DiverseFlow: Sample-Efficient Diverse Mode Coverage in Flows

Mashrur M. Morshed Vishnu Boddeti
Michigan State University
{morshedm, vishnu}@msu.edu

Abstract

Many real-world applications of flow-based generative models desire a diverse set of samples that cover multiple modes of the target distribution. However, the predominant approach for obtaining diverse sets is not sample-efficient, as it involves independently obtaining many samples from the source distribution and mapping them through the flow until the desired mode coverage is achieved. As an alternative to repeated sampling, we introduce DiverseFlow: a training-free approach to improve the diversity of flow models. Our key idea is to employ a determinantal point process to induce a coupling between the samples that drives diversity under a fixed sampling budget. In essence, DiverseFlow allows exploration of more variations in a learned flow model with fewer samples. We demonstrate the efficacy of our method for tasks where sample-efficient diversity is desirable, such as text-guided image generation with polysemous words, inverse problems like large-hole inpainting, and class-conditional image synthesis.

1. Introduction

Consider the task of text-guided image generation from open-ended prompts, like ***"A famous boxer"***. If we use a generative ordinary differential equation (ODE) to obtain a few images for this given prompt, we may observe something unusual: all the resultant images depict *a dog*, as shown in Figure 1 (top-left). Such a result is plausible, as the word "boxer" has multiple meanings: it can either mean a *combat sport athlete* or a particular *dog breed*. However, for the prompt ***"A famous boxer"***, what if the output desired by the user was actually a human athlete, and not a dog? This situation necessitates obtaining additional samples from the model, until the desired alternate meanings are discovered. But instead of repeated sampling, can we directly observe more meanings by finding a more *diverse* set?

Beyond the aforementioned example of text-to-image generation from polysemous[1] prompts, sample diversity is a desirable objective for many other tasks that use generative models. These include inverse problems (e.g., large hole filling) and class-conditioned image generation, to name a few. Diversity or mode coverage is a key pillar in the *generative learning trilemma* [32], in addition to fidelity and latency. For state-of-the-art generative methods such as flow matching models (FM) [20, 22] and diffusion models (DM) [12, 26], significant work has been done on improving the photorealism of samples and the efficiency of the sampling process [11, 16, 20, 28, 30, 34]. However, relatively little attention has been paid to explicitly enhancing the diversity of generated samples under a limited sampling budget.

The standard approach to generate a diverse set of images is to repeatedly obtain independent and identically distributed (IID) samples from a simple or tractable source distribution (e.g., Gaussian distribution), map them to samples

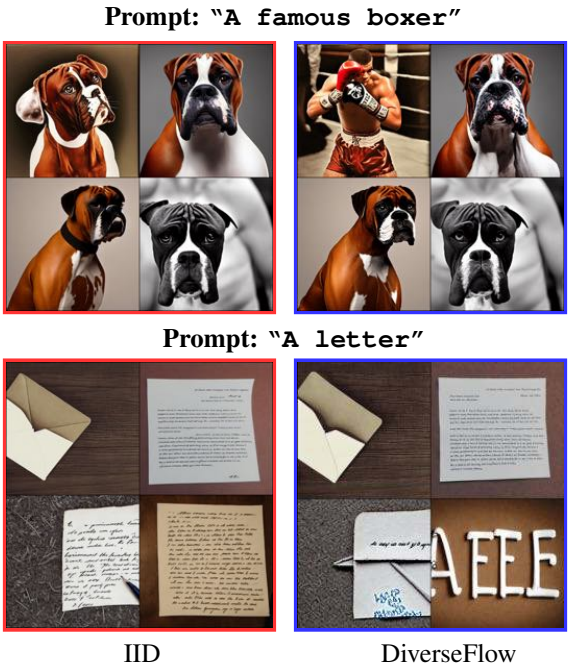


Figure 1. Text-guided image generation with polysemous words. With the same budget of samples, DiverseFlow (right) finds a more diverse set of results compared to IID sampling (left).

[1] Words or phrases with several meanings.

in the target distribution, and continue this process until we observe sufficient mode coverage in the target distribution. This process, while effective, is *sample-inefficient*, requiring the generation of more results than necessary. Importantly, the mapping from the source to the target density does not hold a linear relationship; even specifically selecting diverse samples from the source distribution by design does not necessarily yield diverse samples in the target distribution. These limitations naturally raise the following research question.

> *How can we generate diverse samples from the target density under a limited sampling budget?*

In this paper, we propose DiverseFlow, an inference-time, training-free approach to obtain a diverse set of samples in a desired target density under a fixed sampling budget. We focus on deterministic ODE sampling in continuous-time generative models, specifically FMs, an emerging generative paradigm that enables simulation-free training of continuous normalizing flows (CNFs) and includes diffusion models as a special case.

DiverseFlow measures the diversity of a set of samples through the *volume* they span in the target space. A set of similar samples span a lower volume, while a diverse set naturally spans a larger volume. We impose a volume-based gradient constraint on the flow ODE by drawing on determinantal point processes (DPP) [17, 23], a probabilistic model arising from quantum physics that exactly describes the Pauli exclusion principle: that no two fermions may occupy the same quantum state. In Figure 1 consider the images generated via IID sampling (left column) from a text-to-image generative ODE—for both the prompts ***"A famous boxer"*** and ***"A letter"***, only a single potential meaning is discovered. Unlike the results generated via IID sampling, samples obtained from DiverseFlow (right column) span more diverse modes corresponding to the polysemous words in the prompts. For ***"A famous boxer"***, the samples show both a dog breed and a human athlete; for ***"A letter"***, we observe both *written correspondence* and *alphabet symbols*.

We empirically demonstrate the utility of DiverseFlow across several applications where diversity is inherently desirable. First, we use DiverseFlow to perform **text-guided image synthesis** for words and phrases that may carry a variety of meanings. Second, we perform **large-hole face inpainting** with occlusion masks covering significant regions of the face that may be important to the person's identity. Third, we apply DiverseFlow on **class-conditioned image synthesis** and demonstrate that we can more efficiently explore the data space compared to IID sampling. Lastly, to better characterize and explain the behavior of DiverseFlow, we perform several experiments on synthetic 2D densities.

Summary of Contributions

1. We present a sample-efficient method to obtain a diverse set of samples from a flow ODE in Section 5 and demonstrate it qualitatively in Sections 6.1 to 6.3 and quantitatively in Sections 6.4 and 6.5
2. We provide an empirical analysis that demonstrates that our method is consistent across various flow matching formulations (Section 6.4)
3. We introduce the task of text-to-image synthesis from prompts with polysemous words in the context of analyzing sample diversity, and provide a comparison with relevant methods (Section 6.5)

2. Preliminaries

2.1. Flow Matching

Many generative models can be considered as a *transport map* from some easy-to-sample source distribution to an empirically observed yet unknown target distribution. Recent successes in generative modeling represent this transport map in the form of continuous-time processes, such as stochastic differential equations (SDEs) [12, 29], or ordinary differential equations (ODEs) [2, 20, 22]. Although diffusion models are formulated as SDEs, a significant body of research focuses on converting the diffusion SDE to a deterministic ODE at inference time for faster inference. The diffusion ODE, or probability flow ODE, is a particular case of continuous normalizing flows (CNFs). Flow Matching (FM) [2, 20, 22] is motivated by the idea of directly training CNFs in a scalable and simulation-free manner, just like diffusion models. Moreover, many recent text-to-image generative models, such as Stable Diffusion 3 [10], adopt the FM framework. As such, we present our approach primarily in the context of FM, and our findings can be extended to diffusion and score-based generative models in a straightforward manner.

A CNF reshapes a prior source density p_0 to the empirically observed target density p_1 with an ODE of the form:

$$d\mathbf{x}_t = v_\theta(\mathbf{x}_t, t)dt, \quad \mathbf{x}_0 \sim p_0 \tag{1}$$

where v_θ is a time-dependent velocity field whose parameters θ are learned; we interchangeably use the notation v_t to imply $v_\theta(\cdot, t)$. It becomes possible to obtain samples from p_1 by integrating Equation (1) over time, i.e. by starting at $\mathbf{x}_0 \sim p_0$ for $t = 0$ and solving the ODE till $t = 1$. As our approach is training-free, we do not elaborate on the details of learning to regress the vector field v_t; we encourage interested readers to refer to the works of Lipman et al. [20] and Tong et al. [30] for a primer on training FMs.

