PFGM++: Unlocking the Potential of Physics-Inspired Generative Models

Yilun Xu [1] **Ziming Liu** [1] **Yonglong Tian** [1] **Shangyuan Tong** [1] **Max Tegmark** [1] **Tommi Jaakkola** [1]

Abstract

We introduce a new family of physics-inspired generative models termed *PFGM++* that unifies diffusion models and Poisson Flow Generative Models (PFGM). These models realize generative trajectories for N dimensional data by embedding paths in $N{+}D$ dimensional space while still controlling the progression with a simple scalar norm of the D additional variables. The new models reduce to PFGM when $D{=}1$ and to diffusion models when $D{\to}\infty$. The flexibility of choosing D allows us to trade off robustness against rigidity as increasing D results in more concentrated coupling between the data and the additional variable norms. We dispense with the biased large batch field targets used in PFGM and instead provide an unbiased perturbation-based objective similar to diffusion models. To explore different choices of D, we provide a direct alignment method for transferring well-tuned hyperparameters from diffusion models ($D{\to}\infty$) to any finite D values. Our experiments show that models with finite D can be superior to previous state-of-the-art diffusion models on CIFAR-10/FFHQ $64{\times}64$ datasets/LSUN Churches $256{\times}256$, with median Ds. In class-conditional setting, $D{=}2048$ yields current state-of-the-art FID of 1.74 on CIFAR-10 without additional training. Furthermore, we demonstrate that models with smaller D exhibit improved robustness against modeling errors. Code is available at https://github.com/Newbeeer/pfgmpp

1. Introduction

Physics continues to inspire new deep generative models such as *diffusion models* (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021b; Karras et al., 2022) based on thermodynamics (Jarzynski, 1997) or *Poisson flow generative models* (PFGM) (Xu et al., 2022) derived from electrostatics (Griffiths, 2005). The associated generative processes involve iteratively de-noising samples by following physically meaningful trajectories. Diffusion models learn a noise-level dependent score function so as to reverse the effects of forward diffusion, progressively reducing the noise level σ along the generation trajectory. PFGMs in turn augment N-dimensional data points with an extra dimension and evolve samples drawn from a uniform distribution over a large $N{+}1$-dimensional hemisphere back to the $z{=}0$ hyperplane where the clean data (as charges) reside by tracing learned electric field lines. Diffusion models in particular have been demonstrated across image (Song et al., 2021b; Nichol et al., 2022a; Ramesh et al., 2022), 3D (Zeng et al., 2022; Poole et al., 2022), audio (Kong et al., 2020; Chen et al., 2020) and biological data (Shi et al., 2021; Watson et al., 2022) generation, and have more stable training objectives compared to GANs (Arjovsky et al., 2017; Brock et al., 2019). More recent PFGM (Xu et al., 2022) rival diffusion models on image generation.

In this paper, we introduce a broader family of physics-inspired generative models that we call ***PFGM++***. These models extend the electrostatic view into higher dimensions through multi-dimensional $\mathbf{z} \in \mathbb{R}^D$ augmentations. When interpreting N-dimensional data points $\mathbf{x}$ as positive charges, the electric field lines define a surjection from a uniform distribution on an infinite $N{+}D$-dimensional hemisphere to the data distribution located on the $\mathbf{z}{=}\mathbf{0}$ hyperplane. We can therefore draw generative samples by following the electric field lines, evolving points from the hemisphere back to the $\mathbf{z}{=}\mathbf{0}$ hyperplane. We leverage the symmetry of z to reduce the vector to a scalar $\|\mathbf{z}\|_2 = r$, simplifying the sampling process. The use of symmetry turns the aforementioned surjection into a bijection between an easy-to-sample prior on a large $r = r_{\text{max}}$ hyper-cylinder to the data distribution. The symmetry reduction also permits D to take any positive values, including reals. We derive a new perturbation-based training objective akin to denoising score matching (Vincent, 2011) that avoids the need to use large batches to construct electric field line targets in PFGM. The perturbation-based objective is more efficient, unbiased, and compatible with paired sample training of conditional generation models.

[1]Massachusetts Institute of Technology, MIT, Cambridge, MA, USA. Correspondence to: Yilun Xu <ylxu@mit.edu>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

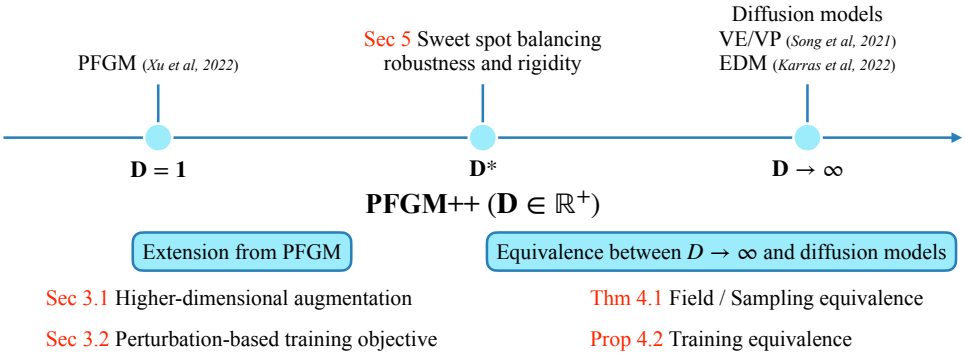


Figure 1. Overview of paper contributions and structure. PFGM++ unify PFGM and diffusion models, as well as the potential to combine their strengths (robustness and rigidity).

The models in the new family differ based on their augmentation dimension D which is now a hyper-parameter. By setting $D{=}1$ we obtain PFGM while $D{\to}\infty$ leads to diffusion models. We establish $D{\to}\infty$ equivalence with popular diffusion models (Song et al., 2021b; Karras et al., 2022) both in terms of their training objectives as well as their inferential processes. We demonstrate that the hyper-parameter D controls the balance between robustness and rigidity: using a small D widens the distribution of noisy training sample norms in comparison to the norm of the augmented variables, leading to a more robust generative process. However, small D also leads to a heavy-tailed problem of training samples, making the training process more challenging (neural networks cannot rigidly predict the fields correctly). Neither $D{=}1$ nor $D{\to}\infty$ offers an ideal balance between being insensitive to missteps (robustness) and allowing effective learning (rigidity). Instead, we adjust D in response to different architectures and tasks. To facilitate quickly finding the best D we provide an alignment method to directly transfer other hyperparameters across different choices of D.

Experimentally, we show that some models with finite D outperform the previous state-of-the-art diffusion models ($D{\to}\infty$), *i.e.*, EDM (Karras et al., 2022), on image generation tasks. In particular, intermediate $D{=}2048/128/131072$ achieve the best performance among other choices of D ranging from 64 to ∞, with min FID scores of $1.91/2.43/6.52$ on CIFAR-10/ FFHQ $64{\times}64$/LSUN Churches $256{\times}256$ datasets in unconditional generation, using $35/79/99$ NFE. In class-conditional generation, $D{=}2048$ achieves new state-of-the-art FID of 1.74 on CIFAR-10. We further verify that in general, decreasing D leads to improved robustness against a variety of sources of errors, *i.e.*, controlled noise injection, large sampling step sizes and post-training quantization.

Our contributions are summarized as follows: **(1)** We propose PFGM++ as a new family of generative models based on expanding augmented dimensions and show that symmetries involved enable us to define generative paths simply based on the scalar norm of the augmented variables (Sec 3.1); **(2)** We propose a perturbation-based objective to dispense with any biased large batch derived electric field targets, allowing unbiased training (Sec 3.2); **(3)** We prove that the score field and the training objective of diffusion models arise in the limit $D{\to}\infty$ (Sec 4); **(4)** We demonstrate the trade-off between robustness and rigidity by varying D (Sec 5). *We also detail the hyperparameter transfer procedures from EDM/DDPM ($D \to \infty$) to finite Ds in Appendix C.2*; **(5)** We empirically show that models with finite D achieve superior performance to diffusion models while exhibiting improved robustness (Sec 6).

2. Background and Related Works

Diffusion Model Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021b; Karras et al., 2022) are often presented as a pair of two processes. A fixed forward process governs the training of the model, which learns to denoise data of different noise levels. A corresponding backward process involves utilizing the trained model iteratively to denoise the samples starting from a fully noisy prior distribution. Karras et al. (2022) propose a unifying framework for popular diffusion models (Variance Exploding (VE)/Variance Preserving (VP) (Song et al., 2021b) and EDM (Karras et al., 2022)), and their sampling process can be understood as traveling in time with a probability flow ordinary differential equation (ODE):

$$\mathrm{d}\mathbf{x} = -\dot{\sigma}(t)\sigma(t)\nabla_{\mathbf{x}} \log p_{\sigma(t)}(\mathbf{x})\mathrm{d}t$$

where $\sigma(t)$ is a predefined noise schedule w.r.t. time, and $\nabla_{\mathbf{x}} \log p_{\sigma(t)}(\mathbf{x})$ is the score of noise-injected data distribution at time t. A neural network $f_\theta(\mathbf{x}, \sigma)$ is trained to learn the score $\nabla_{\mathbf{x}} \log p_{\sigma(t)}(\mathbf{x})$ by minimizing a weighted sum of the denoising score-matching objectives (Vincent, 2011):

$$\mathbb{E}_{\sigma\sim p(\sigma)}\lambda(\sigma)\mathbb{E}_{\mathbf{y}\sim p(\mathbf{y})}\mathbb{E}_{\mathbf{x}\sim p_\sigma(\mathbf{x}|\mathbf{y})} \left[\|f_\theta(\mathbf{x},\sigma) - \nabla_{\mathbf{x}} \log p_\sigma(\mathbf{x}|\mathbf{y})\|_2^2\right] \quad (1)$$

where $p(\sigma)$ defines a training distribution of noise levels, $\lambda(\sigma)$ is a weighting function, $p(\mathbf{y})$ is the data distribution, and $p_\sigma(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{y}, \sigma^2\boldsymbol{I})$ defines a Gaussian perturbation kernel which samples a noisy version $\mathbf{x}$ of the clean data $\mathbf{y}$. Please refer to Table 1 in Karras et al. (2022) for specific instantiations of different diffusion models.

PFGM Inspired by the theory of electrostatics (Griffiths, 2005), Xu et al. (2022) propose Poisson flow generative models (PFGM), which interpret the N-dimensional data $\mathbf{x} \in \mathbb{R}^N$ as electric charges in an $N{+}1$-dimensional space augmented with an extra dimension z: $\tilde{\mathbf{x}} = (\mathbf{x}, z) \in \mathbb{R}^{N+1}$. In particular, the training data is placed on the $z{=}0$ hyperplane, and the electric field lines emitted by the charges define a bijection between the data distribution and a uniform distribution on the infinite hemisphere of the augmented space[1]. To perform generative modeling, PFGM learn the following high-dimensional electric field, which is the derivative of the electric potential in a Poisson equation:

$$\mathbf{E}(\tilde{\mathbf{x}}) = \frac{1}{S_N(1)} \int \frac{\tilde{\mathbf{x}} - \tilde{\mathbf{y}}}{\|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{N+1}} p(\mathbf{y}) \mathrm{d}\mathbf{y} \tag{2}$$

where $S_N(1)$ is the surface area of a unit N-sphere (a geometric constant), and $p(\mathbf{y})$ is the data distribution. Samples are then generated by following the electric field lines, which are described by the ODE $\mathrm{d}\tilde{\mathbf{x}} = \mathbf{E}(\tilde{\mathbf{x}})\mathrm{d}t$. In practice, the network is trained to estimate a normalized version of the following empirical electric field: $\hat{\mathbf{E}}(\tilde{\mathbf{x}}) = c(\tilde{\mathbf{x}}) \sum_{i=1}^n \frac{\tilde{\mathbf{x}} - \tilde{\mathbf{y}}_i}{\|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}_i\|^{N+1}}$, where $c(\tilde{\mathbf{x}}) = 1/\sum_{i=1}^n \frac{1}{\|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}_i\|^{N+1}}$ and $\{\tilde{\mathbf{y}}_i\}_{i=1}^n \sim \tilde{p}(\tilde{\mathbf{y}})$ is a large batch used to approximate the integral in Eq. (2). The training objective is minimizing the ℓ_2-loss between the neural model prediction $f_\theta(\tilde{\mathbf{x}})$ and the normalized field $\mathbf{E}(\tilde{\mathbf{x}})/\|\mathbf{E}(\tilde{\mathbf{x}})\|$ at various positions of $\tilde{\mathbf{x}}$. These positions are heuristically designed to carefully cover the regions that the sampling trajectories pass through.

Phases of Score Field Xu et al. (2023) show that the score field in the forward process of diffusion models can be decomposed into three phases. When moving from the near field (Phase 1) to the far field (Phase 3), the perturbed data get influenced by more modes in the data distribution. They show that the posterior $p_{0|\sigma}(\mathbf{y}|\mathbf{x}) \propto p_\sigma(\mathbf{x}|\mathbf{y})p(\mathbf{y})$ serves as a phase indicator, as it gradually evolves from a delta distribution to uniform distribution when shifting from Phase 1 to Phase 3. The relevant concepts of phases have also been explored in Karras et al. (2022); Choi et al. (2022); Xiao et al. (2022). Similar to the PFGM training objective, Xu et al. (2023) approximates the score field by large batches to reduce the variance of training targets in Phase 2, where multiple data points exert comparable but distinct influences on the scores. These observations inspire us to align the phases of different Ds in Sec 4.

[1]In practice, the hemisphere is projected to a hyperplane $z{=}z_{\max}$, so that all samples have the initial z.

3. PFGM++: A Novel Generative Framework

In this section, we present our new family of generative models PFGM++, generalizing PFGM (Xu et al., 2022) in terms of the augmented space dimensionality. We show that the electric fields in $N{+}D$-dimensional space with $D \in \mathbb{Z}^+$ still constitute a valid generative model (Sec 3.1). Furthermore, we show that the additional D-dimensional augmented variable can be condensed into their scalar norm due to the inherent symmetry of the electric field. To improve the training process, we propose an efficient perturbation-based objective for training PFGM++ (Sec 3.2) without relying on the large batch approximation in the original PFGM.

