Conditional Generation from Unconditional Diffusion Models using Denoiser Representations

Alexandros Graikos
agraikos@cs.stonybrook.edu
Srikar Yellapragada
myellapragad@cs.stonybrook.edu
Dimitris Samaras
samaras@cs.stonybrook.edu

Stony Brook University
NY, USA

Abstract

Denoising diffusion models have gained popularity as a generative modeling technique for producing high-quality and diverse images. Applying these models to downstream tasks requires conditioning, which can take the form of text, class labels, or other forms of guidance. However, providing conditioning information to these models can be challenging, particularly when annotations are scarce or imprecise. In this work, we propose adapting pre-trained unconditional diffusion models to new conditions using the learned internal representations of the denoiser network. We demonstrate the effectiveness of our approach on various conditional generation tasks, including attribute-conditioned generation and mask-conditioned generation. Additionally, we show that augmenting the Tiny ImageNet training set with synthetic images generated by our approach improves the classification accuracy of ResNet baselines by up to 8%. Our approach provides a powerful and flexible way to adapt diffusion models to new conditions and generate high-quality augmented data for various conditional generation tasks. [1]

1 Introduction

Denoising diffusion models have recently gained significant attention in the generative modeling literature due to their impressive results on various synthesis tasks, including image, audio, and molecule synthesis [4, 9, 11, 14, 20, 26]. Providing conditioning information can improve the sample quality of diffusion models, effectively guiding the model towards generating more diverse and representative samples [3, 10, 19].

While text-based conditioning has been widely used in image generation tasks, due to the wide availability of image-caption pairs, it is not always the best approach, as the scale at which the sampling guidance is applied can vary between tasks. For instance, sampling images with a dense per-pixel condition, such as a semantic mask, cannot be adequately substituted with a text condition, no matter how detailed. Additionally, it is often infeasible to

[1] Code is available at https://github.com/cvlab-stonybrook/fewshot-conditional-diffusion

provide finer annotations on the image or pixel level due to the large scale of the data used to train diffusion models. Thus, adapting pre-trained diffusion models to new conditions using the learned intermediate representations of the denoiser network is an attractive alternative.

In this paper, we propose a method to adapt pre-trained unconditional diffusion models to new conditions using the internal representations of the denoiser network. We show that the learned representations are inherently robust to noisy inputs, allowing us to provide guidance during sampling while utilizing a noisy estimate of the final image $\mathbf{x}_0$. Whereas previous methods relied on training large-scale guidance classifiers on the intermediate noisy steps, our method can efficiently learn from a small set of examples by exploiting the existing parameters of the denoiser network. We demonstrate the effectiveness of our approach on various conditional generation tasks, including attribute-conditioned generation and mask-conditioned generation.

When we are presented with more data but not enough to train a conditional diffusion model from scratch, we demonstrate that we can combine the learned unconditional model representations with the fine-tuning of the model. In particular, we focus on synthetic data augmentation, where we generate augmented data using a conditional diffusion model fine-tuned on Tiny ImageNet [15]. We use the internal representations of the unconditional U-Net [24] denoiser network to train a rejection classifier. This classifier is used to reject low-quality samples generated by the fine-tuned conditional diffusion model, which helps to improve the overall quality of the generated images. We then use the augmented data to train a classification model and evaluate its performance on the Tiny ImageNet dataset. Our contributions are as follows:

- We demonstrate how the internal representations of an unconditional diffusion denoiser network can be used to adapt to new conditions with limited examples.
- We verify the effectiveness of our approach on various conditional generation tasks such as attribute-conditioned generation and mask-conditioned generation.
- We show how augmenting the Tiny ImageNet training set with synthetic images generated by our approach significantly improves the classification accuracy over ResNet [6] baselines.

2 Related Work

2.1 Conditional diffusion models

Similar to other generative models, conditional diffusion models perform better than their unconditional counterparts, showing impressive results in text-conditioned generation [20, 23]. Apart from text-to-image models, in [4], the authors demonstrate how exploiting the denoiser network as a learned score function can be used with an auxiliary classifier trained on noisy samples to guide inference with class labels. In [5], the authors show that diffusion models can perform conditional generation by minimizing an energy function defined using the learned denoiser and any auxiliary constraint on the inferred sample.

