Distribution-Aware Data Expansion with Diffusion Models

Haowei Zhu[1], **Ling Yang** [*2], **Jun-Hai Yong**[1], **Hongzhi Yin**[3], **Jiawei Jiang**[4], **Meng Xiao**[5], **Wentao Zhang**[†2], **Bin Wang** [†1]
[1]Tsinghua University [2]Peking University
[3]University of Queensland [4] Wuhan University [5] CNIC, CAS
wentao.zhang@pku.edu.cn wangbins@tsinghua.edu.cn

Abstract

The scale and quality of a dataset significantly impact the performance of deep models. However, acquiring large-scale annotated datasets is both a costly and time-consuming endeavor. To address this challenge, dataset expansion technologies aim to automatically augment datasets, unlocking the full potential of deep models. Current data expansion techniques include image transformation and image synthesis methods. Transformation-based methods introduce only local variations, leading to limited diversity. In contrast, synthesis-based methods generate entirely new content, greatly enhancing informativeness. However, existing synthesis methods carry the risk of distribution deviations, potentially degrading model performance with out-of-distribution samples. In this paper, we propose **DistDiff**, a training-free data expansion framework based on the **dist**ribution-aware **diff**usion model. DistDiff constructs hierarchical prototypes to approximate the real data distribution, optimizing latent data points within diffusion models with hierarchical energy guidance. We demonstrate its capability to generate distribution-consistent samples, significantly improving data expansion tasks. DistDiff consistently enhances accuracy across a diverse range of datasets compared to models trained solely on original data. Furthermore, our approach consistently outperforms existing synthesis-based techniques and demonstrates compatibility with widely adopted transformation-based augmentation methods. Additionally, the expanded dataset exhibits robustness across various architectural frameworks. Our code is available at https://github.com/haoweiz23/DistDiff.

1 Introduction

A substantial number of training samples are essential for unlocking the full potential of deep networks. However, the manual collection and labeling of large-scale datasets are both costly and time-intensive. This makes it difficult to expand data-scarce datasets. Therefore, it is of great value to study how to expand high-quality training data in an efficient and scalable way [6].

Automatic data expansion technology can alleviate the data scarcity problem by augmenting or creating diverse samples, it mitigates the bottleneck associated with limited data, thereby improving model's downstream performance and fostering greater generalization [14, 78]. One simple yet effective strategy is employing image transformation techniques such as cropping, rotation, and erasing to augment samples [59]. Although these methods prove effective and have been widely applied in various fields, their pre-defined perturbations only introduce local variations to the images, thereby falling short in providing a diverse range of content change. In recent

[*]Equal Contribution.
[†]Corresponding author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

times, generative models have gained considerable attention [18, 44, 47, 50, 54, 45], exhibit impressive performance in various areas like image inpainting [36, 53], super-resolution [26, 55], and video generation [25, 41]. Generative models leverage text and image conditions to create images with entirely novel content, harnessing the expansive potential of data expansion [13]. Nevertheless, there is a risk of generating images that deviate from the real data distribution. Therefore, when employing diffusion models for dataset expansion tasks, further research is necessary to ensure a match between synthetic data distributions and real data distributions.

There are several strategies aimed at mitigating the risk of distribution shift, which can be broadly categorized into two groups: training-based and training-free methods. The training-based methods [42, 52, 65] fine-tune pre-trained diffusion models to adapt target dataset, necessitating additional training costs and increasing the likelihood of overfitting on small-scale datasets. Other training-free methods [17, 22] eliminate potentially noisy samples by designing optimizing and filtering strategies, but they still struggle to generate data that conforms to the real data distribution.

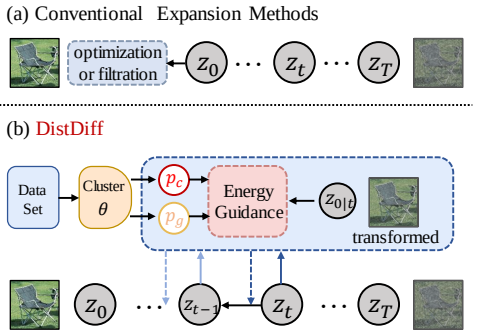


Figure 1: A comparison unveils distinctions between conventional data expansion methods and our innovative distribution-aware diffusion framework, benefiting from hierarchical clustering and multi-step energy guidance.

In this work, we propose a training-free data expansion framework, dubbed **Dist**ribution-Aware **Diff**usion (DistDiff) to optimize generation results. As shown in Figure 1, DistDiff initially approximates the true data distribution using class-level and group-level prototypes obtained through hierarchical clustering. Subsequently, DistDiff utilizes these prototypes to formulate two synergistic energy functions. A residual multiplicative transformation is then applied to the latent data points, enabling the generation of data distinct from the original. Following this, the hierarchical energy guidance process refines intermediate predicted data points, optimizing the diffusion model to generate data samples that are consistent with the underlying distribution. DistDiff ensures fidelity and diversity in the generated samples through distribution-aware energy guidance. Experimental results demonstrate that DistDiff outperforms advanced data expansion techniques, producing better expansion effects and significantly improving downstream model performance. Our contributions can be summarized as follows:

- We introduce a novel diffusion-based data expansion algorithm, named DistDiff, which facilitates distribution-consistent data augmentation without requiring re-training.
- By leveraging hierarchical prototypes to approximate data distribution, we propose an effective distribution-aware energy guidance at both class and group levels in the diffusion sampling process.
- The experimental results illustrate that our DistDiff is capable of generating high-quality samples, surpassing existing image transformation and synthesis methods significantly.

2 Related Work

2.1 Transformation-Based Data Augmentation

Traditional data augmentation techniques [5, 10, 23, 74, 76, 81, 82] typically involve expanding the dataset through distortive transformations, aiming to enhance the model's ability to capture data invariance and mitigate overfitting [59]. For instance, scale invariance is cultivated through random cropping and scaling, while rotation invariance is developed through random rotation and flipping. Region mask-based methods [5, 12, 82] enhance model robustness against target occlusion by strategically obscuring portions of the target area. Interpolation-based methods [23, 74, 76] generate virtual samples by randomly blending content from two images. RandAugment [10] further boosts augmentation effectiveness by sampling from a diverse range of augmentation strategies. However, these methods induce only subtle changes on the original data through transformation, deletion, and blending, leading to a lack of diversity. Moreover, they are predefined and uniformly applied across the entire dataset, which may not be optimal for varying data types or scenarios.

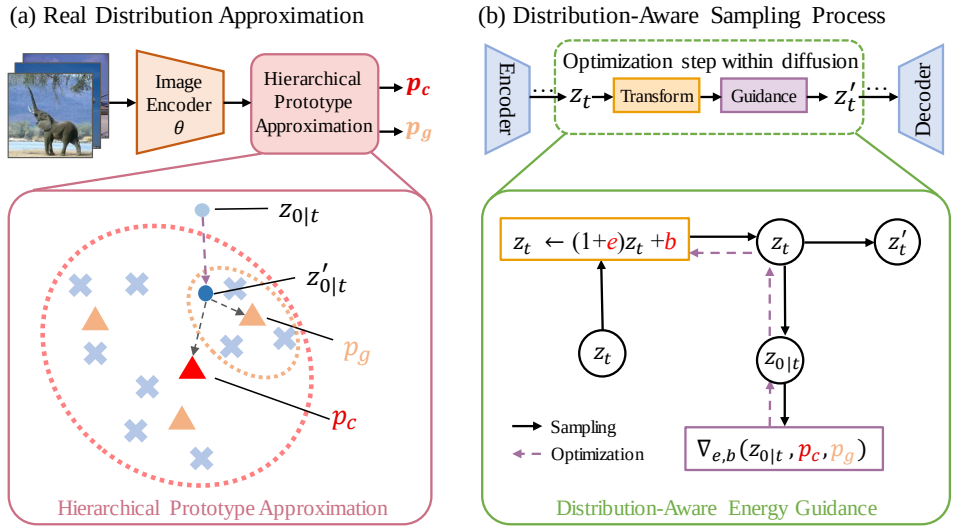


Figure 2: Overview of the DistDiff pipeline. DistDiff enhances the generation process in diffusion models with distribution-aware optimization. It approximates the real data distribution using hierarchical prototypes $\boldsymbol{p}_c$ and $\boldsymbol{p}_g$, optimizing the sampling process through distribution-aware energy guidance. Subsequently, original generated data point $\boldsymbol{z}_t$ is refined for improved alignment with the real distribution.

