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-

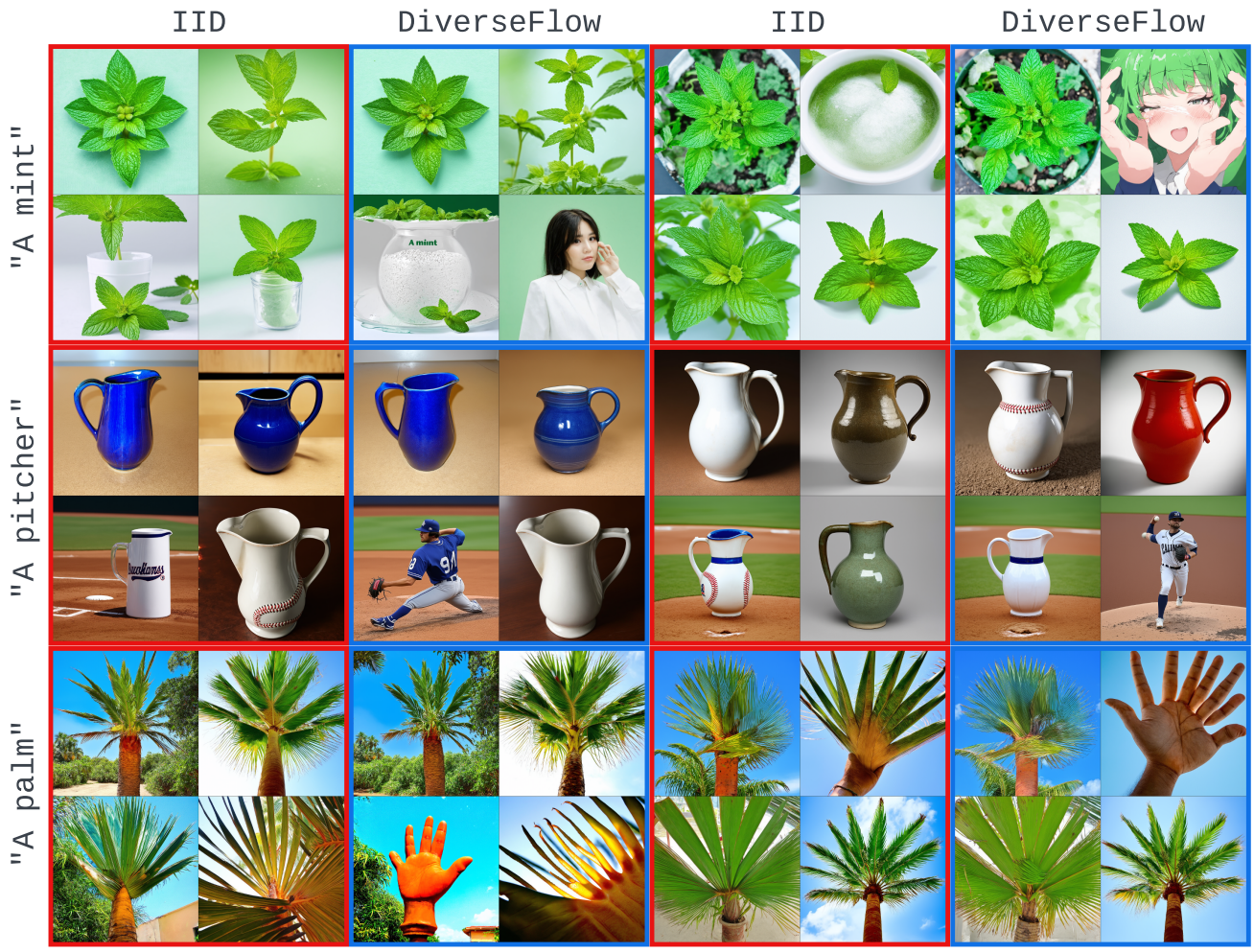


Figure 12. Some examples on SD3 where DiverseFlow discovers alternate meanings that IID sampling doesn't find.

