sqrt{k+1}}\to 0$ as $k\to\infty$).

Thus, the iterates $\mathbf{x}_t^n$ form a Cauchy sequence in $L^2(\Omega;C([0,T];\mathbb{R}^d))$. Since $L^2(\Omega;C([0,T];\mathbb{R}^d))$ is complete, there exists a trajectory $\mathbf{x}_t\in L^2(\Omega;C([0,T];\mathbb{R}^d))$ such that

$$\|\mathbf{x}^n-\mathbf{x}\|_{L^2}:=\mathbb{E}[\sup_{0\le t'\le T}\|\mathbf{x}_{t'}^{n}-\mathbf{x}_{t'}\|^2]^{1/2}\to 0,\qquad\text{as } n\to\infty. \tag{264}$$

Furthermore, for a given n we have

$$\|\mathbf{x}^n-\mathbf{x}\|_{L^2}\le\|\mathbf{x}^n-\mathbf{x}^m\|_{L^2}+\|\mathbf{x}^m-\mathbf{x}\|_{L^2} \tag{265}$$

$$\le\left(\sum_{k=n}^{\infty}\frac{\sigma}{a}\frac{(C_3T)^k}{\sqrt{k!}}\right)+\|\mathbf{x}^m-\mathbf{x}\|_{L^2}, \tag{266}$$

and taking $m\to\infty$ gives

$$\|\mathbf{x}^n-\mathbf{x}\|_{L^2}\le\sum_{k=n}^{\infty}\frac{\sigma}{a}\frac{(C_3T)^k}{\sqrt{k!}}, \tag{267}$$

where

$$C_3=C_1\sqrt{2e}=\frac{2a^2}{\sigma^2}\sqrt{2de}. \tag{268}$$

We now use the fact that $k!\ge(k/e)^k$ for all $k\ge 1$ to get

$$\|\mathbf{x}^n-\mathbf{x}\|_{L^2}\le\sum_{k=n}^{\infty}\frac{\sigma}{a}\left(\frac{\sqrt{e}C_3T}{\sqrt{k}}\right)^k, \tag{269}$$

and hence, if $\sqrt{n} \geq 2\sqrt{e}C_3T$ (equivalently $n \geq 4eC_3^2T^2$), we have

$$\|\mathbf{x}^n - \mathbf{x}\|_{L^2} \leq \sum_{k=n}^{\infty} \frac{\sigma}{a} \left(\frac{1}{2}\right)^k = \frac{\sigma}{a} \cdot \left(\frac{1}{2}\right)^{n-1}. \tag{270}$$

□

Lemma G.8. *The limiting trajectory $\mathbf{x}_t$ satisfies the integral equation*

$$\mathbf{x}_t = e^{-a^2t}\mathbf{x}_0 - a\int_0^t e^{a^2(s-t)} g(\mathbf{x}_s, p_s) ds \tag{271}$$

almost surely for all $t \in [0, T]$. Consequently, by multiplying with e^{a^2t} and differentiating with respect to t, we have that $\mathbf{x}_t$ satisfies the WGF equation 5.

Proof. Let

$$\mathbf{r}_t := \mathbf{x}_t - e^{-a^2t}\mathbf{x}_0 + a\int_0^t e^{a^2(s-t)} g(\mathbf{x}_s, p_s) ds. \tag{272}$$

Then, we have

$$\mathbf{r}_t = \mathbf{x}_t - \mathbf{x}_t^{n+1} + a\int_0^t e^{a^2(s-t)} \big(g(\mathbf{x}_s, p_s) - g(\mathbf{x}_s^n, p_s^n)\big) ds \tag{273}$$

$$\implies \|\mathbf{r}_t\| \leq \left\|\mathbf{x}_t - \mathbf{x}_t^{n+1}\right\| + a\int_0^t e^{a^2(s-t)} \left\|\big(g(\mathbf{x}_s, p_s) - g(\mathbf{x}_s^n, p_s^n)\big)\right\| ds \tag{274}$$

$$\leq \left\|\mathbf{x}_t - \mathbf{x}_t^{n+1}\right\| + a\int_0^T \left\|\big(g(\mathbf{x}_s, p_s) - g(\mathbf{x}_s^n, p_s^n)\big)\right\| ds \tag{275}$$