2.2 Synthesis-Based Data Augmentation

Generative data augmentation aims to leverage generative models to approximate the real data distribution, generating samples with novel content to enhance data diversity. GAN [18] excels at learning data distributions and producing unseen samples in an unsupervised manner [1, 20, 33, 39, 79, 80, 67]. While their efficacy has been demonstrated across diverse downstream tasks, studies indicate that training existing models like ResNet50 [21] on images synthesized by BigGAN [4] yields subpar results compared to training on real images. This disparity in performance can be attributed to the limited diversity and potential domain gap between synthesized samples and real images. Additionally, the training processes of GAN are notoriously unstable, particularly with a low-data regime, and suffer from mode collapse, resulting in a lack of diversity [3, 19, 48]. In contrast, diffusion model-based methods [69] can offer better controllability and superior customization capabilities. Text-to-image models such as Stable Diffusion [50], DALL-E 2 [47] and RPG [71] have demonstrated the creation of compelling high-resolution images [70, 72, 77]. Recently, large-scale text-to-image models have been used for data generation [2, 15, 35, 58, 64, 63]. For example, LECF [22] utilized GLIDE [44] to generate images, filtering low-confidence samples to enhance zero-shot and few-shot image classification performance. SGID [32] leverages image descriptions generated by BLIP [34] to enhance the semantic consistency of generated samples. Feng et al. [17] filters out low-quality samples based on feature similarity between generated and reference images. GIF [78] creates new informative samples through prediction entropy and feature divergence optimization. However, it's crucial to note that datasets generated by existing methods may exhibit distribution shifts, impacting image classification performance significantly. Zhou et al. [83] address this issue by employing diffusion inversion to mitigate distributional shifts. In contrast, we propose a training-free approach, leveraging hierarchical prototypes as optimization targets to guide the generation process, thereby addressing distributional shifts. This approach offers the advantage of avoiding additional computational costs and overfitting issues associated with fine-tuning diffusion models.

3 Method

In this study, we introduce a distribution-aware data expansion framework utilizing Stable Diffusion as a prior generation model. This framework guides the diffusion model based on hierarchical prototype guidance criteria. As illustrated in Figure 2, our DistDiff initially employs an image encoder θ to extract instance features and subsequently derives hierarchical prototypes to approximate the real data distribution. Next, for a given seed image and corresponding text prompts, we extract image's

latent feature and apply stochastic noise to it. Subsequently, in the denoising process, we optimize the latent features using a training-free hierarchical energy guidance process. Our optimization strategy ensures that the generated samples not only match the distribution but also carry new information to enhance model training.

3.1 Task Definition

In the context of a small-scale training dataset, the data expansion task is designed to augment the original dataset $\mathcal{D}_o = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^{n_o}$ with a new set of synthetic samples, referred to as $\mathcal{D}_s = \{\boldsymbol{x}'_i, \boldsymbol{y}'_i\}_{i=1}^{n_s}$. Here $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ represent the sample and its corresponding label, where n_o and n_s respectively denote the original sample quantity and the synthetic sample quantity. The objective is to enhance the performance of a deep learning model trained on both the original dataset $\mathcal{D}_o$ and the expanded dataset $\mathcal{D}_s$ compared to a model trained solely on the original $\mathcal{D}_o$. The crucial aspect lies in ensuring that the generated dataset is highly consistent with the distribution of the original dataset while being as informative as possible.

3.2 Hierarchical Prototypes Approximate Data Distribution

Prototypes have been widely employed in class incremental learning methods to retain information about each class [40, 49]. In this work, we propose two levels of prototypes to capture the original data distribution. Firstly, the class-level prototypes $\boldsymbol{p}_{\mathrm{c}}$ are obtained by averaging feature vectors within the same class. The class vector aggregates high-level statistical information to characterize all samples from the same class as a collective entity. However, as class-level prototypes represent the class feature space as a single vector, potentially reducing informativeness, we further introduce group-level prototypes to capture the structure of the class feature space. Specifically, we divide all samples from the same class into K groups using agglomerative hierarchical clustering, followed by averaging feature vectors within each group to obtain K group prototypes $\boldsymbol{p}_{\mathrm{g}} = \{\boldsymbol{p}_{\mathrm{g}}^1, \boldsymbol{p}_{\mathrm{g}}^2, ..., \boldsymbol{p}_{\mathrm{g}}^K\}$. Instances with similar patterns are grouped together. Transitioning from class-level to group-level, the prototypes encapsulate abstract distribution information of the class at different scales.

Thanks to these hierarchical prototypes, we design two function $\mathcal{D}_{\boldsymbol{\theta}}^{\mathrm{c}}$ and $\mathcal{D}_{\boldsymbol{\theta}}^{\mathrm{g}}$ to evaluate the degree of distribution matching:

$$\mathcal{D}_{\boldsymbol{\theta}}^{\mathrm{c}}(\boldsymbol{x}, \boldsymbol{p}_{\mathrm{c}}) = \|\boldsymbol{\theta}(\boldsymbol{x}) - \boldsymbol{p}_{\mathrm{c}}\|_2, \tag{1}$$

$$\begin{aligned} \mathcal{D}_{\boldsymbol{\theta}}^{\mathrm{g}}(\boldsymbol{x}, \boldsymbol{p}_{\mathrm{g}}) &= \|\boldsymbol{\theta}(\boldsymbol{x}) - \boldsymbol{p}_{\mathrm{g}}^j\|_2, \\ \mathrm{s.t.} j &= \arg\max(\cos(\boldsymbol{\theta}(\boldsymbol{x}), \{\boldsymbol{p}_{\mathrm{g}}^j\}_{j=1}^K)), \end{aligned} \tag{2}$$

where $\boldsymbol{\theta}(\cdot)$ means feature extractor, which could be ResNet [21], CLIP [46] or other deep models.

Note that these two functions evaluate the score of distribution matching from two perspectives. The value will be lower when $\boldsymbol{x}$ is more consistent with the real distribution. As shown in Figure 2 (a), $\mathcal{D}_{\boldsymbol{\theta}}^{\mathrm{c}}$ gauges the distance of sample features to the class center, resulting in low scores for easy samples while high scores for hard samples that are situated at the boundaries of the distribution. On the other hand, $\mathcal{D}_{\boldsymbol{\theta}}^{\mathrm{g}}$ assesses distance from the group-level, offering lower scores for hard samples, while still maintaining relatively high scores for outlier samples. Consequently, these two scores mutually reinforce each other and are indispensable.

3.3 Transform Data Points

Given a reference sample $(\boldsymbol{x}, \boldsymbol{y})$, the pre-trained large-scale diffusion model $\mathcal{G}$ can generate new samples $\boldsymbol{x}'$ with novel content. We formalize this process as $\boldsymbol{x}' = \mathcal{G}(\Phi(\boldsymbol{x}) + \delta)$, where $\Phi(\boldsymbol{x})$ represents the latent feature representation and δ is the perturbation applied to latent features. Drawing inspiration from GIF [78], we introduce residual multiplicative transformation to the latent feature $\boldsymbol{z} = \Phi(\boldsymbol{x})$ using randomly initialized channel-level noise $\boldsymbol{e} \sim \mathcal{U}(0, 1)$ and $\boldsymbol{b} \sim \mathcal{N}(0, 1)$. We impose an ϵ-ball constraint on the transformed feature to control the degree of adjustment within a reasonable range, *i.e.*, $\|\boldsymbol{z} - \tilde{\boldsymbol{z}}\|_\infty \leq \epsilon$, and derive $\tilde{\boldsymbol{z}}$ as follows.

$$\tilde{\boldsymbol{z}} = \mathcal{P}_{z,\epsilon}(\tau(\boldsymbol{z})) = \mathcal{P}_{z,\epsilon}((1 + \boldsymbol{e})\boldsymbol{z} + \boldsymbol{b}), \tag{3}$$

where $\tau(\cdot)$ represents the transformation function and $\mathcal{P}_{z,\epsilon}(\cdot)$ denotes the projection of the transformed feature $\tilde{\boldsymbol{z}}$ onto the ϵ-ball of the original latent feature $\boldsymbol{z}$.

