Generating High Fidelity Data from Low-density Regions using Diffusion Models

Vikash Sehwag[†] Caner Hazirbas[‡] Albert Gordo[‡] Firat Ozgenel[‡] Cristian Canton Ferrer[†]
[†] Princeton University, [‡] Meta AI
vvikash@princeton.edu, {hazirbas, agordo, firatozgenel, ccanton}@fb.com

(a) Real (b) BigGAN-deep (c) DDPM (d) DDPM (Ours)

Figure 1. **Real vs synthetic data.** We compare synthetic images from different generative models with real images from the low-density (1.a.i) and high-density (1.a.ii) neighborhoods of the data manifold, respectively. In 1.b we show uniformly sampled images from BigGAN [4] and in 1.c we display images generated using the conventional uniform sampling process from the diffusion model (DDPM [10, 17]). While diffusion model achieves much higher diversity than GANs, uniform sampling from them rarely generates samples from low-density neighborhoods. (1.d) Our framework guides the sampling process in diffusion models to low-density regions and generates novel high fidelity instances from these regions.[1]

Abstract

Our work focuses on addressing sample deficiency from low-density regions of data manifold in common image datasets. We leverage diffusion process based generative models to synthesize novel images from low-density regions. We observe that uniform sampling from diffusion models predominantly samples from high-density regions of the data manifold. Therefore, we modify the sampling process to guide it towards low-density regions while simultaneously maintaining the fidelity of synthetic data. We rigorously demonstrate that our process successfully generates novel high fidelity samples from low-density regions. We further examine generated samples and show that the model does not memorize low-density data and indeed learns to generate novel samples from low-density regions.

1. Introduction

Most common image datasets have a long-tailed distribution of sample density[2], where the majority of samples lie in high-density neighborhoods of the data manifold. Samples from low-density regions often comprise novel attributes (Figure 1a) and have higher entropy than high-density samples [1]. However, due to their lower likelihood, curating even a small amount of such samples requires a dedicated effort [16].

Our goal is to leverage generative models to generate synthetic images from low-density neighborhoods. A natural requirement for this task is that the model should generalize to low-density regions. While generative adversarial networks (GANs) excel at generating high-fidelity samples, they have poor coverage, thus struggle to generate high-fidelity samples from low-density regions [4] (Figure 1b). In contrast, autoregressive models have a high coverage but

[1] ImageNet [8, 29] has no explicit category for humans, though one might be present in some images. Thus generative models might generate synthetic images that include a human. We further conduct a rigorous analysis to validate whether the network has memorized any such information from training samples.

[2] We refer to the long-tail w.r.t. sample density for each class. It is different from the long-tailed distribution over classes [24], *i.e.*, when some classes are heavily underrepresented than others.

fail to generate high fidelity images [7]. We use diffusion-based models due to their ability to achieve high fidelity and high coverage of the distribution, simultaneously [17, 26].

In training diffusion models, the goal is to approximate data distribution, which is often long-tailed. Diffusion models excel at this task, as we observe that the density distribution of uniformly sampled instances from the diffusion model is very similar to real data.

Thus uniform sampling from these models leads to an imitation of real data density distribution, i.e., a long-tailed density distribution, where it generates samples from high-density regions with a much higher probability than from low-density regions (figure 1c). To alleviate this issue, we first modify the sampling process to include an additional guidance signal to steer it towards low-density neighborhoods. However, at higher magnitudes of this signal, the generative process is steered off the manifold, thus generating low fidelity samples. We circumvent this challenge by including a second guidance signal which incentives diffusion models to generate samples that are close to the real data manifold.

Since a very limited number of training samples are available from low-density regions, it is natural to ask whether diffusion models are generalizing in the low-density regions or simply memorizing the training data. After all, recent works have uncovered such memorization in language-based generative models [5, 6]. We conduct an extensive analysis to justify that diffusion models do not show signs of memorizing training samples from low-density neighborhoods and indeed learn to interpolate in these regions. We make the following key contributions.

- We propose an improved sampling process for diffusion models that can generate samples from low-density neighborhoods of the training data manifold.
- We validate the success of our approach using three different metrics for neighborhood density and provide extensive comparisons with the baseline sampling process in diffusion models.
- We show that our sampling process successfully generates novel samples, which aren't simply memorized training samples, from low-density regions. This observation from our sampling process also uncover that despite a limited number of training images available from low-density regions, diffusion models successfully generalize in low-density regions.

2. Related work

Diffusion-based probabilistic models [10, 17, 26] and its closely related variants [38, 39] are likelihood-based models that learns data distribution by learning the reverse process of the forward diffusion process. Following latest advances [10], diffusion models achieve state-of-the-art performance, outperforming other classes of generative models, such as Generative adversarial networks (GANs), VQ-VAE [28], and Autoregressive models [7] on various metrics in image fidelity and diversity [10, 26]. Some of the key factors behind their success are the innovation on the architecture of the diffusion models [10, 17], simplified formulation for the training objective [17], and use of cascaded diffusion processes [10, 18, 26].