2.2 Conditioning of unconditional diffusion models

Few-shot conditioning settings have utilized latent diffusion models in the past. In [21, 25], the authors propose learning an expressive latent representation that can be utilized to condition diffusion models on a small set of labeled examples. The idea behind their approach is that the compressed latent representation can be efficiently used to adapt to new conditions. More recently, [18, 36] showcase faster and more efficient adaptation of diffusion models to new conditions, but both require a large number of data to train with.

2.3 Diffusion as a pre-training task

In [1], the authors introduced the idea of utilizing the learned representations of a denoiser network for downstream tasks by training a segmentation network on top of the denoiser features with as few as 20 labels. Similarly, in [2] and [30], the denoising task is exploited as a pre-training task for learning semantically meaningful features that can be easily adapted for performing downstream dense prediction tasks.

3 Background

A Gaussian diffusion process [9, 26] can be viewed as a latent variable model of the form

$$p_\theta(\mathbf{x}_0) = \int p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta^{(t)}(\mathbf{x}_{t-1} \mid \mathbf{x}_t) d\mathbf{x}_1 d\mathbf{x}_2 \dots d\mathbf{x}_T \tag{1}$$

where $\mathbf{x}_0$ is the data, $\mathbf{x}_1, \dots, \mathbf{x}_{T-1}$ are intermediate latent variables that represent noisy versions of $\mathbf{x}_0$, and $\mathbf{x}_T$ is the terminal state with $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The forward process gradually adds Gaussian noise to the data and is defined by a noise schedule α_t

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) := N(\mathbf{x}_t; \sqrt{a_t}\mathbf{x}_{t-1}, (1 - a_t)\mathbf{I}). \tag{2}$$

The reverse process is defined by learned Gaussian transitions and gradually denoises the data, starting from $\mathbf{x}_T := \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) := N(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)). \tag{3}$$

A property of the forward process is that it allows sampling of any intermediate $\mathbf{x}_t$ given $\mathbf{x}_0$, as

$$q(\mathbf{x}_t \mid \mathbf{x}_0) = N(\mathbf{x}_t; \sqrt{\bar{a}_t}\mathbf{x}_0, (1 - \bar{a}_t)\mathbf{I}) \text{ or equivalently} \tag{4}$$

$$\mathbf{x}_t = \sqrt{\bar{a}_t}\mathbf{x}_0 + \sqrt{1 - \bar{a}_t}\varepsilon, \text{ where } \varepsilon \sim N(\mathbf{0}, \mathbf{I}) \tag{5}$$

with $\bar{a}_t = \prod_{i=1}^{t} a_i$. To approximate the reverse process, it is common to fix the variance $\Sigma_\theta = \sigma_t \mathbf{I}$ and learn a function that predicts the noise added at each intermediate $\mathbf{x}_t$ by minimizing

$$\mathbb{E}_{\mathbf{x}_0, t}\left[w(t) \| \varepsilon - \varepsilon_\theta(\sqrt{\bar{a}_t}\mathbf{x}_0 + \sqrt{1 - \bar{a}_t}\varepsilon, t) \|^2 \right]. \tag{6}$$

The original DDPM [9] formulation introduced sampling from a Gaussian diffusion model by sequentially applying the predicted noise

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{a_t}} \left(\mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \varepsilon_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z},\ \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{7}$$

starting from a random $\mathbf{x}_T \sim N(\mathbf{0},\mathbf{I})$. Alternatively, in DDIM [27], the authors proposed utilizing Eq. 5 to sample $\mathbf{x}_{t-1}$ from $\mathbf{x}_t$ by first estimating $\mathbf{x}_0$ and then using the known forward process to 'point-back' towards $\mathbf{x}_t$, as

$$\mathbf{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}} \underbrace{\left(\frac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\,\varepsilon_\theta(\mathbf{x}_t,t)}{\sqrt{\bar{\alpha}_t}}\right)}_{\text{estimated } x_0} + \underbrace{\sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\,\varepsilon_\theta(\mathbf{x}_t,t)}_{\text{direction pointing to } \mathbf{x}_t} + \underbrace{\sigma_t \mathbf{z}}_{\text{random noise}} .$$