$$\implies \sup_{0\leq t\leq T} \|\mathbf{r}_t\| \leq \sup_{0\leq t\leq T} \left\|\mathbf{x}_t - \mathbf{x}_t^{n+1}\right\| + a\int_0^T \left\|\big(g(\mathbf{x}_s, p_s) - g(\mathbf{x}_s^n, p_s^n)\big)\right\| ds \tag{276}$$

$$\implies \mathbb{E}[\sup_{0\leq t\leq T} \|\mathbf{r}_t\|^2]^{1/2} \leq \mathbb{E}[\sup_{0\leq t\leq T} \left\|\mathbf{x}_t - \mathbf{x}_t^{n+1}\right\|^2]^{1/2} + a\int_0^T \|g(\mathbf{x}_s, p_s) - g(\mathbf{x}_s^n, p_s^n)\|_{L^2} ds. \tag{277}$$

Now, the first term $\mathbb{E}[\sup_{0\leq t\leq T} \left\|\mathbf{x}_t - \mathbf{x}_t^{n+1}\right\|^2]^{1/2}$ in the RHS converges to 0 as $n \to \infty$. We need to show the same for the second term.

Since $\mathbf{x}^n \to \mathbf{x}$ in L^2 (where we denote $\mathbf{x}^n := \mathbf{x}_t^n$ and $\mathbf{x} := \mathbf{x}_t$), there exists a subsequence n_j such that $\mathbf{x}^{n_j} \to \mathbf{x}$ almost surely, uniformly in $t \in [0, T]$. Hence, for any $n \geq 1$,

$$\sup_{t\leq T} \left\|\mathbf{x}_t^n - \mathbf{x}_t^{n_j}\right\|^4 \to \sup_{t\leq T} \|\mathbf{x}_t^n - \mathbf{x}_t\|^4, \quad \text{as } j \to \infty, \tag{278}$$

and hence by Fatou's lemma,

$$\mathbb{E}\left[\sup_{t\leq T} \|\mathbf{x}_t^n - \mathbf{x}_t\|^4\right] \leq \liminf_j \mathbb{E}\left[\sup_{t\leq T} \left\|\mathbf{x}_t^n - \mathbf{x}_t^{n_j}\right\|^4\right] \tag{279}$$

$$\implies \mathbb{E}\left[\sup_{t\leq T} \|\mathbf{x}_t^n - \mathbf{x}_t\|^4\right]^{1/4} \leq \liminf_j \mathbb{E}\left[\sup_{t\leq T} \left\|\mathbf{x}_t^n - \mathbf{x}_t^{n_j}\right\|^4\right]^{1/4} \tag{280}$$

$$\leq \liminf_j \int_0^T a \left\|g(\mathbf{x}_s^{n-1}, p_s^{n-1}) - g(\mathbf{x}_s^{n_j-1}, p_s^{n_j-1})\right\|_{L^4} ds \tag{281}$$

$$\leq \frac{2am_{d,4}}{\sigma} T, \tag{282}$$

where we used Lemma G.2.

Denoting $\mathbf{u}_n = \sup_{t \le T} \|\mathbf{x}_t^n - \mathbf{x}_t\|$, we thus have $\|\mathbf{u}_n\|_{L^4} = \mathbb{E}[\mathbf{u}_n^4]^{1/4} \le \frac{2am_{d,4}T}{\sigma}$ for all $n \ge 1$. We also have $\|\mathbf{u}_n\|_{L^2} \to 0$ as $n \to \infty$. Thus,

$$\|\mathbf{u}_n\|_{L^3} = \mathbb{E}[\mathbf{u}_n^3]^{1/3} \tag{283}$$

$$= \mathbb{E}[\mathbf{u}_n \cdot \mathbf{u}_n^2]^{1/3} \tag{284}$$

$$\le \mathbb{E}[\mathbf{u}_n^2]^{1/6} \cdot \mathbb{E}[\mathbf{u}_n^4]^{1/6} \to 0 \quad \text{as } n \to \infty. \tag{285}$$