3.1. Electric field in $N{+}D$-dimensional space

While PFGM (Xu et al., 2022) consider the electric field in a $N{+}1$-dimensional augmented space, we augment the data $\mathbf{x}$ with D-dimensional variables $\mathbf{z} = (z_1, \ldots, z_D)$, *i.e.*, $\tilde{\mathbf{x}} = (\mathbf{x}, \mathbf{z})$ and $D \in \mathbb{Z}^+$. Similar to the $N{+}1$-dimensional electric field (Eq. (2)), the electric field at the augmented data $\tilde{\mathbf{x}} = (\mathbf{x}, \mathbf{z}) \in \mathbb{R}^{N+D}$ is:

$$\mathbf{E}(\tilde{\mathbf{x}}) = \frac{1}{S_{N+D-1}(1)} \int \frac{\tilde{\mathbf{x}} - \tilde{\mathbf{y}}}{\|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{N+D}} p(\mathbf{y}) d\mathbf{y} \tag{3}$$

Analogous to the theoretical results presented in PFGM, with the electric field as the drift term, the ODE $d\tilde{\mathbf{x}}{=}\mathbf{E}(\tilde{\mathbf{x}})\mathrm{d}t$ defines a surjection from a uniform distribution on an infinite $N{+}D$-*dim* hemisphere (the measure we used on hemisphere is defined as the "surface area" of the hypersphere, *i.e.*, $\bar{r}^{N+D-1}d\Omega$, where $d\Omega$ is the solid angle on the $N{+}D{-}1$-dimensional sphere with radius $\bar{r}$) and the data distribution on the N-*dim* $\mathbf{z}{=}\mathbf{0}$ hyperplane. However, the mapping has $SO(D)$ symmetry on the surface of D-dim cylinder $\sum_{i=1}^D z_i^2 = r^2$ for any positive r. We provide an illustrative example at the bottom of Fig. 2 ($D{=}2, N{=}1$), where the electric flux emitted from a line segment (red) has rotational symmetry through the ring area (blue) on the $z_1^2 + z_2^2 = r^2$ cylinder. Hence, instead of modeling the individual behavior of each z_i, it suffices to track the norm of augmented variables — $r(\tilde{\mathbf{x}}) = \|\mathbf{z}\|_2$ — due to symmetry. Specifically, note that $\mathrm{d}z_i = \mathbf{E}(\tilde{\mathbf{x}})_{z_i}\mathrm{d}t$, and the time derivative of r is

$$\begin{aligned}\frac{\mathrm{d}r}{\mathrm{d}t} &= \sum_{i=1}^D \frac{z_i}{r}\frac{\mathrm{d}z_i}{\mathrm{d}t} = \int \frac{\sum_{i=1}^D z_i^2}{S_{N+D-1}(1) r \|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{N+D}} p(\mathbf{y})\mathrm{d}\mathbf{y} \\ &= \frac{1}{S_{N+D-1}(1)} \int \frac{r}{\|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{N+D}} p(\mathbf{y})\mathrm{d}\mathbf{y}\end{aligned}$$

Henceforth we replace the notation for augmented data with $\tilde{\mathbf{x}} = (\mathbf{x}, r)$ for simplicity. After the symmetry reduction, the field to be modeled has a similar form as Eq. (3) except that the last D sub-components $\{\mathbf{E}(\tilde{\mathbf{x}})_{z_i}\}_{i=1}^D$ are condensed into a scalar $E(\tilde{\mathbf{x}})_r = \frac{1}{S_{N+D-1}(1)} \int \frac{r}{\|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{N+D}} p(\mathbf{y})\mathrm{d}\mathbf{y}$. Therefore, we can use the physically meaningful r as the anchor

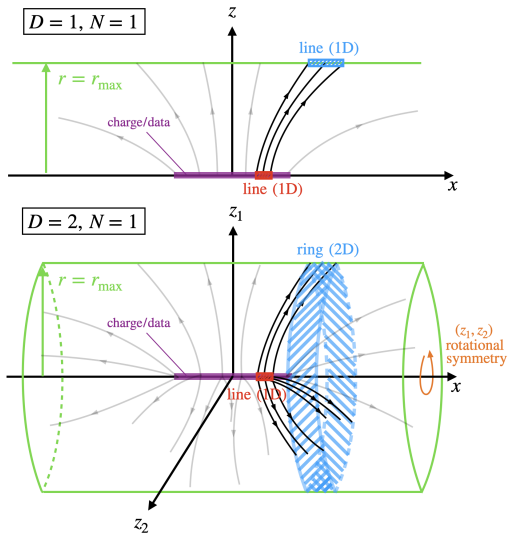


Figure 2. The augmented dimension D affects electric field lines (gray), which connect charge/data on a line (purple) to latent space (green). When $D = 1$ (top) or $D = 2$ (bottom), electric field lines map the same red line segment to a blue line segment or onto a blue ring, respectively. The mapping defined by electric lines has $SO(2)$ symmetry on the surface of $z_1^2 + z_2^2 = r^2$ cylinder.

variable in the ODE $\mathrm{d}\mathbf{x}/\mathrm{d}r$ by change-of-variable:

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}r} = \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t}\frac{\mathrm{d}t}{\mathrm{d}r} = \frac{\mathbf{E}(\tilde{\mathbf{x}})_{\mathbf{x}}}{E(\tilde{\mathbf{x}})_r} \tag{4}$$

Indeed, the ODE $\mathrm{d}\mathbf{x}/\mathrm{d}r$ turns the aforementioned surjection into a bijection between an easy-to-sample prior distribution on the $r{=}r_{\max}$ hyper-cylinder[2] and the data distribution on $r{=}0$ (*i.e.*, $\mathbf{z}{=}\mathbf{0}$) hyperplane. The following theorem states the observation formally:

Theorem 3.1. *Assume the data distribution $p \in \mathcal{C}^1$ and p has compact support. As $r_{max}{\to}\infty$, for $D \in \mathbb{R}^+$, the ODE $\mathrm{d}\mathbf{x}/\mathrm{d}r = \mathbf{E}(\tilde{\mathbf{x}})_{\mathbf{x}}/E(\tilde{\mathbf{x}})_r$ defines a bijection between $\lim_{r_{max}\to\infty} p_{r_{max}}(\mathbf{x}) \propto \lim_{r_{max}\to\infty} r_{max}^D/(\|\mathbf{x}\|_2^2 + r_{max}^2)^{\frac{N+D}{2}}$ when $r = r_{max}$ and the data distribution p when $r = 0$.*

Proof sketch. The r-dependent intermediate distribution of the ODE (Eq. (4)) is $p_r(\mathbf{x}){\propto}\int r^D/\|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{N+D}p(\mathbf{y})d\mathbf{y}$, which satisfies initial/terminal conditions, *i.e.*, $p_{r=0}{=}p$, $\lim_{r_{\max}\to\infty} p_{r_{\max}} \propto \lim_{r_{\max}\to\infty} r_{\max}^D/(\|\mathbf{x}\|_2^2 + r_{\max}^2)^{\frac{N+D}{2}}$, as well as the continuity equation of the ODE, *i.e.*, $\partial_r p_r + \nabla_{\mathbf{x}} \cdot (p_r \mathbf{E}(\tilde{\mathbf{x}})_{\mathbf{x}}/E(\tilde{\mathbf{x}})_r) = 0$. □

We defer the formal proof to Appendix A.1. Note that in the theorem we further extend the domain of D from positive integers to positive real numbers. In practice, the starting condition of the ODE is some sufficiently large $r_{\max}$ such that $p_{r_{\max}}(\mathbf{x}) \propto r_{\max}^D/(\|\mathbf{x}\|_2^2 + r_{\max}^2)^{\frac{N+D}{2}}$. The terminal condition is $r{=}\,0$, which represents the generated samples reaching the data support. The proposed PFGM++ framework thus permits choosing arbitrary D, including $D = 1$ which recovers the original PFGM formulation. Interestingly, we will also show that when $D{\to}\infty$, PFGM++ recover the diffusion models (Sec 4). In addition, as discussed in Sec 5, the choice of D is important, since it controls two properties of the associated electric field, *i.e.*, robustness and rigidity, which affect the sampling performance.

[2]The hyper-cylinder here is consistent with the hemisphere in PFGM (Xu et al., 2022), because hyper-cylinders degrade to hyperplanes for $D = 1$, which are in turn isomorphic to hemispheres.

3.2. New objective with Perturbation Kernel

Although the training process in PFGM can be directly applied to PFGM++, we propose a more efficient training objective to dispense with the large batch in PFGM. The objective from PFGM paper (Xu et al., 2022) requires sampling a large batch of data $\{\mathbf{y}_i\}_{i=1}^n{\sim}p^n(\mathbf{y})$ in each training step to approximate the integral in the electric field (Eq. (3)):

$$\mathbb{E}_{\{\mathbf{y}_i\}_{i=1}^n\sim p^n(\mathbf{y})}\mathbb{E}_{\tilde{\mathbf{x}}\sim\tilde{p}_{\text{train}}(\tilde{\mathbf{x}}|\tilde{\mathbf{y}}_1=(\mathbf{y}_1,\mathbf{0}))}\left[\left\|f_\theta(\tilde{\mathbf{x}}) - \frac{\sum_{i=0}^{n-1}\frac{\tilde{\mathbf{x}}-\tilde{\mathbf{y}}_i}{\|\tilde{\mathbf{x}}-\tilde{\mathbf{y}}_i\|^{N+D}}}{\left\|\sum_{i=0}^{n-1}\frac{\tilde{\mathbf{x}}-\tilde{\mathbf{y}}_i}{\|\tilde{\mathbf{x}}-\tilde{\mathbf{y}}_i\|^{N+D}}\right\|_2 + \gamma}\right\|_2^2\right]$$

where $\tilde{p}_{\text{train}}$ is heuristically designed to cover the regions that the backward ODE traverses and γ in the denominator is a tiny positive number to prevent numerical issues. This objective has several obvious drawbacks: *(1)* The large batch incurs additional overheads; *(2)* Its minimizer is a biased estimator of the electric field (Eq. (3)); *(3)* The large batch is incompatible with typical paired sample training of conditional generation, where each condition is paired with only one sample, such as text-to-image (Rombach et al., 2021; Saharia et al., 2022) and text-to-3D generation (Poole et al., 2022; Nichol et al., 2022b).

To remedy these issues, we propose a perturbation-based objective without the need for the large batch, while achieving an unbiased minimizer and enabling paired sample training of conditional generation. Inspired by denoising score-matching (Vincent, 2011), we design the perturbation kernel to guarantee that the minimizer in the following square loss objective matches the ground-truth electric field in Eq. (3):

$$\mathbb{E}_{r\sim p(r)}\mathbb{E}_{p(\mathbf{y})}\mathbb{E}_{p_r(\mathbf{x}|\mathbf{y})}\left[\|f_\theta(\tilde{\mathbf{x}}) - (\tilde{\mathbf{x}} - \tilde{\mathbf{y}})\|_2^2\right] \tag{5}$$

where $r \in (0, \infty)$, $p(r)$ is the training distribution over r, $p_r(\mathbf{x}|\mathbf{y})$ is the perturbation kernel and $\tilde{\mathbf{y}}{=}(\mathbf{y}, 0)/\tilde{\mathbf{x}}{=}(\mathbf{x}, r)$ are the clean/perturbed augmented data. The minimizer of Eq. (5) is $f_\theta^*(\tilde{\mathbf{x}}){\propto}\int p_r(\mathbf{x}|\mathbf{y})(\tilde{\mathbf{x}} - \tilde{\mathbf{y}})p(\mathbf{y})\mathrm{d}\mathbf{y}$, which matches the direction of electric field $\mathbf{E}(\tilde{\mathbf{x}}){\propto}\int(\tilde{\mathbf{x}} - \tilde{\mathbf{y}})/\|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{N+D}p(\mathbf{y})\mathrm{d}\mathbf{y}$ when setting the perturbation kernel to $p_r(\mathbf{x}){\propto}1/(\|\mathbf{x}\|_2^2 + r^2)^{\frac{N+D}{2}}$. Denoting the r-dependent intermediate marginal distribution as $p_r(\mathbf{x}){=}\int p_r(\mathbf{x}|\mathbf{y})p(\mathbf{y})d\mathbf{y}$, the following proposition states

that the choice of $p_r(\cdot|\mathbf{y})$ guarantee that the minimizer of the square loss to match the direction of the electric field:

Proposition 3.2. *With perturbation kernel $p_r(\mathbf{x}|\mathbf{y}) \propto 1/(\|\mathbf{x}-\mathbf{y}\|_2^2 + r^2)^{\frac{N+D}{2}}$, for $\forall \mathbf{x} \in \mathbb{R}^N, r > 0$, the minimizer $f_\theta^*(\tilde{\mathbf{x}})$ in the PFGM++ objective (Eq. (5)) matches the direction of electric field $\mathbf{E}(\tilde{\mathbf{x}})$ in Eq. (3). Specifically, $f_\theta^*(\tilde{\mathbf{x}}) \propto (S_{N+D-1}(1)/p_r(\mathbf{x}))\mathbf{E}(\tilde{\mathbf{x}})$.*

We defer the proof to Appendix A.2. The proposition indicates that the minimizer $f_\theta^*(\tilde{\mathbf{x}})$ can match the direction of $\mathbf{E}(\tilde{\mathbf{x}})$ with sufficient data and model capacity. The current training target in Eq. (5) is the directional vector between the clean data $\tilde{\mathbf{y}}$ and perturbed data $\tilde{\mathbf{x}}$ akin to denoising score-matching for diffusion models (Song et al., 2021b; Karras et al., 2022). In addition, the new objective allows for conditional generations under a one-sample-per-condition setup. Since the perturbation kernel is isotropic, we can decompose $p_r(\cdot|\mathbf{y})$ in hyperspherical coordinates to $\mathcal{U}_\psi(\psi)p_r(R)$, where $\mathcal{U}_\psi$ is the uniform distribution over the angle component and the distribution of the perturbed radius $R = \|\mathbf{x}-\mathbf{y}\|_2$ is

$$p_r(R) \propto \frac{R^{N-1}}{(R^2+r^2)^{\frac{N+D}{2}}}$$

We defer the practical sampling procedure of the perturbation kernel to Appendix B. The mean of the r-dependent radius distribution $p_r(R)$ is around $r\sqrt{N/D}$. Hence we explicitly normalize the target in Eq. (5) by $r/\sqrt{D}$, to keep the norm of the target around the constant $\sqrt{N}$, similar to diffusion models (Song et al., 2021b). In addition, we drop the last dimension of the target because it is a constant — $(\tilde{\mathbf{x}}-\tilde{\mathbf{y}})_r/(r/\sqrt{D}) = \sqrt{D}$. Together, the new objective is

$$\mathbb{E}_{r\sim p(r)}\mathbb{E}_{p(\tilde{\mathbf{y}})}\mathbb{E}_{p_r(\tilde{\mathbf{x}}|\tilde{\mathbf{y}})}\left[\left\|f_\theta(\tilde{\mathbf{x}}) - \frac{\mathbf{x}-\mathbf{y}}{r/\sqrt{D}}\right\|_2^2\right] \quad (6)$$

which is essentially a rescaled version of Eq. (5). After training the neural network through objective Eq. (6), we can use the ODE (Eq. (4)) anchored by r to generate samples, *i.e.*, $\mathrm{d}\mathbf{x}/\mathrm{d}r = \mathbf{E}(\tilde{\mathbf{x}})_\mathbf{x}/E(\tilde{\mathbf{x}})_r = f_\theta(\tilde{\mathbf{x}})/\sqrt{D}$, starting from the prior distribution $p_{r_{\max}}$. We would like to highlight that PFGM++ maintain the same memory requirements as PFGM (i$D=1$) or diffusion models ($D=\infty$) during both training and sampling. This is achieved by condensing the high-dimensional augmented variable $\mathbf{z}$ into the scalar r.

4. Diffusion Models as $D\to\infty$ Special Cases

Diffusion models generate samples by simulating ODE/SDE involving the score function $\nabla_\mathbf{x}\log p_\sigma(\mathbf{x})$ at different intermediate distributions p_σ (Song et al., 2021b; Karras et al., 2022), where σ is the standard deviation of the Gaussian kernel. In this section, we show that both sampling and training schemes in diffusion models are equivalent to those in $D\to\infty$ case under the PFGM++ framework. To begin with, we show that the electric field (Eq. (3)) in PFGM++ has the same direction as the score function when D tends to infinity, and their sampling processes are also identical.

Theorem 4.1. *Assume the data distribution $p \in \mathcal{C}^1$. Consider taking the limit $D\to\infty$ while holding $\sigma = r/\sqrt{D}$ fixed. Then, for all $\mathbf{x}$,*

$$\lim_{\substack{D\to\infty\\ r=\sigma\sqrt{D}}}\left\|-\frac{\sqrt{D}}{E(\tilde{\mathbf{x}})_r}\mathbf{E}(\tilde{\mathbf{x}})_\mathbf{x} - \sigma\nabla_\mathbf{x}\log p_{\sigma=r/\sqrt{D}}(\mathbf{x})\right\|_2 = 0$$

where $\mathbf{E}(\tilde{\mathbf{x}} = (\mathbf{x}, r))_\mathbf{x}$ is given in Eq. (3). Further, given the same initial point, the trajectory of the PFGM++ ODE ($\mathrm{d}\mathbf{x}/\mathrm{d}r = \mathbf{E}(\tilde{\mathbf{x}})_\mathbf{x}/E(\tilde{\mathbf{x}})_r$) matches the diffusion ODE (Karras et al., 2022) ($\mathrm{d}\mathbf{x}/\mathrm{d}t = -\dot{\sigma}(t)\sigma(t)\nabla_\mathbf{x}\log p_{\sigma(t)}(\mathbf{x})$) in the same limit.