A denoiser $\varepsilon_\theta(\mathbf{x}_t,t)$ can be interpreted as a learned score function of the noise-perturbed $\mathbf{x}_t$ [4, 28]

$$\nabla_{\mathbf{x}_t} \log p_\theta(\mathbf{x}_t) = -\frac{1}{\sqrt{1-\bar{\alpha}_t}}\varepsilon_\theta(\mathbf{x}_t,t). \tag{8}$$

For a conditioning variable $\mathbf{y}$, the score of the posterior $p(\mathbf{x}_t \mid \mathbf{y}) \propto p(\mathbf{x}_t)p(\mathbf{y} \mid \mathbf{x}_t)$ can be expressed as

$$\begin{aligned} \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t \mid \mathbf{y}) &= \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log p(\mathbf{y} \mid \mathbf{x}_t) \\ &= -\frac{1}{\sqrt{1-\bar{\alpha}_t}}\varepsilon_\theta(\mathbf{x}_t,t) + \nabla_{\mathbf{x}_t} \log p(\mathbf{y} \mid \mathbf{x}_t). \end{aligned} \tag{9}$$

Thus by modifying the predicted noise ε_θ at every step using the gradient of the log-likelihood w.r.t. $\mathbf{x}_t$ we can draw samples from the posterior distribution $p(\mathbf{x}_0 \mid \mathbf{y})$

$$\hat{\varepsilon}_\theta(\mathbf{x}_t,t) = \varepsilon_\theta(\mathbf{x}_t,t) - \lambda\sqrt{1-\bar{\alpha}_t}\nabla_{\mathbf{x}_t} \log p(\mathbf{y} \mid \mathbf{x}_t) \tag{10}$$

where λ is the tunable *guidance sterngth* hyperparameter.

4 Methodology

4.1 Learning the guidance signal using denoiser representations

To provide conditional guidance during inference, we can compute the gradient of the log-likelihood of a condition $\mathbf{y}$ w.r.t. $\mathbf{x}_t$. Recent works employed a separate classifier $p(\mathbf{y} \mid \mathbf{x}_t)$ trained on **noisy** samples $\mathbf{x}_t$ to provide semantic guidance, in the form of class labels. This work argues that retraining the guidance network from scratch is unnecessary. Instead, we propose using the intermediate denoiser representations, with the estimate of $\mathbf{x}_0$ at every step, to guide sampling.

Recall that from Eq. 5 we can compute an estimate of the final sample $\mathbf{x}_0$ from

$$\hat{\mathbf{x}}_0 = \frac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\,\varepsilon_\theta(\mathbf{x}_t,t)}{\sqrt{\bar{\alpha}_t}}. \tag{11}$$

Using this point estimate $p(\mathbf{x}_0 \mid \mathbf{x}_t) = \delta(\mathbf{x}_0 - \hat{\mathbf{x}}_0)$ we can rewrite the likelihood as

$$\begin{aligned} p(\mathbf{y} \mid \mathbf{x}_t) &= \sum_{\mathbf{x_0}} p(\mathbf{y} \mid \mathbf{x}_t, \mathbf{x}_0) p(\mathbf{x}_0 \mid \mathbf{x}_t) && (12) \\ &= \sum_{\mathbf{x_0}} p(\mathbf{y} \mid \mathbf{x}_0) p(\mathbf{x}_0 \mid \mathbf{x}_t) && \text{(cond. ind.)} \\ &= p(\mathbf{y} \mid \hat{\mathbf{x}}_0) && \text{(point estimate)} \end{aligned}$$