At any timestep t during sampling, an intermediate sample $\mathbf{x}_t$ in the flow trajectory can be used to obtain an approx-

imation of the target as follows:

$$\hat{\mathbf{x}}_1 = \mathbf{x}_t + v_\theta(\mathbf{x}_t, t)(1-t) \tag{2}$$

Equation (2) is equivalent to simply taking a large Euler step at any time instance t and is naturally more accurate as t approaches $t = 1$. Further, Equation (2) is also well suited for ODEs with 'straight' paths, where the direction of the time-varying velocity v_t remains near-constant in time (such as the work of Liu et al. [22]). Similarly, we can estimate the source sample by simply taking a step in the reverse direction:

$$\hat{\mathbf{x}}_0 = \mathbf{x}_t - v_\theta(\mathbf{x}_t, t)t \tag{3}$$

2.2. Determinantal Point Processes

Determinantal point processes (DPPs) [4, 17, 23] are probabilistic models of repulsion between points. They were originally termed as *fermion processes* as they describe the Pauli exclusion principle or antibunching effect in fermions. To define a DPP, we must first consider a set of points, $\mathcal{Y}$, and a point process $\mathcal{P}(\mathcal{Y})$—a probability measure on $2^{\mathcal{Y}}$ (the set of all possible subsets of $\mathcal{Y}$). $\mathcal{P}$ is *determinantal* when the probability of choosing a random subset $Y \subset \mathcal{Y}$ according to $\mathcal{P}$ is given by:

$$\mathcal{P}(Y \subset \mathcal{Y}) = \frac{\det(\mathbf{L}_Y)}{\sum_{Y \subset \mathcal{Y}} \det(\mathbf{L}_Y)} = \frac{\det(\mathbf{L}_Y)}{\det(\mathbf{L}+\mathbf{I})} \tag{4}$$

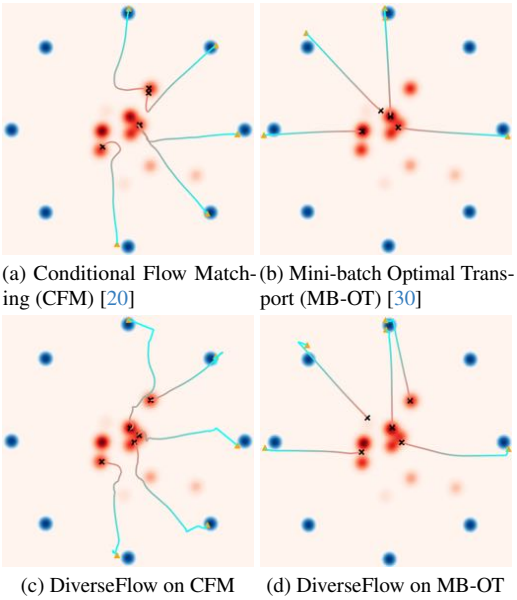
where $\mathbf{L} \in \mathbb{R}^{|\mathcal{Y}| \times |\mathcal{Y}|}$ is a kernel matrix, and $\mathbf{L}_Y$ is the sub-kernel matrix indexed by the elements of Y. Equation (4) has an intuitive geometric interpretation if we consider the kernel $\mathbf{L}$ to be constructed from cosine similarity: the determinant of $\mathbf{L}_Y$ is the Gram-determinant, describing the squared volume of the N-dimensional parallelotope spanned by the set of vectors Y. Thus, a DPP naturally assigns higher probabilities to more orthogonal (and thus diverse) subsets that span larger volumes. We leverage DPPs to define a coupled likelihood measure over a set of samples in a flow trajectory.

3. Related Work

Efficiently finding diverse sets is useful in several application areas of machine learning. For instance, Batra et al. [3] show that the M-Best MAP (maximum a posteriori) solutions in Markov random fields are often distant from the ground truth and highly similar. They thus propose the *Diverse M-Best* problem—finding a set of M highly probable solutions satisfying some minimum dissimilarity threshold—that partly inspires our study in Section 4. [33] utilize DPPs in conjunction with variational autoencoders (VAE) for diverse trajectory forecasting; a set of diverse future pedestrian trajectories improves safety-critical perception systems in autonomous vehicles. Motivated by potential drug discovery and material design applications, Jain et al. [14] propose finding diverse Pareto-optimal candidates in a multi-objective setting with generative flow networks.

The work by Corso et al. [6] which explores diverse non-IID sampling for diffusion models is most similar in spirit to DiverseFlow. However, DiverseFlow is notably different in the following aspects: (1) Our diversity objective is derived from determinantal point processes, a diversity-promoting probability measure of the joint occurrence of a set of samples. Corso et al. [6] is instead inspired by stein variational gradient descent (SVGD) [21]. 2) The diversity measure in DiverseFlow (volume, or determinant of similarity kernel) assigns a zero likelihood to a set if any duplicate elements are present; presence of duplicates is tolerated in the diversity measure in Particle Guidance (row-wise sum of similarity kernel)

4. Diverse Source Samples Do Not Yield Diverse Target Samples



(a) Conditional Flow Matching (CFM) [20] (b) Mini-batch Optimal Transport (MB-OT) [30]

(c) DiverseFlow on CFM (d) DiverseFlow on MB-OT

Figure 2. Finding $K = 5$ samples from the target distribution with $N = 10$ modes. In the example, **only 3 modes** are found by 5 IID samples (a, b). DiverseFlow discovers **5 modes** with the same sampling budget in (d, e).

Problem Setting: We start with a synthetic example to illustrate our problem of interest. Consider that we have empirical observations from a target distribution $\pi_1 \in \mathbb{R}^2$, which is a random mixture of Gaussians, such as the example shown in Figure 4. We design $\pi_1 = \sum_{i=1}^{N} w_i \mathcal{N}(\boldsymbol{\mu}_i, \sigma_i^2 \boldsymbol{I})$ to contain $N = 10$ randomly selected modes $\mathcal{N}(\boldsymbol{\mu}_i, \sigma_i^2 \boldsymbol{I})$, each with a random mixture weight w_i; we observe that in our example, there are 6 high probability modes and 4 low probability ones. Suppose we have a sampling budget

of K samples. This leads to three possible scenarios: (i) $K < N$, (ii) $K = N$, and (iii) $K > N$. Among the aforementioned, case (i) (fewer samples than modes) is the most likely characteristic of any real-world dataset.

Let us have a prior distribution π_0 and some generative model Ψ, such that, in the limit of infinite samples, $\Psi(x_0 \sim \pi_0) \sim \pi_1$. Then, the objective of *sample-efficient diverse sampling* is to obtain samples from $\min(K, N)$ modes from π_1, given a fixed set of K samples in π_0.

If diverse samples are desired from the target density of the flow, one may make the elementary assumption that *if the particles are distant at the source distribution, after being transported by the flow, they remain distant in the target distribution*. This assumption is not necessarily true, as we show in Figure 2. By design, we choose a uniform mixture of eight Gaussians as the source π_0 to obtain diverse source samples. In Figure 2a, we can observe that source points from distinct modes can still converge to the same target mode with IID sampling. Thus, an alternative procedure is necessary to obtain a diverse set from a flow in a sample-efficient manner. We further explore this toy problem in Section 6.4.

5. Diverse Sampling from Flows

From Figure 2, we observe that independently (or heuristically) chosen source samples may not map to a diverse set of target samples. In this case, we can select a new set of source samples and repeat the sampling process till eventually covering at least K modes. However, this approach does not satisfy our fixed sampling budget constraint. An alternative solution to repeated independent sampling is defining and leveraging a diversity measure of the target samples to drive sample diversity. For the set of source samples $\{\mathbf{x}_0^{(1)}, \mathbf{x}_0^{(2)}, \ldots, \mathbf{x}_0^{(k)}\}$, we could optimize a set of perturbations $\{\boldsymbol{\delta}^{(1)}, \boldsymbol{\delta}^{(2)}, \ldots, \boldsymbol{\delta}^{(k)}\}$ such that the new set $\{\mathbf{x}_0^{(1)} + \boldsymbol{\delta}^{(1)}, \mathbf{x}_0^{(2)} + \boldsymbol{\delta}^{(2)}, \ldots, \mathbf{x}_0^{(k)} + \boldsymbol{\delta}^{(k)}\}$ maps to a diverse set of target particles. However, this approach would require multiple simulations of the whole ODE and backpropagating over all the timesteps, which increases the computational complexity of the sampling process over the standard IID sampling.

This leads us to our proposed approach: we avoid multiple simulations and instead optimize the flow trajectory for diversity *while solving the ODE*. For any sample in the flow trajectory $\mathbf{x}_t$, we have an estimate of the target sample $\hat{\mathbf{x}}_1$ through Equation (2). Suppose we have a differentiable objective $\mathcal{L}(\{\hat{\mathbf{x}}_1^{(1)}, \hat{\mathbf{x}}_1^{(2)}, \ldots, \hat{\mathbf{x}}_1^{(k)}\})$ that assigns a likelihood to the joint outcome $\{\hat{\mathbf{x}}_1^{(1)}, \hat{\mathbf{x}}_1^{(2)}, \ldots, \hat{\mathbf{x}}_1^{(k)}\}$. Further, let $\mathcal{L}$ assign a higher likelihood if the joint outcome is a diverse set, and a diminished likelihood if the set is similar. We can then leverage $\mathcal{L}$ to drive diversity among the target samples by modifying the flow velocity of the i-th particle as,

$$\tilde{v}_t^{(i)} = v_t^{(i)} - \gamma(t) \nabla_{\mathbf{x}_t^{(i)}} \log \mathcal{L}(\{\hat{\mathbf{x}}_1^{(1)}, \hat{\mathbf{x}}_1^{(2)}, \ldots, \hat{\mathbf{x}}_1^{(k)}\}) \quad (5)$$

where $\gamma(t)$ is a time-varying scale that controls the strength of the diversity gradient. Setting $\gamma(t) = 0$ reduces to the standard IID sampling scenario, while $\gamma(t) > 0$ will encourage diversity between the generated samples. In practice, $\gamma(t)$ follows the schedule of the probability path normalized by the norm of the DPP gradient.

5.1. Determinantal Gradient Constraints

We desire objective $\mathcal{L}$ in Equation (5) to be higher if the items in the set are diverse and lower if they are similar to each other. We interpret diversity in terms of the *volume* spanned by the set. Consider that we have k samples in $\mathbb{R}^d$ (assume $k < d$). An objective that prefers diversity can be defined as the volume of the k-dimensional parallelotope in $\mathbb{R}^d$ spanned by the sample vectors; this volume becomes diminished when there are similar samples (and even zero, for identical samples). The determinant describes volumes well; a diverse set must span a large volume in the sample space and have a corresponding large determinant.