Now, the key challenge lies in optimizing $\boldsymbol{e}$ and $\boldsymbol{b}$ to create new samples that align closely with the real data distribution. Another intuitive approach is to directly optimize latent features instead of performing residual multiplication transformations. However, directly optimizing latent features leads to minimal perturbations, making it challenging to achieve performance gains. We discuss this alternative approach in Section 4.4.

3.4 Distribution-Aware Diffusion Generation

In a typical diffusion sampling process, the model iteratively predicts noise to progressively map the noisy $\boldsymbol{z}_T$ into clean $\boldsymbol{z}_0$. While existing data expansion methods [17, 22, 32] treat the generative model as a black box, focusing on filtering or optimizing the final generated $\boldsymbol{z}_0$. The importance of the intermediate sampling stage is ignored, which plays a crucial role in ensuring data quality, especially as the image begins to take on a stable shape appearance. Diverging from prior approaches, we advocate for intervention at the intermediate denoising step for optimization.

Specifically, we first introduce energy guidance into standard reverse sampling process to optimize the transformation using Equation 4. As our energy guidance step is applied to the transformed data point, the transformed data point $\tilde{\boldsymbol{z}}_t$ is denoted as $\boldsymbol{z}_t$ for simplicity.

$$\begin{aligned} \boldsymbol{e}' &= \boldsymbol{e} - \rho \nabla_{\boldsymbol{e}} \varepsilon(\boldsymbol{z}_t, c), \\ \boldsymbol{b}' &= \boldsymbol{b} - \rho \nabla_{\boldsymbol{b}} \varepsilon(\boldsymbol{z}_t, c), \end{aligned} \tag{4}$$

where ρ is the learning rate and $\varepsilon(\boldsymbol{z}_t, \boldsymbol{c})$ is the energy function measuring the compatibility between the transformed noisy data point $\boldsymbol{z}_t$ and the given condition $\boldsymbol{c}$, representing the real data distribution in this work. Equation 4 guides the sampling process and generates distribution-consistent samples. After that, the optimized $\boldsymbol{z}_t'$ is obtained via Equation 3. However, directly measuring the distance between intermediate results $\boldsymbol{z}_t$ with condition $\boldsymbol{c}$ is impractical due to the difficulty in finding a pre-trained network that provides meaningful guidance when the input is noisy.

To address this issue, we leverage the capability that the diffusion model can predict the noise added to $\boldsymbol{z}_t$, and thus predict a clean data point $\boldsymbol{z}_{0|t}$, as shown in Equation 5. Then, the new energy function $\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{z}_{0|t}, \boldsymbol{c})$ based on the predicted clean data point is constructed to approximate $\varepsilon(\boldsymbol{z}_t, \boldsymbol{c})$.

$$\boldsymbol{z}_{0|t} = \frac{\boldsymbol{z}_t - \sqrt{1-\alpha_t}\psi(\boldsymbol{z}_t, t)}{\sqrt{\alpha_t}}, \tag{5}$$

where α_t represents the noise scale and ψ is the learned denoising network. Finally, we employ hierarchical prototypes $\boldsymbol{p}_{\mathrm{c}}$ and $\boldsymbol{p}_{\mathrm{g}}$ as conditions to construct our energy guidance in the following manner:

$$\begin{aligned} \boldsymbol{e}' &= \boldsymbol{e} - \rho \nabla_{e} (\mathcal{D}^{\mathrm{c}}_{\boldsymbol{\theta}}(\boldsymbol{z}_{0|t}, \boldsymbol{p}_{\mathrm{c}}) + \mathcal{D}^{\mathrm{g}}_{\boldsymbol{\theta}}(\boldsymbol{z}_{0|t}, \boldsymbol{p}_{\mathrm{g}})), \\ \boldsymbol{b}' &= \boldsymbol{b} - \rho \nabla_{b} (\mathcal{D}^{\mathrm{c}}_{\boldsymbol{\theta}}(\boldsymbol{z}_{0|t}, \boldsymbol{p}_{\mathrm{c}}) + \mathcal{D}^{\mathrm{g}}_{\boldsymbol{\theta}}(\boldsymbol{z}_{0|t}, \boldsymbol{p}_{\mathrm{g}})). \end{aligned} \tag{6}$$

Unlike existing methods that exclusively optimize the final sampling result $\boldsymbol{z}_0$, our approach focuses on optimizing intermediate denoising steps within the sampling process. The detailed algorithm is shown in Appendix C. This novel strategy leads to substantial improvements in optimization results and will be further explored in Section 4.4.

4 Experiments

4.1 Experimental Setups

Datasets We assess the performance of DistDiff across six image classification datasets, encompassing diverse tasks such as general object classification (Caltech-101 [16], CIFAR100-Subset [31], ImageNette [27]), fine-grained classification (Cars [30]), textual classification (DTD [8]) and medical imaging (PathMNIST [68]). More details are provided in Appendix B.1.

Compared Methods We conduct a comparative analysis between DistDiff and conventional image transformation methods, as well as diffusion-based expansion methods. Traditional image transformation techniques considered in the comparison comprise AutoAugment [9], RandAugment [10], Random Erasing [82], GridMask [5], and interpolation-based techniques like MixUp [76] and CutMix

[74]. For generative-based methods, we include the direct application of stable diffusion for data expansion, as well as the most recent state-of-the-art method, Stable Diffusion 1.4 [50], LECF [22], GIF-SD [78]. The implementation details of these techniques are provided in Appendix B.3 and B.4.

4.2 Implementation Details

In our experimental setup, we implement DistDiff based on Stable Diffusion 1.4 [50]. The images created by Stable Diffusion have a resolution of 512×512 for all datasets. Throughout the diffusion process, we employ the DDIM [60] sampler for a 50-step latent diffusion, with hyper-parameters for noise strength set at 0.5 and classifier free guidance scale at 7.5. The ϵ in Equation 3 is 0.2 by default. We use a ResNet-50 [21] model trained from scratch on the original datasets as our guidance model. We assign $K = 3$ to each class when constructing group-level prototypes, the learning rate ρ is 10.0, and optimization step M is set to 20 unless specified otherwise. After expansion, we concatenate the original dataset with synthetic data to create expanded datasets. We then train the classification model from random initialization for 100 epochs using these expanded datasets. During model training, we process images through random cropping to 224×224 using random rotation and random horizontal flips. Our optimization strategy involves using the SGD optimizer with a momentum of 0.9, and cosine decay with an initial learning rate of 0.1. All results are averaged over three runs with different random seeds. More implementation details can be found in the Appendix B.2.

4.3 Main Results

Comparison with Synthesis-Based Methods To investigate the effectiveness of methods for generating high-quality datasets for downstream classification model training, we initially compare our method, DistDiff, with existing synthesis-based methods on Caltech-101in terms of classification performance. Figure 3 highlights the superiority of our method over state-of-the-art techniques. Compared to the original stable diffusion, DistDiff exhibits an average improvement of 6.25%, illustrating that our method retains more distribution-aligned information from the original datasets. Additionally, GIF-SD [78], which uses a pre-trained CLIP [46] model to enhance class-maintained information and employs KL-divergence to encourage batch-wise sample diversity, is also surpassed by our method by 5.46% in accuracy. This can be attributed to DistDiff guiding the generation process from the distribution level with hierarchical prototypes, providing better optimization signals.

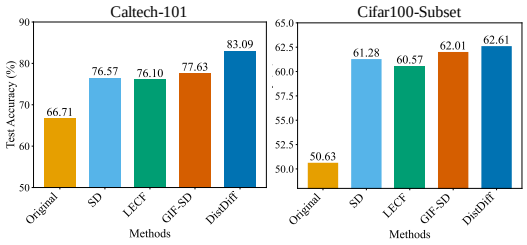


Figure 3: Our method outperforms state-of-the-art data expansion methods when trained on expanded datasets, underscoring the importance of a high-quality generator in training a classifier.