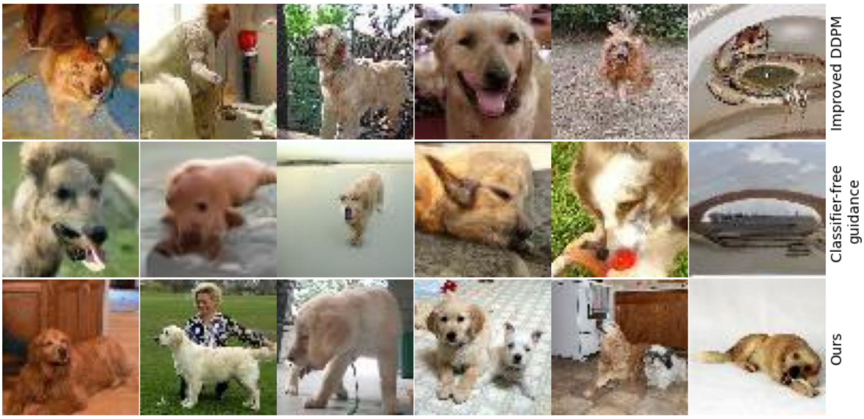

Figure 1: Synthetic images from the fine-tuned class conditional sampling (first row) vs Classifier-free guided sampling (second row) vs. our model with rejection (third row). Samples are from class 9: Golden Retriever on the Tiny-Imagenet. Our method generated more realistic and diverse images compared to others.

and plug-in Eq. 10 to get

$$\hat{\varepsilon}_\theta(\mathbf{x}_t, t) = \varepsilon_\theta(\mathbf{x}_t, t) - \sqrt{1 - \bar{\alpha}_t} \nabla_{\mathbf{x}_t} \log p(\mathbf{y} \mid \mathbf{x}_0(\mathbf{x}_t)). \tag{13}$$

By expressing the likelihood in terms of an estimate of $\mathbf{x}_0$, we allow our guidance network to be trained only on 'clean' samples, thus reducing the overall complexity of the post-hoc adaptation process. However, this assumes that the estimates $\hat{\mathbf{x}}_0$ perfectly follow the marginal distribution $p_\theta(\mathbf{x}_0)$, which is not true, notably for the first few estimates in the denoising process. In practice, the guidance network $p(\mathbf{y} \mid \hat{\mathbf{x}}_0)$ will initially steer the sample $\mathbf{x}_t$ towards the wrong directions, which in most cases cannot be mitigated by the following denoising updates.

Using the unconditional denoiser network as a feature extractor, we can both provide guidance that is robust to the initial inaccurate estimates of $\mathbf{x}_0$ and quickly learn the guidance directions from a small set of labeled samples. The denoiser network is trained with varying noise levels in its inputs and has learned to extract features of different scales at different timesteps. Additionally, the intermediate features of the denoiser U-Net are shown to contain important information for downstream tasks [□]. Therefore, we argue that apart from robustness, using the denoiser features enables us to learn the conditioning models with a small set of examples.

4.2 Combining adaptation with denoiser representations

When presented with more data, we can combine model adaptation methods with the unconditional denoiser features to generate higher-quality and diverse images. We begin by fine-tuning the unconditional diffusion model on the labeled examples we are given by naively adding the conditioning connections to the pre-trained model. It is important to note that here we assume that the image-label pairs that are provided do not cover the entirety of the dataset, which would make the task trivial.

Class	Ours	DiffAE [21]	D2C [25]	DDIM-I	NVAE [31]
Male	15.34	11.52	13.44	29.03	41.07
Female	9.94	7.29	9.51	15.17	16.57
Blond	13.07	16.10	17.61	29.09	31.24
Non-Blond	10.97	8.48	8.94	19.76	16.73

Table 1: FID scores for attribute-conditioned generation on the CelebA-64 dataset.

Even with more image-condition pairs, the conditional model fails to generate realistic images, as demonstrated in Fig. 1. We use the unconditional denoiser representations to train a classifier of the labels of the given samples and improve the results by applying rejection sampling. The classifier model decides whether to accept a generated sample using the predicted probability value. We find this method preferable to just providing guidance during sampling since tuning the weight of the guidance becomes increasingly difficult, with an inherent trade-off between quality and diversity that becomes infeasible to hand-tune and we leave to future work.

5 Experiments

5.1 Few-shot guidance for face attributes

We first demonstrate how we can generate conditional samples with a limited number of image-attribute pairs. We utilize an unconditional diffusion model trained on the CelebA-64 [17] dataset. Similarly to [25], we train an attribute classifier by giving 50 positive and 50 negative examples of a single attribute, such as blonde or male.