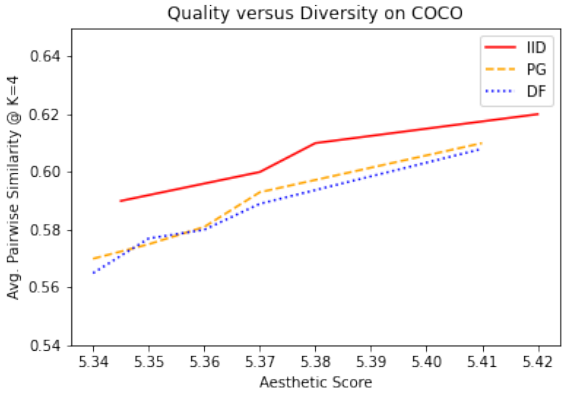


Figure 13. Diversity and Quality on COCO validation set.

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-

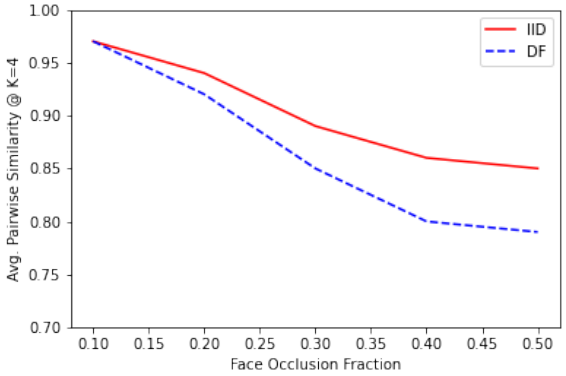


Figure 15. Similarity between faces decreases with increasing occlusion mask size. DiverseFlow finds more dissimilar faces with larger occlusions, but has little effect on small occlusions.

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 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.

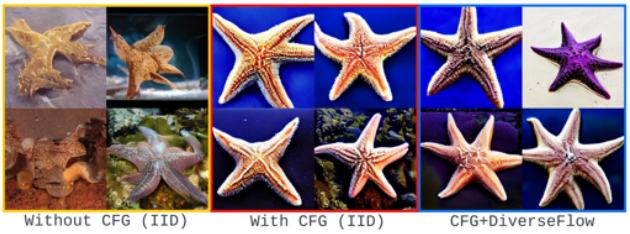


Figure 16. Generation for Class 327 (Starfish). DiverseFlow finds a significantly different result (a purple sea star) in top-right sample.

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.

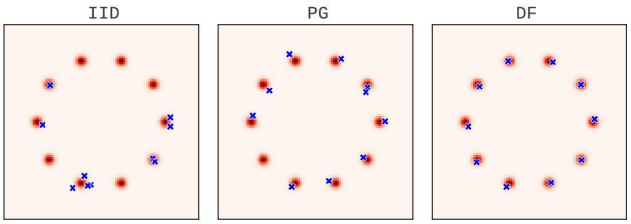


Figure 17. Finding modes on uniform mixture of Gaussian with true score.

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.

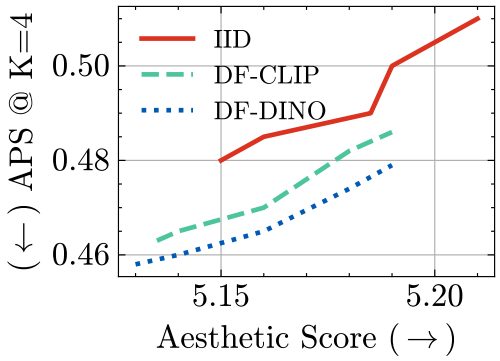


Figure 18. CLIP versus DINO feature extractor.

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 $\mathbf{L}$):

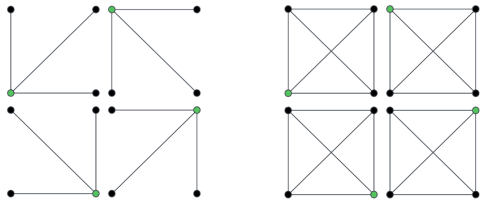
$$\begin{aligned}\mathbb{E}\left[\,|\,\{\hat{\mathbf{x}}_1^{(1)}, \hat{\mathbf{x}}_1^{(2)}, \ldots, \hat{\mathbf{x}}_1^{(k)}\}\,|\,\right] &= \sum_{a=1}^{k} \frac{\lambda(\mathbf{L})_a}{\lambda(\mathbf{L})_a + 1} \\ &= \mathrm{Tr}(\mathbf{I} - (\mathbf{L}+\mathbf{I})^{-1})\end{aligned} \tag{12}$$

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.