Now, applying Lemma G.6 with $r = 2, s = 6$ and $u = 3$ gives

$$\|g(\mathbf{x}_s, p_s) - g(\mathbf{x}_s^n, p_s^n)\|_{L^2} \le L \cdot \|\mathbf{x}_s^n - \mathbf{x}_s\|_{L^3} \le L \cdot \|\mathbf{u}_n\|_{L^3}, \tag{286}$$

where $L = \frac{a}{\sigma^2}(\sqrt{6d} + \sqrt{2})$. Thus,

$$a \int_0^T \|g(\mathbf{x}_s, p_s) - g(\mathbf{x}_s^n, p_s^n)\|_{L^2} \, ds \le aTL \|\mathbf{u}_n\|_{L^3} \to 0 \quad \text{as } n \to \infty, \tag{287}$$

and hence both terms on the RHS of equation 277 converge to 0 as $n \to \infty$. Since the LHS does not depend on n, we have $\mathbb{E}[\sup_{0 \le t \le T} \|\mathbf{r}_t\|^2] = 0$, and hence $\sup_{t \le T} \|\mathbf{r}_t\| = 0$ almost surely.

Therefore, $\mathbf{r}_t = \mathbf{x}_t - e^{-a^2 t}\mathbf{x}_0 + a\int_0^t e^{a^2(s-t)} g(\mathbf{x}_s, p_s) ds = 0$ simultaneously for all $t \in [0, T]$, almost surely. □

H Additional Experimental Details

H.1 Implementation Details

We make the following minor modifications to Algorithm 1 during actual implementation.

- We introduce a warmup phase (phase 0) for the score network s_ϕ, where s_ϕ is trained for N_{warm} steps for sufficiently large N_{warm} on the initial trajectory, in order to learn its smoothed scores more accurately.
- We swap Phase 1 and Phase 2 in Algorithm 1, i.e. phase 1 in our implementation is the trajectory map update,phase 2 is the score update, and phase 3 is the decoder update.

Phases 1,2 and 3 alternate, while phase 0 is run only once (for the initial trajectory). Algorithm 2 contains the pseudo-code corresponding to our implementation.

Algorithm 2 Implemented PC-VAE Training Procedure

Require: Pre-trained VAE with encoder E (frozen) and decoder D with trainable parameters ω; $T > 0$; $\epsilon \in (0, 1)$; iteration counts $N_{\text{warm}}, N_{\text{picard}}, N_1, N_2, N_3$.

Initialize $\Psi_\theta(x, t) = x + tF_\theta(x, t)$ with $F_\theta \equiv 0$, and score network $s_\phi(x, t)$.

Phase 0: Score network warmup (freeze θ; unfreeze ϕ)
for $k = 1, \ldots, N_{\text{warm}}$ **do**
    Sample a minibatch of images $\mathbf{u}_0$, times $t \in [0, T]$, and $\mathbf{z} \sim \mathcal{N}(0, I)$; set $\mathbf{x}_0 = E(\mathbf{u}_0)$.
    Update ϕ using $\mathcal{L}_{\text{score}}(\phi)$ in equation 7.
end for

for $n = 0, \ldots, N_{\text{Picard}} - 1$ **do**
    Create a frozen copy $\widehat{\theta} \leftarrow \theta$

    Phase 1: Transport map training (freeze ϕ; unfreeze θ)
    for $k = 1, \ldots, N_1$ **do**
        Sample a minibatch of images $\mathbf{u}_0$, times $t \in [0, T]$, and $\mathbf{z} \sim \mathcal{N}(0, I)$; set $\mathbf{x}_0 = E(\mathbf{u}_0)$.
        Update θ using $\mathcal{L}_{\text{flow}}(\theta)$ in equation 8.
    end for

    Phase 2: Score network training (freeze θ; unfreeze ϕ)
    for $k = 1, \ldots, N_2$ **do**
        Sample a minibatch of images $\mathbf{u}_0$, times $t \in [0, T]$, and $\mathbf{z} \sim \mathcal{N}(0, I)$; set $\mathbf{x}_0 = E(\mathbf{u}_0)$.
        Update ϕ using $\mathcal{L}_{\text{score}}(\phi)$ in equation 7.
    end for

    Phase 3: Adapting the decoder (freeze θ, ϕ; unfreeze ω)
    for $k = 1, \ldots, N_3$ **do**
        Sample a minibatch of images $\mathbf{u}_0$; set $\mathbf{x}_0 = E(\mathbf{u}_0)$.
        Update ω by minimizing reconstruction losses on the terminal latents $\Psi_\theta(\mathbf{x}_0, T)$.
    end for
end for

return (Ψ_θ, D_ω).