To train the classifier, we extract the bottleneck and second upsampling block features from the denoiser U-Net at $T = 700$. The classifier network is a simple 3-layer convolutional network trained for 100 steps. During sampling, we apply classifier guidance with a weight of $\lambda = 1$. The results for the conditional generation of images with the attributes *male*, *female*, *blond*, and *non-blond* are shown in Table 1.

In order to compare with D2C [25] and DiffAE [21], we train a classifier using the latent representation of each and generate samples by unconditionally sampling from the model and applying rejection sampling. When comparing with DiffAE and D2C, we show that we can provide meaningful guidance and achieve comparable FID scores without requiring to compress the entire image into a latent representation and without rejection, which significantly increases the computation needed. For DDIM-I, we use our approach for providing guidance with an estimate of $\hat{\mathbf{x}}_0$, but now using a separately trained classifier network trained only on “clean” images. The subpar performance validates the assumption that the intermediate denoiser representations are more robust to the initial inaccurate estimates of the final image. All FID scores are computed using the images of the CelebA-64 test set with the target attribute.

5.2 Few-shot guidance for semantic segmentations

5.2.1 Mask-conditional generation on faces

We also demonstrate how we can generate conditional samples with a small set of paired images and segmentation masks. We utilize the dataset of [1] and an unconditional diffusion model trained on FFHQ-256 [12]. In this setting, we train on 20 images and their corresponding segmentation masks to generate realistic-looking, semantically accurate images.

We again train a per-pixel classifier network using intermediate U-Net representations extracted at $T = 500$ from decoder upsampling blocks. The classifier is a simple per-pixel multi-layer perceptron trained for 50 steps. During sampling, we apply classifier guidance of $\lambda = 2 \times 10^{-4}$, which is significantly lower than for the single attribute case of 5.1, which we attribute to the finer per-pixel adjustments that we now have to make.

In Table 2, we provide the results of generating images conditioned on segmentation masks taken from CelebA-Mask [16]. We compute the mIoU over all classes between the generated images and input segmentations using a separately trained Oracle segmentation model on a test set of 500 CelebA-Mask images. We also compute the FID scores between the two. When comparing to the naive approach of using a separately trained segmentation network to provide guidance using the estimates of $\hat{\mathbf{x}}_0$, we see that although we get slightly correct semantic results, the quality is subpar. On the contrary, when using the DiffAE latent to train a segmentation network and sample with rejection sampling, we can improve the quality at the loss of semantic correctness and speed. The latent representation over-compresses the image information and fails to accurately reproject the segmentation onto the entire image. Our method provides more diverse and high-quality samples without sacrificing semantic correctness. We note that we ran our method with ten denoising steps to make the comparison to the faster DiffAE fair. The results significantly improve with more steps, as shown in the last column of Fig. 2 (a).

Model	mIoU ↑	FID ↓
DDIM-I	0.28	106.90
DiffAE [21]	0.23	87.68
Ours	**0.31**	**74.74**

Table 2: Mean Intersection over Union and FID on the 500-image CelebA-Mask test set for mask-conditioned generation.

Additionally, we compare our approach to the widely popular ControlNet [36] which fine-tunes copies of the denoiser layers to process the conditioning information. We demonstrate that it does not work in low-data regimes, at which our method is specifically targeted, by repeating our segmentation experiment using an FFHQ-256 ControlNet model. It is evident from Fig. 2 (b) that ControlNet is unable to produce realistic images, as the added layers cannot interpret the conditioning signal and combine it with the unconditional representations with the limited set of examples given. We, on the other hand, exploit the fact that the conditioning information is highly correlated with the learned unconditional denoiser representations and we can provide guidance even in this data-limited setting.

5.2.2 Mask-conditional generation with large diffusion models

We apply our method on Stable Diffusion (SD) [23] where we use the LSUN-Cat segmentation masks from [1] and condition the denoiser with just 30 examples. We freeze the text conditioning to *"a photo of a cat, full body, realistic photo"* and learn a mapping from denoiser features to segmentations to provide the guidance signal. The generated samples in Fig. 2 (c) showcase our method's ability to work seamlessly with larger models.