Proof sketch. By re-expressing the $\mathbf{x}$ component $\mathbf{E}(\tilde{\mathbf{x}})_\mathbf{x}$ in the electric field and the score $\nabla_\mathbf{x}\log p_\sigma$ in diffusion models, the proof boils down to show that $\lim_{D\to\infty, r=\sigma\sqrt{D}} p_r(\mathbf{x}|\mathbf{y}) \propto \exp(-\|\mathbf{x}-\mathbf{y}\|_2^2/2\sigma^2)$ for $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^{N+D}$:

$$\begin{aligned}
&\lim_{D\to\infty, r=\sigma\sqrt{D}} \frac{1}{(\|\mathbf{x}-\mathbf{y}\|_2^2 + r^2)^{\frac{N+D}{2}}} \\
\propto &\lim_{D\to\infty, r=\sigma\sqrt{D}} e^{-\frac{(N+D)}{2}\ln(1+\frac{\|\mathbf{x}-\mathbf{y}\|^2}{r^2})} \\
= &\lim_{D\to\infty, r=\sigma\sqrt{D}} e^{-\frac{(N+D)\|\mathbf{x}-\mathbf{y}\|_2^2}{2r^2}} = e^{-\frac{\|\mathbf{x}-\mathbf{y}\|_2^2}{2\sigma^2}}
\end{aligned} \quad (7)$$

The equivalence of trajectories can be proven by change-of-variable $\mathrm{d}\sigma = \mathrm{d}r/\sqrt{D}$. Their prior distributions are also the same since $\lim_{D\to\infty} p_{r_{\max}=\sigma_{\max}\sqrt{D}}(\mathbf{x}) = \mathcal{N}(\mathbf{0}, \sigma_{\max}\boldsymbol{I})$. □

We defer the formal proof to Appendix A.3. Since $\|\mathbf{x}-\mathbf{y}\|_2^2/r^2 \approx N/D$ when $\mathbf{x}\sim p_r(\mathbf{x}), \mathbf{y}\sim p(\mathbf{y})$, Eq. (7) approximately holds under the condition $D \gg N$. Remarkably, the theorem states that PFGM++ recover the field and sampling of previous popular diffusion models, such as VE/VP (Song & Ermon, 2020) and EDM (Karras et al., 2022), by choosing the appropriate schedule and scale function in Karras et al. (2022).

In addition to the field and sampling equivalence, we demonstrate that the proposed PFGM++ objective (Eq. (6)) with perturbation kernel $p_r(\mathbf{x}|\mathbf{y}) \propto 1/(\|\mathbf{x}-\mathbf{y}\|_2^2 + r^2)^{\frac{N+D}{2}}$ recovers the weighted sum of the denoising score matching objective (Vincent, 2011) for training continuous diffusion model (Karras et al., 2022; Song et al., 2021b) when $D\to\infty$. All previous objectives for training diffusion models can be subsumed in the following form (Karras et al., 2022), under different parameterizations of the neural networks f_θ:

$$\mathbb{E}_{\sigma\sim p(\sigma)}\lambda(\sigma)\mathbb{E}_{p(\mathbf{y})}\mathbb{E}_{p_\sigma(\mathbf{x}|\mathbf{y})}\left[\left\|f_\theta(\mathbf{x},\sigma) - \frac{\mathbf{x}-\mathbf{y}}{\sigma}\right\|_2^2\right] \quad (8)$$

where $p_\sigma(\mathbf{x}|\mathbf{y}) \propto \exp(-\|\mathbf{x}-\mathbf{y}\|_2^2/2\sigma^2)$. The objective of the diffusion models resembles the one of PFGM++ (Eq. (6)). Indeed, we show that when $D\to\infty$, the minimizer of the proposed PFGM++ objective at $\tilde{\mathbf{x}}=(\mathbf{x}, r)$ is $f_\theta^*(\mathbf{x}, r = \sigma\sqrt{D}) = -\sigma\nabla_{\mathbf{x}} \log p_\sigma(\mathbf{x})$, the same as the minimizer of diffusion objective at the noise level $\sigma=r/\sqrt{D}$.

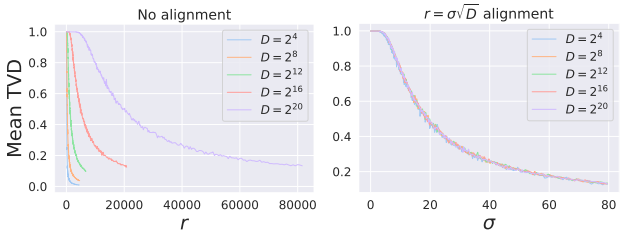
Proposition 4.2. *When $r=\sigma\sqrt{D}, D\to\infty$, the minimizer in the PFGM++ objective (Eq. (6)) is* ***equivalent to*** *the minimizer in the weighted sum of denoising score matching objective (Eq. (8))*

We defer the proof to Appendix A.4. The proposition states that the training objective of diffusion models is essentially the same as PFGM++'s when $D\to\infty$. Combined with Theorem 4.1, PFGM++ thus recover both the training and sampling processes of diffusion models when $D\to\infty$.

Transfer hyperparameters to finite Ds The training hyperparameters of diffusion models ($D\to\infty$) have been highly optimized through a series of works (Ho et al., 2020; Song et al., 2021b; Karras et al., 2022). It motivates us to transfer hyperparameters, such as $r_{\max}$ and $p(r)$, of $D\to\infty$ to finite Ds. Here we present an alignment method that enables a "zero-shot" transfer of hyperparameters across different Ds. Our alignment method is inspired by the concept of phases in Xu et al. (2023), which demonstrates that the score field in the forward process of diffusion models can be decomposed into three successive phases. As we move from the near field (Phase 1) to the far field (Phase 3), the perturbed data become influenced by more modes in the data distribution. The authors show that the posterior $p_{0|\sigma}$ serves as a phase indicator, as it gradually evolves from a delta distribution to a uniform distribution when transitioning from Phase 1 to Phase 3.

We aim to align the phases for two distinct $D_1, D_2 > 0$. In Appendix C.1, we demonstrate that when $r \propto \sqrt{D}$, the phase of the intermediate distribution p_r is approximately invariant to all $D > 0$ (including $D\to\infty$). In other words, when $r_{D_1}/r_{D_2} = \sqrt{D_1/D_2}$, the phases of $p_{r_{D_1}}$ and $p_{r_{D_2}}$, under D_1 and D_2 respectively, are roughly aligned. Theorem 4.1 further shows that the relation $r=\sigma\sqrt{D}$ makes PFGM++ equivalent to diffusion models when $D\to\infty$. Together, the $r=\sigma\sqrt{D}$ formula aligns the phases of p_σ in diffusion models and $p_{r=\sigma\sqrt{D}}$ in PFGM++ for $\forall D>0$. Such alignment enables directly transferring the finely tuned hyperparameters $\sigma_{\max}, p(\sigma)$ in previous state-of-the-art diffusion models (Karras et al., 2022) with $r_{\max}=\sigma_{\max}\sqrt{D}, p(r)=p(\sigma=r/\sqrt{D})/\sqrt{D}$. *We put the practical hyperparameter transfer procedures in Appendix C.2.*

We empirically verify the alignment formula on the CIFAR-10 (Krizhevsky, 2009). Xu et al. (2023) shows that the posterior $p_{0|r}(\mathbf{y}|\mathbf{x}) \propto p_r(\mathbf{x}|\mathbf{y})p(\mathbf{y})$ gradually grows towards a uniform distribution from the near to the far field. As a result, the mean total variational distance (TVD) between a uniform distribution and the posterior serves as an indicator of the phase of p_r: $\mathbb{E}_{p_r(\mathbf{x})}\text{TVD}\left(U(\cdot) \parallel p_{0|r}(\cdot|\mathbf{x})\right)$. Fig. 3 reports the mean TVD before and after the $r=\sigma\sqrt{D}$ alignment. We observe that the mean TVDs of a wide range of Ds take similar values after the alignment, suggesting that the phases of $p_{r=\sigma\sqrt{D}}$ are roughly aligned for different Ds.



(a) No alignment (b) $r=\sigma\sqrt{D}$ alignment

Figure 3. Mean TVD between the posterior $p_{0|r}(\cdot|\mathbf{x})$ ($\mathbf{x}$ is perturbed sample) and the uniform prior, w/o **(a)** and w/ **(b)** the phase alignment ($r=\sigma\sqrt{D}$).

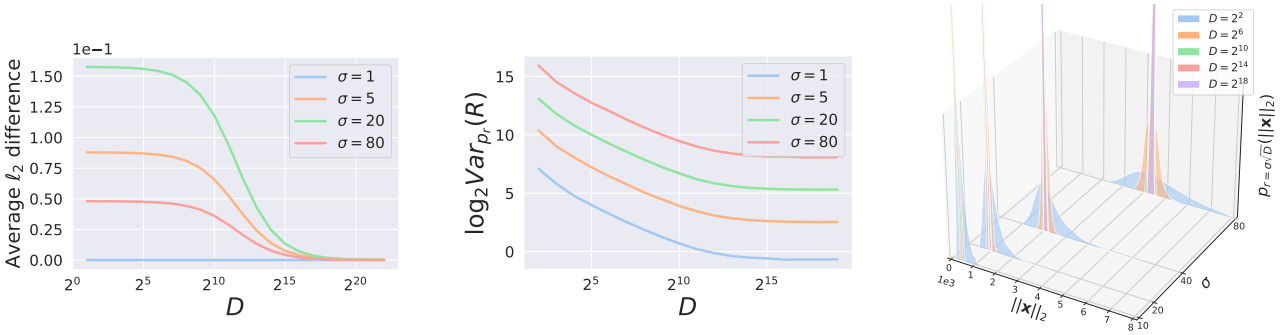
5. Balancing Robustness and Rigidity

In this section, we first delve into the behaviors of PFGM++ with different Ds (Sec 5.1) based on the alignment formula. Then we demonstrate how to leverage D to balance the robustness and rigidity of models (Sec 5.2). We defer all experimental details in this section to Appendix D.1.

5.1. Behavior of perturbation kernel when varying D

According to Theorem 4.1, when $D\to\infty$, the field in PFGM++ has the same direction as the score function, *i.e.*, $\sqrt{D}\mathbf{E}(\tilde{\mathbf{x}})_{\mathbf{x}}/E(\tilde{\mathbf{x}})_r=\sigma\nabla_{\mathbf{x}}\log p_{\sigma=r/\sqrt{D}}(\mathbf{x})$. In addition to the theoretical analysis, we provide further empirical study to characterize the convergence towards diffusion models as D increases. Fig. 4(a) reports the average ℓ_2 difference between the two quantities, *i.e.*, $\mathbb{E}_{p_\sigma(\mathbf{x})}[\|-\sqrt{D}\mathbf{E}(\tilde{\mathbf{x}})_{\mathbf{x}}/E(\tilde{\mathbf{x}})_r-\sigma\nabla_{\mathbf{x}}\log p_\sigma(\mathbf{x})\|_2]$ with $r=\sigma\sqrt{D}$. We observe that the difference monotonically decreases as a function of D, and converges to 0 as predicted by theory. For $\sigma=1$, the distance remains 0 since the empirical posterior $p_{0|r}$ (a categorical distribution) concentrates around a single example for all D. This is because the distance between the perturbed data $\mathbf{x}$ and a specific data point is much smaller than the distance between $\mathbf{x}$ and any other data points in the training set. The posterior will gradually allocate all the mass on a certain datapoint for all D when decreasing σ.

Next, we examine the behavior of the perturbation kernel after the phase alignment. Recall that the isotropic perturbation kernel $p_r(\mathbf{x}|\mathbf{y}) \propto 1/(\|\mathbf{x}-\mathbf{y}\|_2^2+r^2)^{\frac{N+D}{2}}$ can be decomposed into a uniform angle component and a ra-



(a) (b) (c)

Figure 4. **(a)** Average ℓ_2 difference between scaled electric field and score function, versus D. **(b)** Log-variance of radius distribution versus D. **(c)** Density of radius distributions $p_{r=\sigma\sqrt{D}}(R)$ with varying σ and D.

dius distribution $p_r(R) \propto R^{N-1}/(R^2+r^2)^{\frac{N+D}{2}}$. Fig. 4(b) shows the variance of the radius distribution significantly decreases as D increases. The results imply that with relatively large r, the norm of the training sample in $p_r(\mathbf{x})$ becomes increasingly concentrated around a specific value as D increases, reaching its highest level of concentration as $D\to\infty$ (diffusion models). Fig. 4(c) further shows the density of training sample norms in $p_{r=\sigma\sqrt{D}}(\mathbf{x})$ on CIFAR-10. We can see that the range of the high-mass region gradually shrinks when D increases.

5.2. Balancing the trade-off by controlling D

As noted in Xu et al. (2022), diffusion models ($D\to\infty$) are more susceptible to estimation errors compared to PFGM (D=1) due to the strong correlation between σ and the training sample norm, as demonstrated in Fig. 4(c). When D and r are large, the marginal distribution $p_r(\mathbf{x})$ is approximately supported on the sphere with radius $r\sqrt{N/D}$. The backward ODE can lead to unexpected results if the sampling trajectories deviate from this norm-r relation present in training samples. This phenomenon was empirically confirmed by Xu et al. (2022) for PFGM/diffusion models (D=1 and $D\to\infty$ cases) using a weaker architecture NCSNv2 (Song & Ermon, 2020), where PFGM was shown to be significantly more robust than diffusion models.

Smaller D, however, implies a heavy-tailed input distribution. Fig. 4(c) illustrates that the examples used as the input to the neural network have a broader range of norms when D is small. In particular, when $D<2^5$, the variance of perturbation radius can be larger than 2^{10} (Fig. 4(b)). This broader input range can be challenging for any finite-capacity neural network. Although Xu et al. (2022) introduced heuristics to bypass this issue in the D=1 case, *e.g.*, restricting the sampling/training regions, these heuristics also prevent the sampling process from faithfully recovering the data distribution.

Thus, we can view D as a parameter to optimize so as to balance the robustness of generation against rigidity that helps learning. Increased robustness allows practitioners to use smaller neural networks, e.g., by applying post-training quantization (Han et al., 2015; Banner et al., 2018). In other words, smaller D allows for more aggressive quantization/larger sampling step sizes/smaller architectures. These can be crucial in real-world applications where computational resources and storage are limited. On the other hand, such gains need to be balanced against easier training afforded by larger values of D. The ability to optimize the balance by varying D can be therefore advantageous. We expect that there exists a sweet spot of D in the middle striking the balance, as the model robustness and rigidity go in opposite directions.

6. Experiments

6.1. Image generation

We consider the widely used benchmarks CIFAR-10 32×32 (Krizhevsky, 2009), FFHQ 64×64 (Karras et al., 2018) and LSUN Churches 256 × 256 (Yu et al., 2015) for image generation. For training, we utilize the improved NCSN++/DDPM++ architectures, preconditioning techniques and hyperparameters from the state-of-the-art diffusion model EDM (Karras et al., 2022). Specifically, we use the alignment method developed in Sec 4 to transfer their tuned critical hyperparameters $\sigma_{\max}, \sigma_{\min}, p(\sigma)$ in the $D\to\infty$ case to finite D cases. According to the experimental results in Karras et al. (2018), the log-normal training distribution $p(\sigma)$ has the most substantial impact on the final performances. For ODE solver during sampling, we use Heun's 2nd method (Ascher & Petzold, 1998) as in EDM.

We compare models trained with $D\to\infty$ (EDM) and $D\in\{64, 128, 2048, 3072000\}$. In our experiments, we exclude the case of D=1 (PFGM) because the perturbation kernel is extremely heavy-tailed (Fig. 4(b)), making it difficult to integrate with our perturbation-based objective without the restrictive region heuristics proposed in Xu et al. (2022).

Table 1. CIFAR-10 sample quality (FID) and number of function evaluations (NFE).

	Min FID ↓	Top-3 Avg FID ↓	NFE ↓
DDPM (Ho et al., 2020)	3.17	-	1000
DDIM (Song et al., 2021a)	4.67	-	50
VE-ODE (Song et al., 2021b)	5.29	-	194
VP-ODE (Song et al., 2021b)	2.86	-	134
PFGM (Xu et al., 2022)	2.48	-	104
PFGM++ (unconditional)			
$D = 64$	1.96	1.98	35
$D = 128$	1.92	1.94	35
$D = 2048$	**1.91**	**1.93**	35
$D = 3072000$	1.99	2.02	35
$D \to \infty$ (Karras et al., 2022)	1.98	2.00	35
PFGM++ (class-conditional)			
$D = 2048$	**1.74**	-	35
$D \to \infty$ (Karras et al., 2022)	1.79	-	35

Table 2. FFHQ 64×64 sample quality (FID) with 79 NFE in unconditional setting

	Min FID ↓	Top-3 Avg FID ↓
$D = 128$	**2.43**	2.48
$D = 2048$	2.46	**2.47**
$D = 3072000$	2.49	2.52
$D \to \infty$ (Karras et al., 2022)	2.53	2.54

We also exclude the small $D = 64$ for the higher-resolution dataset FFHQ. Since the data dimension of LSUN Churches is relatively high (N=196608), we only try D=131072 to validate our ideas while saving computations. We include several popular generative models for reference and defer more training and sampling details to Appendix D.