H.2 Network Parameters

In this subsection we outline the configurations of all the PC-VAE experiments reported in Section 6.

Table 4 defines the exact layer dimensions, attention resolutions and total parameter counts for both the VAE components and the bottleneck Diffusion Transformer.

Table 4: Encoder and decoder configurations across datasets

Parameter	**CelebA-HQ & LSUN Church**		**LSUN Bedroom**	
	Encoder	Decoder	Encoder	Decoder
in_channels	3	–	3	–
out_channels	–	3	–	3
resolution	256	256	256	256
ch	128	128	128	128
ch_mult	(1, 2, 2, 4, 4, 4, 4)	(1, 2, 2, 4, 4, 4, 4)	(1, 2, 2, 4, 4, 4)	(1, 2, 2, 4, 4, 4)
num_res_blocks	3	3	3	3
attn_resolutions	(8, 4)	(4,)	(16, 8, 4)	(8,)
dropout	0.0	0.0	0.0	0.0
z_channels	8	8	4	4
double_z	True	False	True	False
attn_type	vanilla	vanilla	vanilla	vanilla
Bottleneck DiT Configuration				
input_size	4	4	8	8
patch_size	1	1	1	1
num_channels	16	8	8	4
hidden_size	256	256	256	256
depth	4	4	4	4
num_heads	4	4	4	4
mlp_ratio	4.0	4.0	4.0	4.0
Total Parameters	**94.94M**	**118.77M**	**78.41M**	**97.51M**

Table 5 lists the structural hyperparameters for the score and transport networks (which are both Diffusion Transformers).

Table 5: Score and trajectory network configurations across datasets

Parameter	**CelebA-HQ & LSUN Church**		**LSUN Bedroom**	
	Score Net	Ψ Net	Score Net	Ψ Net
input_size	4	4	8	8
patch_size	1	1	1	1
num_channels	8	8	4	4
hidden_size	512	512	512	512
depth	12	8	12	12
num_heads	8	8	8	8
mlp_ratio	4.0	4.0	4.0	4.0
dropout	0.1	0.1	0.1	0.1
conditioning_mode	timestep	timestep	timestep	timestep
Total Parameters	**57.64M**	**38.74M**	**57.64M**	**57.64M**

Table 6 lists the number of training steps for each phase and the number of Picard iterations.

Table 6: Picard experiment configurations. Phase numbers follow Algorithm 2; num_periods denotes number of Picard steps.

Parameter	**CelebA-HQ**	**LSUN Church**	**LSUN Bedroom**
Latent shape	$4 \times 4 \times 8$	$4 \times 4 \times 8$	$8 \times 8 \times 4$
phase0_steps	50000	50000	40000
phase1_steps	10000	10000	10000
phase2_steps	10000	10000	10000
phase3_steps	20000	20000	20000
num_periods	50	50	50
ϵ	0.05	0.1	0.1
T	5	5	5

To ensure proper rigorous comparison, we reproduced the baseline results for Shortcut Models using the official codebase on all datasets. Table 7 provides the set of hyperparameters used to train and evaluate the Shortcut models.

Table 7: Hyperparameters used to evaluate Shortcut Models (DiT-B/2) on LSUN-Church, LSUN-Bedroom, and CelebA-HQ (256×256).