We also evaluated DistDiff against LECF [22], which enhances language prompts and filters samples with low confidence. We use LE enhanced to synthesize $5\times$ synthetic datasets and filter with different thresholds. We assessed multiple Clip Filtering strengths in LECF and found that LECF did not achieve better performance compared to the original Stable Diffusion. This is due to our SD baseline already generates high-quality samples, and the additional filtering post-process may lead to data loss. Besides, both LECF and GIF-SD use auxiliary models to filter/guide the diffusion generation results. However, there are two main strengths of our DistDiff compared to these methods. First, DistDiff generates images end-to-end without requiring filter-based postprocessing. Second, our method is not sensitive to the classification performance of the auxiliary model, which is proved in the Appendix D.2. This suggests that DistDiff is simpler and more generalizable for data expansion tasks.

Comparison with Transformation-Based Augmentation Methods In Table 1, we present a comparison of our methods with widely adopted data augmentation techniques for Caltech-101image classification. Our DistDiff method surpasses transformation-based augmentation methods by introducing a broader range of new content into images. Additionally, we demonstrate the compatibility of our approach with transformation-based data augmentation methods, leading to further improvements.

Scaling in Number of Data We evaluate the scalability of our approach by assessing its advantages in classification model training across four datasets. We compare the performance of DistDiff with the original real dataset and strong augmentation method AutoAug [9] with varying numbers of

Table 1: Comparison of transformation-based augmentation methods on Caltech-101. Our approach, combined with default augmentation (crop, flip, and rotate), consistently outperforms existing advanced transform-based methods and can be further improved by combining these techniques.

	Default	AutoAug	RandAug	Random Erasing	GridMask	MixUp	CutMix
Original	66.71	74.34	74.07	74.22	73.88	78.64	70.13
DistDiff	83.38	82.93	83.21	83.05	83.48	81.06	85.27

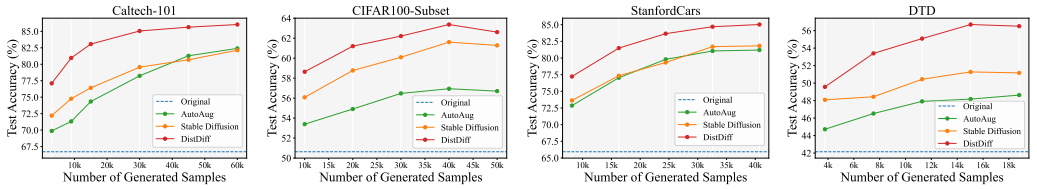


Figure 4: Performance comparison across different scale data sizes. Our method demonstrates significant improvements in classification model performance in both low-data and large-scale data scenarios, outperforming the transformation method AutoAug and the synthesized method Stable Diffusion 1.4.

generated examples, as depicted in Figure 4. As the data expansion scale increases, the corresponding improvement in accuracy also enlarges. The accuracy on Caltech-101achieved with our $5\times$ expansion surpasses even the $20\times$ expanded dataset obtained by AutoAug and diffusion baseline. This indicates that DistDiff exhibits superior efficiency in data expansion compared to existing methods.

Versatility to Various Architectures We conduct an in-depth assessment of the expanded datasets generated by DistDiffacross four distinct backbones: ResNet-50 [21], ResNeXt-50 [66], WideResNet-50 [75], and MobileNetv2 [56]. These backbones are trained from scratch on $5\times$ expanded Caltech-101dataset by DistDiff. The results presented in Table 3 affirm that our innovative methodology is effective and versatile across a spectrum of architectures.

Comparison with Stronger Classification Models As we know, data expansion is typically applied in scenarios with data scarcity. However, if we use models pre-trained on large-scale datasets, the performance on the original training set can be significantly enhanced. In such cases, does our expanded dataset still offer improvements? To validate our method, we fine-tuned a ResNet-50 model pre-trained on ImageNet-1k [11] for ImageNette, Caltech-101, and StanfordCars, and a LAION [57] pre-trained CLIP-ViT-B32 [6] model for PathMNIST. As shown in Table 2, the model achieved a high accuracy of 99.4% on ImageNette, which is a subset of its pretrained datasets. Further data expansion resulted in a slight decrease in performance. Similarly, on the general image dataset Caltech-101, which shares significant overlap with ImageNet data, our method demonstrated only slight improvement. However, on the more challenging fine-grained dataset StanfordCars, our method demonstrated obvious 3.56% accuracy improvement. For the medical image dataset PathMNIST, which exhibits a significantly different distribution, using DistDiff for data expansion effectively boosted classification performance by 5.18%. This highlights the importance of scaling up data when transferring pre-trained models to downstream tasks that exhibit significant distribution shifts.

Table 2: Comparison of using stronger pre-trained baseline models. On ImageNette [28] , Caltech-101 [16], and StanfordCars [30] datasets, we employ an ImageNet-1k [11] pre-trained ResNet-50 [21] model. For the PathMNIST [68] dataset, we fine-tune using the stronger CLIP-ViT-B/32 baseline.

Dataset	ImageNette	Caltech-101	StanfordCars	PathMNIST
Original	99.40	96.87	87.61	84.29
Expanded $5\times$ by SD	98.51 (-0.89)	96.91 $(+0.04)$	90.19 $(+2.58)$	86.81 $(+2.52)$
Expanded $5\times$ by DistDiff	99.30 (-0.10)	97.00 $(+0.13)$	91.17 $(+3.56)$	89.47 $(+5.18)$

Qualitative Analysis In addition to the quantitative experiment results, we also gain a more intuitive understanding of the diverse changes facilitated by our method through visualization of the generated results. As shown in Figure 5, the images generated using our distribution-aware guidance approach exhibit high fidelity and diverse synthetic changes, including object texture, background, and color contrast. More visualization results can be found in Appendix D.3.

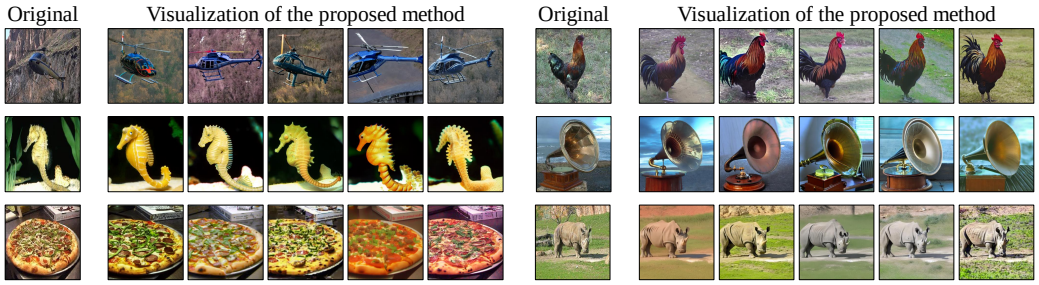


Figure 5: The visualization of synthetic samples generated by our method, showcasing high fidelity, diversity, and alignment with the original data distribution.

4.4 Ablation Study

Hierarchical Prototypes We delved further into how each component of DistDiff impacts its data expansion performance. As depicted in Table 4, utilizing both $\boldsymbol{p}_{\mathrm{c}}$ and $\boldsymbol{p}_{\mathrm{g}}$ contributes to enhancing the model's expansion performance, showcasing their ability to optimize the generated sample distribution at the class-level and group-level, respectively. Moreover, combining $\boldsymbol{p}_{\mathrm{c}}$ and $\boldsymbol{p}_{\mathrm{g}}$ results in a further performance improvement, validating the effectiveness of integrating representations from different hierarchical levels. Additionally, with the introduction of our approach, the Fréchet Inception Distance (FID) values notably decrease, indicating that our proposed FID effectively optimizes the model to generate samples more aligned with the real distribution, thereby reducing the domain gap between the generated dataset and the real dataset.

Table 3: Performance comparison of models trained on original Caltech-101 datasets and 5x expanded datasets by DistDiff.