Results: In Table 1, 2 and 3, we report the sample quality measured by the FID score (Heusel et al., 2017) (lower is better), and inference speed measured by the number of function evaluations. As in EDM, we report the minimum FID score over checkpoints. Since we empirically observe a large variation of FID scores on FFHQ across checkpoints (Appendix D.4), we also use the average FID score over the Top-3 checkpoints as another metric. Our main findings are **(1) Median Ds outperform previous best diffusion models (Karras et al., 2022) under PFGM++ framework.** We observe that the D=2048/128/131072 cases achieve the best performance among our choices on CIFAR-10/FFHQ/LSUN Churches, with min FID score of 1.91/2.43/6.52 in unconditional setting, using the perturbation-based objective. In addition, median Ds obtain better Top-3 average FID scores than EDM across datasets in unconditional setting and achieve a current state-of-the-art FID score of 1.74 in CIFAR-10 class-conditional setting. **(2) There is a sweet spot between $(1, \infty)$.** Neither small D nor infinite D obtains the best performance, which confirms that there is a sweet spot in the middle, well-balancing rigidity and robustness. **(3)**

Table 3. LSUN Churches 256×256 sample quality (FID) with 99 NFE in unconditional setting

	Min FID ↓	Top-3 Avg FID ↓
$D = 131072$	**6.52**	6.58
$D \to \infty$ (Karras et al., 2022)	6.63	6.66

Model with $D \gg N$ recovers diffusion models. We find that model with sufficiently large D roughly matches the performance of diffusion models, as predicted by the theory. Further results in Appendix E.1 show that D=3072000 and diffusion models obtain the same FID score when using a more stable training target (Xu et al., 2023) to mitigate the variations between different runs and checkpoints.

6.2. Model robustness versus D

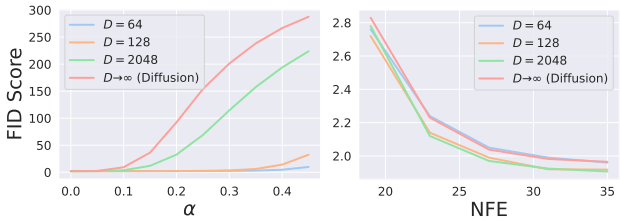


Figure 5. FID score versus **(left)** α and **(right)** NFE on CIFAR-10.

In Section 5, we show that the model robustness degrades with an increasing D by analyzing the behavior of perturbation kernels. To further validate the phenomenon, we conduct three sets of experiments with different sources of errors on CIFAR-10. We defer more details to Appendix D.5. Firstly, we perform controlled experiments to compare the robustness of models quantitatively. To simulate the errors, we inject noise into the intermediate point $\mathbf{x}_r$ in each of the 35 ODE steps: $\mathbf{x}_r = \mathbf{x}_r + \alpha \boldsymbol{\epsilon}_r$ where $\boldsymbol{\epsilon}_r \sim \mathcal{N}(\mathbf{0}, r/\sqrt{D}\boldsymbol{I})$, and α is a positive number controlling the amount of noise. Fig. 5(a) demonstrates that as α increases, FID score exhibits a much slower degradation for smaller D. In particular, when D=64, 128, the sample quality degrades gracefully. We further visualize the generated samples in Appendix E.2. It shows that when α=0.2, models with D=64, 128 can still produce clean images while the sampling process of diffusion models ($D \to \infty$) breaks down.

In addition to the controlled scenario, we conduct two more realistic experiments: **(1)** We introduce more estimation error of neural networks by applying post-training quantization (Sung et al., 2015), which can directly compress neural networks without fine-tuning. Table 4 reports the FID score with varying quantization bit-widths for the convolution weight values. We can see that finite Ds have better robustness than the infinite case, and a lower D exhibits

a larger performance gain when applying lower bit-widths quantization. **(2)** We increase the discretization error during sampling by using smaller NFEs, *i.e.*, larger sample steps. As shown in Fig. 5(b), gaps between D=128 and diffusion models gradually widen, indicating greater robustness against the discretization error. The rigidity issue of smaller D also affects the robustness to discretization error, as D=64 is consistently inferior to D=128.

Table 4. FID score versus quantization bit-widths on CIFAR-10.

Quantization bits:	9	8	7	6	5
$D = 64$	1.96	**1.96**	**2.12**	**2.94**	**28.50**
$D = 128$	1.93	1.97	2.15	3.68	34.26
$D = 2048$	**1.91**	1.97	**2.12**	5.67	47.02
$D \to \infty$	1.97	2.04	2.16	5.91	50.09

7. Conclusion and Future Directions

We present a new family of physics-inspired generative models called PFGM++, by extending the dimensionality of augmented variable in PFGM from 1 to $D \in \mathbb{R}^+$. Remarkably, PFGM++ includes diffusion models as special cases when $D \to \infty$. To address issues related to large batch training, we propose a perturbation-based objective. In addition, we show that D effectively controls the robustness and rigidity in the PFGM++ family. The multi-dimensional augmentation is crucial for empirical improvement, as it allows us to search for better models tailored to specific tasks and architectures, and enables the perturbation-based training objective (avoid the heavy-tailed problem when $D = 1$ as in PFGM (Xu et al., 2022)). On the other hand, the perturbation-based objective reduces training overheads and makes PFGM++ applicable to typical conditional generation settings. Empirical results show that models with finite values of D can perform better than previous state-of-the-art diffusion models, while also exhibiting improved robustness.

There are many potential avenues for future research in the PFGM++ framework. For example, it may be possible to identify the "sweet spot" value of D for different architectures and tasks by analyzing the behavior of errors. Since PFGM++ enables adjusting robustness, another direction is to apply aggressive network compression techniques, *i.e.*, pruning and low-bit training, to smaller D. Furthermore, there may be opportunities to develop stochastic samplers for PFGM++, with the reverse SDE in diffusion models as a special case. Lastly, PFGM++ may yield more significant performance improvements over diffusion models (the $D \to \infty$ case) in fields with less optimized network architectures. Our theoretical and experimental results demonstrate that PFGM++ exhibit superior robustness compared to diffusion models when using a smaller D. This increased robustness can translate to more substantial improvements on weaker architectures.we expect PFGM++ to have more significant performance gains than diffusion models in domains other than image generation, where network architectures have already been extensively optimized. We will leave the application of PFGM++ to other fields for future work.

Acknowledgements

We would like to thank Ji Lin and Haotian Tang for helpful discussions about post-training quantization. YX and TJ acknowledge support from MIT-DSTA Singapore collaboration, from NSF Expeditions grant (award 1918839) "Understanding the World Through Code", and from MIT-IBM Grand Challenge project. ZL and MT would like to thank the Center for Brains, Minds, and Machines (CBMM) for hospitality. ZL and MT are supported by The Casey and Family Foundation, the Foundational Questions Institute, the Rothberg Family Fund for Cognitive Science and IAIFI through NSF grant PHY-2019786. ST and TJ also acknowledge support from the ML for Pharmaceutical Discovery and Synthesis Consortium (MLPDS).

References

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, 2017.

Ascher, U. M. and Petzold, L. R. Computer methods for ordinary differential equations and differential-algebraic equations. 1998.

Banner, R., Nahshan, Y., and Soudry, D. Post training 4-bit quantization of convolutional networks for rapid-deployment. In *Neural Information Processing Systems*, 2018.

Brock, A., Donahue, J., and Simonyan, K. Large scale gan training for high fidelity natural image synthesis. *ArXiv*, abs/1809.11096, 2019.

Chen, N., Zhang, Y., Zen, H., Weiss, R. J., Norouzi, M., and Chan, W. Wavegrad: Estimating gradients for waveform generation. *ArXiv*, abs/2009.00713, 2020.

Choi, J., Lee, J., Shin, C., Kim, S., Kim, H., and Yoon, S. Perception prioritized training of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11472–11481, 2022.

Griffiths, D. J. Introduction to electrodynamics, 2005.

Han, S., Mao, H., and Dally, W. J. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. *arXiv: Computer Vision and Pattern Recognition*, 2015.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NIPS*, 2017.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *ArXiv*, abs/2006.11239, 2020.

Jarzynski, C. Equilibrium free-energy differences from nonequilibrium measurements: A master-equation approach. *Physical Review E*, 56:5018–5035, 1997.

Karras, T., Laine, S., and Aila, T. A style-based generator architecture for generative adversarial networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4396–4405, 2018.

Karras, T., Aittala, M., Aila, T., and Laine, S. Elucidating the design space of diffusion-based generative models. *ArXiv*, abs/2206.00364, 2022.

Kong, Z., Ping, W., Huang, J., Zhao, K., and Catanzaro, B. Diffwave: A versatile diffusion model for audio synthesis. *ArXiv*, abs/2009.09761, 2020.

Krizhevsky, A. Learning multiple layers of features from tiny images. 2009.

Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., and Chen, M. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *ICML*, 2022a.

Nichol, A., Jun, H., Dhariwal, P., Mishkin, P., and Chen, M. Point-e: A system for generating 3d point clouds from complex prompts. *ArXiv*, abs/2212.08751, 2022b.

Poole, B., Jain, A., Barron, J. T., and Mildenhall, B. Dreamfusion: Text-to-3d using 2d diffusion. *ArXiv*, abs/2209.14988, 2022.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical text-conditional image generation with clip latents. *ArXiv*, abs/2204.06125, 2022.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10674–10685, 2021.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E. L., Ghasemipour, S. K. S., Ayan, B. K., Mahdavi, S. S., Lopes, R. G., Salimans, T., Ho, J., Fleet, D. J., and Norouzi, M. Photorealistic text-to-image diffusion models with deep language understanding. *ArXiv*, abs/2205.11487, 2022.

Shi, C., Luo, S., Xu, M., and Tang, J. Learning gradient fields for molecular conformation generation. In *ICML*, 2021.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pp. 2256–2265. PMLR, 2015.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. *ArXiv*, abs/2010.02502, 2021a.

Song, Y. and Ermon, S. Improved techniques for training score-based generative models. *ArXiv*, abs/2006.09011, 2020.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. *ArXiv*, abs/2011.13456, 2021b.

Sung, W., Shin, S., and Hwang, K. Resiliency of deep neural networks under quantization. *ArXiv*, abs/1511.06488, 2015.

Vincent, P. A connection between score matching and denoising autoencoders. *Neural Computation*, 23:1661–1674, 2011.

Watson, J. L., Juergens, D., Bennett, N. R., Trippe, B. L., Yim, J., Eisenach, H. E., Ahern, W., Borst, A. J., Ragotte, R. J., Milles, L. F., Wicky, B. I. M., Hanikel, N., Pellock, S. J., Courbet, A., Sheffler, W., Wang, J., Venkatesh, P., Sappington, I., Torres, S. V., Lauko, A., Bortoli, V. D., Mathieu, E., Barzilay, R., Jaakkola, T., DiMaio, F., Baek, M., and Baker, D. Broadly applicable and accurate protein design by integrating structure prediction networks and diffusion generative models. *bioRxiv*, 2022.

Xiao, Z., Kreis, K., and Vahdat, A. Tackling the generative learning trilemma with denoising diffusion GANs. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=JprM0p-q0Co.

Xu, Y., Liu, Z., Tegmark, M., and Jaakkola, T. Poisson flow generative models. *ArXiv*, abs/2209.11178, 2022.

Xu, Y., Tong, S., and Jaakkola, T. Stable target field for reduced variance score estimation in diffusion models. *ArXiv*, abs/2302.00670, 2023.

Yu, F., Zhang, Y., Song, S., Seff, A., and Xiao, J. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

Zeng, X., Vahdat, A., Williams, F., Gojcic, Z., Litany, O., Fidler, S., and Kreis, K. Lion: Latent point diffusion models for 3d shape generation. *ArXiv*, abs/2210.06978, 2022.

Appendix

A. Proofs

A.1. Proof of Theorem 3.1

Theorem 3.1. *Assume the data distribution $p \in \mathcal{C}^1$ and p has compact support. As $r_{max} \to \infty$, for $D \in \mathbb{R}^+$, the ODE $\mathrm{d}\mathbf{x}/\mathrm{d}r = \mathbf{E}(\tilde{\mathbf{x}})_{\mathbf{x}}/E(\tilde{\mathbf{x}})_r$ defines a bijection between $\lim_{r_{max}\to\infty} p_{r_{max}}(\mathbf{x}) \propto \lim_{r_{max}\to\infty} r_{max}^D/(\|\mathbf{x}\|_2^2 + r_{max}^2)^{\frac{N+D}{2}}$ when $r = r_{max}$ and the data distribution p when $r = 0$.*

Proof. Let $q_r(\mathbf{x}) = \frac{S_{D-1}}{S_{N+D-1}} \int r^D/\|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{N+D} p(\mathbf{y}) d\mathbf{y}$, where S_n is the surface area of the n-sphere. We will show that $q_r \propto \int r^D/\|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{N+D} p(\mathbf{y}) d\mathbf{y}$ is equal to the r-dependent marginal distribution p_r by verifying (1) the starting distribution is correct when r=0; (2) the continuity equation holds, *i.e.*, $\partial_r q_r + \nabla \cdot (q_r \mathbf{E}(\tilde{\mathbf{x}})_{\mathbf{x}}/E(\tilde{\mathbf{x}})_r) = 0$. The starting distribution is $\lim_{r\to 0} q_r(\mathbf{x}) \propto \lim_{r\to 0} \int r^D/\|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{N+D} p(\mathbf{y}) d\mathbf{y} \propto p(\mathbf{x})$, which confirms that q_r=p. The continuity equation can be expressed as:

$$
\begin{aligned}
&\partial_r q_r + \nabla \cdot (q_r \mathbf{E}(\tilde{\mathbf{x}})_{\mathbf{x}}/E(\tilde{\mathbf{x}})_r) \\
&= \partial_r \left(\int \frac{r^D}{\|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{N+D}} p(\mathbf{y}) d\mathbf{y} \right) + \nabla \cdot \left(\int \frac{r^D}{\|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{N+D}} p(\mathbf{y}) d\mathbf{y} \frac{\int \frac{\tilde{\mathbf{x}} - \tilde{\mathbf{y}}}{\|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{N+D}} p(\mathbf{y}) d\mathbf{y}}{\int \frac{r}{\|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{N+D}} p(\mathbf{y}) d\mathbf{y}} \right) \\
&= \int \left(\frac{D r^{D-1}}{\|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{N+D}} - \frac{(N+D) r}{\|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{N+D-2}} \right) p(\mathbf{y}) d\mathbf{y} + \nabla \cdot \left(r^{D-1} \int \frac{\tilde{\mathbf{x}} - \tilde{\mathbf{y}}}{\|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{N+D}} p(\mathbf{y}) d\mathbf{y} \right) \\
&= \int \left(\frac{D r^{D-1}}{\|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{N+D}} - \frac{(N+D) r}{\|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{N+D-2}} \right) p(\mathbf{y}) d\mathbf{y} + \nabla \cdot \left(r^{D-1} \int \frac{\tilde{\mathbf{x}} - \tilde{\mathbf{y}}}{\|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{N+D}} p(\mathbf{y}) d\mathbf{y} \right) \\
&= \int \left(\frac{D r^{D-1}}{\|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{N+D}} - \frac{(N+D) r}{\|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{N+D-2}} \right) p(\mathbf{y}) d\mathbf{y} \\
&\quad + r^{D-1} \sum_{i=1}^{N} \int \frac{\|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{N+D} - \|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{N+D-2} (\mathbf{x}_i - \mathbf{y}_i)^2 (N+D)}{\|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{2(N+D)}} p(\mathbf{y}) d\mathbf{y} \\
&= \int \left(\frac{D r^{D-1}}{\|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{N+D}} - \frac{(N+D) r^{D+1}}{\|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{N+D-2}} \right) p(\mathbf{y}) d\mathbf{y} \\
&\quad + r^{D-1} \int \frac{N\|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{N+D} - \|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{N+D-2} \|\mathbf{x} - \mathbf{y}\|^2 (N+D)}{\|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{2(N+D)}} p(\mathbf{y}) d\mathbf{y} \\
&= r^{D-1} \int \frac{\|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{N+D} D - (N+D) r^2 \|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{N+D-2} + N\|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{N+D} - \|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{N+D-2} \|\mathbf{x} - \mathbf{y}\|^2 (N+D)}{\|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{2(N+D)}} p(\mathbf{y}) d\mathbf{y} \\
&= r^{D-1} \int \frac{(N+D)(\|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{N+D} - \|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{N+D-2} \|\mathbf{x} - \mathbf{y}\|^2) - (N+D) r^2 \|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{N+D-2}}{\|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{2(N+D)}} p(\mathbf{y}) d\mathbf{y} \\
&= r^{D-1} \int \frac{(N+D) r^2 \|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{N+D-2} - (N+D) r^2 \|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{N+D-2}}{\|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{2(N+D)}} p(\mathbf{y}) d\mathbf{y} \\
&= 0
\end{aligned}
$$