To define a measure over a set of samples, we draw on the idea of determinantal point processes (DPP). We first define a kernel $\mathbf{L}(\{\hat{\mathbf{x}}_1^{(1)}, \hat{\mathbf{x}}_1^{(2)}, \ldots, \hat{\mathbf{x}}_1^{(k)}\})$ as follows:

$$\mathbf{L}^{(ij)} = \exp\left(-h \frac{\|\hat{\mathbf{x}}_1^{(i)} - \hat{\mathbf{x}}_1^{(j)}\|_2^2}{\text{med}(\mathbf{U}(\mathbf{D}))}\right) \quad (6)$$

where $\mathbf{D}$ denotes a distance matrix with $\mathbf{D}_{ij} = \|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|_2^2$, $\mathbf{U}(\mathbf{D})$ denotes the upper triangle entries of $\mathbf{D}$, h denotes a kernel spread parameter, and $\text{med}(\mathbf{U}(\mathbf{D}))$ denotes the median of those entries. Given $\mathbf{L}$, we may define a DPP-based likelihood as:

$$\begin{aligned} \mathcal{L}(\{\hat{\mathbf{x}}_1^{(1)}, \hat{\mathbf{x}}_1^{(2)}, \ldots, \hat{\mathbf{x}}_1^{(k)}\}) &= \frac{\det(\mathbf{L})}{\det(\mathbf{L} + \mathbf{I})} \\ &= \prod_{a=1}^{k} \frac{\lambda(\mathbf{L})_a}{1 + \lambda(\mathbf{L})_a} \end{aligned} \quad (7)$$

where $\lambda(\mathbf{L})_a$ is the a^{th} eigenvalue of the kernel $\mathbf{L}$. The log-likelihood is then,

$$\mathcal{LL} = \log \mathcal{L} = \log\det(\mathbf{L}) - \log\det(\mathbf{L} + \mathbf{I}) \quad (8)$$

Note that the Euclidean distance $\|\hat{\mathbf{x}}_1^{(i)} - \hat{\mathbf{x}}_1^{(j)}\|_2^2$ is not very meaningful in the high-dimensional raw image space [1]. Therefore, in practice, the distance should be computed in a robust feature space, i.e., $\|F(\hat{\mathbf{x}}_1^{(i)}) - F(\hat{\mathbf{x}}_1^{(j)})\|_2^2$, where F is some domain-specific feature extractor, such as the vision transformer (ViT) [9] for images.

Quality Constraint: The DPP defined in Equation (7) acts as a repulsive force that is unaware of sample quality. A quality term can be incorporated into the DPP kernel to regularize the trajectory diversification. Although flows can be defined between any arbitrary two distributions, let us consider the special case when the source is a Gaussian, i.e., $p_0 \sim \mathcal{N}(0, \mathbf{I})$. Suppose we have a quality vector $\mathbf{q}_t = \{q^{(1)}(t), q^{(2)}(t), \ldots, q^{(k)}(t)\}$, where any $q^{(i)}(t) \in [0, 1]$. We can then define a new kernel $\mathbf{L}_q = \mathbf{L} \odot \mathbf{q}_t \mathbf{q}_t^T$, where each $q^{(i)}(t)$ penalizes a sample $\mathbf{x}_t^{(i)}$ if it deviates too much from the flow. To define this, we obtain an estimate of the source sample $\hat{\mathbf{x}}_0^{(i)}(t)$ for any given sample $\mathbf{x}_t^{(i)}$ via Equation (3), and check if it lies within a desired percentile-radius ρ of the Gaussian p_0. Specifically, we define the time-dependent sample quality as

$$q^{(i)}(t) = \begin{cases} 1 & \text{if } \|\hat{\mathbf{x}}_0^{(i)}(t)\|_2^2 \leq \rho^2 \\ \max\left(\epsilon, e^{-\left(\|\hat{\mathbf{x}}_0^{(i)}(t)\|_2^2 - \rho^2\right)}\right) & \text{otherwise} \end{cases} \tag{9}$$

where ϵ is a 'minimum quality' we assign to prevent a zero determinant.

5.2. Coupled Ordinary Differential Equations

At any timestep t, the measure of diversity in Equation (8) can be adopted to modify the flow of the i-th particle. We compute the gradient of the samples with respect to the diversity measure and use it to modify the ODE as follows:

$$d\mathbf{x}_t^{(i)} = \left[v_\theta(\mathbf{x}_t^{(i)}, t) - \gamma(t) \nabla_{\mathbf{x}_t^{(i)}} \log \mathcal{L}(\{\hat{\mathbf{x}}_1^{(1)}, \ldots, \hat{\mathbf{x}}_1^{(k)}\})\right] dt \tag{10}$$

Where $\gamma(t)$ is a time-varying scaling factor. Unlike the IID sampling scenario where we have K independent ODEs, Equation (10) corresponds to a system of coupled non-linear ordinary differential ordinary equations. To see this, first note that the estimate $\hat{\mathbf{x}}_1^{(i)}$ depends on the current sample $\mathbf{x}_t^{(i)}$ i.e., $\hat{\mathbf{x}}_1^{(i)} = \mathbf{x}_t^{(i)} + v_\theta(\mathbf{x}_t^{(i)}, t)(1-t)$. Second, the DPP log-likelihood $\mathcal{LL}(\{\hat{\mathbf{x}}_1^{(1)}, \hat{\mathbf{x}}_1^{(2)}, \ldots, \hat{\mathbf{x}}_1^{(k)}\})$ induces a time-dependent coupling between the K trajectories of $\mathbf{x}_t^{(i)}, i = 1, \ldots, K$ and seeks to diversify the target samples. Although higher-order ODE solvers [16] can be employed to solve the coupled ODEs, we use the standard Euler method.

6. Experiments

We demonstrate the utility of DiverseFlow in flow-based generative models by considering three applications where sample diversity is naturally desirable: text-guided image generation with polysemous words, large-hole inpainting, and class-conditional image generation. We also analyze the effect of DiverseFlow on different flow matching formulations w.r.t. its ability to span diverse modes through a synthetically constructed 2D density example.

(a) "A buck"

(b) "A famous boxer"

(c) "Van Gogh painting"

Figure 3. For each prompt, the left image (red box) denotes standard IID sampling with classifier-free guidance, while the right image (blue box) shows the result after incorporating DiverseFlow. DiverseFlow finds more diverse sets given the same source points—clearly distinguishable as new semantic meanings in the case of prompts with multiple meanings.

6.1. Image Generation from Polysemous Prompts

In text-to-image generation, the conditional data distribution corresponding to a text prompt may contain many variations, and it is a desirable objective to generate images that span those variations in a sample-efficient manner. As we have previously shown in Figure 1, the result desired by a user (a human boxer athlete) may not be the result generated by the model (a dog breed); if diverse results are obtained, it allows a user to obtain the desired results with fewer attempts.

We pose a scenario where diverse sets are easily observable: when an open-ended text prompt is *polysemous* and carries *multiple meanings*, such as the examples we show in Figure 1 and Figure 3. In Figure 3(a), the prompt "A buck" may commonly refer to a male deer. However, it may also informally refer to a United States dollar. Using the

same four source points, which are deterministically mapped to four deer images by IID sampling, DiverseFlow finds a different set of samples—one that includes a dollar-like coin, albeit embossed with a deer head. We also observe minor differences between the two sets of images, such as changes in pose and background in the top-right and bottom-right deers. For Figure 3(c), while 'Van Gogh painting' is not quite a polysemous word, it can still have two interpretations: a painting *painted by* Van Gogh, or a painting *of* Van Gogh. The regular samples contain minimal diversity, as they include two sets of repeated paintings of Van Gogh. With DiverseFlow, not only can we get a set of four distinct paintings, but we also have a portrait of Van Gogh, which is one of the additional meanings of the prompt. We present additional qualitative samples in the supplementary material.

6.2. Diverse Inpainting on Faces

Another inverse problem where diverse solutions are desirable is face inpainting, where we seek to inpaint the missing parts of the face with diverse plausible facial textures and structures. To demonstrate inpainting with FM models, we first incorporate Manifold Constrained Gradient (MCG) [5] in an off-the-shelf unconditional Rectified-Flow model. In addition to the manifold constraints, we employ determinantal gradient constraints to enhance diversity (further detailed in Algorithm 1). In Figure 5 (b), we observe that the inpainted faces of the four women have similar expressions (largely neutral). DiverseFlow improves the diversity of the set by yielding a highly different expression in the top-right image. In (d) and (e), we also observe changes in facial hair and expressions due to diversification.

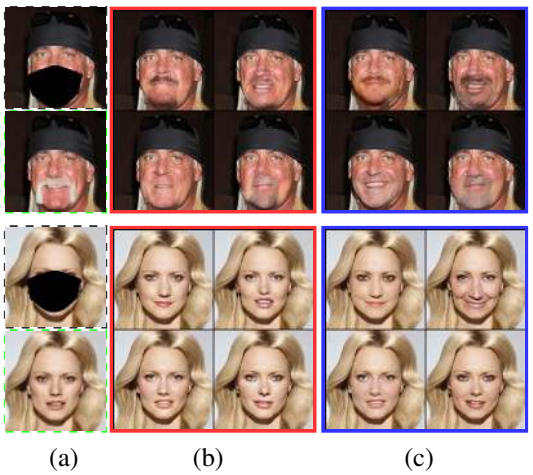


Figure 5. Inpainting on CelebAHQ-256 × 256. (a) Dashed boxes show masked input (top) and ground truth (bottom) respectively. (b) Rectified-Flow [22] + MCG [5] (c) With DiverseFlow.