Sampling from diffusion models is quite slow since it requires an iterative denoising operation. Reducing this overhead by developing fast sampling techniques is a topic of tremendous research interest [19, 21, 37, 42]. Orthogonal to this direction, our interest is in sampling data from low-density neighborhoods. We further show that our sampling approach can be easily integrated with fast sampling techniques.

To measure neighborhood density around a sample, we use the Gaussian model of training data in the embedding space of a pre-trained classifier. Modeling images in embedding space is a common approach, particularly due to their alignment with human perception [45], in numerous vision applications, such as outlier detection [32] and instance selection [9].

Across generative models, given a distribution learned by the model, there have been previous attempts in sampling from a targeted data distribution. Discriminator rejection sampling (DRS) and its successors [2, 11] consider rejection sampling using the discriminator in a generative adversarial network (GAN). Similarly, Razavi *et al*. [28] exploits a pre-trained classifier to reject samples that are classified with low confidence. Most often the goal is to filter out low fidelity samples, thus improving the quality of synthetic data. In contrast, our goal is to generate high fidelity samples from low-density regions of the data manifold. These samples are rarely generated by the model under uniform sampling, thus sampling them using a naive classifier-based rejection sampling approach leads to high-cost overheads. We instead opt to modify the generative process of diffusion models to guide it towards low-density neighborhoods of the data manifold.

The most closely related work to ours is from Li *et al*. [23], which smoothes class embeddings of a BigGAN model to generate diverse images. In contrast, we focus on diffusion-based generative models. We also demonstrate the limitation of their approach with diffusion models in Appendix A.6.

3. Low-density sampling from diffusion models

In this section, we first provide an overview of the sampling process in diffusion-based generative models. Next, we describe our modification in the sampling process for low-density sampling.

3.1. Overview of diffusion models

Denoising diffusion probabilistic models (DDPM) [17] model the data distribution by learning the reverse process (generative process) for a forward diffusion process. The forward process is often a Markov chain with Gaussian transitions, *i.e.*, $q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$. Given a large number of timesteps (T), this diffusion process sufficiently destroys the information in input samples ($\mathbf{x}_0$) such that $p(\mathbf{x}_T) := \mathcal{N}(\mathbf{x}_t; \mathbf{0}, \mathbf{I})$.

Reverse or generative process is also assumed to be a Markov process with Gaussian transitions that learns the inverse mapping, *i.e.*, $p(\mathbf{x}_{t-1}|\mathbf{x}_t)$, at each time step. This process is usually modeled with a deep neural network, parameterized by θ, that learns the Gaussian transition such that $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$.

$$p_\theta(\mathbf{x}_0) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) \tag{1}$$

The model is trained by maximizing the variational lower bounds on the negative log likelihood over the training data.

In order to sample synthetic data from diffusion models, we first sample a latent vector $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and iteratively denoise it using the following procedure in reverse process.

$$\mathbf{x}_{t-1} = \boldsymbol{\mu}_\theta(\mathbf{x}_t, t) + \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)\mathbf{z}, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{2}$$

We refer to this approach as *baseline* sampling process.

3.2. Generating synthetic images from low-density regions on the data manifold

In this section, we present our approach to generating samples from low-density regions of data manifold using diffusion-based models.

3.2.1 Identifying low-density regions on data manifold

Given a data distribution $q(\mathbf{x})$, low-density regions or neighborhoods are part of the data manifold that have significantly lower sample density than the others. To develop techniques to sample from these regions, the first step is to characterize them.

Limitation of likelihood estimates from the diffusion model. A natural choice to characterize manifold density is to use the likelihood estimate from the diffusion model itself (Equation 1). After all, we expect the likelihood of getting a sample from high-density regions being higher than the low-density regions. However, due to its intractability for diffusion-based models, the likelihood estimates from the model are only an approximation of exact likelihood [17, 36]. We find that these likelihood estimates are not a reliable predictor of manifold density as they fail to align with multiple commonly used metrics or with human judgment (Appendix A.2). This trend aligns with a similar limitation of likelihood estimates in autoregressive models [25].

We shift our focus to discriminative models since they are well-known to learn meaningful embeddings that align with human perception for images [45]. We measure the manifold density by estimating the likelihood of data in the embedding space. Let $(g \circ f)(.)$ be a discriminative model, where f extracts embeddings for the input image and g is the head classifier, most often a linear model. We model embeddings of each class using a Gaussian model and estimate the log-likelihood of a given image ($\mathbf{x}_i$) with class label $\mathbf{y}_i$ from this model. We refer to the negative log-likelihood as *Hardness score (H)*.

$$H(\mathbf{x}, y) = \frac{1}{2}\Big[(f(\mathbf{x}) - \mu_y)^T \Sigma_y^{-1} (f(\mathbf{x}) - \mu_y) + \ln(\det(\Sigma_y)) + k\,\ln(2\pi)\Big] \tag{3}$$

μ_y and Σ_y refer to sample mean and sample covariance for embeddings of class y and k is the dimension of embedding space. We provide further analysis in Appendix A.3 to justify that the decrease in manifold density leads to an increase in hardness score.