It means that q_r satisfies the continuity equation for any $r \in \mathbb{R}_{\geq 0}$. Together, we conclude that $q_r = p_r$. Lastly, note that the terminal distribution is

$$
\begin{aligned}
\lim_{r_{\max}\to\infty} p_{r_{\max}}(\mathbf{x}) &\propto \lim_{r_{\max}\to\infty} \int \frac{r_{\max}^D}{\|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{N+D}} p(\mathbf{y}) d\mathbf{y} = \lim_{r_{\max}\to\infty} \int \frac{r_{\max}^D}{(\|\mathbf{x} - \mathbf{y}\|^2 + r_{\max}^2)^{\frac{N+D}{2}}} p(\mathbf{y}) d\mathbf{y} \\
&= \lim_{r_{\max}\to\infty} \frac{r_{\max}^D}{(\|\mathbf{x}\|^2 + r_{\max}^2)^{\frac{N+D}{2}}} + \lim_{r_{\max}\to\infty} \int \left(\frac{r_{\max}^D}{(\|\mathbf{x} - \mathbf{y}\|^2 + r_{\max}^2)^{\frac{N+D}{2}}} - \frac{r_{\max}^D}{(\|\mathbf{x}\|^2 + r_{\max}^2)^{\frac{N+D}{2}}} \right) p(\mathbf{y}) d\mathbf{y} \\
&= \lim_{r_{\max}\to\infty} \frac{r_{\max}^D}{(\|\mathbf{x}\|^2 + r_{\max}^2)^{\frac{N+D}{2}}} \qquad (p \text{ has a compact support})
\end{aligned}
$$

□

A.2. Proof of Theorem 3.2

Proposition A.1. *With perturbation kernel $p_r(\mathbf{x}|\mathbf{y}) \propto 1/(\|\mathbf{x}-\mathbf{y}\|_2^2 + r^2)^{\frac{N+D}{2}}$, for $\forall \mathbf{x} \in \mathbb{R}^N, r > 0$, the minimizer $f_\theta^*(\tilde{\mathbf{x}})$ in the PFGM++ objective (Eq. (5)) matches the direction of electric field $\mathbf{E}(\tilde{\mathbf{x}})$ in Eq. (3). Specifically, $f_\theta^*(\tilde{\mathbf{x}}) \propto (S_{N+D-1}(1)/p_r(\mathbf{x}))\mathbf{E}(\tilde{\mathbf{x}})$.*

Proof. The minimizer at $\tilde{\mathbf{x}}$ in Eq. (5) is

$$f_\theta^*(\tilde{\mathbf{x}}) = \int p_r(\mathbf{y}|\mathbf{x})(\tilde{\mathbf{x}} - \tilde{\mathbf{y}})d\tilde{\mathbf{y}} = \frac{\int p_r(\mathbf{x}|\mathbf{y})(\tilde{\mathbf{x}} - \tilde{\mathbf{y}})p(\mathbf{y})d\mathbf{y}}{p_r(\mathbf{x})} \tag{9}$$

The choice of perturbation kernel is

$$p_r(\mathbf{x}|\mathbf{y}) \propto \frac{1}{\|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{N+D}} = \frac{1}{(\|\mathbf{x}-\mathbf{y}\|_2^2 + r^2)^{\frac{N+D}{2}}}$$

By substituting the perturbation kernel in Eq. (9), we have:

$$\begin{aligned} f_\theta^*(\tilde{\mathbf{x}}) &= \frac{\int \frac{\tilde{\mathbf{x}}-\tilde{\mathbf{y}}}{(\|\mathbf{x}-\mathbf{y}\|_2^2+r^2)^{\frac{N+D}{2}}} p(\mathbf{y})d\mathbf{y}}{p_r(\mathbf{x})} \\ &= \frac{\int \frac{\tilde{\mathbf{x}}-\tilde{\mathbf{y}}}{\|\tilde{\mathbf{x}}-\tilde{\mathbf{y}}\|_2{}^{N+D}} p(\mathbf{y})d\mathbf{y}}{p_r(\mathbf{x})} \\ &= (S_{N+D-1}(1)/p_r(\mathbf{x}))\mathbf{E}(\tilde{\mathbf{x}}) \end{aligned}$$

□

A.3. Proof of Theorem 4.1

Theorem 4.1. *Assume the data distribution $p \in \mathcal{C}^1$. Consider taking the limit $D \to \infty$ while holding $\sigma = r/\sqrt{D}$ fixed. Then, for all $\mathbf{x}$,*

$$\lim_{\substack{D\to\infty \\ r=\sigma\sqrt{D}}} \left\| -\frac{\sqrt{D}}{E(\tilde{\mathbf{x}})_r}\mathbf{E}(\tilde{\mathbf{x}})_{\mathbf{x}} - \sigma\nabla_{\mathbf{x}} \log p_{\sigma=r/\sqrt{D}}(\mathbf{x}) \right\|_2 = 0$$

where $\mathbf{E}(\tilde{\mathbf{x}} = (\mathbf{x}, r))_{\mathbf{x}}$ is given in Eq. (3). Further, given the same initial point, the trajectory of the PFGM++ ODE ($\mathrm{d}\mathbf{x}/\mathrm{d}r = \mathbf{E}(\tilde{\mathbf{x}})_{\mathbf{x}}/E(\tilde{\mathbf{x}})_r$) matches the diffusion ODE (Karras et al., 2022) ($\mathrm{d}\mathbf{x}/\mathrm{d}t = -\dot{\sigma}(t)\sigma(t)\nabla_{\mathbf{x}} \log p_{\sigma(t)}(\mathbf{x})$) in the same limit.

Proof. The $\mathbf{x}$ component in the Poisson field can be re-expressed as

$$\begin{aligned} \mathbf{E}(\tilde{\mathbf{x}})_{\mathbf{x}} &= \frac{1}{S_{N+D-1}(1)} \int \frac{\mathbf{x}-\mathbf{y}}{\|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{N+D}} p(\mathbf{y})d\mathbf{y} \\ &\propto \int p_r(\mathbf{x}|\mathbf{y})(\mathbf{x}-\mathbf{y})p(\mathbf{y})d\mathbf{y} \end{aligned}$$

where the perturbation kernel $p_r(\mathbf{x}|\mathbf{y}) \propto 1/(\|\mathbf{x}-\mathbf{y}\|_2^2 + r^2)^{\frac{N+D}{2}}$. The direction of the score can also be written down in a similar form:

$$\nabla_{\mathbf{x}} \log p_\sigma(\mathbf{x}) = \frac{\int p_\sigma(\mathbf{x}|\mathbf{y})\frac{\mathbf{y}-\mathbf{x}}{\sigma^2}p(\mathbf{y})d\mathbf{y}}{p_\sigma(\mathbf{x})} \propto \int p_\sigma(\mathbf{x}|\mathbf{y})(\mathbf{x}-\mathbf{y})p(\mathbf{y})d\mathbf{y}$$

where $p_\sigma(\mathbf{x}|\mathbf{y}) \propto \exp -\frac{\|\mathbf{x}-\mathbf{y}\|_2^2}{2\sigma^2}$. Since $p \in \mathcal{C}^1$, and obviously $p_r(\mathbf{x}|\mathbf{y}) \in C^1$, then $\lim_{D\to\infty} \int p_r(\mathbf{x}|\mathbf{y})(\mathbf{x}-\mathbf{y})p(\mathbf{y})d\mathbf{y} = \int \lim_{D\to\infty} p_r(\mathbf{x}|\mathbf{y})(\mathbf{x}-\mathbf{y})p(\mathbf{y})d\mathbf{y}$. It suffices to prove that the perturbation kernel $p_r(\mathbf{x}|\mathbf{y})$ point-wisely converge to the

Gaussian kernel $p_\sigma(\mathbf{x}|\mathbf{y})$, *i.e.*, $\lim_{D\to\infty} p_r(\mathbf{x}|\mathbf{y}) = p_\sigma(\mathbf{x}|\mathbf{y})$, to ensure $\mathbf{E}(\mathbf{x})_\mathbf{x} \propto \nabla_\mathbf{x} \log p_\sigma(\mathbf{x})$. Given $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^N$,

$$
\begin{aligned}
\lim_{D\to\infty} p_r(\mathbf{x}|\mathbf{y}) &\propto \lim_{D\to\infty} \frac{1}{(\|\mathbf{x}-\mathbf{y}\|_2^2 + r^2)^{\frac{N+D}{2}}} \\
&= \lim_{D\to\infty} (\|\mathbf{x}-\mathbf{y}\|_2^2 + r^2)^{-\frac{N+D}{2}} \\
&\propto \lim_{D\to\infty} (1 + \frac{\|\mathbf{x}-\mathbf{y}\|_2^2}{r^2})^{-\frac{N+D}{2}} \\
&= \lim_{D\to\infty} (1 + \frac{\|\mathbf{x}-\mathbf{y}\|_2^2}{D\sigma^2})^{-\frac{N+D}{2}} && (r = \sigma\sqrt{D}) \\
&= \lim_{D\to\infty} \exp\left(-\frac{N+D}{2}\ln(1 + \frac{\|\mathbf{x}-\mathbf{y}\|_2^2}{D\sigma^2})\right) \\
&= \lim_{D\to\infty} \exp\left(-\frac{N+D}{2}\frac{\|\mathbf{x}-\mathbf{y}\|_2^2}{D\sigma^2}\right) && (\lim_{D\to\infty} \tfrac{\|\mathbf{x}-\mathbf{y}\|_2^2}{D\sigma^2} = 0) \\
&= \exp -\frac{\|\mathbf{x}-\mathbf{y}\|_2^2}{2\sigma^2} \\
&\propto p_\sigma(\mathbf{x}|\mathbf{y})
\end{aligned}
$$

Hence $\lim_{D\to\infty} p_r(\mathbf{x}|\mathbf{y}) = p_\sigma(\mathbf{x}|\mathbf{y})$, and we establish that $\mathbf{E}(\tilde{\mathbf{x}})_\mathbf{x} \propto \nabla_\mathbf{x} \log p_\sigma(\mathbf{x})$. We can rewrite the drift term in the PFGM++ ODE as

$$
\begin{aligned}
\lim_{\substack{D\to\infty \\ r=\sigma\sqrt{D}}} \sqrt{D}\mathbf{E}(\tilde{\mathbf{x}})_\mathbf{x}/E(\tilde{\mathbf{x}})_r &= \lim_{\substack{D\to\infty \\ r=\sigma\sqrt{D}}} \frac{\sqrt{D}\int p_r(\mathbf{x}|\mathbf{y})(\mathbf{x}-\mathbf{y})p(\mathbf{y})d\mathbf{y}}{\int p_r(\mathbf{x}|\mathbf{y})(-r)p(\mathbf{y})d\mathbf{y}} \\
&= \lim_{\substack{D\to\infty \\ r=\sigma\sqrt{D}}} \frac{\sqrt{D}\int p_r(\mathbf{x}|\mathbf{y})(\mathbf{y}-\mathbf{x})p(\mathbf{y})d\mathbf{y}}{rp_r(\mathbf{x})} \\
&= \lim_{\substack{D\to\infty \\ r=\sigma\sqrt{D}}} \frac{\sqrt{D}\int p_\sigma(\mathbf{x}|\mathbf{y})(\mathbf{y}-\mathbf{x})p(\mathbf{y})d\mathbf{y}}{rp_\sigma(\mathbf{x})} \\
&= \sigma\nabla_\mathbf{x} \log p_\sigma(\mathbf{x}) \qquad (\nabla_\mathbf{x} \log p_\sigma(\mathbf{x}) = \frac{\int p_\sigma(\mathbf{x}|\mathbf{y})\frac{\mathbf{y}-\mathbf{x}}{\sigma^2}p(\mathbf{y})d\mathbf{y}}{p_\sigma(\mathbf{x})})
\end{aligned} \tag{10}
$$

which establishes the first part of the theorem. For the second part, by the change-of-variable $d\sigma = dr/\sqrt{D}$, the PFGM++ ODE is

$$
\begin{aligned}
\lim_{\substack{D\to\infty \\ r=\sigma\sqrt{D}}} \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\sigma} &= \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}r} \cdot \frac{\mathrm{d}r}{\mathrm{d}\sigma} \\
&= \lim_{\substack{D\to\infty \\ r=\sigma\sqrt{D}}} \mathbf{E}(\tilde{\mathbf{x}})_\mathbf{x} \cdot E(\tilde{\mathbf{x}})_r^{-1} \cdot \sqrt{D} \\
&= \lim_{\substack{D\to\infty \\ r=\sigma\sqrt{D}}} \frac{\sigma\nabla_\mathbf{x} \log p_\sigma(\mathbf{x})}{\sqrt{D}} \cdot \sqrt{D} \qquad \text{(by Eq. (10))} \\
&= \sigma\nabla_\mathbf{x} \log p_\sigma(\mathbf{x})
\end{aligned}
$$

which is equivalent to the diffusion ODE. □

A.4. Proof of Proposition 4.2

Proposition A.2. *When $r = \sigma\sqrt{D}, D \to \infty$, the minimizer in the PFGM++ objective (Eq. (6)) is* ***equaivalent to*** *the minimizer in the weighted sum of denoising score matching objective (Eq. (8))*

Proof. For $\forall \mathbf{x} \in \mathbb{R}^N$, the minimizer in PFGM++ objective (Eq. (6)) at point $\tilde{\mathbf{x}} = (\mathbf{x}, r)$ is

$$\begin{aligned} f^*_{\theta,\text{PFGM++}}(\tilde{\mathbf{x}}) &= \lim_{\substack{D\to\infty \\ r=\sigma\sqrt{D}}} \frac{\int p_r(\mathbf{x}|\mathbf{y}) \frac{\mathbf{x}-\mathbf{y}}{r/\sqrt{D}} p(\mathbf{y}) d\mathbf{y}}{p_r(\mathbf{x})} \\ &= \lim_{\substack{D\to\infty \\ r=\sigma\sqrt{D}}} \frac{\int p_\sigma(\mathbf{x}|\mathbf{y}) \frac{\mathbf{x}-\mathbf{y}}{r/\sqrt{D}} p(\mathbf{y}) d\mathbf{y}}{p_\sigma(\mathbf{x})} \qquad \text{(By Theorem 4.1, } \lim_{D\to\infty} p_r(\mathbf{x}|\mathbf{y}) = p_\sigma(\mathbf{x}|\mathbf{y})\text{)} \\ &= \frac{\int p_\sigma(\mathbf{x}|\mathbf{y}) \frac{\mathbf{x}-\mathbf{y}}{\sigma} p(\mathbf{y}) d\mathbf{y}}{p_\sigma(\mathbf{x})} \end{aligned} \tag{11}$$

On the other hand, the minimizer in denoising score matching at point $\mathbf{x}$ in noise level $\sigma = r/\sqrt{N+D}$ is

$$f^*_{\theta,\text{DSM}}(\mathbf{x}, \sigma) = \frac{\int p_\sigma(\mathbf{x}|\mathbf{y}) \frac{\mathbf{x}-\mathbf{y}}{\sigma} p(\mathbf{y}) d\mathbf{y}}{p_\sigma(\mathbf{x})} \tag{12}$$

Combining Eq. (11) and Eq. (12), we have

$$\lim_{\substack{D\to\infty \\ r=\sigma\sqrt{D}}} f^*_{\theta,\text{PFGM++}}(\mathbf{x}, \sigma\sqrt{N+D}) = f^*_{\theta,\text{DSM}}(\mathbf{x}, \sigma)$$