Backbone	Original	DistDiff
ResNet-50 [21]	66.71	83.09
ResNeXt-50 [66]	67.60	83.75
WideResNet-50 [75]	66.51	83.51
MobileNetV2 [56]	74.39	83.85

Table 4: Comparison of accuracy and FID in expanding Caltech-101 by $5\times$, with and without hierarchical prototypes in DistDiff.

$\boldsymbol{p}_{\mathrm{c}}$	$\boldsymbol{p}_{\mathrm{g}}$	Accuracy ↑	FID-3K ↓
		76.57 ± 0.35	72.56
	✓	82.70 ± 0.07	68.82
✓		82.84 ± 0.54	68.66
✓	✓	83.09 ± 0.11	67.72

Augmentation within Diffusion In the application of energy guidance, perturbation is introduced at the M-th step, and these subsequent predicted data points are optimized. Theoretically, optimizing predictions at different stages yields distinct effects. Our exploration focused on the optimization step M, and the experimental results are illustrated in Table 5. When M is small, indicating optimization at a later stage (i.e., the refinement stage), the change in generated results is already minimal, resulting in relatively consistent optimization outcomes. Conversely, when M increases, corresponding to optimization at an intermediate stage (i.e., the semantic stage), the generated results are in the stage of forming semantics and exhibit significant changes. Hence, this stage plays a crucial role in determining the final generated results. Furthermore, there is a decline in performance during the early chaos stage (M=25), as the data points in this initial phase are too chaotic to establish an optimal target for optimization. We observed that achieving higher data expansion performance is possible when optimized in the semantic stage, with optimal results obtained when M=20.

Table 5: Comparison of optimization in different phases.

M	Accuracy
1	81.54 ± 0.32
10	82.21 ± 0.22
20	82.36 ± 0.05
25	82.11 ± 0.55

Table 6: Ablation of the number K of $\boldsymbol{p}_{\mathrm{g}}$ in DistDiff.

K	Accuracy
2	82.69 ± 0.51
3	83.09 ± 0.11
4	83.08 ± 0.13
5	83.08 ± 0.21

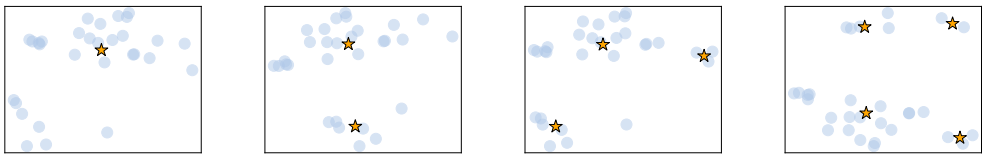
Table 7: Results of introducing more optimization steps.

Step	Accuracy
1	82.54 ± 0.54
2	82.94 ± 0.43
3	82.77 ± 0.19
4	82.55 ± 0.30

More Optimization Steps Furthermore, a natural idea arises regarding the potential improvement in effectiveness through the optimization of more optimization steps. Therefore, we further explored increasing the number of steps in the semantic stages. As shown in Table 7, increasing the number of optimization steps in semantic stages enhances performance. However, further increases in optimization steps lead to a decline in performance. This can be attributed to excessive optimization strength in energy guidance, which causes data distortion.

Compared with Direct Guidance on Latent Point We evaluate our transform guidance strategy against an alternative strategy that directly guides the latent data points while ignoring the residual transform preprocess, a method found useful in previous works [24, 73]. We initially conducted a grid search for this alternative strategy to find the optimal learning rate, ρ (i.e., [0.1, 1, 10, 20]), and guide step, M (i.e., [1, 10, 20]). We found the best result of 76.77% was achieved with $\rho = 10$ and $M = 10$, which is only slightly better than the original Stable Diffusion but lags behind our DistDiff, which achieved 83.19%. This indicates that applying the residual multiplicative transformation to the latent feature offers more optimization potential.

Determination of Group-Level Prototype Number K The determination of the number K of group-level prototypes is crucial for accurately approximating the real data distribution. In Table 6, we compare the outcomes associated with varying numbers of prototypes. The results highlight that the optimal number of prototypes is found at $K = 3$. We posit that an insufficient number of prototypes may impede the characterization of the real distribution, leading to diminished performance. Conversely, an excessive number of prototypes may lead to overfitting of noisy sample points, also resulting in suboptimal performance. Furthermore, we present a visualization analysis of group-level prototypes in Figure 6. The visual representation demonstrates that an appropriate number of group-level prototypes can effectively cover the real distribution space, aligning with the underlying motivation of our DistDiff.



(a) $K = 1$ (b) $K = 2$ (c) $K = 3$ (d) $K = 4$

Figure 6: The visualization of group-level prototypes alongside original sample features. Here $\bullet$ is the sample point and $\star$ is group-level prototype. By selecting an appropriate number K, these prototypes effectively span the feature space, providing an approximation of the real data distribution.

Computational Efficiency Our DistDiff is not only training-free but also highly efficient in processing. As illustrated in Figure 2, DistDiff introduces only a few optimization steps in the original diffusion process. We analyze the time costs of our methods. Stable Diffusion generates per sample in 12.65 seconds on average, while our DistDiff achieves the same in 13.13 seconds, with the increased time costs being negligible. We can conclude that DistDiff achieves a notable improvement over stable diffusion models, with only a slight increase in computation costs.

5 Conclusion

This paper presents DistDiff, a distribution-aware data expansion method employing a stable diffusion model for data expansion. The proposed method optimizes the diffusion process to align the synthesized data distribution with the real data distribution. Specifically, DistDiff constructs hierarchical prototypes to effectively represent the real data distribution and refines intermediate features within the sampling process using energy guidance. We evaluate our method through extensive experiments on six datasets, showcasing its superior performance over existing methods.

6 Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 62072271.

References

[1] Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017.

[2] Shekoofeh Azizi, Simon Kornblith, Chitwan Saharia, Mohammad Norouzi, and David J Fleet. Synthetic data from diffusion models improves imagenet classification. *arXiv preprint arXiv:2304.08466*, 2023.

[3] Hritik Bansal and Aditya Grover. Leaving reality to imagination: Robust classification via generated datasets. *arXiv preprint arXiv:2302.02503*, 2023.

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[5] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. Gridmask data augmentation. *arXiv preprint arXiv:2001.04086*, 2020.

[6] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2829, June 2023.

[7] Jaemin Cho, Abhay Zala, and Mohit Bansal. Dall-eval: Probing the reasoning skills and social biases of text-to-image generation models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3043–3054, 2023.

[8] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3606–3613, 2014.

[9] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

[10] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 702–703, 2020.

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

[12] Terrance DeVries and Graham W. Taylor. Improved regularization of convolutional neural networks with cutout, 2017.

[13] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[14] Lisa Dunlap, Alyssa Umino, Han Zhang, Jiezhi Yang, Joseph E Gonzalez, and Trevor Darrell. Diversify your vision datasets with automatic diffusion-based augmentation. *arXiv preprint arXiv:2305.16289*, 2023.

[15] Lisa Dunlap, Alyssa Umino, Han Zhang, Jiezhi Yang, Joseph E Gonzalez, and Trevor Darrell. Diversify your vision datasets with automatic diffusion-based augmentation. *Advances in Neural Information Processing Systems*, 36, 2024.

[16] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pp. 178–178. IEEE, 2004.

[17] Chun-Mei Feng, Kai Yu, Yong Liu, Salman Khan, and Wangmeng Zuo. Diverse data augmentation with diffusions for effective test-time prompt tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2704–2714, 2023.

[18] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11): 139–144, 2020.

[19] Sven Gowal, Sylvestre-Alvise Rebuffi, Olivia Wiles, Florian Stimberg, Dan Andrei Calian, and Timothy A Mann. Improving robustness using generated data. *Advances in Neural Information Processing Systems*, 34:4218–4233, 2021.

[20] Swaminathan Gurumurthy, Ravi Kiran Sarvadevabhatla, and R Venkatesh Babu. Deligan: Generative adversarial networks for diverse and limited data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 166–174, 2017.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[22] Ruifei He, Shuyang Sun, Xin Yu, Chuhui Xue, Wenqing Zhang, Philip Torr, Song Bai, and Xiaojuan Qi. Is synthetic data from generative models ready for image recognition? *arXiv preprint arXiv:2210.07574*, 2022.