Figure 6. Class-conditional ImageNet samples from LFM [7]. We show samples for two classes, (top row) 'Mushroom' (class 947) and (bottom row) 'Macaw' (class 88). (a, d) No CFG. (b, e) LFM with CFG. (d, f) LFM with CFG and DiverseFlow.

6.3. Diverse Class-Conditional Image Synthesis

Suppose we can access a class-conditioned flow matching (FM) model trained on an unknown image dataset. To explore the *unobservable* true dataset, we may use a set of class-conditional samples from the FM model. We adopt a latent flow matching (LFM) model [7], trained to generate 256×256 resolution images from the ImageNet [8] dataset. Much like latent diffusion, LFM employs classifier-free guidance to create high-quality samples. However, this naturally poses a cost to diversity, as we show in Figure 6.

By incorporating DiverseFlow, we can maintain the high quality of the samples and simultaneously explore more modes in the dataset. In Figure 6, we demonstrate two ImageNet classes that may have diversity: 'Mushroom' and 'Macaw.' For mushrooms, we observe that LFM primarily generates two species of mushrooms. However, by applying DiverseFlow, we successfully find a new species within our limited set: an *Amanita muscaria*, also known as the *fly agaric*—easily distinguishable by the white spots on its red cap. In another example, we see that while LFM generates four scarlet macaws, using the same source samples, DiverseFlow helps us find a different blue and yellow macaw. We further evaluate the precision and recall [18] for ImageNet-256 synthesis in Figure 7, showing that DiverseFlow improves the recall while retaining similar precision, suggesting improved diversity.

6.4. DiverseFlow Across Various FM Formulations

We ask the question: does DiverseFlow remain consistent across different formulations of flow matching? We adopt the same toy bivariate mixture of Gaussian distributions as in the example shown in Figure 2, across four different FM formulations: (i) Conditional Flow Matching (CFM) [20],

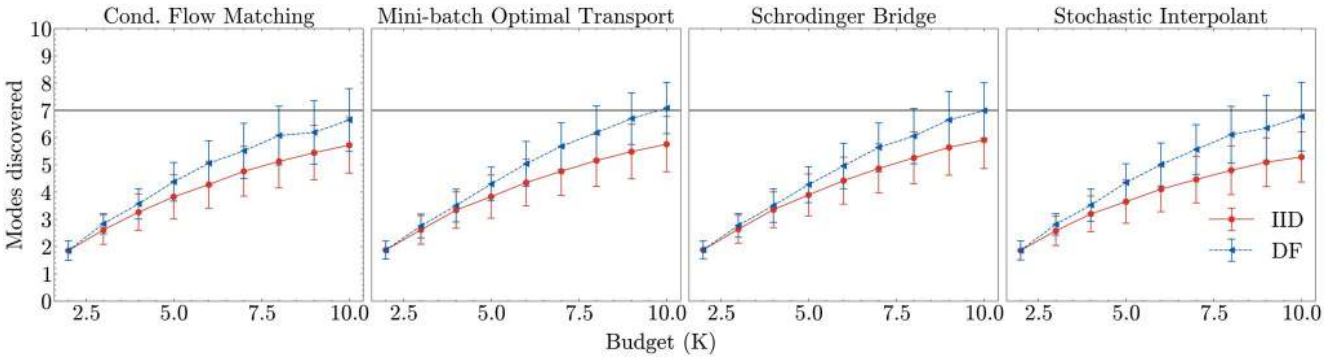


Figure 4. Comparing different FM formulations in terms of the number of modes spanned by IID sampling versus with DiverseFlow. More details about the experiment are provided in the supplementary.

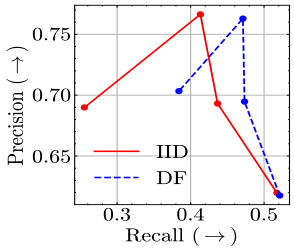


Figure 7. Precision vs. recall on ImageNet-256.

Method	CFG	P (↑)	R (↑)
LFM	1	0.62	0.52
LFM	1.5	0.69	0.44
LFM	2	0.77	0.41
LFM	4	0.69	0.26
LFM + DF	1	0.62	0.52
LFM+DF	1.5	0.69	0.47
LFM+DF	2	0.76	0.46
LFM+DF	4	0.70	0.38

Table 1. Precision (P) and recall (R) score of DiverseFlow and baselines.

(ii) Mini-batch Optimal Transport CFM (MB-OT) [30], (iii) Schrödinger Bridge CFM (SB-CFM) [31], and (iv) Stochastic Interpolants (SI-CFM) [2]. We then perform a numerical experiment to quantify the average number of modes discovered by each FM variant with increasing the sampling budget K. Figure 4 reports the results. For a maximum sampling budget of 10, MB-OT discovers only 5.64 modes on average, which is expected since the dataset contains 6 high-probability modes. By incorporating DiverseFlow, we can find 7.11 modes on average. We observe that MB-OT and SB-CFM benefit most from DiverseFlow. We hypothesize this is because MB-OT and SB-CFM are formulated with the notion of optimal paths, which results in a more accurate estimation of $\hat{x}_1$ and thus a better diversity gradient.

6.5. Comparison To Particle Guidance

(i) Preventing Training Data Copy: Previously, Corso et al. [6] demonstrate that their method can alleviate Stable Diffusion's training data regurgitation problem [27] to some extent. In Figure 8, we demonstrate similar capabilities; in (b), the top left and bottom right examples are copies of the training data. Subsequently, (c) and (d) find a new example for the top-left sample. **(ii) Diversity vs Quality:** We also compare the diversity versus quality of DiverseFlow against Particle Guidance in Figure 10 over 30 polysemous prompts repeated over 10 random seeds. While varying the strength of DiverseFlow to improve diversity, we also vary the values of the classifier-free guidance strength from 7.5 to 10 to boost quality. Quality is measured by *Aesthetic Score* (higher is better) [25], and diversity is measured by *average pairwise similarity* of a set (lower is better) [6]. We observe that though DiverseFlow obtains better diversity at similar quality to IID sampling, the aesthetic score of Particle Guidance is unusually low. In Figure 9 we highlight the cause: for the challenging task of polysemous text-to-image generation, the images generated from Particle Guidance (a) suffers

(a) Training Data (b) IID Sampling

(c) Particle Guidance [6] (d) DiverseFlow

Figure 8. For the prompt "`VAN GOGH CAFE TERASSE copy.jpg`", we show (a) the training data, (b) IID sampling with two copies in top-left and bottom-right, (c) Particle Guidance [6], and (d) DiverseFlow.

Figure 9. Particle Guidance (middle rows) can suffer from artefacts for polysemous prompts. DiverseFlow retains quality while achieving diversity: for "A mint", a minted coin with a mint leaf was discovered.

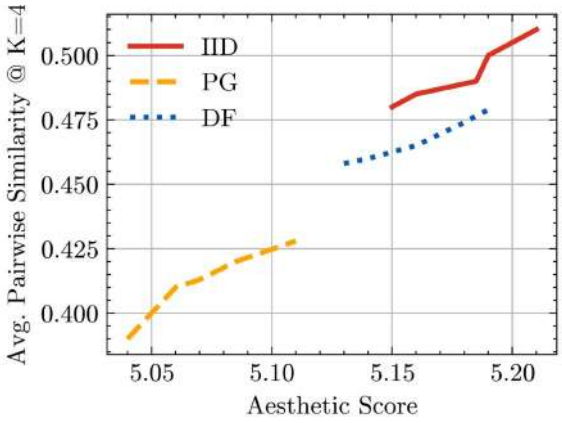


Figure 10. In-batch similarity (↓) vs aesthetic score (↑)

from image artefacts, which achieves a high diversity score but a poor quality score. Additional details about the experiment, including the set of 30 prompts, are provided in the

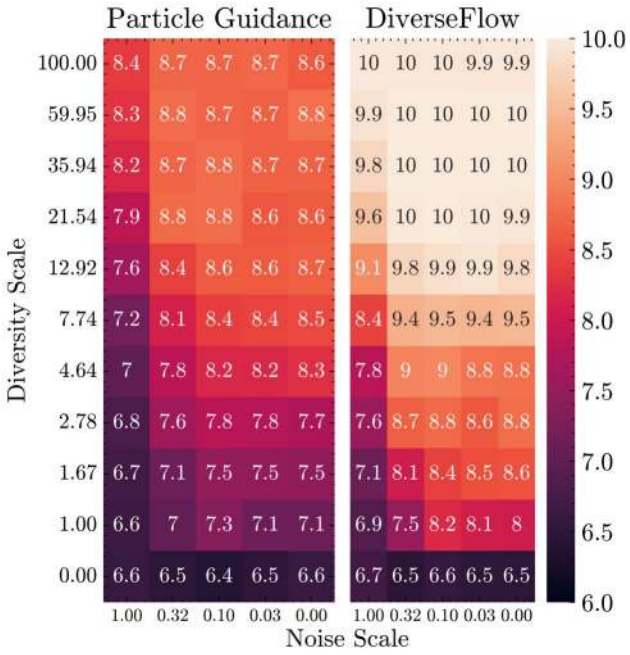


Figure 11. Average modes discovered over 100 trials, for varying noise scales and strengths of DiverseFlow and ParticleGuidance.

supplementary. **(iii) Mode Discovery:** In high dimensional data, the number of modes, N, is significantly greater than the budget K, so finding unique modes is still highly probable. To better highlight the differences between DiverseFlow and Particle Guidance, we adopt a simpler toy experiment: finding modes in a symmetric *uniform* Mixture of Gaussian distribution. Corso et al. [6] provides the result that IID sampling with budget $K = 10$ discovers **about 6.5 modes** on average, while Particle Guidance with a Euclidean kernel discovers **almost 9 out of 10 modes**. We verify this result in Figure 11, finding that Particle Guidance discovers up to **8.8 modes** (averaged over 100 trials). However, by using DiverseFlow, it is possible to discover **all 10 modes** on average, showing that our approach has a stronger diversification effect.