□

B. Practical Sampling Procedures of Perturbation Kernel and Prior Distribution

In this section, we discuss how to simple from the perturbation kernel $p_r(\mathbf{x}|\mathbf{y}) \propto 1/(\|\mathbf{x}-\mathbf{y}\|_2^2 + r^2)^{\frac{N+D}{2}}$ in practice. We first decompose $p_r(\cdot|\mathbf{y})$ in hyperspherical coordinates to $\mathcal{U}_\psi(\psi) p_r(R)$, where $\mathcal{U}_\psi$ is the uniform distribution over the angle component and the distribution of the perturbed radius $R = \|\mathbf{x}-\mathbf{y}\|_2$ is

$$p_r(R) \propto \frac{R^{N-1}}{(R^2+r^2)^{\frac{N+D}{2}}} \tag{13}$$

The sampling procedure of the radius distribution encompasses three steps:

$$\begin{aligned} R_1 &\sim \text{Beta}(\alpha = \frac{N}{2}, \beta = \frac{D}{2}) \\ R_2 &= \frac{R_1}{1-R_1} \\ R_3 &= \sqrt{r^2 R_2} \end{aligned}$$

Next, we prove that $p(R_3) = p_r(R_3)$. Note that the pdf of the inverse beta distribution is

$$p(R_2) \propto R_2^{\frac{N}{2}-1} (1+R_2)^{-\frac{N+D}{2}}$$

By change-of-variable, the pdf of $R_3 = \sqrt{r_{max}^2 R_2}$ is

$$
\begin{aligned}
p(R_3) &\propto R_2^{\frac{N}{2}-1}(1+R_2)^{-\frac{N}{2}-\frac{D}{2}} * \frac{2R_3}{r_{max}^2} \\
&\propto \frac{R_3 R_2^{\frac{N}{2}-1}}{(1+R_2)^{\frac{N+D}{2}}} \\
&= \frac{(R_3/r)^{N-1}}{(1+(R_3^2/r^2))^{\frac{N+D}{2}}} \\
&\propto \frac{R_3^{N-1}}{(1+(R_3^2/r^2))^{\frac{N+D}{2}}} \\
&\propto \frac{R_3^{N-1}}{(r^2+R_3^2)^{\frac{N+D}{2}}} \propto p_r(R_3) \qquad \text{(By Eq. (13))}
\end{aligned}
$$

Note that R_1 has mean $\frac{N}{N+D}$ and variance $O(\frac{ND}{(N+D)^3})$. Hence when $D = O(N)$, $p_r(R)$ would highly concentrate on a specific value, resolving the heavy-tailed problem. We can sample the uniform angel component by $\mathbf{u} = \mathbf{w}/\|\mathbf{w}\|, \mathbf{w} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_{N\times N})$. Together, sampling from the perturbation kernel $p_r(\mathbf{x}|\mathbf{y})$ is equivalent to setting $\mathbf{x} = \mathbf{y} + R_3\mathbf{u}$. On the other hand, the prior distribution is

$$
p_{r_{\text{max}}}(\mathbf{x}) \propto \lim_{r_{\text{max}}\to\infty} \int r_{\text{max}}^D / \|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^{N+D} p(\mathbf{y}) d\mathbf{y} = \lim_{r_{\text{max}}\to\infty} r_{\text{max}}^D / (\|\mathbf{x}\|^2 + r_{\text{max}}^2)^{\frac{N+D}{2}}
$$

We observe that $p_{r_{\text{max}}}(\mathbf{x})$ the same as the perturbation kernel $p_{r_{\text{max}}}(\mathbf{x}|\mathbf{y} = \mathbf{0})$. Hence we can sample from the prior following $\mathbf{x} = R_3\mathbf{u}$ with $R_3, \mathbf{u}$ defined above and $r = r_{\text{max}}$.

C. $r = \sigma\sqrt{D}$ for Phase Alignment

C.1. Analysis

In this section, we examine the phase of intermediate marginal distribution p_r under different Ds to derive an alignment method for hyper-parameters. Consider a N-dimensional dataset $\mathcal{D}$ in which the average distance to the nearest neighbor is about l. We consider an arbitrary datapoint $\mathbf{x}_1 \in \mathcal{D}$ and denote its nearest neighbor as $\mathbf{x}_2$. We assume $\|\mathbf{x}_1 - \mathbf{x}_2\|_2 = l$, and uniform prior on $\mathcal{D}$.

To characterize the phases of $p_r, \forall r > 0$, we study the perturbation point $\mathbf{y} \sim p_r(\mathbf{y}|\mathbf{x}_1)$. According to Appendix B, the distance $\|\mathbf{x}_1 - \mathbf{y}\|$ is roughly $r\sqrt{\frac{N}{D-1}}$. Since $p_r(\mathbf{y}|\mathbf{x}_1)$ is isotropic, with high probability, the two vectors $\mathbf{y} - \mathbf{x}_1, \mathbf{x}_2 - \mathbf{x}_1$ are approximately orthogonal. In particular, the vector product $(\mathbf{y} - \mathbf{x}_1)^T(\mathbf{x}_1 - \mathbf{x}_2) = O(\frac{1}{\sqrt{N}}\|\mathbf{y} - \mathbf{x}_1\|\|\mathbf{x}_1 - \mathbf{x}_2\|) = O(\frac{rl}{\sqrt{D}})$ w.h.p. It reveals that $\|\mathbf{y} - \mathbf{x}_2\| = \sqrt{l^2 + r^2\frac{N}{D-1} + O(\frac{rl}{\sqrt{D}})}$. Fig. 6 depicts the relative positions of $\mathbf{x}_1, \mathbf{x}_2$ and the perturbed point $\mathbf{y}$.

The ratio of the posterior of the $\mathbf{x}_2$ and $\mathbf{x}_1$ — $\frac{p_r(\mathbf{x}_2|\mathbf{y})}{p_r(\mathbf{x}_1|\mathbf{y})}$ — is an indicator of different phases of field (Xu et al., 2023): point in the nearer field tends to have a smaller ratio. Indeed, the ratio would gradually decay from 1 to 0 when moving from the

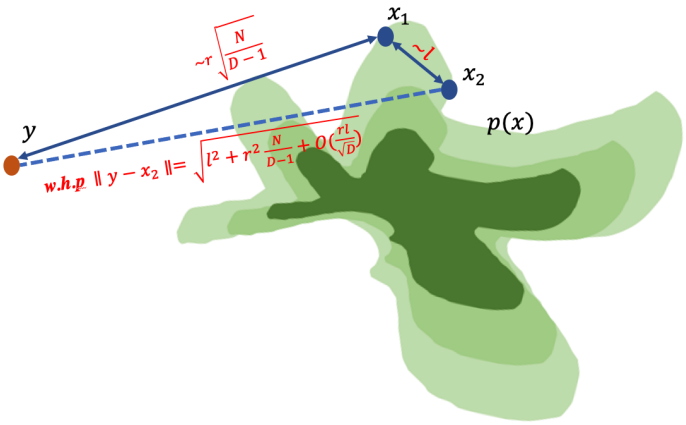


Figure 6. Illustration of the phase alignment analysis

far to the near field. We can calculate the ratio of the coefficients after approximating the distance $\|\mathbf{y} - \mathbf{x}_2\|$:

$$
\begin{aligned}
\frac{p_r(\mathbf{x}_2|\mathbf{y})}{p_r(\mathbf{x}_1|\mathbf{y})} = \frac{p_r(\mathbf{y}|\mathbf{x}_2)}{p_r(\mathbf{y}|\mathbf{x}_1)} &= \left(\frac{l^2 + r^2\frac{N}{D-1} + O(\frac{rl}{\sqrt{D}}) + r^2}{r^2\frac{N}{D-1} + r^2}\right)^{\frac{N+D}{2}} \\
&= \left(1 + \frac{l^2 + O(\frac{rl}{\sqrt{D}})}{r^2\frac{N}{D-1} + r^2}\right)^{\frac{N+D}{2}} \\
&= \exp\left(\ln(1 + \frac{l^2 + O(\frac{rl}{\sqrt{D}})}{r^2\frac{N}{D-1} + r^2}) \cdot \frac{N+D}{2}\right) \\
&\approx \exp\left(\frac{l^2 + O(\frac{rl}{\sqrt{D}})}{r^2\frac{N}{D-1} + r^2} \cdot \frac{N+D}{2}\right) \\
&= \exp\left(\frac{l^2 + O(\frac{rl}{\sqrt{D}})}{r^2} \cdot \frac{N+D}{2(N+D-1)} \cdot (D-1)\right) \\
&\approx \exp\left(\frac{l^2 + O(\frac{rl}{\sqrt{D}})}{r^2} \cdot D\right)
\end{aligned} \tag{14}
$$

Hence the relation $r \propto \sqrt{D}$ should hold to keep the ratio invariant of the parameter D. On the other hand, by Theorem 4.1 we know that p_σ is equivalent to $p_{r=\sigma\sqrt{D}}$ when $D \to \infty$. To achieve phase alignment on the dataset, one should roughly set $r = \sigma\sqrt{D}$.

C.2. Practical Hyperparameter Transfer from Diffusion Models

C.2.1. Transfer EDM training and sampling

We list out and compare the EDM training algorithm (Alg 1) and the PFGM++ with transferred hyper-parameters (Alg 2). The major modification is to replace the Gaussian noise $\mathbf{n}_i \sim \mathcal{N}(0, \sigma^2\boldsymbol{I})$ with the additive noise $R_i\mathbf{v}_i \sim \mathcal{U}_\psi(\psi)p_r(R)$, where $r = \sigma\sqrt{D}$. We highlight the major modifications in blue.

We also show the sampling algorithms of EDM (Alg 3) and PFGM++ (Alg 4). Note that we only change the prior sampling process while the for-loop is identical for both algorithms, since EDM (Karras et al., 2022) sets $\sigma = t$, and $\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}r} = \frac{\mathbf{x} - f_\theta(\mathbf{x}, r)}{r} = \frac{\mathbf{x} - f_\theta(\mathbf{x}, r)}{\sigma\sqrt{D}} = \frac{\mathrm{d}\mathbf{x}}{\sqrt{D}\mathrm{d}\sigma} = \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\sigma}\frac{\mathrm{d}\sigma}{\mathrm{d}r} = \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\sigma} = \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t}$. Thus we can use the original samplers of EDM without further modification.

Algorithm 1 EDM training

Sample a batch of data $\{\mathbf{y}_i\}_{i=1}^{\mathcal{B}}$ from $p(\mathbf{y})$
Sample standard deviations $\{\sigma_i\}_{i=1}^{\mathcal{B}}$ from $p(\sigma)$
Sample noise vectors $\{\mathbf{n}_i \sim \mathcal{N}(0, \sigma_i^2 \boldsymbol{I})\}_{i=1}^{\mathcal{B}}$
Get perturbed data $\{\hat{\mathbf{y}}_i = \mathbf{y}_i + \mathbf{n}_i\}_{i=1}^{\mathcal{B}}$
Calculate loss $\ell(\theta) = \sum_{i=1}^{\mathcal{B}} \lambda(\sigma_i) \| f_\theta(\hat{\mathbf{y}}_i, \sigma_i) - \mathbf{y}_i \|_2^2$
Update the network parameter θ via Adam optimizer

Algorithm 2 PFGM++ training with hyperparameter transferred from EDM

Sample a batch of data $\{\mathbf{y}_i\}_{i=1}^{\mathcal{B}}$ from $p(\mathbf{y})$
Sample standard deviations $\{\sigma_i\}_{i=1}^{\mathcal{B}}$ from $p(\sigma)$
Sample r from p_r: $\{r_i = \sigma_i \sqrt{D}\}_{i=1}^{\mathcal{B}}$
Sample radiuses $\{R_i \sim p_{r_i}(R)\}_{i=1}^{\mathcal{B}}$
Sample uniform angles $\{\mathbf{v}_i = \frac{\mathbf{u}_i}{\|\mathbf{u}_i\|_2}\}_{i=1}^{\mathcal{B}}$, with $\mathbf{u}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
Get perturbed data $\{\hat{\mathbf{y}}_i = \mathbf{y}_i + R_i \mathbf{v}_i\}_{i=1}^{\mathcal{B}}$
Calculate loss $\ell(\theta) = \sum_{i=1}^{\mathcal{B}} \lambda(\sigma_i) \| f_\theta(\hat{\mathbf{y}}_i, \sigma_i) - \mathbf{y}_i \|_2^2$
Update the network parameter θ via Adam optimizer

Algorithm 3 EDM sampling (Heun's 2nd order method)

$\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \sigma_{\max}^2 \boldsymbol{I})$
for $i = 0, \ldots, T-1$ **do**
  $\mathbf{d}_i = (\mathbf{x}_i - f_\theta(\mathbf{x}_i, t_i))/t_i$
  $\mathbf{x}_{i+1} = \mathbf{x}_i + (t_{i+1} - t_i)\mathbf{d}_i$
  if $t_{i+1} > 0$ **then**
    $\mathbf{d}'_i = (\mathbf{x}_{i+1} - f_\theta(\mathbf{x}_{i+1}, t_{i+1}))/t_{i+1}$
    $\mathbf{x}_{i+1} = \mathbf{x}_i + (t_{i+1} - t_i)(\frac{1}{2}\mathbf{d}_i + \frac{1}{2}\mathbf{d}'_i)$
  end if
end for

Algorithm 4 PFGM++ training with hyperparameter transferred from EDM

Set $r_{\max} = \sigma_{\max}\sqrt{D}$
Sample radius $R \sim p_{r_{\max}}(R)$ and uniform angle $\mathbf{v} = \frac{\mathbf{u}}{\|\mathbf{u}\|_2}$, with $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
Get initial data $\mathbf{x}_0 = R\mathbf{v}$
for $i = 0, \ldots, T-1$ **do**
  $\mathbf{d}_i = (\mathbf{x}_i - f_\theta(\mathbf{x}_i, t_i))/t_i$
  $\mathbf{x}_{i+1} = \mathbf{x}_i + (t_{i+1} - t_i)\mathbf{d}_i$
  if $t_{i+1} > 0$ **then**
    $\mathbf{d}'_i = (\mathbf{x}_{i+1} - f_\theta(\mathbf{x}_{i+1}, t_{i+1}))/t_{i+1}$
    $\mathbf{x}_{i+1} = \mathbf{x}_i + (t_{i+1} - t_i)(\frac{1}{2}\mathbf{d}_i + \frac{1}{2}\mathbf{d}'_i)$
  end if
end for

C.2.2. Transfer DDPM (continuous) training and sampling

Here we demonstrate the "zero-shot" transfer of hyperparameters from DDPM to PFGM++, using the $r = \sigma\sqrt{D}$ formula. We highlight the modifications in blue. In particular, we list the DDPM training/sampling algorithms (Alg 5/Alg 7), and their counterparts in PFGM++ (Alg 6/Alg 8) for comparions. Let β_T and β_1 be the maximum/minimum values of β in DDPM (Ho et al., 2020). Similar to Song et al. (2021b), we denote $\alpha_t = e^{-\frac{1}{2}t^2(\bar{\beta}_{\max} - \bar{\beta}_{\min}) - t\bar{\beta}_{\min}}$, with $\bar{\beta}_{\max} = \beta_T \cdot T$ and $\bar{\beta}_{\min} = \beta_1 \cdot T$. For example, on CIFAR-10, $\bar{\beta}_{\min} = 1e-1$ and $\bar{\beta}_{\max} = 20$ with $T = 1000$. We would like to note that the t_is in the sampling algorithms (Alg 7 and Alg 8) monotonically decrease from 1 to 0 as i increases.