To sample from low-density regions, our approach is to guide the diffusion model to generate samples with high hardness scores, *i.e.*, equivalent to achieving low likelihood in the correct class. We maximize the following contrastive guiding loss for this task.

$$L_{g_1}(\mathbf{x}_i, y_i) = \log\left[\frac{exp(H(\mathbf{x}_i, y_i)/\tau)}{\sum_{j=1}^{C} exp(H(\mathbf{x}_j, j)/\tau)}\right] \tag{4}$$

where τ is the temperature and C is the total number of classes.

This formalization of guiding loss function is fairly similar to cross-entropy loss on output softmax probabilities, *i.e.*, $(g \circ f)(.)$. Thus we also consider an equivalent loss function where instead of hardness score, we minimize the output softmax probability in the correct class.

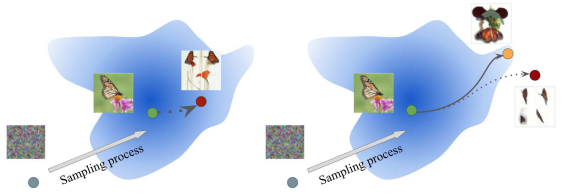
Incorporating guiding loss in sampling process. The next step is to guide the sampling process to low-density regions by minimizing the log-likelihood of generated samples at each time step. We modify the sampling process as follows.

$$\mathbf{x}_{t-1} = \boldsymbol{\mu}_\theta(\mathbf{x}_t, t) + \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)\,\mathbf{z} + \alpha\,\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)\,\nabla^* L_{g_1}(\mathbf{x}_{t-1}, y) \tag{5}$$

where $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, ∇^* refers to normalized gradients, and α is a scaling hyperparameter. We normalize gradients to disentangle the choice of scaling hyperparameter, α, from the diffusion process time steps, t (Appendix A.4). This formulation of sampling process is similar to Dhariwal *et al.* [10], with the modification that our loss function is designed to guide towards low-density regions and that we use normalized gradients.

3.2.2 Maintaining fidelity when minimizing likelihood

We find that the sampling process in Equation 5 is highly successful at smaller values of α. However, with higher values of α, the guidance term dominates the Gaussian transition term from the diffusion model and steers the sampling process off data-manifold, thus generating very low fidelity images (as illustrated in Figure 2). Its effect is exacerbated by model distribution often not being a good approximation of data distribution in low-density regions, in particular, due to the reason that a very limited number of training samples are available from low-density regions.



(a) Small α (●) (b) Large α (●), With β (●)

Figure 2. An illustration demonstrating that small α values successfully guide the sampling process to low-density regions (lighter colors) on the data manifold. However, at large values of α, using additional guidance (by using a non-zero β) from the binary discriminator (Eq. 7) helps in staying close to data manifold. We provide a demonstration of it in figure 3.

We include another term in the sampling process to compel it to stay close to the data manifold. In particular, we train a binary discriminator, with hardness score H', that distinguishes between synthetic and real samples. While sampling, we enforce synthetic images to stay close to the real data manifold by maximizing the following loss value.

$$L_{g_2}(\mathbf{x}_i) = -\log\left[\frac{exp(H'(\mathbf{x}_i, 1)/\tau)}{\sum_{j=0}^{1} exp(H'(\mathbf{x}_j, j)/\tau)}\right] \quad (6)$$

Here class zero and one represents synthetic and real images, respectively. In low-density regions, where model distribution is likely a poor approximation of real data distribution, this objective forces the diffusion model to generate samples that are closest to the real data manifold. Our final sampling process is following.

$$\begin{aligned}\mathbf{x}_{t-1} = \boldsymbol{\mu}_\theta(\mathbf{x}_t, t) + \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)\,\mathbf{z} \\ + \alpha\,\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)\,\nabla^* L_{g_1}(\mathbf{x}_{t-1}, y) \\ + \beta\,\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)\,\nabla^* L_{g_2}(\mathbf{x}_{t-1})\end{aligned} \quad (7)$$

where $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, ∇^* refers to normalized gradients, and α, β are scaling hyperparameters. To further demonstrate the combined effect of α and β, we provide synthetic images with a grid search over both hyperparameters in Figure 3. We also detail our final approach in Algorithm 1.

Algorithm 1: Sampling from low-density regions.

Input : Class label (y), α, β
Function : $Normalize\,(\mathbf{u})$: return $\mathbf{u}/\|\mathbf{u}\|$
$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $i \leftarrow T$ **to** 1 **do**
  if $t > 1$ **then**
    $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\mathbf{s} \leftarrow \mathbf{I}$
  else
    $\mathbf{z} \leftarrow \mathbf{0}$, $\mathbf{s} \leftarrow \mathbf{0}$
  end
  $\mathbf{u}_1 = \alpha\,\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)\,Normalize\left(\nabla L_{g_1}(\mathbf{x}_{t-1}, y)\right)$
  $\mathbf{u}_2 = \beta\,\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)\,Normalize\left(\nabla L_{g_2}(\mathbf{x}_{t-1})\right)$
  $\mathbf{x}_{t-1} = \boldsymbol{\mu}_\theta(\mathbf{x}_t, t) + \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)\,\mathbf{z} + \mathbf{s}(\mathbf{u}_1 + \mathbf{u}_2)$
end
return $\mathbf{x}_0$

4. Experimental results

Experimental setup. We use a U-Net-based architecture with adaptive group normalization for the diffusion model [10]. We consider the encoder from U-Net for the classifier architecture. Both classifier and diffusion model are conditioned on the diffusion process timestep. We consider $T = 1000$ for the diffusion process. When sampling, we use 250 timesteps, as it speeds up the sampling process while incurring negligible cost in the image quality.