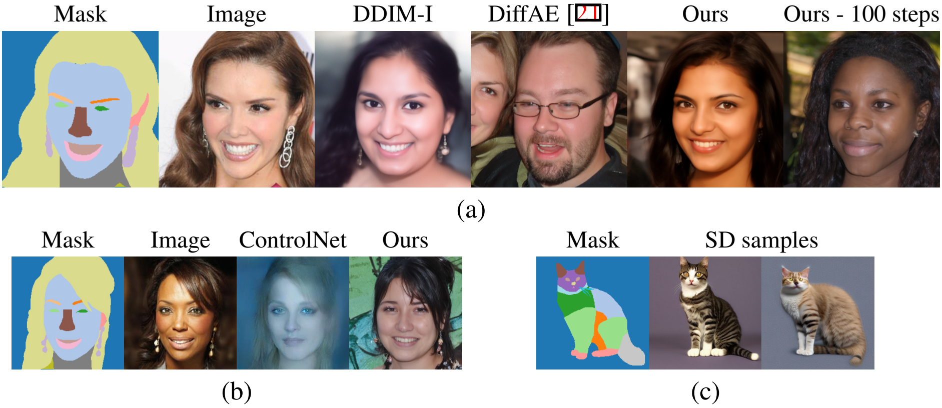

Figure 2: (a) Example of segmentation-conditioned generated images for each method. (b) Comparison with ControlNet [36] using 100 inference steps. (c) Generating mask-conditioned samples from Stable Diffusion [23].

5.3 Synthetic data augmentation

5.3.1 Setup

To evaluate the effectiveness of our proposed approach, we conduct experiments on Tiny Imagenet [29]. It has 200 object categories and 100,000 images, with 500 images per category for training and 10,000 images for validation. The images are 64x64 pixels.

We start with an unconditional DDPM trained on Imagenet images with a cosine noise schedule and fine-tune it as a class-conditioned diffusion model by adding a classifier embedding as proposed in [19]. The classifier embedding layer weights are randomly initialized, and we train this model for 150k steps with a batch size of 128 and a learning rate of 10^{-4}.

Since the Tiny Imagenet dataset is much smaller than Imagenet, the resulting conditional model fails to generate realistic images, as seen in Figure 1. As a solution, we also train a rejection classifier to improve the quality and diversity of generated images. We extract Tiny-Imagenet features from the unconditional DDPM's U-Net bottleneck layer. Two feature vectors extracted at timestep $T = 300$ and $T = 800$ are concatenated to obtain a $1024 \times 4 \times 4$ embedding. The rejection classifier is a 4-layer CNN with 1.2M parameters built on top of the U-Net features. We set a threshold of 0.2 and discard any generated image for which the classifier predicts a probability lower than the threshold.

We use classifier-free guidance sampling [7] to generate images from our class conditional model, taking a linear combination of unconditional and conditional estimates. We use a low guidance weight of 0.01 to encourage higher diversity and generate additional synthetic images for the classes in Tiny-Imagenet to construct training datasets with $100,000-400,000$ synthetic images. Figure 3 shows sample images generated by our approach.

5.3.2 Classification Accuracy Score

Classification Accuracy Score (CAS) [22] can be a better proxy than Inception and FID for measuring the quality of synthetic datasets. CAS was originally designed for Imagenet; we adopted it for Tiny-Imagenet by computing the Top-1 accuracy on the Tiny-Imagenet validation set using a classifier trained on the synthetic data.

Data type	Data Source	Top-1 accuracy (%)
Real	Tiny-Imagenet	52.24
Synthetic	Improved DDPM [19]	35.37
Synthetic	Classifier-free guidance [8]	36.12
Synthetic	**Ours**	**45.09**

Table 3: Classification Accuracy scores of a Resnet-18 on synthetic data. Results indicate rejection is critical in making our synthetic data effective.