[23] Dan Hendrycks, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. *arXiv preprint arXiv:1912.02781*, 2019.

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

[25] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet, and Tim Salimans. Imagen video: High definition video generation with diffusion models. 2022.

[26] 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(1), jan 2022.

[27] Jeremy Howard. A smaller subset of 10 easily classified classes from imagenet, and a little more french, 2020. *URL https://github. com/fastai/imagenette*.

[28] Jeremy Howard. A smaller subset of 10 easily classified classes from imagenet, and a little more french, 2019. URL https://github.com/fastai/imagenette.

[29] Khawar Islam, Muhammad Zaigham Zaheer, Arif Mahmood, and Karthik Nandakumar. Diffusemix: Label-preserving data augmentation with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 27621–27630, 2024.

[30] Jonathan Krause, Jia Deng, Michael Stark, and Li Fei-Fei. Collecting a large-scale dataset of fine-grained cars. 2013.

[31] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[32] Bohan Li, Xinghao Wang, Xiao Xu, Yutai Hou, Yunlong Feng, Feng Wang, and Wanxiang Che. Semantic-guided image augmentation with pre-trained models. *arXiv preprint arXiv:2302.02070*, 2023.

[33] Daiqing Li, Huan Ling, Seung Wook Kim, Karsten Kreis, Sanja Fidler, and Antonio Torralba. Bigdataset-gan: Synthesizing imagenet with pixel-wise annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21330–21340, 2022.

[34] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pp. 12888–12900. PMLR, 2022.

[35] Ziyi Li, Qinye Zhou, Xiaoyun Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. Guiding text-to-image diffusion model towards grounded generation. *arXiv preprint arXiv:2301.05221*, 3(6):7, 2023.

[36] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11461–11471, 2022.

[37] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference, 2023.

[38] Siwei Lyu. Deepfake detection: Current challenges and next steps. pp. 1–6, 2020.

[39] Giovanni Mariani, Florian Scheidegger, Roxana Istrate, Costas Bekas, and Cristiano Malossi. Bagan: Data augmentation with balancing gan. *arXiv preprint arXiv:1803.09655*, 2018.

[40] Marc Masana, Xialei Liu, Bartłomiej Twardowski, Mikel Menta, Andrew D Bagdanov, and Joost Van De Weijer. Class-incremental learning: survey and performance evaluation on image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5513–5533, 2022.

[41] Kangfu Mei and Vishal Patel. Vidm: Video implicit diffusion models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(8):9117–9125, Jun. 2023.

[42] Taehong Moon, Moonseok Choi, Gayoung Lee, Jung-Woo Ha, and Juho Lee. Fine-tuning diffusion models with limited data. In *NeurIPS 2022 Workshop on Score-Based Methods*, 2022.

[43] Ranjita Naik and Besmira Nushi. Social biases through the text-to-image generation lens. In *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 786–808, 2023.

[44] Alex 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. *arXiv preprint arXiv:2112.10741*, 2021.

[45] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. In *ACM SIGGRAPH 2023 Conference Proceedings*, pp. 1–11, 2023.

[46] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

[47] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

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

[49] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 2001–2010, 2017.

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

[51] Candace Ross, Boris Katz, and Andrei Barbu. Measuring social biases in grounded vision and language embeddings. *arXiv preprint arXiv:2002.08911*, 2020.

[52] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22500–22510, 2023.

[53] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*. Association for Computing Machinery, 2022.

[54] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.

[55] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4713–4726, 2023. doi: 10.1109/TPAMI.2022.3204461.

[56] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

[57] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022.

[58] Jordan Shipard, Arnold Wiliem, Kien Nguyen Thanh, Wei Xiang, and Clinton Fookes. Diversity is definitely needed: Improving model-agnostic zero-shot classification via stable diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 769–778, 2023.

[59] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

[60] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[61] Brandon Trabucco, Kyle Doherty, Max Gurinas, and Ruslan Salakhutdinov. Effective data augmentation with diffusion models. *arXiv preprint arXiv:2302.07944*, 2023.

[62] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, Steven Liu, William Berman, Yiyi Xu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models, 2023. URL https://github.com/huggingface/diffusers.

[63] Weijia Wu, Yuzhong Zhao, Mike Zheng Shou, Hong Zhou, and Chunhua Shen. Diffumask: Synthesizing images with pixel-level annotations for semantic segmentation using diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1206–1217, 2023.

[64] Weijia Wu, Yuzhong Zhao, Hao Chen, Yuchao Gu, Rui Zhao, Yefei He, Hong Zhou, Mike Zheng Shou, and Chunhua Shen. Datasetdm: Synthesizing data with perception annotations using diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

[65] Enze Xie, Lewei Yao, Han Shi, Zhili Liu, Daquan Zhou, Zhaoqiang Liu, Jiawei Li, and Zhenguo Li. Difffit: Unlocking transferability of large diffusion models via simple parameter-efficient fine-tuning. *arXiv preprint arXiv:2304.06648*, 2023.

[66] 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*, pp. 1492–1500, 2017.

[67] Austin Xu, Mariya I Vasileva, Achal Dave, and Arjun Seshadri. Handsoff: Labeled dataset generation with no additional human annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7991–8000, 2023.

[68] Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10(1):41, 2023.

[69] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys*, 56(4):1–39, 2023.

[70] Ling Yang, Jingwei Liu, Shenda Hong, Zhilong Zhang, Zhilin Huang, Zheming Cai, Wentao Zhang, and Bin Cui. Improving diffusion-based image synthesis with context prediction. *Advances in Neural Information Processing Systems*, 36, 2024.

[71] Ling Yang, Zhaochen Yu, Chenlin Meng, Minkai Xu, Stefano Ermon, and Bin Cui. Mastering text-to-image diffusion: Recaptioning, planning, and generating with multimodal llms. *arXiv preprint arXiv:2401.11708*, 2024.

[72] Ling Yang, Zhilong Zhang, Zhaochen Yu, Jingwei Liu, Minkai Xu, Stefano Ermon, and Bin CUI. Cross-modal contextualized diffusion models for text-guided visual generation and editing. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=nFMS6wF2xq.

[73] Jiwen Yu, Yinhuai Wang, Chen Zhao, Bernard Ghanem, and Jian Zhang. Freedom: Training-free energy-guided conditional diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 23174–23184, 2023.

[74] 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*, pp. 6023–6032, 2019.

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

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

[77] Xinchen Zhang, Ling Yang, Yaqi Cai, Zhaochen Yu, Jiake Xie, Ye Tian, Minkai Xu, Yong Tang, Yujiu Yang, and Bin Cui. Realcompo: Dynamic equilibrium between realism and compositionality improves text-to-image diffusion models. *arXiv preprint arXiv:2402.12908*, 2024.

[78] Yifan Zhang, Daquan Zhou, Bryan Hooi, Kai Wang, and Jiashi Feng. Expanding small-scale datasets with guided imagination. *arXiv preprint arXiv:2211.13976*, 2022.

[79] Yuxuan Zhang, Huan Ling, Jun Gao, Kangxue Yin, Jean-Francois Lafleche, Adela Barriuso, Antonio Torralba, and Sanja Fidler. Datasetgan: Efficient labeled data factory with minimal human effort. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10145–10155, 2021.

[80] Bo Zhao and Hakan Bilen. Synthesizing informative training samples with gan. *arXiv preprint arXiv:2204.07513*, 2022.

[81] Long Zhao, Ting Liu, Xi Peng, and Dimitris Metaxas. Maximum-entropy adversarial data augmentation for improved generalization and robustness. *Advances in Neural Information Processing Systems*, 33: 14435–14447, 2020.

[82] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 13001–13008, 2020.

[83] Yongchao Zhou, Hshmat Sahak, and Jimmy Ba. Training on thin air: Improve image classification with generated data. *arXiv preprint arXiv:2305.15316*, 2023.