We consider two commonly used image datasets: CIFAR-10 [22] and ImageNet [8]. When training the binary discriminator, H', we first uniformly sample synthetic images equal to the size of the training dataset, *i.e.*, 50K

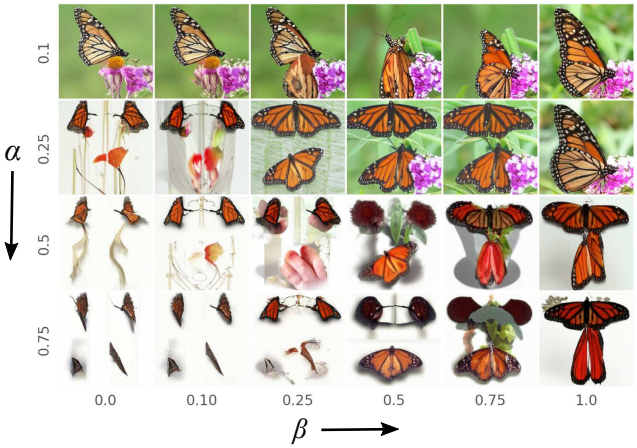


Figure 3. **Controlling hardness and fidelity.** Effect of increasing α (y-axis) and β (x-axis) on synthetic images. Increasing α forces the model to sample from low-density regions while β forces the sampling process to stay close to real data manifold. Salient impact of β includes improving foreground semantics to correctly represent the class and preserving background information.

images for the CIFAR-10 dataset and 1.2M images for the ImageNet dataset. We conduct a hyperparameter search for α and β between 0.01 and 1.0. In most analyses, we sample 50K synthetic images for ImageNet and 10K synthetic images for the CIFAR-10 dataset, i.e., equal to the size of validation set for each dataset. We provide additional experimental details in Appendix A.1.

When sampling we optimize the likelihood estimate, *i.e.*, hardness score, calculated in the embedding space of the U-Net encoder model. To measure generalization to other representation spaces, we consider multiple other models to calculate hardness scores post sample generation. We present results with the ResNet-50 model in the main paper and the rest in the Appendix B.1.

4.1. Generating synthetic data using proposed α-β guided sampling process

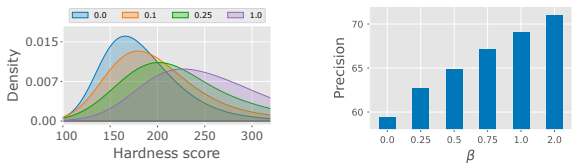
Validating the effect of hyperparameter α and β. Our sampling process is designed such that we can sample images from the low-density regions by increasing α and improving the fidelity of these images using β. Our first goal is to validate the desired effect of both hyperparameters.

While using $\beta = 0$, we first increase α value from 0 to 1.0 and measure the hardness score of sampled images at each value (Figure 4a). Our results demonstrate that increasing α shifts the hardness score distribution to the right, i.e., higher probability of sampling images that have lower estimated likelihood.

Next we fix $\alpha = 0.5$ and increase β from zero to two. We use precision [30] to measure the fidelity of synthetic images. It broadly measures the fraction of images that are realistic or equivalently, the coverage of synthetic data by the support of training data distribution. Our results show that increasing β does improve the realism of generated synthetic images (Figure 4b).

Finally, we analyze the joint effect of parameters α and β. We perform a grid search over both α and β and generate images for each pair of values. To avoid the impact of stochasticity, we use the same seed for all runs of the sampling process. We present the sampled images in Figure 3.

These visualizations validate our argument that solely increasing α to very high values degrades image fidelity. This is because a higher value of α encourages sampling of low-likelihood images. However, the model can satisfy this constraint by simply generating a poor-quality image. Increasing β addresses this issue, in particular on high values of α, where it restores the key attributes of the image thus effectively moving it closer to the data manifold. We find that a $1:1$ ratio between α and β strikes a modest trade-off between sample hardness and fidelity and use $\alpha = \beta = 0.5$ for further experiments.



(a) α increases the hardness score. (b) β improves the fidelity

Figure 4. **Validating effect of hyperparameters.** Quantitative results validating the desired effect of hyperparameters α and β.