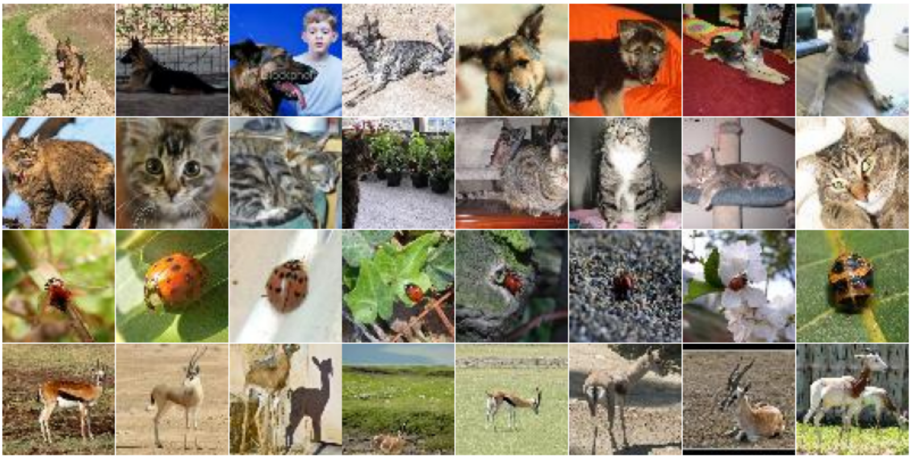

Figure 3: Tiny-Imagenet samples generated by our method. The classes are 28: German Shepherd, 30: Tabby Cat, 36: Ladybug, and 52: Gazelle. Our model can generate diverse images across different classes, proving the effectiveness of rejection sampling.

Table 3 reports the CAS for our synthetic data. In all cases, we train a ResNet-18 model for 40 epochs with a batch size of 256 using the Adam optimizer [13] and a learning rate of 0.001. We generate synthetic data using the fine-tuned class conditional DDPM (Improved DDPM), combining the unconditional and conditional models (Classifier-free guidance), and using our method. The classifier trained on our synthetic data achieves a validation accuracy of **45.09 %**, significantly more than the other baselines, showcasing our capability to generate more diverse and higher-quality data.

5.3.3 Combining real and synthetic data

Having verified the quality of our synthetic data, we also try improving the baseline classification accuracy on Tiny-Imagenet by augmenting the real data with increasing amounts of diffusion-generated synthetic data. We follow the same training procedure as 5.3.2 , using three network architectures: ResNet-18, Wide-ResNet-50 [34], and ResNeXt-50 [32]. Results can be found in Table 4.

We observe that augmenting the training set with our synthetic images improves the accuracy of ResNet baselines by up to **8%**. ResNeXt-50 performs the best, improving accuracy from 53.98% with real data only to **63.15%** with the additional augmented data. We see that

Architecture	Mixup + Cutmix	Real only	Real + 1x Generated	Real + 2x Generated	Real + 3x Generated
Resnet-18	No	52.24	56.13	58.13	**59.37**
	Yes	52.9	58.9	62.01	**62.75**
Wide-ResNet-50	No	53.27	58.57	61.71	**62.82**
	Yes	56.56	62.71	66.42	**66.82**
ResNeXt-50	No	53.98	59.33	62.27	**63.15**
	Yes	57.98	64.4	66.85	**67.05**

Table 4: Comparison of Top-1 Accuracy (%) with our synthetic data. "Mixup + Cutmix" means both techniques were applied with probability 0.5. Our augmentation approach complements image-level augmentations.

performance continues to improve as we increase the amount of synthetic data, with the expected diminishing returns. The improvement is also consistent across all three network architectures, which validates the effectiveness of our data generation method. However, we did notice diminishing returns in the improvement of validation accuracy when increasing the dataset size from 3× to 7×. This suggests that there may be a saturation point in the model's capacity to extract valuable information from augmented data.

5.3.4 Comparison with image level augmentations

To evaluate the effectiveness of our synthetic data compared to image-level augmentation methods, we conducted experiments on Tiny Imagenet using Mixup [35] and Cutmix [33] techniques. We applied Mixup and Cutmix with a probability of 0.5 on all images in the training set (both real and synthetic) and followed the same training procedure from 5.3.3. The results of these experiments are summarized in Table 4.