A Limitations and Societal Impacts

Limitations Although DistDiff achieves better FID scores and enhances classifier performance without the need for training, it may incur additional computation time, which accumulates as data scales up. Additionally, our method requires an extra guide model, increasing development costs. Integrating a fast sampler [37] and lightweight guide model represents a promising direction for maximizing the effectiveness of diffusion-based data expansion methods in classifier training.

Societal Impacts Generative models [47, 50] can significantly reduce the costs associated with manual data collection and annotation. Our approach builds an efficient method that enhances the distributional consistency of these generative models for downstream tasks without requiring training, thereby improving performance on downstream classifiers. This capability can assist organizations and researchers with limited data resources in developing more effective machine learning models.

However, since generative models are pre-trained on vast, diverse vision-language datasets from the internet, these data may contain social biases and stereotypes [7, 43, 51], leading to discriminatory generated outputs. Therefore, integrating mechanisms to detect and mitigate biases is crucial. Nevertheless, our method guides the generation process decisions based on task-specific hierarchical prototypes, fostering AI systems that better align with downstream task distributions and exhibit fewer biases.

Another potential concern is the misuse of generated data, which could be exploited for malicious purposes such as deepfakes [38], resulting in misinformation dissemination and adverse societal impacts. Proper constraints on the proliferation and correct usage of generative models are crucial. This necessitates the establishment of relevant regulations and guidelines to ensure the responsible development and utilization of synthetic data models.

B More Implementation Details

B.1 Datasets

Table 8 provides the detailed statistics of six experimental datasets, including Caltech-101 [16], CIFAR100-Subset [31], StandardCars [30], ImageNette [28], DTD [8], and PathMNIST[68]. Cifar100-Subset and PathMNIST are subsets that randomly sampled from the original trainset with 100 samples per class. Specifically, our evaluation encompasses datasets that span diverse scenarios, incorporating generic objects, fine-grained categories, and textural images. These diverse datasets enable a comprehensive assessment of both the robustness and effectiveness of our proposed methods. Additionally, we also provide the text template employed in the conditional diffusion process in Table 9.

Table 8: Summary of our six experimental datasets.

NAME	CLASSES	SIZE (TRAIN / TEST)	DESCRIPTION
CALTECH-101	100	3000 / 6085	RECOGNITION OF GENERIC OBJECTS
CIFAR100-SUBSET	100	10000 / 10000	RECOGNITION OF GENERIC OBJECTS
STANDARDCARS	196	8144 / 8041	FINE-GRAINED CLASSIFICATION OF CARS
IMAGENETTE	10	9469 / 3925	RECOGNITION OF GENERIC OBJECTS
DTD	47	3760 / 1880	TEXTURE CLASSIFICATION
PATHMNIST	9	900 / 7180	RECOGNITION OF COLON PATHOLOGY IMAGE

B.2 Our Method

We implement our methods using the PyTorch framework with Python 3.10.6. We utilize the diffuser [62] to implement our base diffusion models. For all datasets, we generate images with a noise strength of 0.5, except for the medical image dataset PathMNIST, where we use a noise strength of 0.2. The guidance step M in our method is 20 by default, except for 10 for PathMNIST. The half-precision floating-point is used in the generation process to reduce memory costs. The generation, training, and evaluation processes are conducted on a single GeForce RTX 3090 GPU. We report

Table 9: Text templates for six experimental datasets.

NAME	TEMPLATE
CALTECH-101	"A PHOTO OF A [CLASS]."
CIFAR100-SUBSET	"A PHOTO OF A [CLASS]."
STANDARDCARS	"A PHOTO OF A [CLASS] CAR."
IMAGENETTE	"A PHOTO OF A [CLASS]."
DTD	"[CLASS] TEXTURE."
PATHMNIST	"A COLON PATHOLOGICAL IMAGE OF [CLASS]."

the best test accuracy averaged over three runs and calculate the FID metric using 3000 samples. In order to reduce generation time costs, we pre-compute and save the latent embeddings for all training samples. We use Stable Diffusion 1.4 as our base generation model[3]. We follow previous data expansion methods by expanding each sample certain ratios. The sampling category distribution is consistent with the original dataset category distribution.

B.3 Synthesis-Based Augmentation Contenders

Due to benchmark and training setting differences in existing works [78, 2, 33], in order to fairly compare existing methods with ours, we reproduced two state-of-the-art generation-based methods: GIF-SD [78] and LECF [2] on our benchmarks and base diffusion model. For GIF-SD, we used a pretrained CLIP-ViT-B32 model to facilitate its class-maintained optimization, and a batch size of 4 to facilitate its KL-divergence based diverse sampling. For LECF, we generated 200 prompts for each class prompt, and generated samples with sizes equal to our method's, filtering these samples with a specified threshold using a pretrained ResNet50-CLIP model. As shown in Section 4.3, our methods surpass these contenders.

B.4 Transformation-Based Augmentation Contenders

We outline the transformation-based augmentation methods compared in our experiments. The effectiveness of all these methods are evaluated on the same datasets and with the same configuration (e.g., learning rate, epochs, batch size, etc.) as our method.

Default: We use random crop, random horizontal flip, and random rotation with a 15-degree angle as default augmentation strategies.

AutoAug [9]: We employ the AutoAugment function in PyTorch with the widely used ImageNet policy, which includes a diverse range of color and shape transformations. During training, one transformation strategy is randomly selected and applied.

RandAug [10]: Similar to AutoAug, we randomly choose two operations from a predefined policy list and apply them to the training samples.

Random Erasing [82]: We use the random erase function in PyTorch for Random Erasing. This function randomly selects a rectangular region within an image and erases its pixels with a 50% probability. The proportion of erased area relative to the input image ranges from 0.02 to 0.33, and the aspect ratio of the erased area ranges from 0.3 to 3.3.

GridMask [5]: GridMask is implemented with its official configuration. The probability of applying GridMask linearly increases from 0 to 0.8 as training epochs progress up to the 80th epoch, after which it remains constant until 100 epochs.

MixUp [76]: Synthetic interpolated images are generated within each data batch. We sample the interpolation strength from a beta distribution (beta = 1). The loss function is also adjusted accordingly.

CutMix [74]: Similar to MixUp, CutMix replaces specified regions in original images with input from another image. The loss function is modified accordingly.

[3]We use the pretrained weights "CompVis/stable-diffusion-v1-4" from Hugging Face. https://huggingface.co/CompVis/stable-diffusion-v1-4

C Pseudocode

We present the pseudocode for our algorithm in Algorithm 1, illustrating the hierarchical energy guidance within the diffusion sampling process.

Algorithm 1 Optimization Process of our proposed DistDiff

Input: Hierarchical prototypes $\boldsymbol{p}_{\mathrm{c}}$ and $\boldsymbol{p}_{\mathrm{g}}$; Latent data point $\boldsymbol{z}_T$ which is encoded from the sample $\boldsymbol{x}$; Optimization step M; Pre-defined parameters β_t; Perturbation constraint ϵ; Pre-trained feature extractor $\boldsymbol{\theta}$; Denoising network ψ.
for $t = T-1, \ldots, 0$ **do**
  $\delta \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 0$, else $\delta = 0$.
  $\boldsymbol{z}_t = (1 + \frac{1}{2}\beta_{t+1})\boldsymbol{z}_{t+1} + \beta_{t+1}\psi(\boldsymbol{z}_{t+1}, t+1) + \sqrt{\beta_{t+1}}\delta$
  if $t = M$ **then**
    Initialize $\boldsymbol{e} \sim \mathcal{U}(0,1)$, $\boldsymbol{b} \sim \mathcal{N}(0,1)$.
    $\boldsymbol{z}_t = (1+\boldsymbol{e})\boldsymbol{z}_t + \boldsymbol{b}$
    $\boldsymbol{z}_{0|t} = (\boldsymbol{z}_t - \sqrt{1-\alpha_t}\psi(\boldsymbol{z}_t, t))/\sqrt{\alpha_t}$
    $\mathcal{D}^{\mathrm{c}}_{\boldsymbol{\theta}}(\boldsymbol{z}_{0|t}, \boldsymbol{p}_{\mathrm{c}}) = \|\boldsymbol{\theta}(\boldsymbol{z}_{0|t}) - \boldsymbol{p}_{\mathrm{c}}\|_2$
    $\mathcal{D}^{\mathrm{g}}_{\boldsymbol{\theta}}(\boldsymbol{z}_{0|t}, \boldsymbol{p}_{\mathrm{g}}) = \|\boldsymbol{\theta}(\boldsymbol{z}_{0|t}) - \boldsymbol{p}_{\mathrm{g}}\|_2$
    $\boldsymbol{g}_t = \mathcal{D}^{\mathrm{c}}_{\boldsymbol{\theta}}(\boldsymbol{z}_{0|t}, \boldsymbol{p}_{\mathrm{c}}) + \mathcal{D}^{\mathrm{g}}_{\boldsymbol{\theta}}(\boldsymbol{z}_{0|t}, \boldsymbol{p}_{\mathrm{g}})$
    update $\boldsymbol{e}', \boldsymbol{b}' \leftarrow \arg\min_{e,b} \boldsymbol{g}_t$ (Equation 6)
    update $\boldsymbol{z}_t \leftarrow \boldsymbol{z}'_t = \mathcal{P}_{z,\epsilon}((1+\boldsymbol{e}')\boldsymbol{z}_t + \boldsymbol{b}')$
  end if
end for
Output: $\boldsymbol{z}_0$.