Category	**Parameter**	**LSUN-Church**	**LSUN-Bedroom**	**CelebA-HQ**
Data	Resolution	256×256	256×256	256×256
	Preprocessing	Center crop	Center crop	Random h-flip
	Conditioning / CFG scale	Uncond. (0.0)	Uncond. (0.0)	Uncond. (0.0)
Latent VAE	Autoencoder	Frozen SD-VAE (`sd-vae-ft-mse`)		
	Latent shape ($H \times W \times C$)	$32 \times 32 \times 4$	$32 \times 32 \times 4$	$32 \times 32 \times 4$
DiT-B/2	Patch size p	2	2	2
	Hidden dimension d	768	768	768
	Transformer blocks / Heads	12 / 12	12 / 12	12 / 12
	MLP ratio	$4\times$	$4\times$	$4\times$
	Dropout / Class dropout	0.0 / 1.0	0.0 / 1.0	0.0 / 1.0
Shortcut	Max denoising steps M	128	128	128
	Time / Step-size sampling	Discrete-dt / Unif.	Discrete-dt / Unif.	Discrete-dt / Unif.
	Self-consistency ratio (k)	$1/8$	$1/8$	$1/8$
	Bootstrap target network	EMA	EMA	EMA
Training	Optimizer	AdamW	AdamW	AdamW
	Learning rate / Schedule	10^{-4} (Const.)	10^{-4} (Const.)	10^{-4} (Const.)
	AdamW (β_1, β_2) / Weight decay	$(0.9, 0.999)$ / 0.1	$(0.9, 0.999)$ / 0.1	$(0.9, 0.999)$ / 0.1
	EMA decay rate	0.999	0.999	0.999
	Batch size / Training steps	64 / 410k	64 / 410k	64 / 410k
	Hardware ($8\times$ A100-40GB)	$\sim$ 12.2 h	$\sim$ 11.5 h	$\sim$ 12.0 h
Evaluation	Sampling step budgets (NFE)	$\{1, 4, 32\}$	$\{1, 4, 32\}$	$\{1, 4, 32\}$
	Final FID sample count	50,000	50,000	50,000

H.3 Ablations

We perform ablations for PC-VAE, for three parameters: ϵ (smoothing parameter), T (terminal time), and latent shape $H \times W \times C$. Table 8 contains the ablation baseline configurations for each dataset.

Table 8: Ablation baseline configurations. Phase numbers follow Algorithm 2; num_periods denotes number of Picard steps.

Parameter	CelebA-HQ	LSUN Church	LSUN Bedroom
Latent shape	$4 \times 4 \times 8$	$4 \times 4 \times 8$	$8 \times 8 \times 4$
phase0_steps	50000	50000	50000
phase1_steps	10000	10000	10000
phase2_steps	10000	10000	10000
phase3_steps	20000	20000	20000
num_periods	50	50	50
ϵ	0.1	0.1	0.1
T	5	5	5

The network architectures are the same as in H.2, with the only difference being that for LSUN Bedroom the depth of Ψ_θ is 8 (instead of 12 earlier).

Table 9 contains the ablations across all 3 datasets for each of the chosen parameters. We did not include latent shape $4 \times 4 \times 8$ for LSUN Bedroom as the pretrained VAE had poor reconstruction.

Table 9: Ablations evaluated across CelebA-HQ, LSUN Church and Bedroom.

Parameter	Value	FID (↓)		
		CelebA-HQ	LSUN Church	LSUN Bedroom
ϵ	$\epsilon = 0.05$	9.95	6.53	8.83
	$\epsilon = 0.1$	10.68	6.49	8.03
	$\epsilon = 0.2$	16.77	8.99	10.03
Latent shape	$4 \times 4 \times 8$	10.68	6.49	–
	$8 \times 8 \times 4$	18.56	8.68	8.03
	$16 \times 16 \times 4$	24.87	12.90	11.04
T	$T = 5$	10.68	6.49	8.03
	$T = 7.5$	12.37	6.70	8.77
	$T = 10$	14.25	7.00	10.20

H.4 Additional Latent Distribution Diagnostics

The following diagnostics are computed from the LSUN Bedroom PC-VAE run reported in Section 6.