Our observations reveal that Mixup and Cutmix techniques further enhance the accuracy of ResNet baselines. This improvement is consistent across all the network architectures and different amounts of synthetic data. These findings demonstrate that our augmentation approach complements image-level augmentations and can be effectively combined to achieve even better performance.

6 Conclusion

In this paper, we proposed a method for adapting pre-trained unconditional diffusion models to new conditions using the internal learned representations of the denoiser network. The effectiveness of the proposed approach is demonstrated by providing guidance for attribute-conditioned generation and mask-conditioned generation, as well as filtering samples for synthetic data augmentation. We showed that augmenting the Tiny ImageNet training set with synthetic images generated by our approach significantly improves the classification accuracy over ResNet baselines. This result highlights the potential of our approach to improving the performance on datasets with limited labels by using a generative diffusion model as a pre-training task or feature extractor.

Acknowledgements This work is supported by NSF grants IIS-2123920 and IIS-2212046, Stony Brook Profund 2022 seed funding and a gift from Adobe Research.

References

[1] Dmitry Baranchuk, Andrey Voynov, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Label-efficient semantic segmentation with diffusion models. In *ICLR*. OpenReview.net, 2022.

[2] Emmanuel Asiedu Brempong, Simon Kornblith, Ting Chen, Niki Parmar, Matthias Minderer, and Mohammad Norouzi. Denoising pretraining for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4175–4186, 2022.

[3] Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning, 2022. URL https://arxiv.org/abs/2208.04202.

[4] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat gans on image synthesis. In *NeurIPS*, pages 8780–8794, 2021.

[5] Alexandros Graikos, Nikolay Malkin, Nebojsa Jojic, and Dimitris Samaras. Diffusion models as plug-and-play priors. In *Thirty-Sixth Conference on Neural Information Processing Systems*, 2022. URL https://arxiv.org/pdf/2206.09012.pdf.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning. *Image Recognition*, 7, 2015.

[7] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

[8] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. URL https://arxiv.org/abs/2207.12598.

[9] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

[10] Jonathan Ho, Chitwan Saharia, William Chan, David J. Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *J. Mach. Learn. Res.*, 23:47:1–47:33, 2022.

[11] Emiel Hoogeboom, Victor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. In *ICML*, volume 162 of *Proceedings of Machine Learning Research*, pages 8867–8887. PMLR, 2022.

[12] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[13] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[14] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. In *ICLR*. OpenReview.net, 2021.

[15] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

[16] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[17] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[18] Chong Mou, Xintao Wang, Liangbin Xie, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. *arXiv preprint arXiv:2302.08453*, 2023.

[19] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 8162–8171. PMLR, 2021.

[20] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. GLIDE: towards photorealistic image generation and editing with text-guided diffusion models. In *ICML*, volume 162 of *Proceedings of Machine Learning Research*, pages 16784–16804. PMLR, 2022.

[21] Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. In *CVPR*, pages 10609–10619. IEEE, 2022.

[22] Suman Ravuri and Oriol Vinyals. Classification accuracy score for conditional generative models. *Advances in neural information processing systems*, 32, 2019.

[23] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10674–10685. IEEE, 2022.

[24] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.

[25] Abhishek Sinha, Jiaming Song, Chenlin Meng, and Stefano Ermon. D2C: diffusion-decoding models for few-shot conditional generation. In *NeurIPS*, pages 12533–12548, 2021.

[26] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2256–2265. JMLR.org, 2015. URL http://proceedings.mlr.press/v37/sohl-dickstein15.html.

[27] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*. OpenReview.net, 2021.

[28] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*. OpenReview.net, 2021.

[29] tinyimagenet. Tiny ImageNet. https://www.kaggle.com/c/tiny-imagenet, 2017.

[30] Nurislam Tursynbek and Marc Niethammer. Unsupervised discovery of 3d hierarchical structure with generative diffusion features. *arXiv preprint arXiv:2305.00067*, 2023.

[31] Arash Vahdat and Jan Kautz. NVAE: A deep hierarchical variational autoencoder. In *NeurIPS*, 2020.

[32] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[33] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

[34] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[35] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[36] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023.