7. Conclusion

In this paper, we present DiverseFlow, a way to enforce diversity among a limited set of samples generated from a flow by coupling them through a determinantal point process. We demonstrate multiple applications where our method can be useful. In essence, though flow models admit a deterministic mapping from source to sample, DiverseFlow allows us to modify this mapping at inference-time and obtain samples with improved diversity. Diversity enhances the utility of the underlying generative flow in the case of ill-posed problems, where many solutions exist and it is desirable to find a multitude of such solutions.

References

[1] Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. On the surprising behavior of distance metrics in high dimensional space. In *Database Theory—ICDT 2001: 8th International Conference London, UK, January 4–6, 2001 Proceedings 8*, pages 420–434. Springer, 2001. 4

[2] Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023. 2, 7, 4

[3] Dhruv Batra, Payman Yadollahpour, Abner Guzman-Rivera, and Gregory Shakhnarovich. Diverse m-best solutions in markov random fields. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V 12*, pages 1–16. Springer, 2012. 3

[4] Alexei Borodin and Grigori Olshanski. Distributions on partitions, point processes, and the hypergeometric kernel. *Communications in Mathematical Physics*, 211:335–358, 2000. 3

[5] Hyungjin Chung, Byeongsu Sim, Dohoon Ryu, and Jong Chul Ye. Improving diffusion models for inverse problems using manifold constraints. *Advances in Neural Information Processing Systems*, 35:25683–25696, 2022. 6, 1

[6] Gabriele Corso, Yilun Xu, Valentin De Bortoli, Regina Barzilay, and Tommi Jaakkola. Particle guidance: non-iid diverse sampling with diffusion models. *International Conference on Learning Representations*, 2023. 3, 7, 8, 1

[7] Quan Dao, Hao Phung, Binh Nguyen, and Anh Tran. Flow matching in latent space. *arXiv preprint arXiv:2307.08698*, 2023. 6, 3

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 6

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations*, 2021. 4

[10] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning*, pages 12606–12633. PMLR, 2024. 2, 1

[11] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 1

[12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1, 2

[13] Noboru Isobe, Masanori Koyama, Jinzhe Zhang, Kohei Hayashi, and Kenji Fukumizu. Extended flow matching: a method of conditional generation with generalized continuity equation. *arXiv preprint arXiv:2402.18839*, 2024. 5

[14] Moksh Jain, Sharath Chandra Raparthy, Alex Hernández-Garcıa, Jarrid Rector-Brooks, Yoshua Bengio, Santiago Miret, and Emmanuel Bengio. Multi-objective gflownets. In *International Conference on Machine Learning*, pages 14631–14653. PMLR, 2023. 3

[15] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *International Conference on Learning Representations*, 2018. 1

[16] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 2022. 1, 5

[17] Alex Kulesza, Ben Taskar, et al. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286, 2012. 2, 3

[18] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in neural information processing systems*, 32, 2019. 6

[19] Yiheng Li, Heyang Jiang, Akio Kodaira, Masayoshi Tomizuka, Kurt Keutzer, and Chenfeng Xu. Immiscible diffusion: Accelerating diffusion training with noise assignment. *Advances in neural information processing systems*, 2024. 5

[20] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 1, 2, 3, 6

[21] Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. *Advances in neural information processing systems*, 29, 2016. 3

[22] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022. 1, 2, 3, 6

[23] Odile Macchi. The coincidence approach to stochastic point processes. *Advances in Applied Probability*, 7(1):83–122, 1975. 2, 3

[24] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1

[25] Christoph Schuhmann and LAION. Improved aesthetic predictor: Clip+mlp aesthetic score predictor, 2022. https://github.com/christophschuhmann/improved-aesthetic-predictor. 7

[26] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015. 1

[27] Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Diffusion art or digital forgery? investigating data replication in diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6048–6058, 2023. 7

[28] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *Interneational Conference on Learning Representations*, 2021. 1

[29] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *International Conference on Learning Representations*, 2021. 2

[30] Alexander Tong, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Kilian Fatras, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2023. 1, 2, 3, 7, 5

[31] Alexander Y Tong, Nikolay Malkin, Kilian Fatras, Lazar Atanackovic, Yanlei Zhang, Guillaume Huguet, Guy Wolf, and Yoshua Bengio. Simulation-free schrödinger bridges via score and flow matching. In *International Conference on Artificial Intelligence and Statistics*, pages 1279–1287. PMLR, 2024. 7

[32] Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion gans. *International Conference on Learning Representations*, 2022. 1

[33] Ye Yuan and Kris Kitani. Diverse trajectory forecasting with determinantal point processes. *International Conference on Learning Representations*, 2020. 3, 5

[34] Qinqing Zheng, Matt Le, Neta Shaul, Yaron Lipman, Aditya Grover, and Ricky TQ Chen. Guided flows for generative modeling and decision making. *arXiv preprint arXiv:2311.13443*, 2023. 1

[35] Yinglin Zheng, Hao Yang, Ting Zhang, Jianmin Bao, Dongdong Chen, Yangyu Huang, Lu Yuan, Dong Chen, Ming Zeng, and Fang Wen. General facial representation learning in a visual-linguistic manner. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18697–18709, 2022. 2

DiverseFlow: Sample-Efficient Diverse Mode Coverage in Flows

Supplementary Material

In the supplementary, we primarily focus on two aspects: First, we present additional experimental details and ablation of the content in the main body of the paper. Second, we discuss some of the potential limitations of our method, as well as challenges and open questions.

8. Additional Experimental Details

8.1. Polysemous Prompts

Rationale: One question that may arise is why we use polysemous prompts to primarily evaluate the effects of DiverseFlow, instead of some other existing regular text-to-image task. There are two reasons: (i) Diversity is clearly distinguishable both qualitatively and quantitatively for polysemous prompts (ii) Text-to-image generation from polysemous prompts is an inherently challenging task for generative flow ODEs.

In our early experiments, we considered the validation set from the COCO dataset as a way of evaluating diversity in text-to-image generation. However, although we observed an increase in diversity (average pairwise dissimilarity in a set), the difference was difficult to observe visually from images. For instance, in Figure 14, it is difficult to tell if diversity has improved from the original IID sample result. We also find in Figure 13 that DiverseFlow and Particle Guidance achieve similar results, where it is difficult to distinguish between either.

In order to show an impactful example, we pose the task of text-to-image generation from ambiguous prompts that may carry multiple distinct meanings, with the assumption that multiple meanings would correspond to sufficiently disentangled modes in the data. In the case of polysemous prompts, the difference between diverse and non-diverse results is clear to the observer, and is significantly highlighted in the metrics. We also find that for the same guidance strength, Particle Guidance is highly prone to aliasing artifacts (as we show in Figure 9); instead of finding diverse samples, it achieves higher dissimilarity by introducing noise in the image (hence the low similarity and low quality in Figure 10). This suggests that open-ended prompts are inherently more difficult than well-defined and constrained prompts.

Setup: For direct comparison to Particle Guidance [6], we utilize the probability flow ODE formulation of Stable Diffusion v1.5 [24] as our underlying generative flow. We also apply DiverseFlow on the larger Stable Diffusion v3 model [10], which is based on the rectified flow approach of Liu et al. [22]. We show some results for SD-v3 in Table 2, and in Figure 19 and Figure 12.

Prompt Selection: We adopt a set of 30 polysemous prompts, which are given in Table 2. To find such prompts, we prompted an LLM for 50 polysemous nouns, and then we manually filtered 30 good polysemous words with clearly distinct meanings.

Implementation: We use 30 Euler steps to sample from SD-v1.5, and 28 Euler steps for SD-v3, with a classifier-free guidance strength of 8 and 7 respectively, which are the default settings of both models. For the feature extractor, we experiment with both CLIP-ViT-B16 and DINO-ViT-B8, and find better results with DINO. From Table 2, it can be seen that polysemous prompts are a challenging task; for many prompts, it is not yet possible to find the diverse meanings. For example, for "a spring", both SD-v1.5 and SD-v3 only yield images of the season, and not the coiled object. DiverseFlow helps discover 5 and 4 additional meanings for SD-v1.5 and SD-v3 respectively. For the images in Figure 9 and the results in Figure 10, we use a scaling factor of $8\sigma(t)$ for Particle Guidance, same as used by the authors in their paper. For DiverseFlow, we use $\frac{20\sigma(t)}{\|\nabla \log \mathcal{L}(\mathbf{x_t}^{(1)}, \mathbf{x_t}^{(2)}, \ldots, \mathbf{x_t}^{(k)})\|}$.