Estimated Fisher Information. At the end of each Picard step n, we compute an estimate of the relative Fisher information between $p_{\mathbf{x}_T^n,\epsilon}$ and $q = \mathcal{N}(0, I)$, where $\mathbf{x}_T^n = \Psi_\theta(\mathbf{x}_0, T)$. The exact Fisher information is given by

$$I(p_{\mathbf{x}_T^n,\epsilon} \| q) = \mathbb{E}\left[\left\| \nabla \log \frac{p_{\mathbf{x}_T^n,\epsilon}(a\mathbf{x}_T^n + \sigma\mathbf{z})}{q(a\mathbf{x}_T^n + \sigma\mathbf{z})} \right\|^2\right] \tag{288}$$

$$= \mathbb{E}\left[\left\| \nabla \log p_{\mathbf{x}_T^n,\epsilon}(a\mathbf{x}_T^n + \sigma\mathbf{z}) + a\mathbf{x}_T^n + \sigma\mathbf{z} \right\|^2\right]. \tag{289}$$

For our estimate of the Fisher information, the smoothed score $\nabla \log p_{\mathbf{x}_T^n,\epsilon}$ is approximated by the score network $s_\phi(\cdot, T)$, and we perform an empirical average. Figure 3 shows the estimated Fisher information plot versus the Picard step. We observe that the relative Fisher information decreases with n, which indicates that the smoothed distributions $p_{\mathbf{x}_T^n,\epsilon}$ become closer to q.

Empirical covariance matrix. We also compute the empirical covariance matrix of the latent embeddings. Since the target distribution q has identity covariance, the empirical covariance matrix provides a second-order diagnostic for prior alignment. In Figure 4, we observe that the empirical covariance matrix of the PC-VAE embeddings (i.e., $\Psi(E(\mathbf{u}_0), T), \mathbf{u}_0 \sim p_{\text{data}}$) is indeed close to identity.

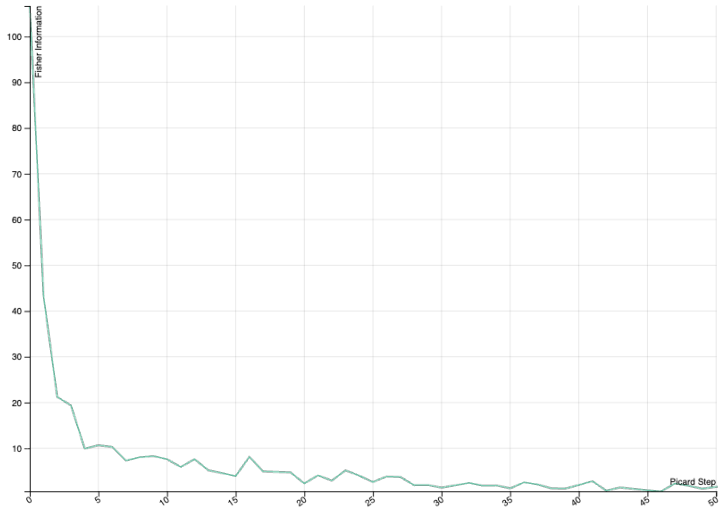

Figure 3: Estimated relative Fisher Information (vs) Picard step

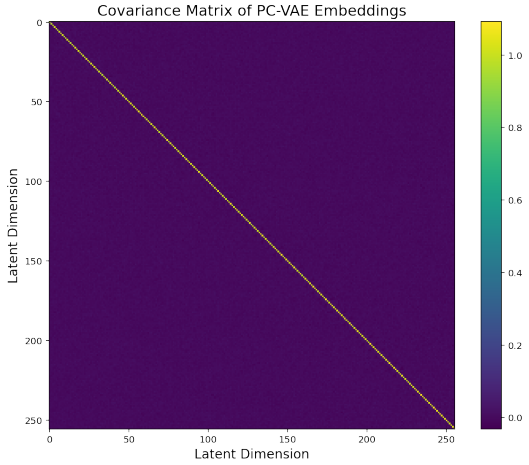

Figure 4: Empirical covariance matrix of PC-VAE embeddings, visualized as a heatmap.

I GENERATED SAMPLES

In this section we provide extended qualitative results for all 3 datasets used. The following figures contain uncurated, randomly generated images to illustrate the generation quality.