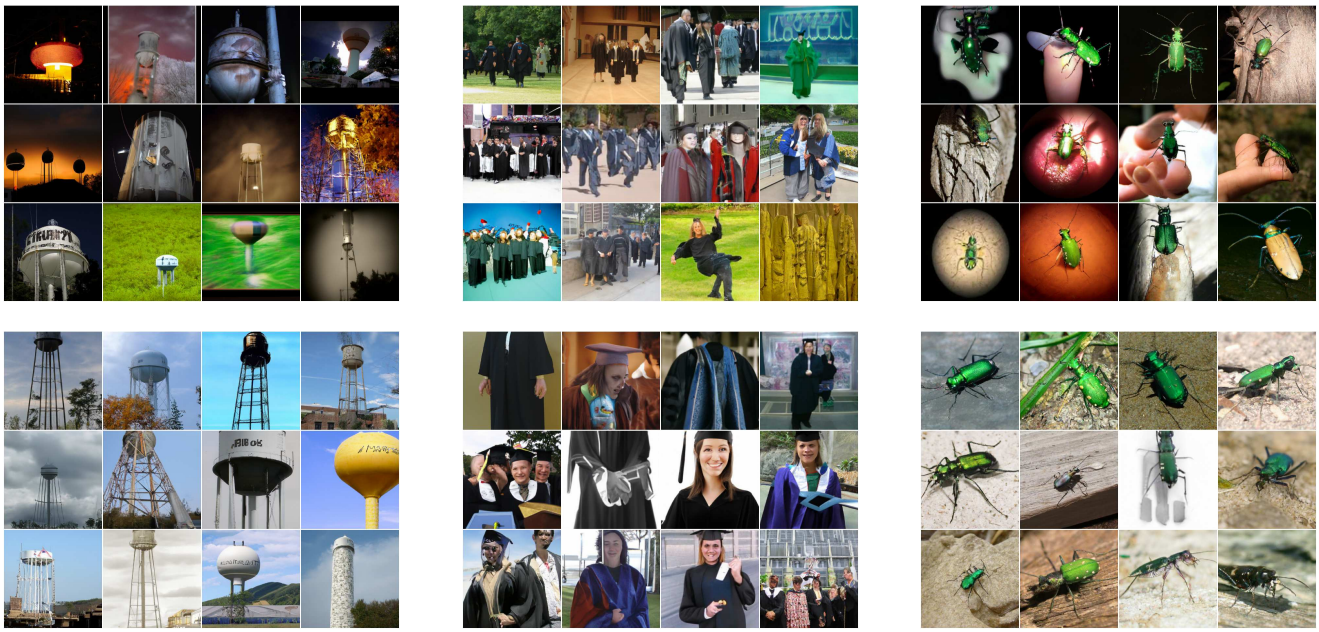
Comparing our sampling process with the baseline sampling process. We compare the synthetic images generated from the baseline and our sampling approach in Figure 5, 6. We use identical experimental setup, including seeds for random number generators, for both sampling processes thus leaving guidance terms to be the only factor impacting final images. Images from our approach are visually distinguishable from the baseline approach since the diffusion model introduces significant changes in the foreground object semantics and background to satisfy the constraints on hardness and fidelity. We provide additional visualizations in Appendix B.3.

4.2. Quantitative comparison of neighborhood density

To validate that our sampling process does generate data from low-density regions, we quantitatively compare the manifold density in the neighborhood of synthetic images with different baselines.

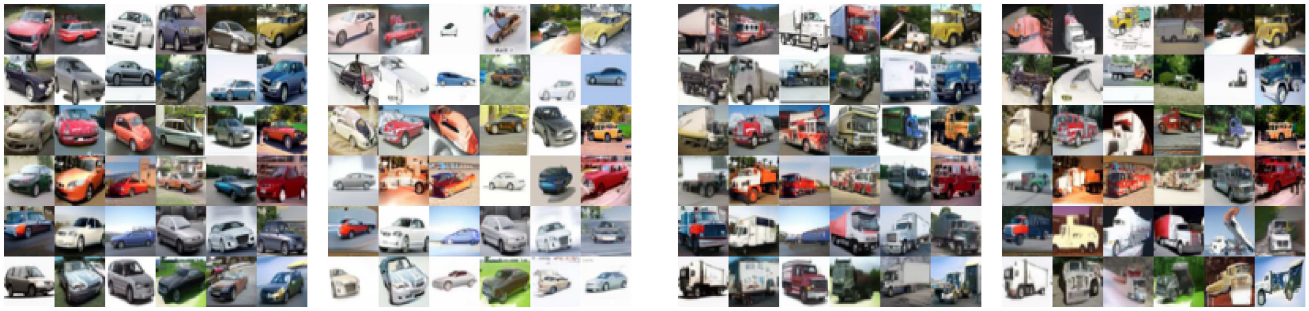
Metrics to measure neighborhood density. We use hardness score as the first validation metric since we maximize it in the sampling process. However, our sampling process might maximize hardness score without actually moving the sampling process to low-density regions. Thus we consider two additional metrics, namely Average nearest neighbor (AvgkNN) and local outlier factor (LOF) [3] to further validate the success of our approach. AvgkNN measures density using proximity to nearest neighbors. We choose five nearest neighbors, which is a common choice [9]. In contrast, the local outlier factor improves on the nearest-neighbor distance metric to compare density around a given sample to density around its neighbors. Higher values of the local outlier factor indicate the sample lies in a much lower density region than its neighbors. We calculate all distances in the feature space of a ResNet50 network which is pre-trained on the ImageNet dataset. We ablate on the choice of feature extractor in Appendix B.1 and show that our conclusions don't change with this choice. For this analysis, we sample 50K synthetic images using recommended values of α and β from Section 4.1. We compare our approach with three baselines 1) BigGAN-deep 2) Real images from the ImageNet validation set and 3) synthetic images generated using baseline sampling from the DDPM model. We present our results in Figure 7.