8.2. Inpainting

Rationale: In masked face datasets, occlusion masks may occur in various areas of the face and in various sizes. In the case of small occlusion masks, or masks on insignificant ares (such as a cheek), there is a minimal scope for generating diverse results. We thus fix a large central mask that approximately covers 50% of the face surface area, consisting primarily of the mouth and nose regions, as shown in Figure 5.

Setup: We sample 500 random images from the CelebA-HQ 256×256 dataset, and apply the same fixed mask to all images. Additionally, we vary how much of the face is occluded by the mask by scaling the size of the mask, to approximately cover 10% to 50% of the visible face. We then measure the average pairwise similarity between the generated faces ($K = 4$ inpainting results per occluded face). In Figure 15, we show that the effect of DiverseFlow is limited for small occlusions, and is distinct for larger occlusions.

Implementation: To implement inpainting with an FM model, we utilize (i) an *unconditional* off-the-shelf face image generating FM, and (ii) a continuous-time ODE inpainting algorithm. We adopt a RectifiedFlow model pre-trained on CelebAHQ-256×256 [15], from https://github.com/gnobitab/RectifiedFlow. Next, we extend the manifold constrained gradient (MCG) algorithm [5] from diffusion models to FM models, in Algo-

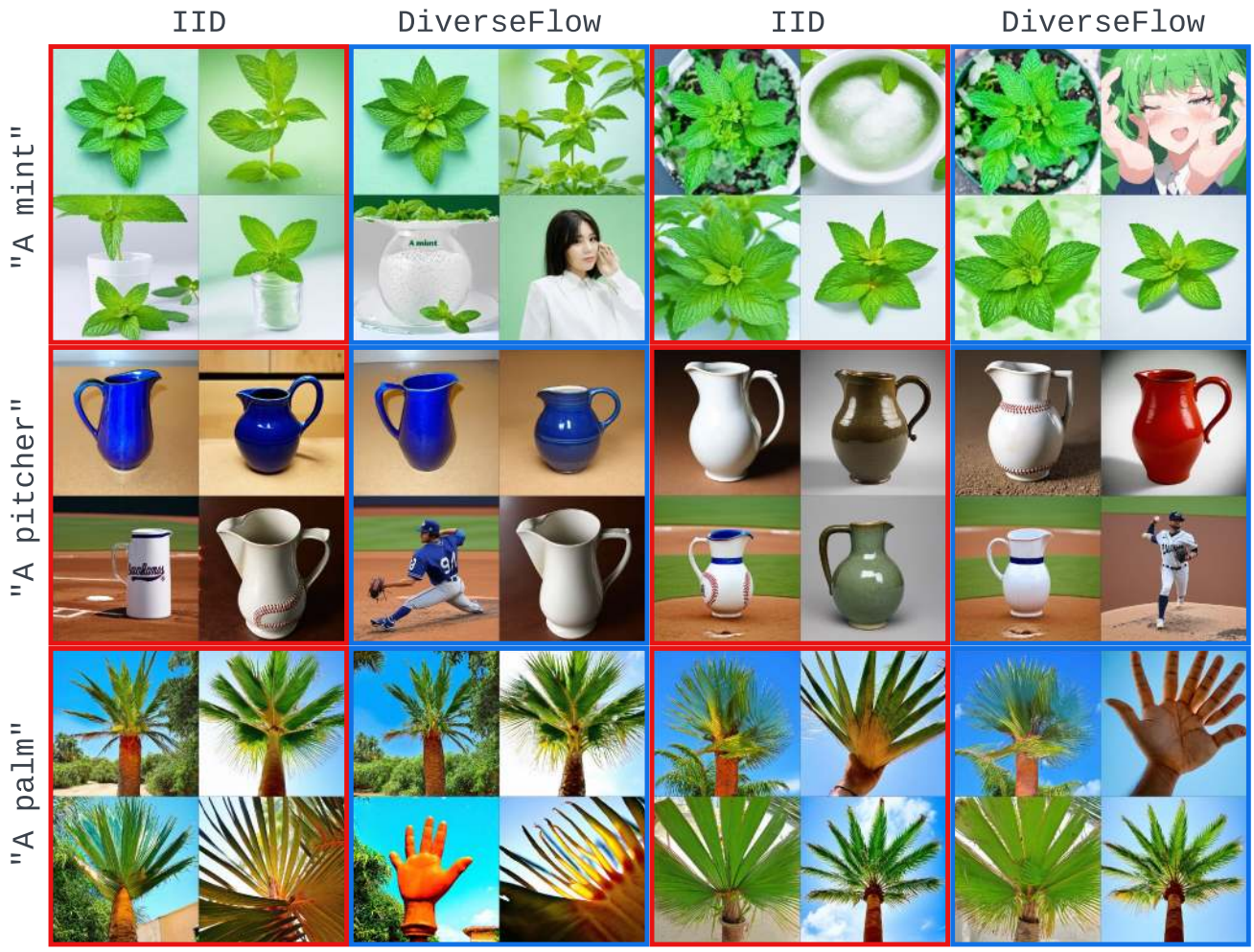


Figure 12. Some examples on SD3 where DiverseFlow discovers alternate meanings that IID sampling doesn't find.

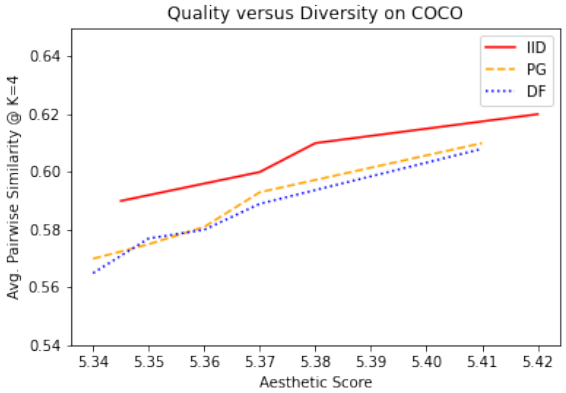


Figure 13. Diversity and Quality on COCO validation set.

rithm 1. We use $\gamma(t) = 10\frac{\sqrt{1-t}}{\|\nabla \log \mathcal{L}\|}$ as a time-varying scale for the DPP gradient. Additionally, we use 200 Euler steps for sampling; more steps are needed in comparison to text-to-image generation for the MCG inpainting algorithm to converge.

Figure 14. "A woman holding a cake is looking at an excited child in a high chair": results from IID sampling, Particle Guidance, and DiverseFlow respectively.

For the feature encoder F, we use the FaRL model [35], which is a CLIP-like model trained on LAIONFace [35]. FaRL is trained in a mask-aware manner, and we downsam-

polysemous word	SD-v1.5	SD-v1.5+DF	SD-v3	SD-v3 + DF
boxer	✓	✓	✗	✓
crane	✓	✓	✓	✓
bat	✗	✗	✗	✗
letter	✓	✓	✓	✓
buck	✓	✓	✗	✗
seal	✓	✓	✗	✗
mouse	✗	✗	✗	✗
horn	✓	✓	✓	✓
chest	✗	✗	✗	✗
nail	✓	✓	✓	✓
ruler	✗	✓	✗	✓
ball	✗	✗	✗	✗
file	✓	✓	✓	✓
ring	✗	✗	✗	✗
deck	✗	✗	✗	✗
nut	✗	✗	✗	✗
bolt	✗	✓	✓	✓
bow	✗	✗	✗	✗
pupil	✗	✗	✗	✗
palm	✗	✓	✓	✓
pitcher	✗	✗	✓	✓
fan	✗	✓	✗	✗
club	✓	✓	✓	✓
anchor	✗	✗	✗	✗
mint	✓	✓	✗	✓
iron	✗	✓	✗	✓
bank	✗	✗	✗	✗
glass	✗	✗	✗	✗
pen	✗	✗	✗	✗
spring	✗	✗	✗	✗
total	10	15	9	13

Table 2. List of polysemous prompts and possible discovered diverse meanings over 100 samples.

ple the inpainting mask to additionally create an attention mask, to ensure that the feature encoder F does not focus on the irrelevant areas.

8.3. Class-Conditioned Image Generation

Rationale: Many ImageNet categories involve animal species that exhibit keen biodiversity. However, to observe the variation between species or animal families, we need to ensure diverse results. However, regular IID sampling can often be very strongly biased towards the dominant mode or variation (for instance, the scarlet Macaw in Figure 6, or the coral-shade starfish in Figure 16). By improving the diversity of the generative model, we can easily discover more varieties with fewer number of samples.

Setup: We only show a few qualitative samples for class-label to image generation from ImageNet. In particular, we pick the classes 'Macaw', 'Mushroom', and 'Starfish' as they are prominently demonstrated as examples in the project page of the underlying flow model (https://vinairesearch.github.io/LFM).

Implementation: For the ImageNet samples, we show in Figure 6, we use pre-trained LFM models [7], specifically the 'imnet_f8_ditb2' weights from https://vinairesearch.github.io/LFM. We primarily used DINO-ViT-B8 as the feature extractor F.