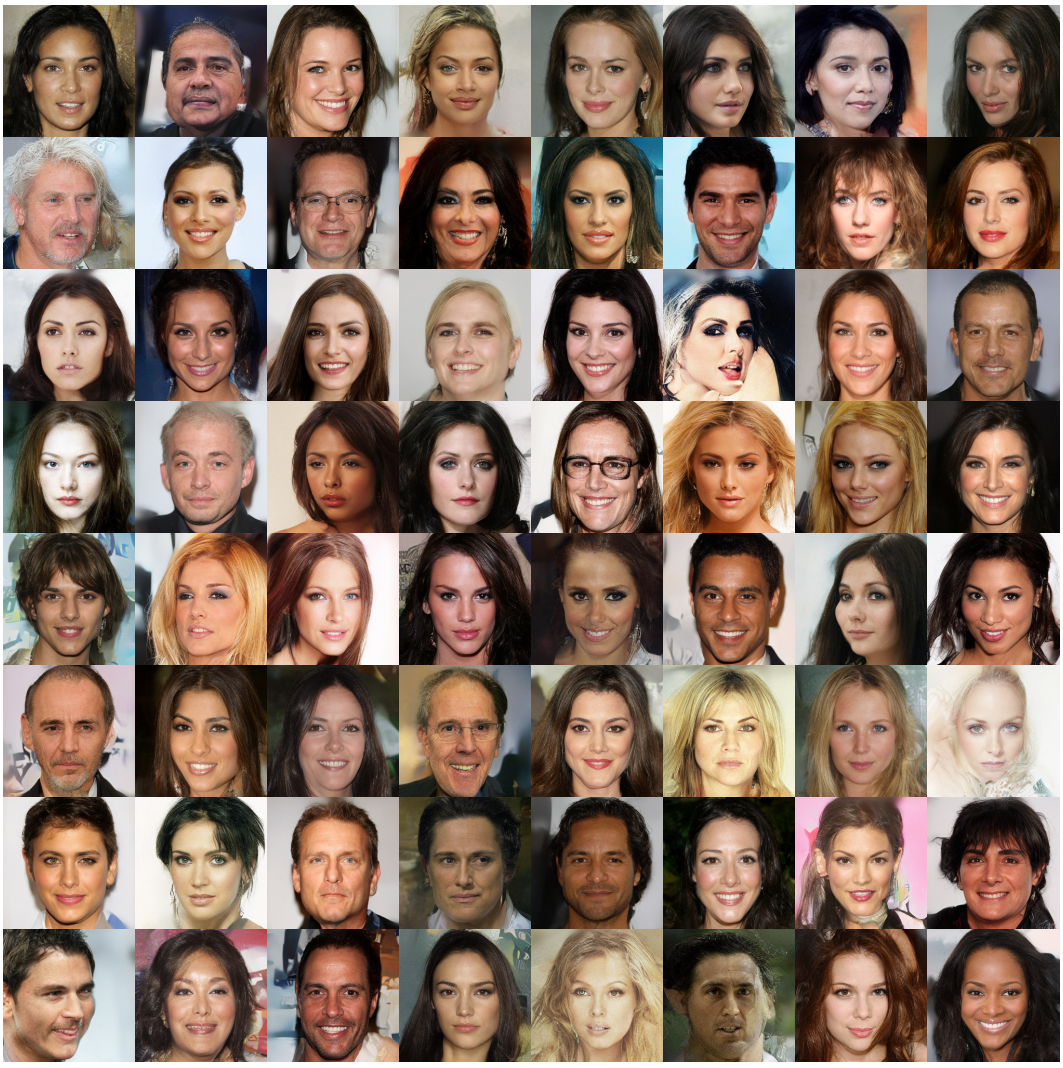

Figure 5: Uncurated samples generated by PC-VAE model trained on **CelebA-HQ**. Samples are drawn randomly without cherry-picking or rejection sampling.



Figure 6: Uncurated samples generated by PC-VAE model trained on **LSUN Bedroom**. Samples are drawn randomly without cherry-picking or rejection sampling.



Figure 7: Uncurated samples generated by PC-VAE model trained on **LSUN Church**. Samples are drawn randomly without cherry-picking or rejection sampling.