All three metrics validate the success of our approach. Under all three metrics, our sampling process has a higher



(a) Water tower (b) Academic robe (c) Tiger beetle

Figure 5. **Comparing samples from proposed and baseline sampling process.** We compare synthetic images from our proposed sampling approach (top) with the baseline sampling process (bottom) on the ImageNet dataset. We use identical random seed for both stochastic sampling processes. Therefore, generation of each pair of images among the two approaches starts from the identical latent vectors and the only difference is the additional guidance terms in our approach.



(a) Automobile (b) Truck

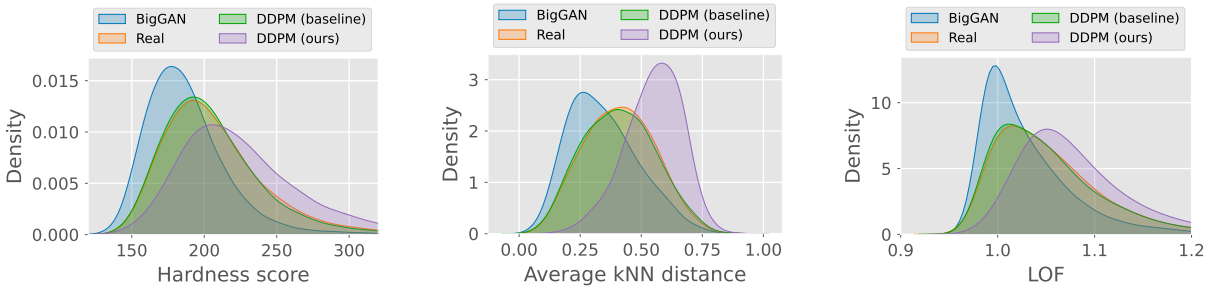
Figure 6. **Comparison on CIFAR-10 dataset.** We compare synthetic images from the baseline sampling process (left) with our proposed sampling approach (right) on the CIFAR-10 dataset. We use the identical seed for random number generators for both processes.

probability of generating synthetic images from low-density neighborhoods. It also validates the claims that the sample density in real data itself follows a long-tail distribution and an unmodified sampling process, i.e., baseline sampling process, from diffusion models closely approximates this distribution. In comparison, BigGAN samples are predominantly from low-density regions. Among the three metrics, the difference between our approach and baseline is most significant in AvgkNN distance. When ablating on the choice of the guidance loss function, we find that under sufficient hyper-parameter ablation, one can obtain equivalent results when optimizing likelihood in embedding space or softmax probabilities after the logit layer (Appendix A.5).

Equivalent reduction in computational cost. Assume that we want to sample images from low-density neighborhoods, *i.e.*, the hardness score of each synthetic sample is greater than a threshold. A naive rejection sampling-based approach is to sample images uniformly at random and reject images that do not satisfy the criterion. However, due

Table 1. **Reduction in sampling cost.** Comparing the sample generation time of our method with uniform sampling. Each entry represents the time taken (in days) to generate 5K 256×256 resolution synthetic images from the corresponding hardness score range on a single A100 GPU.

Score-range	$200 - 240$	$240 - 280$	$280 - 320$
Baseline	1.99	5.74	16.79
Ours	1.88 ($\times$**1.1**)	2.03 ($\times$**2.8**)	2.78 ($\times$**6.0**)



(a) Hardness score (b) AvgkNN distance (c) Local outlier factor

Figure 7. **Comparing neighborhood density.** We measure the density in the neighborhood of a given set of instances using three different metrics. All three metrics share a common trend: while baseline sampling generates synthetic samples that have similar density distribution as real data, our sampling process generates samples from low density neighborhoods with higher probability.

to the long-tail nature of sample density, the likelihood of a sample being from low-density regions is low, thus we would need to reject many samples to curate desired samples. Due to the iterative nature of the sampling process, generating synthetic data from diffusion models is computationally expensive, thus making rejection sampling a highly computationally costly approach (Table 1). Our approach does not depend on rejection sampling, thus it is up to $2-6\times$ faster than the former approach (Table 1).

5. Is our sampling process generating memorized samples from training data?

Since a limited number of samples are available from low-density regions in our long-tailed datasets, the generative model might memorize these samples and fails to generate novel samples from these regions. Therefore we conduct a rigorous analysis to identify whether our sampling process is exploiting any memorization that might be happening in diffusion models.