8.4. Mode Finding

We train a set of four identical models from scratch for the four FM variants used in Figure 4. Each model is an *unconditional* generative model and is defined as an MLP consisting of 4 fully connected layers, each except the first having 256 hidden units; the first layer has a hidden size of 256 + 1 to account for the time input. We use the `torchcfm` library (https://github.com/atong01/conditional-

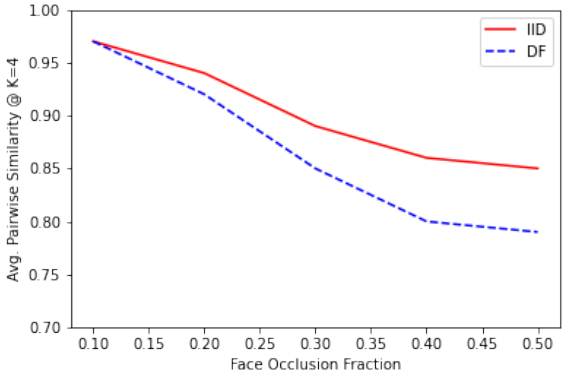


Figure 15. Similarity between faces decreases with increasing occlusion mask size. DiverseFlow finds more dissimilar faces with larger occlusions, but has little effect on small occlusions.

Algorithm 1 MCG Flow Inpainting with Euler Method

Require: Inpainting input $\mathbf{Y} \in \mathbb{R}^{H\times W\times 3}$, inpainting mask $\mathbf{M} \in \mathbb{Z}_2^{H\times W\times 3}$, number of sampling steps N, time-varying velocity field v_θ

$\mathbf{X}_0 \sim \mathcal{N}(0, \mathbf{I})$ ▷ Sample a particle from source distribution $\mathbf{Z}_0$

for i=$0 \dots N-1$ **do**

$t_i, t_{i+1} \leftarrow \frac{i}{N}, \frac{i+1}{N}$ ▷ Uniform timesteps, $t \in 0 \dots 1$

$\Delta_t \leftarrow t_{i+1} - t_i$

$\mathbf{V}_i \leftarrow v_\theta(\mathbf{X}_i, t)$ ▷ Predicted velocity at timestep t

$\hat{\mathbf{X}}_N \leftarrow \mathbf{X}_i + \mathbf{V}_i(1-t)$ ▷ Estimated target particle $\hat{\mathbf{X}}_N \sim \mathbf{Z}_1$

$\mathbf{V}_i \leftarrow \mathbf{V}_i - \gamma(t) * \nabla_{\mathbf{X}_i} \mathcal{L}\mathcal{L}(\hat{\mathbf{X}}_N)$ ▷ DiverseFlow step

$\nabla_{\text{MCG}} \leftarrow \frac{\partial}{\partial \mathbf{X}_i} \|\mathbf{Y} \odot \mathbf{M} - \hat{\mathbf{X}}_N \odot \mathbf{M}\|_2^2$ ▷ Manifold Constrained Gradient

$\mathbf{X}_{i+1} \leftarrow \mathbf{X}_i + \mathbf{V}_i \Delta_t$ ▷ Euler step

$\mathbf{X}'_{i+1} \leftarrow \mathbf{X}_{i+1} - \alpha_{t_i} \nabla_{\text{MCG}}$ ▷ Apply MCG correction; $\alpha_{t_i} = \sqrt{1-t_i}$

$\mathbf{Y}_{i+1} \leftarrow \mathbf{X}_0(1-t') + \mathbf{Y}t'$ ▷ Linearly interpolate between $\mathbf{X}_0$ and $\mathbf{Y}$ at t_{i+1}

$\mathbf{X}''_{i+1} \leftarrow \mathbf{X}'_{i+1} \odot (1-\mathbf{M}) + \mathbf{Y}_{i+1} \odot \mathbf{M}$ ▷ Replace known region with $\mathbf{Y}_{i+1}$

end for

return $\mathbf{X}_N$

`flow-matching`) for the conditional path construction.

We solve the ODE with an Euler solver with 100 steps. We start with a budget of $K = 2$ (as for $K = 1$, the ODE must always find at least 1 mode) and increase K till $K = N = 10$, where $N = 10$ is the true number of modes in the dataset. For each K, we repeat 1000 trials (by taking random seeds 0-999). We use $\gamma(t) = 2\frac{\sqrt{1-t}}{\|\nabla \log \mathcal{L}\|}$. Since the data is 2D, we do not use any feature encoder F.

We find ~ 7 modes on average with DiverseFlow, while

Figure 16. Generation for Class 327 (Starfish). DiverseFlow finds a significantly different result (a purple sea star) in top-right sample.

IID sampling finds ~ 5.6 modes. With regular IID sampling, the least diverse seems to be the Stochastic Interpolant [2]. Additionally, for the quantity 'maximum modes found at any trial' we observe that in over 1000 trials with a budget of $K = 10$, IID sampling does not find a single instance of all 10 modes in any CFM formulation.

8.5. Mode-finding With Ideal Score

In Figure 11, no model is trained, and we have access to a true score function of a mixture of uniform Gaussian distribution, as shown in Figure 17. We scale the DPP gradient by $\gamma(t) = W\frac{\sigma(t)}{\|\nabla \log \mathcal{L}\|}$, where $\sigma(t)$ is the variance schedule path, and W is a variable temperature parameter (Diversity Scale or Y-axis in Figure 11). We also vary the noise levels from 1 (regular SDE) to 0 (probability flow ODE); it can be observed in Figure 11 that both Particle Guidance and DiverseFlow find the best result at noise level of 0.1.

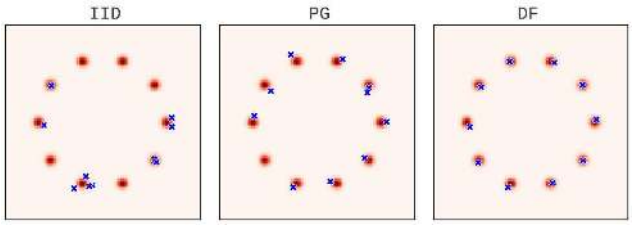


Figure 17. Finding modes on uniform mixture of Gaussian with true score.

8.6. Choice of Feature Extractor

Figure 18 shows an ablation over the effect of using a CLIP vs. a DINO feature extractor. We observe that DINO achieves better diversity and quality on the polysemous prompts (Table 2). This may be due to the fact that using DINO results in a purely image-based feature similarity. However, CLIP features are trained with image-text similarity, and might struggle with polysemous images. For example, an image of a human boxer and an image of a boxer dog can bothmap to similar CLIP latents, despite having stark visual differences.

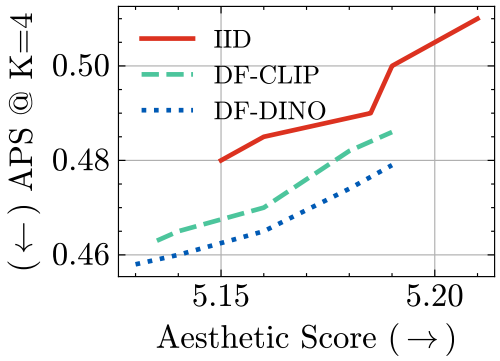


Figure 18. CLIP versus DINO feature extractor.

8.7. Connections to Particle Guidance

It is possible to formulate Particle Guidance in DiverseFlow's framework. Consider the DPP kernel $\mathbf{L}$ that we define in Equation (6). Particle Guidance defines a time-varying 'log potential' that takes the form:

$$\log \Phi_t^{(i)}(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(k)}) = \sum_j \mathbf{L}^{(ij)} \tag{11}$$

That is, the log potential for each particle is its pairwise similarity with every other particle. However, it is not readily apparent why the log potential is this pairwise sum (Equation 4 in particle guidance paper). In our work, the DPP is a probability measure that yields an approximate likelihood of the joint distribution $p(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(k)})$. Therefore, the log potential is simply the log-likelihood of the DPP. One geometric way to interpret the two approaches may be observed in Figure 20.

Thus, the log potential for each particle in particle guidance is distinct. However in our work, the potential is the same for any particle, as it is defined over the determinant. The kernel-sum utilized in Particle Guidance can also be interpreted as an approximate joint likelihood function, except, unlike the DPP, it assigns a non-zero likelihood to the occurrence of duplicate elements. It is thus a softer form of diversification, which can be observed in Figure 11. Finally, particle guidance does not consider a quality factor on the kernel, unlike DPP-based methods.

8.8. Connections to training-based approaches

There are several training-based approaches that implicitly improve diversity. For instance, assigning more optimal coupling [19, 30] can reduce the distance between data and noise, which makes it unlikely for different source samples to be coupled with the same target—thereby improving both the quality and diversity in expectation.

One may question whether it is possible to directly train coupled ODEs, such as the one defined in Equation (10). To do so, it is necessary to make modifications in the model architecture, such that each individual point becomes aware of other points in the set/batch. Video-based generative models introduce a similar type of coupling between frames by adding temporal attention, and can be used as inspiration. In essence, converting DiverseFlow to a trainable approach would require learning a time-varying $K \times d$ matrix field, which is a relatively unexplored area in generative learning; relevant research that explores this direction is the recent work of Isobe et al. [13], that extends flow matching over matrix fields. We hope to explore training-based approaches that incorporate determinantal point processes in future work.

9. Limitations and Challenges

9.1. Soft-DPP Objective:

Recall that the DPP assigns a zero likelihood to a set $\{x^{(1)}, x^{(2)}, \ldots, x^{(k)}\}$ as long as any $x^{(i)} = x^{(j)}$, that is, duplicate elements are not tolerated.