Algorithm 5 DDPM training

Sample a batch of data $\{\mathbf{y}_i\}_{i=1}^{\mathcal{B}}$ from $p(\mathbf{y})$
Sample time $\{t_i = t'_i/T\}_{i=1}^{\mathcal{B}}$ with $t'_i \sim \mathcal{U}(\{1, \ldots, T\})$
Get perturbed data $\{\hat{\mathbf{y}}_i = \sqrt{\alpha_{t_i}}\mathbf{y}_i + \sqrt{1-\alpha_{t_i}}\boldsymbol{\epsilon}_i\}_{i=1}^{\mathcal{B}}$, where $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
Calculate loss $\ell(\theta) = \sum_{i=1}^{\mathcal{B}} \lambda(t_i) \| f_\theta(\hat{\mathbf{y}}_i, t_i) - \boldsymbol{\epsilon}_i \|_2^2$
Update the network parameter θ via Adam optimizer

Algorithm 6 PFGM++ training with hyperparameter transferred from DDPM

Sample a batch of data $\{\mathbf{y}_i\}_{i=1}^{\mathcal{B}}$ from $p(\mathbf{y})$
Sample time $\{t_i\}_{i=1}^{\mathcal{B}}$ from $\mathcal{U}[0, 1]$
Get σ_i from t_i: $\{\sigma_i = \sqrt{\frac{1-\alpha_{t_i}}{\alpha_{t_i}}}\}$
Sample r from p_r: $\{r_i = \sigma_i\sqrt{D}\}_{i=1}^{\mathcal{B}}$
Sample radiuses $\{R_i \sim p_{r_i}(R)\}_{i=1}^{\mathcal{B}}$
Sample uniform angles $\{\mathbf{v}_i = \frac{\mathbf{u}_i}{\|\mathbf{u}_i\|_2}\}_{i=1}^{\mathcal{B}}$, with $\mathbf{u}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
Get perturbed data $\{\hat{\mathbf{y}}_i = \sqrt{\alpha_{t_i}}(\mathbf{y}_i + R_i\mathbf{v}_i)\}_{i=1}^{\mathcal{B}}$
Calculate loss $\ell(\theta) = \sum_{i=1}^{\mathcal{B}} \lambda(t_i) \| f_\theta(\hat{\mathbf{y}}_i, t_i) - \frac{\sqrt{D}R_i\mathbf{v}_i}{r} \|_2^2$
Update the network parameter θ via Adam optimizer

Algorithm 7 DDIM sampling

$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
for $i = T, \dots, 1$ **do**
  $\mathbf{x}_{i-1} = \sqrt{\frac{\alpha_{t_{i-1}}}{\alpha_{t_i}}}\mathbf{x}_i + (\sqrt{1-\alpha_{t_{i-1}}} - \sqrt{\frac{\alpha_{t_{i-1}}}{\alpha_{t_i}}}\sqrt{1-\alpha_{t_i}}) f_\theta(\mathbf{x}_i, t_i)$
end for

Algorithm 8 PFGM++ sampling transferred from DDIM

Set $\sigma_{\text{max}} = \sqrt{\frac{1-\alpha_1}{\alpha_1}}, r_{\text{max}} = \sigma_{\text{max}}\sqrt{D}$
Sample radius $R \sim p_{r_{\text{max}}}(R)$ and uniform angle $\mathbf{v} = \frac{\mathbf{u}}{\|\mathbf{u}\|_2}$, with $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
Get initial data $\mathbf{x}_T = \sqrt{\alpha_1} R\mathbf{v}$
for $i = T, \dots, 1$ **do**
  $\mathbf{x}_{i-1} = \sqrt{\frac{\alpha_{t_{i-1}}}{\alpha_{t_i}}}\mathbf{x}_i + (\sqrt{1-\alpha_{t_{i-1}}} - \sqrt{\frac{\alpha_{t_{i-1}}}{\alpha_{t_i}}}\sqrt{1-\alpha_{t_i}}) f_\theta(\mathbf{x}_i, t_i)$
end for

D. Experimental Details

We show the experimental setups in section 5, as well as the training, sampling, and evaluation details for PFGM++. All the experiments are run on four NVIDIA A100 GPUs or eight NVIDIA V100 GPUs.

D.1. Experiments for the Analysis in Sec 5

In the experiments of section 4 and section 5.1, we need to access the posterior $p_{0|r}(\mathbf{y}|\mathbf{x}) \propto p_r(\mathbf{x}|\mathbf{y})p(\mathbf{y})$ to calculate the mean TVD. We sample a large batch $\{\mathbf{y}_i\}_{i=1}^n$ with $n = 1024$ on CIFAR-10 to empirically approximate the posterior:

$$p_{0|r}(\mathbf{y}_i|\mathbf{x}) = \frac{p_r(\mathbf{x}|\mathbf{y}_i)p(\mathbf{y}_i)}{p_r(\mathbf{x})} \approx \frac{p_r(\mathbf{x}|\mathbf{y}_i)}{\sum_{j=1}^n p_r(\mathbf{x}|\mathbf{y}_j)} = \frac{1/(\|\mathbf{x}-\mathbf{y}_i\|_2^2 + r^2)^{\frac{N+D}{2}}}{\sum_{j=1}^n 1/(\|\mathbf{x}-\mathbf{y}_j\|_2^2 + r^2)^{\frac{N+D}{2}}}$$

We sample a large batch of 256 to approximate all the expectations in section 5, such as the average TVDs.

D.2. Training Details

We borrow the architectures, preconditioning techniques, optimizers, exponential moving average (EMA) schedule, and hyper-parameters from previous state-of-the-art diffusion model EDM (Karras et al., 2022). We apply the alignment method in section 4 to transfer their well-tuned hyper-parameters.

For architecture, we use the improved NCSN++ (Karras et al., 2022) for the CIFAR-10 dataset (batch size 512), and the improved DDPM++ for the FFHQ dataset (batch size 256). Since (Karras et al., 2022) does not experiment on LSUN Churches dataset, we set the number of blocks to 2, and the feature maps ($\times \frac{1}{128}$) to 1-1-2-2-2-2-2 without augmentation, inspired by the architecture in (Song et al., 2021b). For optimizers, following EDM, we adopt the Adam optimizer with a learning rate of $10e-4$. We further incorporate the EMA schedule, learning rate warm-up, and data augmentations in EDM. Please refer to Appendix F in EDM paper (Karras et al., 2022) for details.

The most prominent improvements in EDM are the preconditioning and the new training distribution for σ, *i.e.*, $p(\sigma)$. Specifically, adding these two techniques to the vanilla diffusion objective (Eq. (8)), their effective training objective can be written as:

$$\mathbf{E}_{\sigma\sim p(\sigma)}\lambda(\sigma)c_{\text{out}}(\sigma)^2\mathbf{E}_{p(\mathbf{y})}\mathbf{E}_{p_\sigma(\mathbf{x}|\mathbf{y})}\left[\left\|F_\theta(c_{\text{in}}(\sigma)\cdot\mathbf{x}, c_{\text{noise}}(\sigma)) - \frac{1}{c_{\text{out}}(\sigma)}(\mathbf{y} - c_{\text{skip}}(\sigma)\cdot\mathbf{x})\right\|_2^2\right] \tag{15}$$

with the predicted normalized score function in the vanilla diffusion objective (Eq. (8)) re-parameterized as

$$f_\theta(\mathbf{x}, \sigma) = \frac{c_{\text{skip}}(\sigma)\mathbf{x} + c_{\text{out}}(\sigma)F_\theta(c_{\text{in}}(\sigma)\mathbf{x}, c_{\text{noise}}(\sigma)) - x}{\sigma} \approx \sigma\nabla_{\mathbf{x}}\log p_\sigma(x)$$

$c_{\text{in}}(\sigma) = 1/\sqrt{\sigma^2+\sigma_{\text{data}}^2}, c_{\text{out}}(\sigma) = \sigma\cdot\sigma_{\text{data}}/\sqrt{\sigma^2+\sigma_{\text{data}}^2}, c_{\text{skip}}(\sigma) = \sigma_{\text{data}}^2/(\sigma^2+\sigma_{\text{data}}^2), c_{\text{noise}}(\sigma) = \frac{1}{4}\ln(\sigma)$, with $\sigma_{\text{data}} = 0.5$. $\{c_{\text{in}}(\sigma), c_{\text{out}}(\sigma), c_{\text{skip}}(\sigma), c_{\text{data}}, c_{\text{noise}}(\sigma)\}$ are all the hyper-parameters in the preconditioning. The training distribution $p(\sigma)$ is the log-normal distribution with $\ln(\sigma) \sim \mathcal{N}(-1.2, 1.2^2)$, and the loss weighting $\lambda(\sigma) = 1/c_{\text{out}}(\sigma)^2$.

Recall that the hyper-parameter alignment rule $r = \sigma\sqrt{D}$ can transfer the hyper-parameter from diffusion models ($D\to\infty$) to finite Ds. Hence we can directly set $\sigma = r/\sqrt{D}$ in those hyper-parameters for preconditioning. In addition, the training distribution $p(r)$ can be derived via the change-of-variable formula, *i.e.*, $p(r) = p(\sigma = r/\sqrt{D})/\sqrt{D}$. The final PFGM++ objective after incorporating these techniques into Eq. (6) is:

$$\mathbf{E}_{r\sim p(r)}\lambda(r/\sqrt{D})c_{\text{out}}(r/\sqrt{D})^2\mathbf{E}_{p(\mathbf{y})}\mathbf{E}_{p_r(\mathbf{x}|\mathbf{y})}\left[\left\|F_\theta(c_{\text{in}}(r/\sqrt{D})\cdot\mathbf{x}, c_{\text{noise}}(r/\sqrt{D})) - \frac{1}{c_{\text{out}}(\sigma)}(\mathbf{y} - c_{\text{skip}}(r/\sqrt{D})\cdot\mathbf{x})\right\|_2^2\right]$$

with the predicted normalized electric field in the vanilla PFGM++ objective (Eq. (6)) re-parameterized as

$$f_\theta(\tilde{\mathbf{x}}) = \frac{c_{\text{skip}}(r/\sqrt{D})\mathbf{x} + c_{\text{out}}(r/\sqrt{D})F_\theta(c_{\text{in}}(r/\sqrt{D})\mathbf{x}, c_{\text{noise}}(r/\sqrt{D})) - x}{r/\sqrt{D}} \approx \sqrt{D}\frac{\mathbf{E}(\tilde{\mathbf{x}})_{\mathbf{x}}}{E(\tilde{\mathbf{x}})_r}$$

D.3. Sampling Details

For sampling, following EDM (Karras et al., 2022), we also use Heun's 2nd method (improved Euler method) (Ascher & Petzold, 1998) as the ODE solver for $\mathrm{d}\mathbf{x}/\mathrm{d}r = \mathbf{E}(\tilde{\mathbf{x}})_{\mathbf{x}}/E(\tilde{\mathbf{x}})_r = f_\theta(\tilde{\mathbf{x}})/\sqrt{D}$.

We adopt the same parameterized scheme in EDM to determine the evaluation points during N-step ODE sampling:

$$r_i = (r_{\text{max}}{}^{\frac{1}{\rho}} + \frac{i}{N-1}(r_{\text{min}}{}^{\frac{1}{\rho}} - r_{\text{max}}{}^{\frac{1}{\rho}}))^\rho \quad \text{and} \quad r_N = 0$$

where ρ controls the relative density of evaluation points in the near field. We set $\rho = 7$ as in EDM, and $r_{\text{max}} = \sigma_{\text{max}}\sqrt{D} = 80\sqrt{D}, r_{\text{min}} = \sigma_{\text{min}}\sqrt{D} = 0.002\sqrt{D}$ ($\sigma_{\text{max}}, \sigma_{\text{min}}$ are the hyper-parameters in EDM, controlling the starting/terminal evaluation points) following the $r = \sigma\sqrt{D}$ alignment rule.

D.4. Evaluation Details

For the evaluation, we compute the Fréchet distance between 50000 generated samples and the pre-computed statistics of CIFAR-10 and FFHQ. On CIFAR-10, we follow the evaluation protocol in EDM (Karras et al., 2022), which repeats the generation three times with different seeds for each checkpoint and reports the minimum FID score. However, we observe that the FID score has a large fluctuation across checkpoints, and the minimum FID score of EDM in our re-run experiment does not align with the original results reported in (Karras et al., 2022). Fig. 7(a) shows that the FID score could have a variation of ± 0.2 during the training of a total of 200 million images (Karras et al., 2022). To better evaluate the model performance, Table 2 reports the average FID over the Top-3 checkpoints instead. In Fig. 7(b), we further demonstrate the moving average of the FID score with a window of 10000K images. It shows that $D = 2048$ consistently outperforms other baselines in the same training iterations, in agreement with the results in Table 2.

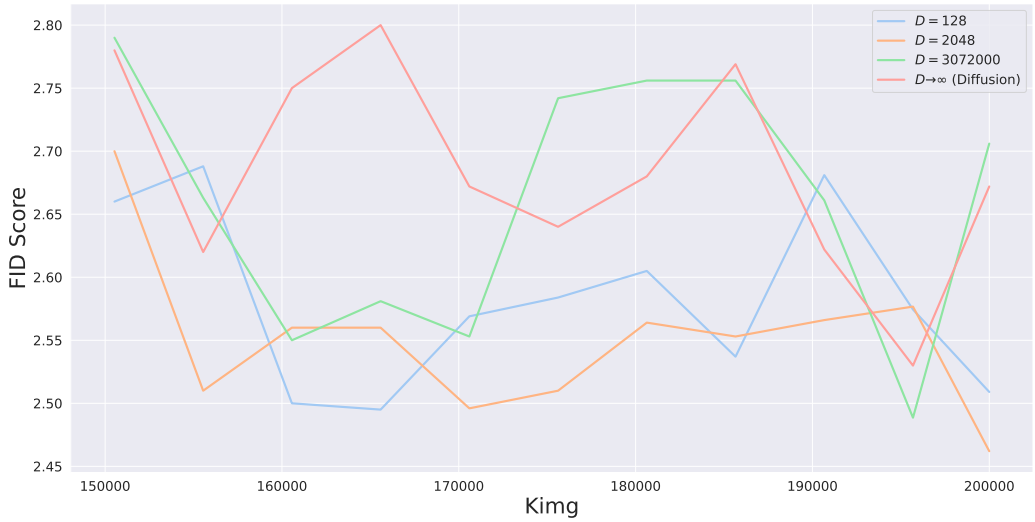
We further report the variation of FID scores in Table 5 for the best checkpoint across different D values, by repeating the sampling process three times using different seeds. We observe that the standard deviation of FID is approximately in the range of $0.5\% \sim 1\%$ of the average FID, which is much smaller than the performance gain of $D = 128/2048$ in terms of Min or Average FID. Additionally, in Table 1 and Table 2 in the main text, we can see that the median $D = 128/2048$ consistently improves over the baseline ($D = \infty$) when using the Top-3 Average FID of checkpoints as a metric.

Table 5. Min, Average and standard deviation of FID on CIFAR-10 using three different sets of random seeds for sampling

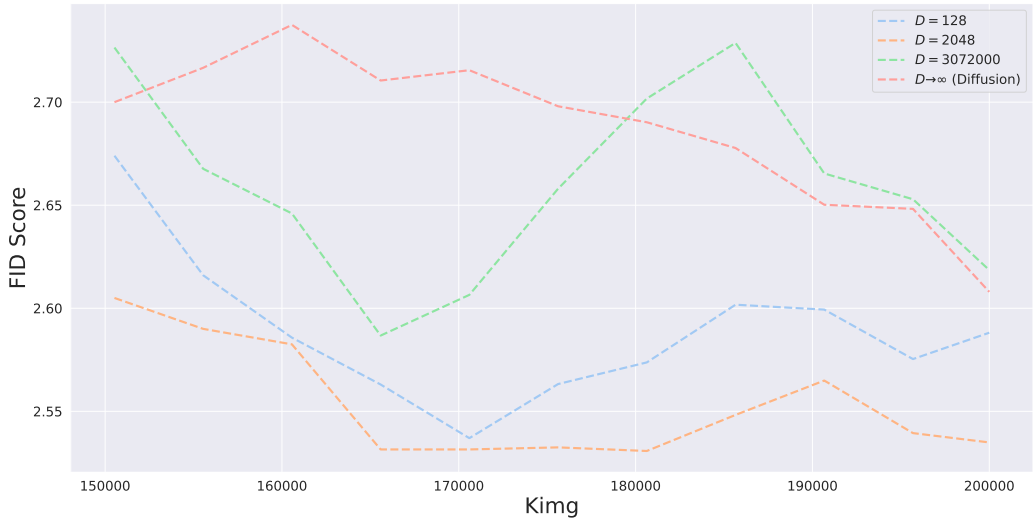
	Min FID ↓	Average FID ↓	Standard deviation
$D = 2048$	1.92	1.94	0.02
$D = 2048$	1.91	1.92	0.01
$D \to \infty$ (Karras et al., 2022)	1.98	2.00	0.02

D.5. Experiments for Robustness

Controlled experiments with α In the controlled noise setting, we inject noise into the intermediate point $\mathbf{x}_r$ in each of the 35 ODE steps by $\mathbf{x}_r = \mathbf{x}_r + \alpha\boldsymbol{\epsilon}_r$ where $\boldsymbol{\epsilon}_r \sim \mathcal{N}(\mathbf{0}, r/\sqrt{D}\boldsymbol{I})$. Since p_r has roughly the same phase as $p_{\sigma=r/\sqrt{D}}$ in diffusion models, we pick $r/\sqrt{D}$ standard deviation of $\boldsymbol{\epsilon}_r$ when the intermediate step is r.