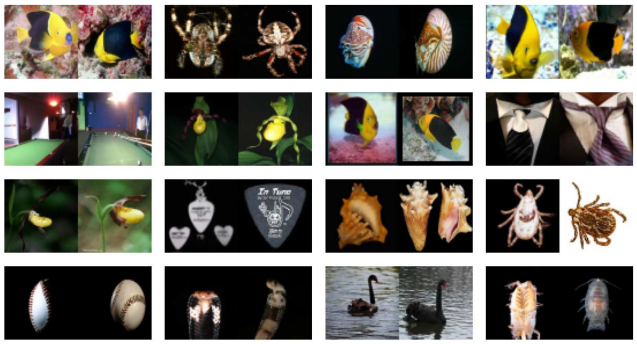


Figure 8. **Is low-density synthetic data being memorized?** Pairs of synthetic and real images with smallest euclidean distance in the feature space. In each pair, left and right image correspond to synthetic and real image, respectively. Our search space for these examples includes all pairs of 50K synthetic images and 1.2M real images. While the synthetic images share multiple attributes with the nearest real image, they are not identical to the real images.

Figure 9. **Is low-density synthetic data novel?** For each synthetic image, we analyze the class label of five nearest neighbors from real data. While each synthetic image has high fidelity and correctly represents the class, it often lies closer to samples of other classes in feature space. Even when the class label is not different, the synthetic image differs significantly from closest real samples.

Analyzing nearest-neighbor distance. We argue that if training data is being memorized, synthetic images will be substantially similar to training data. We measure this similarity by euclidean distance in the embedding space of well trained image classifiers. Thus, if a synthetic image is simply memorized from training data, its nearest-neighbor distance from real data will be very small.

We sample 50K images using our sampling approach and measure their nearest neighbor distance from 1.2M real images in the training set of the ImageNet dataset. We compare these values with the nearest neighbor distance for real data in the validation set. If our approach has memorized training samples, its nearest neighbor distance should be much smaller than real samples. However, the average dis-

tance for our samples is 0.42, much higher than 0.29 for real samples. It supports our hypothesis that our sampling process is not simply generating memorized training samples.

Analyzing synthetic-real data pairs for signs of memorization. Moving beyond comparing distribution statistics, now we analyze individual samples for signs of memorization. In particular, our goal is to manually analyze synthetic images and their closest neighbors for signs of memorization. Even more, we want to analyze pairs that would have the highest likelihood of being memorized, i.e., synthetic samples which are closest to real data. Across all 60B pairs ($50\text{K} \times 1.2\text{M}$) of synthetic and real images, we manually analyze the top-500 pairs with the smallest pairwise distance.

We observe that while images in these pairs share multiple attributes, such as object shape, texture, and identity, they are not being memorized. Instead, they are some semantic variation of the real images, highlighting that the diffusion model learned the data manifold instead of memorizing these samples. We present the top twelve pairs in figure 8 and the rest of them in Appendix B.2.

Novel samples from low-density regions. To validate that our sampling process is indeed generating novel images from low-density regions, we also consider the class label of its nearest neighbors from real data. In multiple cases, we find that the nearest neighbors have different class label than the synthetic sample. We provide few such examples in Figure 9. This phenomenon likely arises due to poorly learned representation by the embedding extractor in low-density regions, primarily due to the scarcity of training samples in these regions.

6. Discussion

We present an improved version of the sampling process in diffusion-based generative models that enables sampling from low-density neighborhoods of the data manifold. We achieve this by guiding the sampling process using two additional classifiers at each timestep. Our sampling process successfully generates novel samples from low-density regions. Our work also identifies another compelling advantage of diffusion models. Despite being trained on a small number of samples from low-density regions, diffusion models successfully interpolate in these regions, i.e., don't memorize the training data from these regions.

We analyze the impact of our guiding loss by juxtaposing samples from baseline and our sampling process (Figure 5, 6). These results demonstrate that the generative model exploits novel transformations in response to guiding loss objectives. We further analyze this effect, by progressively increasing α while keeping all other parameters fixed (Figure 10). Higher values of α forces the model to generate low-likelihood samples. We find that the network sometimes exploits transformations such as photometric changes, zoom, viewpoint, and switching the background to reduce the likelihood of synthetic samples.

Figure 10. **Progressive sampling.** We incrementally increase α across different runs of the sampling process. It highlights how the guiding loss progressively moves the synthetic images to low density regions.

The sampling process in diffusion models iterates for hundreds of steps to generate a single sample. This challenge is often solved using a fast sampling process, which trades off sample quality for speed [19, 37]. To demonstrate that our approach can also integrate with fast sampling techniques, we integrate our modified sampling process with the fast sampling approach from Song et al. [37]. We find no strikingly different trade-off between fidelity and sampling steps for the baseline and our approach (Appendix A.7). At a very low number of sampling steps, such as ten, both approaches struggle to generate high-quality images. However, with increasing the number of timesteps, the fidelity of both baseline and our approach quickly improves.

7. Limitations and broader impact

We guide the sampling process by navigating the data manifold through the feature space of image classifiers. While proximity in feature space of deep neural networks aligns with human perception [45], deep neural networks are also well known to be biased towards certain attributes, such as texture [12] and background [34, 44]. Our sampling process can exploit these biases, such as by simply removing the background, to induce a large change in the likelihood in feature space. We also conduct an examination to investigate signs of memorization and whether our sampling process is exploiting them. While we didn't observe any memorization on the ImageNet dataset, diffusion models might memorize samples on even more complex and non-curated datasets than ImageNet. In event of such memorization, our sampling process might exploit it.