The exact log-likelihood defined in Equation (7) can be thus be undefined on the rare occasion when we have near-identical elements in the set. The work of Yuan and Kitani [33] presents a relaxed objective to address this problem. Instead of maximizing $\sum_a \log(\lambda_a/(1+\lambda_a))$, we can maximize the expectation of the cardinality of the DPP (a bound on the rank of $\mathbf{L}$):

$$\begin{aligned}\mathbb{E}\left[\,|\,\{\hat{\mathbf{x}}_1^{(1)}, \hat{\mathbf{x}}_1^{(2)}, \ldots, \hat{\mathbf{x}}_1^{(k)}\}\,|\,\right] &= \sum_{a=1}^{k} \frac{\lambda(\mathbf{L})_a}{\lambda(\mathbf{L})_a + 1} \\ &= \mathrm{Tr}(\mathbf{I} - (\mathbf{L}+\mathbf{I})^{-1})\end{aligned} \tag{12}$$

For high-dimensional problems (such as text-to-image generation), we find that the exact likelihood Equation (7) is suitable, as it is highly unlikely for random source points to be identical in high-dimensional space.

9.2. Limited by Underlying Model

From a modeling perspective, while DiverseFlow seeks to enhance the sample diversity of flow-matching models under a fixed sampling budget, it is still limited by the distribution modes the underlying FM models have learned. For instance, the word "mouse" may refer to: (i) a mammal (rodent), (ii) a computer peripheral. DiverseFlow could not generate any samples of the computer mouse with just the prompt "a mouse" (Figure 9); we hypothesize that the learned likelihood of the animal significantly dominates the latter meaning. Again, with SD-v3, we could not find any examples of coins for "a mint" which we could find for SD-v1.5. Thus, the discovery of diverse modes is still clearly dependent on the model being used. In Figure 19, we show some examples where the polysemous meaning was not discovered, and in Figure 12, we show discovered polysemous meanings.

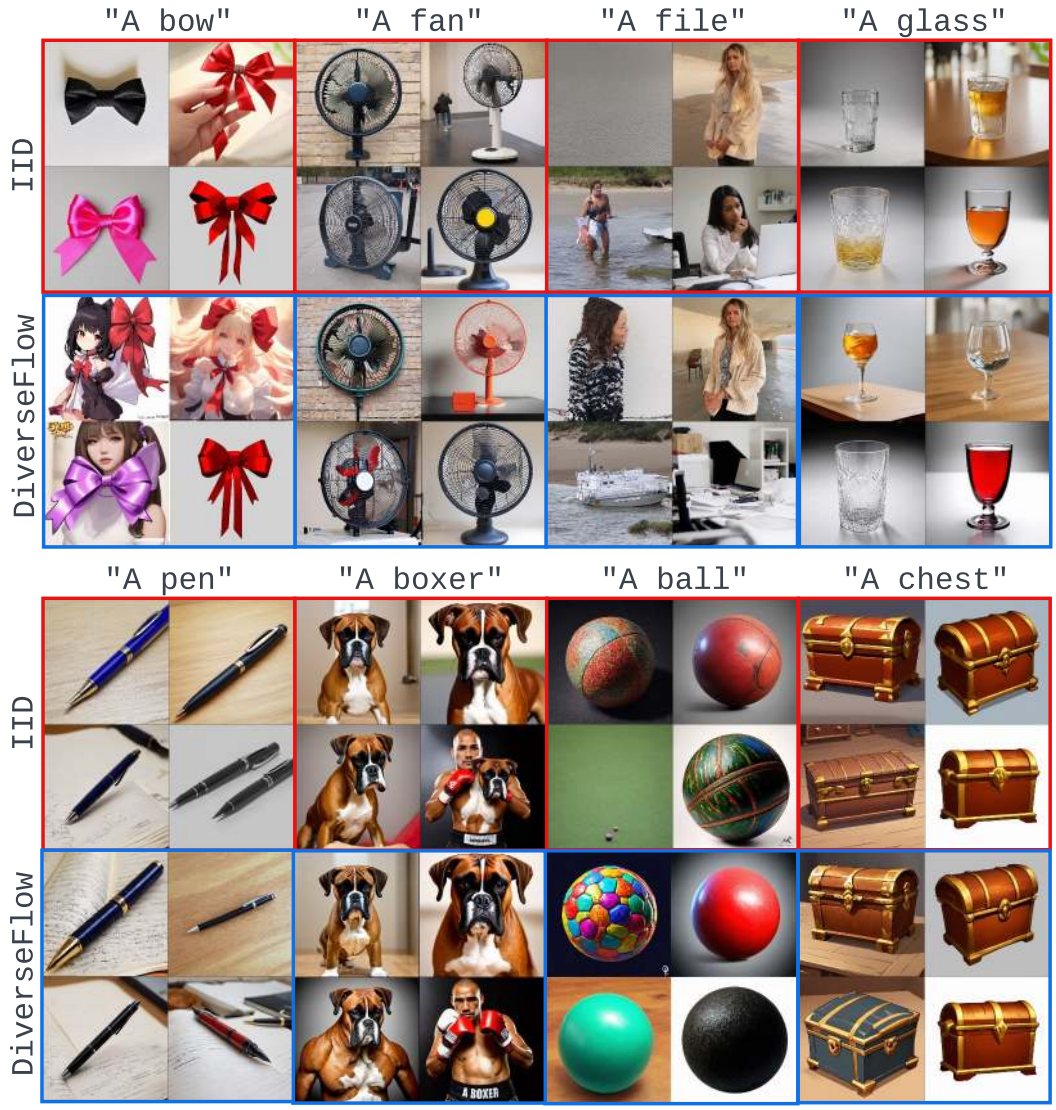
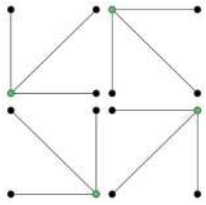
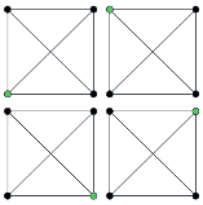


Figure 19. Some examples on SD3 where significantly polysemous meanings are not discovered. However, DiverseFlow still yields more diverse samples compared to IID samples.



(a) Computing the sum of pairwise distances from current particle (green) to every other particle (black)



(b) Computing the determinant needs to consider the distance of every point from every other point

Figure 20. A geometric look at Particle Guidance and DiverseFlow.

9.3. Computational Cost

From a computational perspective, for high-resolution generative modeling, estimating the diversity gradient $\nabla_{\mathbf{x}_t}\mathcal{LL}$ can be memory intensive. With either Stable Diffusion or LFM, it is necessary to backpropagate over (i) the KL-regularized AutoEncoder, (ii) the feature encoding ViT, F, and (iii) the high-resolution sample $\hat{\mathbf{x}_1}$—thus practically limiting us to a batch of 4 samples at a time. We note that Particle Guidance faces a similar challenge.

One way to overcome the memory limit is to utilize a progressively growing kernel: we can sample a set of 4 images, and then sample another 4, where the kernel is 8×8, and another 4, where the kernel is 12×12, and so on. Thus,

Algorithm 2 Progressively Growing Kernel

Require: number of progressions R, number of sampling steps N, time-varying velocity field v_θ, budget K
  $C = \{\}$ ▷ Initialize Cache
  for r=$0 \dots R-1$ **do**
    $\mathbf{X}_0 \sim \mathcal{N}(0, \mathbf{I})$ ▷ Sample source
    $S \leftarrow \mid C \mid$ ▷ Cache size
    for i=$0 \dots N-1$ **do**
      $t_i, t_{i+1} \leftarrow \frac{i}{N}, \frac{i+1}{N}$ ▷ Uniform timesteps
      $\Delta_t \leftarrow t_{i+1} - t_i$
      $\mathbf{V}_i \leftarrow v_\theta(\mathbf{X}_i, t)$ ▷ velocity at timestep t
      $\hat{\mathbf{X}}_N \leftarrow \mathbf{X}_i + \mathbf{V}_i(1-t)$ ▷ Estimated target
      $\mathbf{X} \leftarrow \{\hat{\mathbf{X}}_N^{(1)}, \dots, \hat{\mathbf{X}}_N^{(K)}, C^{(1)}, \dots, C^{(S)}\}$ ▷ Add cached samples to the set
      $\mathbf{V}_i \leftarrow \mathbf{V}_i + \gamma(t) * \nabla_{\mathbf{X}_i} \mathcal{LL}(\mathbf{X})$ ▷ DiverseFlow step
      $\mathbf{X}_{i+1} \leftarrow \mathbf{X}_i + \mathbf{V}_i \Delta_t$ ▷ Euler step
    end for
    $C \leftarrow \mathbf{X}_N$ ▷ Add to cache
  end for
  return C

only the kernel size will grow to $K \times r$ at any iteration r, but we will still compute the gradient with respect to K samples. We provide a pseudocode for this procedure in Algorithm 2.

9.4. Entangled Modes

We find that in many cases, the diverse results obtained by DiverseFlow consist of multiple semantic meanings entangled into one image (for instance, coin with deer head, or or coin with mint leaves). However, we find that these entangled modes are a characteristic of the generative models for polysemous prompts, and thus also a limitation of the underlying model.

An open question for the research community can be how to induce diversity so that disentangled and distinct modes are discovered for ambiguous prompts, rather than entangled ones. Further, numerically measuring the entanglement of different concepts in a generated image could be an initial step towards solving this problem.