(a) w/o moving average



(b) w/ moving average

Figure 7. FID score in the training course when varying D, **(a)** w/o and **(b)** w/ moving average.

Post-training quantization In the post-training quantization experiments on CIFAR-10, we quantize the weights of convolutional layers excluding the 32×32 layers, as we empirically observe that these input/output layers are more critical for sample quality.

E. Extra Experiments

E.1. Stable Target Field

Xu et al. (2023) propose a Stable Target Field objective for training the diffusion models:

$$\nabla_{\mathbf{x}} \log p_\sigma(\mathbf{x}) \approx \mathbf{E}_{\mathbf{y}_1 \sim p_{0|t}(\cdot|\mathbf{x})} \mathbf{E}_{\{\mathbf{y}_i\}_{i=2}^n \sim p^{n-1}} \left[\sum_{k=1}^n \frac{p_{t|0}(\mathbf{x}|\mathbf{y}_k)}{\sum_j p_{t|0}(\mathbf{x}|\mathbf{y}_j)} \nabla_{\mathbf{x}} \log p_{t|0}(\mathbf{x}|\mathbf{y}_k) \right]$$

where they sample a large batch of samples $\{\mathbf{y}_i\}_{i=2}^n$ from the data distribution to approximate the score function at $\mathbf{x}$. They show that the new target can enhance the stability of converged models in different runs/seeds. PFGM++ can be trained in a similar fashion by replacing the target $\frac{\mathbf{x}-\mathbf{y}}{r/\sqrt{D}}$ in perturbation-based objective (Eq. (6)) with

$$\frac{1}{r/\sqrt{D}} \left(\mathbf{x} - \mathbf{E}_{p_{0|r}(\mathbf{y}|\mathbf{x})}[\mathbf{y}] \right) \approx \frac{1}{r/\sqrt{D}} \left(\mathbf{x} - \mathbf{E}_{\mathbf{y}_1 \sim p_{0|r}(\cdot|\mathbf{x})} \mathbf{E}_{\{\mathbf{y}_i\}_{i=2}^n \sim p^{n-1}} \left[\sum_{k=1}^n \frac{1/(\|\mathbf{x}-\mathbf{y}_k\|_2^2 + r^2)^{\frac{N+D}{2}}}{\sum_j 1/(\|\mathbf{x}-\mathbf{y}_j\|_2^2 + r^2)^{\frac{N+D}{2}}} \mathbf{y}_k \right] \right)$$

When $n = 1$, the new target reduces to the original target. Similar to (Xu et al., 2023), one can show that the bias of the new target together with its trace-of-covariance shrinks to zero as we increase the size of the large batch. This new target can alleviate the variations between random seeds. With the new STF-style target, Table 6 shows that when setting $D = 3072000 \gg N = 3072$, the model obtains the same FID score as the diffusion models (EDM (Karras et al., 2022)). It aligns with the theoretical results in Sec 4, which states that PFGM++ recover the diffusion model when $D \to \infty$.

Table 6. FID and NFE on CIFAR-10, using the Stable Target Field (Xu et al., 2023) in training objective.

	FID ↓	NFE ↓
$D = 3072000$	1.90	35
$D \to \infty$ (Karras et al., 2022)	1.90	35

E.2. Extended CIFAR-10 Samples when varying α

To see how the sample quality varies with α, we visualize the generative samples of models trained with $D \in \{64, 128, 2048\}$ and $D \to \infty$. We pick $\alpha \in \{0, 0.1, 0.2\}$. Fig. 8 shows that the smaller Ds produce better samples compared to larger D. Diffusion models ($D \to \infty$) generate noisy images that appear to be out of the data distribution when $\alpha = 0.2$, in contrast to the clean images by $D = 64, 128$.

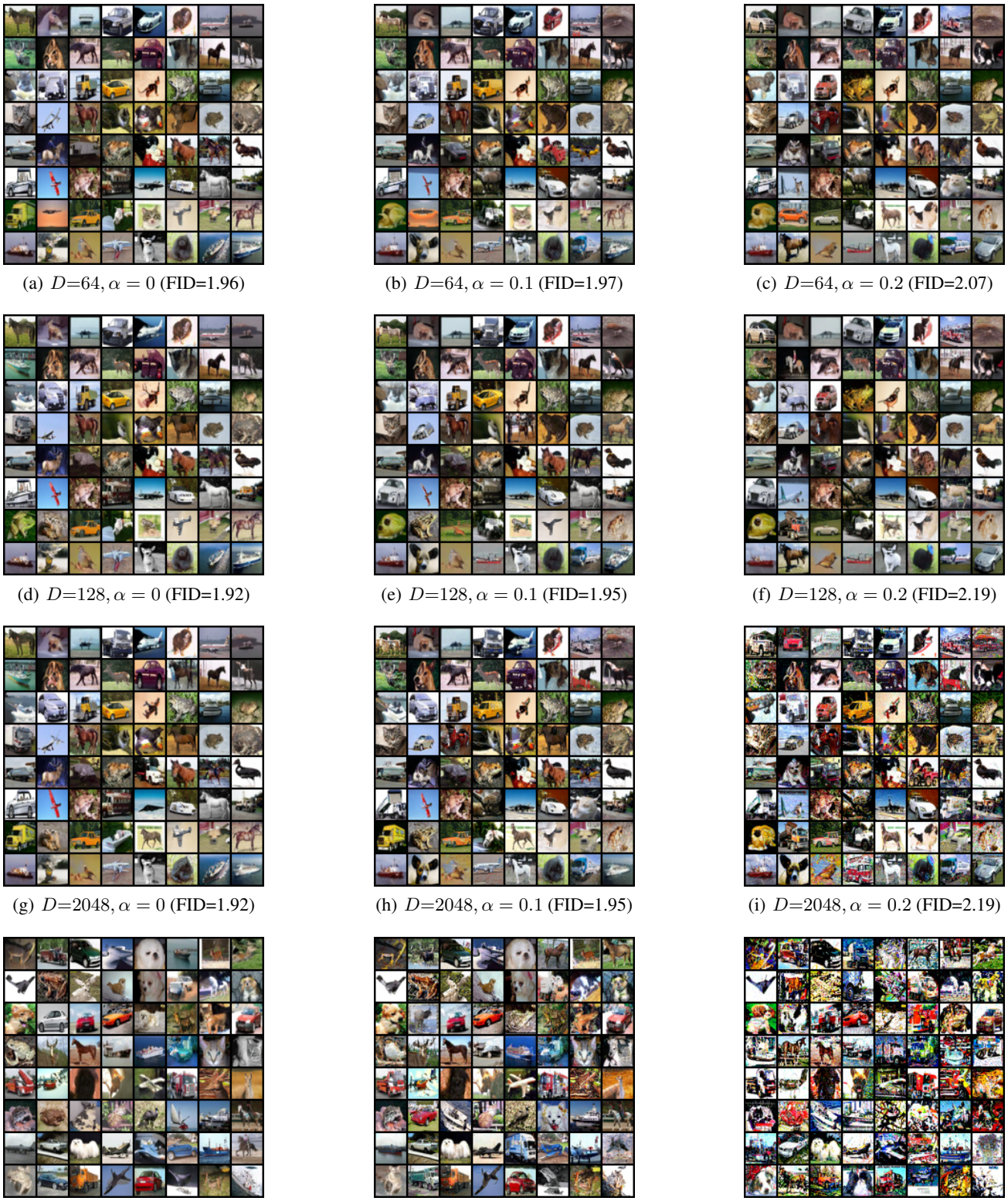
E.3. Extended FFHQ Samples

In Fig. 9, we provide samples generated by the $D = 128$ case and EDM (the $D \to \infty$ case).

F. Toy Dataset

In this section, we construct a 1000-dimensional toy dataset to systematically investigate the behaviors of models with different D values. We synthesize the data in three steps: first we randomly sample the data $\mathbf{y}$ from a 10-dimensional Gaussian mixture $\frac{1}{2}\mathcal{N}(\mathbf{1}, 0.2^2 * \boldsymbol{I}_{10\times10}) + \frac{1}{2}\mathcal{N}(-\mathbf{1}, 0.2^2 * \boldsymbol{I}_{10\times10})$. Next, we map the 10-dimensional data to 1000-dimensional space using a random matrix $W \in \mathbb{R}^{1000\times10}$: $\hat{\mathbf{y}} = W\mathbf{y}$. The entries in W are i.i.d sampled from standard normal distribution. Finally, we perturbed the data with a small Gaussian noise: $\mathbf{x} = \hat{\mathbf{y}} + \epsilon$, where $\epsilon \sim \mathcal{N}(\mathbf{1}, 0.01^2 * \boldsymbol{I}_{1000\times1000})$. The synthetic dataset contains 2000 data points sampled using this procedures.

We design a four-layer UNet architecture, with widths corresponding to *data dimension—latent dimension—latent dimension—latent dimension—data dimension*. The latent dimension directly controls the capacity of the neural network. We also incorporate the residual connection, time-embedding and preconditioning techniques in EDM (Karras et al., 2022).



(a) D=64, $\alpha = 0$ (FID=1.96) (b) D=64, $\alpha = 0.1$ (FID=1.97) (c) D=64, $\alpha = 0.2$ (FID=2.07)

(d) D=128, $\alpha = 0$ (FID=1.92) (e) D=128, $\alpha = 0.1$ (FID=1.95) (f) D=128, $\alpha = 0.2$ (FID=2.19)

(g) D=2048, $\alpha = 0$ (FID=1.92) (h) D=2048, $\alpha = 0.1$ (FID=1.95) (i) D=2048, $\alpha = 0.2$ (FID=2.19)

(j) $D \to \infty, \alpha = 0$ (FID=1.98) (k) $D \to \infty, \alpha = 0.1$ (FID=9.27) (l) $D \to \infty, \alpha = 0.2$ (FID=92.41)

Figure 8. Generated samples on CIFAR-10 with varied hyper-parameter for noise injection (α). Images from top to bottom rows are produced by models trained with $D = 64/128/2048/\infty$. We use the same random seeds for finite Ds during image generation.

(a) $D = 128$ (FID=2.43)

(b) EDM ($D \to \infty$) (FID=2.53)

Figure 9. Generated images on FFHQ 64×64 dataset, by **(left)** $D = 128$ and **(right)** EDM ($D \to \infty$).

We examine the generated samples when varying D and the latent dimension. We visualize the first two coordinates $(\mathbf{x}_0, \mathbf{x}_1)$ of the true data (Fig. 10) and generated data (Fig. 11) for illustration. In Fig. 11, we show that when the latent dimension is set to 4, both the $D = 100$ and $D = \infty$ (diffusion model) fail to recover the data distribution, while model with intermediate $D = 1000$ well captures the underlying data distribution. On weaker architecture (smaller latent dimension), the non-robustness of large D and the non-rigidity of small D would be amplified. It corroborates the arguments that median Ds better balance the robustness and rigidity. As we enlarge the neural network capacity by increasing the latent dimension to 32, all the models with different Ds faithfully recover the data distribution. For quantitative comparison, in Table 7 we report the maximum mean discrepancy between the generated data and the true data for different models. We exclude the $D = 1$ case (PFGM) since the the perturbation kernel is extremely heavy-tailed in 1000-dimensional space, preventing the use of the perturbation-based objective.

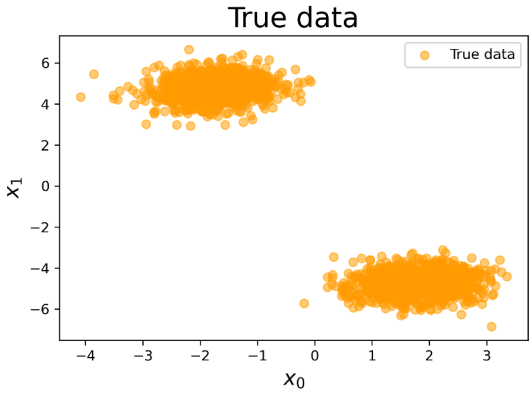
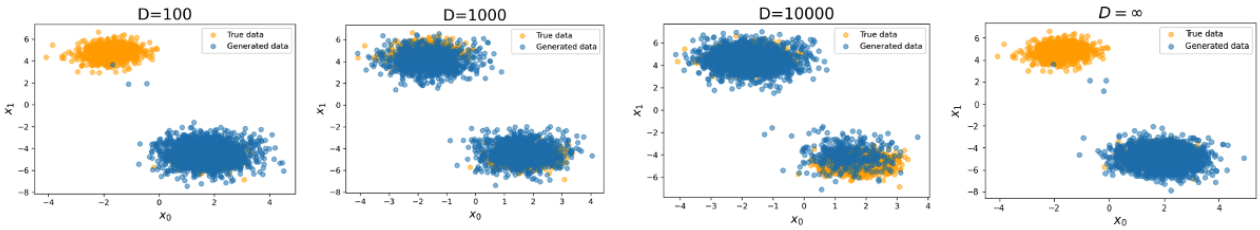
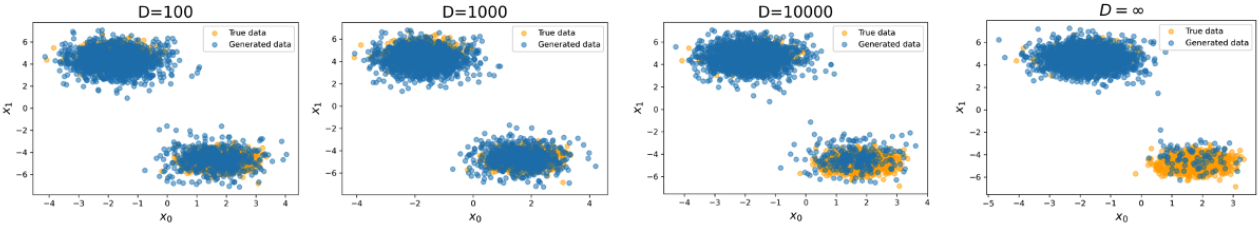


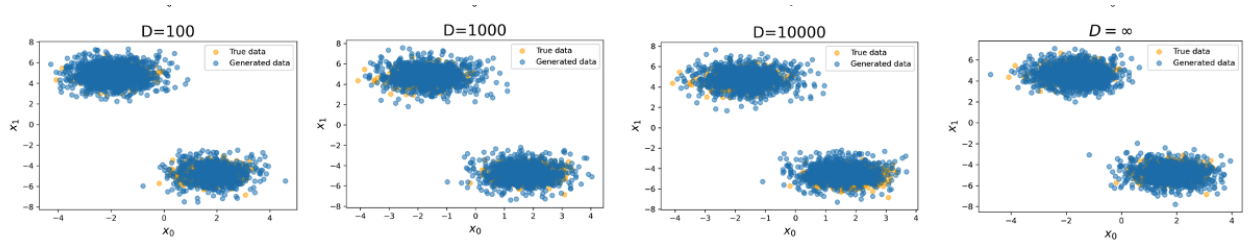
Figure 10. Visualization of the first two coordinates $(\mathbf{x}_0, \mathbf{x}_1)$ for the 1000-dimensional synthetic data.



(a) Latent dimension $= 4$



(b) Latent dimension $= 8$



(c) Latent dimension $= 32$

Figure 11. Visualization of the first two coordinates $(\mathbf{x}_0, \mathbf{x}_1)$ for the generated data (blue) versus true data (orange). From the top row to the bottom row: the latent dimension of the neural network is set to 4 (a), 8 (b), and 32 (c).

Table 7. Maximum mean discrepancy between the generated data and the true data.

	$D = 100$	$D = 1000$	$D = 10000$	$D = \infty$
Latent Dimension $= 4$	1.75	**0.17**	0.79	1.82
Latent Dimension $= 8$	0.33	**0.16**	0.43	1.46
Latent Dimension $= 32$	0.12	**0.01**	0.14	0.07

G. Potential Negative Social Impact

The deep generative model is a burgeoning field and has significant potential for shaping our society. Our work presents a novel family of generative models, the PFGM++, which subsume previous high-performing models and provide greater flexibility. The PFGM++ have many potential applications, particularly in areas that require both robustness and high-quality output. However, it is important to note that the usage of these models can have both positive and negative implications, depending on the specific application. For instance, the PFGM++ can be used to create realistic image and audio samples, but it can also contribute to the development of deepfake technology and potentially lead to social scams. Additionally, the data-collecting process for generative models may infringe upon intellectual property rights. To address these concerns, further research is needed to provide robustness guarantees for generative models and to foster collaborations with experts in socio-technical fields.