Deep neural networks often struggle to generalize to novel and rarely observed samples from the distribution [16, 20]. We believe that our work can further assist in improving the distributional robustness of these networks. Our sampling process also reveals that diffusion models successfully generalize to low-density regions of data manifold which further strengthens the argument that these models hold the potential to provide tremendous benefits in representation learning [13, 33].

References

[1] Chirag Agarwal, Daniel D'souza, and Sara Hooker. Estimating example difficulty using variance of gradients. *arXiv preprint arXiv:2008.11600*, 2020. 1

[2] Samaneh Azadi, Catherine Olsson, Trevor Darrell, Ian Goodfellow, and Augustus Odena. Discriminator rejection sampling. In *International Conference on Learning Representations*, 2018. 2

[3] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104, 2000. 5

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019. 1

[5] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, pages 267–284, 2019. 2

[6] Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, Aug. 2021. 2

[7] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019. 2

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2009. 1, 4

[9] Terrance DeVries, Michal Drozdzal, and Graham W Taylor. Instance selection for gans. *Conference and Workshop on Neural Information Processing Systems*, 33:13285–13296, 2020. 2, 5

[10] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. *Conference and Workshop on Neural Information Processing Systems*, 2021. 1, 2, 3, 4, 11, 13

[11] Xin Ding, Z Jane Wang, and William J Welch. Subsampling generative adversarial networks: Density ratio estimation in feature space with softplus loss. *IEEE Transactions on Signal Processing*, 2020. 2

[12] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019. 8

[13] Sven Gowal, Sylvestre-Alvise Rebuffi, Olivia Wiles, Florian Stimberg, Dan Andrei Calian, and Timothy Mann. Improving robustness using generated data. *arXiv preprint arXiv:2110.09468*, 2021. 8

[14] Matej Grcić, Ivan Grubišić, and Siniša Šegvić. Densely connected normalizing flows. *Advances in Neural Information Processing Systems*, 34, 2021. 11

[15] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. *ICCV*, 2021. 11

[16] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 1, 8, 11

[17] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Conference and Workshop on Neural Information Processing Systems*, 2020. 1, 2, 3

[18] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *arXiv preprint arXiv:2106.15282*, 2021. 2

[19] Alexia Jolicoeur-Martineau, Ke Li, Rémi Piché-Taillefer, Tal Kachman, and Ioannis Mitliagkas. Gotta go fast when generating data with score-based models. *arXiv preprint arXiv:2105.14080*, 2021. 2, 8

[20] Pang Wei Koh, Shiori Sagawa, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR, 2021. 8

[21] Zhifeng Kong and Wei Ping. On fast sampling of diffusion probabilistic models. *arXiv preprint arXiv:2106.00132*, 2021. 2

[22] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. 4

[23] Qi Li, Long Mai, Michael A Alcorn, and Anh Nguyen. A cost-effective method for improving and re-purposing large, pre-trained gans by fine-tuning their class-embeddings. In *Proceedings of the Asian Conference on Computer Vision*, 2020. 2, 13

[24] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 1

[25] Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do deep generative models know what they don't know? In *International Conference on Learning Representations*, 2018. 3, 11

[26] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, 2021. 2

[27] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*. 2019. 11

[28] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. In *Conference and Workshop on Neural Information Processing Systems*, pages 14866–14876, 2019. 2

[29] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 2015. 1

[30] Mehdi SM Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. In *NeurIPS*, 2018. 5

[31] Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*, 2017. 11

[32] Vikash Sehwag, Mung Chiang, and Prateek Mittal. Ssd: A unified framework for self-supervised outlier detection. In *International Conference on Learning Representations*, 2020. 2

[33] Vikash Sehwag, Saeed Mahloujifar, Tinashe Handina, Sihui Dai, Chong Xiang, Mung Chiang, and Prateek Mittal. Robust learning meets generative models: Can proxy distributions improve adversarial robustness? *arXiv preprint arXiv:2104.09425*, 2021. 8

[34] Vikash Sehwag, Rajvardhan Oak, Mung Chiang, and Prateek Mittal. Time for a background check! uncovering the impact of background features on deep neural networks. *arXiv preprint arXiv:2006.14077*, 2020. 8

[35] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 13

[36] 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. 3

[37] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2020. 2, 8, 13, 14

[38] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Conference and Workshop on Neural Information Processing Systems*, 2019. 2

[39] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. In *Conference and Workshop on Neural Information Processing Systems*, 2020. 2

[40] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 13

[41] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems*, pages 10506–10518, 2019. 11

[42] Daniel Watson, Jonathan Ho, Mohammad Norouzi, and William Chan. Learning to efficiently sample from diffusion probabilistic models. *arXiv preprint arXiv:2106.03802*, 2021. 2

[43] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019. 11

[44] Kai Yuanqing Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. In *International Conference on Learning Representations*, 2021. 8

[45] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. 2, 3, 8

[46] Yue Zhao, Zain Nasrullah, and Zheng Li. Pyod: A python toolbox for scalable outlier detection. *Journal of Machine Learning Research*, 20(96):1–7, 2019. 11