D More Experimental Results

D.1 Model Performance

DistDiff is Robust to Guidance Model As mentioned in Section 3.2, we employ a extra feature extractor as our guidance model. To assess the impact of different guidance models, we compared two backbones: a ResNet-50 [21] trained on the original dataset from scratch (weak backbone) and CLIP [6], a strong backbone pre-trained on large datasets and fine-tuned on the original dataset. Table 10 presents the accuracy of the guidance models and the corresponding tuned classifier (ResNet-50). Our DistDiff method demonstrates robustness, showing negligible changes in accuracy (0.25%) across different guidance models. This highlights the robustness of our approach. To ensure fairness in comparison without knowledge leakage from large pre-trained models, we default to using a randomly initialized ResNet-50 trained on the original dataset as our guidance model.

Table 10: Comparison of guidance models on Caltech-101 dataset. We compared the accuracy of two guidance models on the original Caltech-101 dataset. Additionally, we evaluated the performance of a downstream classifier trained on the $5\times$ expanded dataset using corresponding guide model.

Guide Model	**Accuracy (%)**	
	Guide Model	**Downstream Classifier**
Weak (Random initialized and trained ResNet)	66.71 ± 0.47	82.94 ± 0.43
Strong (Pre-trained and finetuned CLIP)	92.24 ± 0.25	83.19 ± 0.69

Determination of Guidance Scale ρ. We compared different learning rates ρ, and the experimental results are shown in Table 11. A learning rate that is too low results in insufficient optimization. Conversely, a learning rate that is too high causes over-optimization, leading to image distortion. Both result in suboptimal performance. We used $\rho = 10$, which achieves the best results, as the default learning rate.

Table 11: Comparison of different learning rate ρ.

ρ	0.1	1.0	10.0	20.0
Accuracy (%)	82.49 ± 0.33	82.74 ± 0.32	83.09 ± 0.11	82.46 ± 0.35

Comparison of Varying Gradient Weight. We compared different contribution levels of two hierarchical prototypes. Specifically, we scaled the gradient of p_g by a certain coefficient, λ_g. The results in Table 12 indicate that an appropriate scaling weight can further enhance overall performance.

Table 12: Comparison of different gradient weights λ_g.

λ_g	0.1	0.3	0.5	0.7	0.9	1.0	2.0
Accuracy (%)	82.61	82.79	83.14	83.12	**83.38**	83.09	82.73

More Efficiency Comparison The high efficiency of DistDiff stems from its direct guidance on the pre-trained model without the need to retrain the diffusion models. We provide computational time comparison in Table 13. Compared to the stable diffusion baseline model, we only introduce an extra 0.48 seconds of inference time per sample, which is only 4% of the original inference time. Our method incurs minimal additional time costs and generates samples directly, unlike LECF, which needs extra post-processing to filter low-confidence samples. This makes LECF slower in generating the same number of samples. We are incorporating this cost analysis into the paper.

Table 13: Inference Efficiency comparison with existing methods on Caltech-101 dataset. * denotes that the actual time required of LECF to derive one sample after filter post-processing. Evaluation processes are conducted on a single GeForce RTX 3090 GPU.

Method	SD	LECF	GID-SD	Ours
Inference Time (s)	12.65	12.73 (17.20 *)	13.47	13.13

More Baseline Comparison We add two more baseline: DA-Fusion [61] and DiffuseMix [29], for comparison. As shown in Table 14. Our method outperforms both baselines. DA-Fusion modifies images while respecting semantic attributes but lacks fine-grained control and incurs additional time costs (around 14 minutes per class on a 3090 GPU) . Our training-free method uses hierarchical guidance for better alignment with the real distribution. DiffuseMix, which uses bespoke prompts and fractal images for augmentation, treats all datasets equally and may not handle distribution shifts well. Our method shows superior performance compared to these approaches.

More Ablation Experiments We add more ablation experiments on StanfordCars dataset. Our method shows more benefits with fine-grained datasets with greater class variance, as illustrated in Table 15. Additionally, having more groups is not always beneficial; an excessive number of groups may cause prototypes to fit noise points or outliers, which could degrade performance.

DistDiff is Effective on Large-Scale Dataset In this work, we follow previous work, such as GIF-SD and DA-Fusion, in conducting experiments on small datasets, as data augmentation strategies are often necessary in data-scarcity scenarios. Besides, we also conducted experiments on ImageNet with $0.2\times$ expanding ratios, resulting in approximately 256K generated samples. As shown in Table 16, our method demonstrates improvements on large-scale datasets.

D.2 Further Analysis

Trade-Off Between Fidelity and Diversity The data expansion task requires both high fidelity and diversity for effective model training. However, this principle does not universally apply across all scenarios. We assessed the trade-off between high fidelity and diversity by adjusting the diffusion model's strength in adding noise to original images. As the noise strength increases, the diversity of generated data enhances, accompanied by a decrease in FID scores. The resulting accuracy and FID scores are presented in Figure 7, where we observed an inverse relationship between FID (fidelity metric) and accuracy across the Caltech-101 and PathMNIST datasets. We found that introducing

Table 14: Comparison of accuracy in expanding Caltech-101 by $5\times$.

Method	DA-Fusion	DiffuseMix	Ours
Accuracy (%)	79.14	82.33	83.09

Table 15: Prototypes comparison of accuracy in expanding StanfordCars by $2\times$. We trained ResNet50 with a 448×448 resolution for 128 epochs.

Method	SD	$\boldsymbol{p}_c$	$\boldsymbol{p}_c + \boldsymbol{p}_g(K=1)$	$\boldsymbol{p}_c + \boldsymbol{p}_g(K=2)$	$\boldsymbol{p}_c + \boldsymbol{p}_g(K=3)$	$\boldsymbol{p}_c + \boldsymbol{p}_g(K=4)$
Accuracy	88.45	89.55	89.69	90.36	90.69	90.62

Table 16: Comparison of accuracy in expanding ImageNet by $0.2\times$. We trained ResNet18 with a 224×224 resolution for 90 epochs.

Method	Original	SD	Ours
Accuracy (s)	69.30	69.66	69.95

more new content on general datasets, such as Caltech-101, can benefit model training by emphasizing the need for diversity. In contrast, on medical datasets with substantial distribution shifts, maintaining the original data distribution and minimizing disturbances proves to be more effective.

D.3 More Visualization Results

In this section, we provide visualization comparison between original stable diffusion (SD) and our DistDiff in Figure 8. Visualization results indicate that our method exhibits finer texture details, more diverse style variations, and stronger foreground-background contrasts, making our samples align real sample distributions. However, the main content of our images still shows minimal deviation from SD, making it challenging for users to discern differences through visual observation alone. In Section 4.4, we provide quantitative analysis to validate that our method generates data with more consistent distributions, thereby enhancing performance in downstream classification tasks.

In addition, we visualize more generated samples across six datasets in Figure 9 to demonstrate the effectiveness of DistDiff. These visualizations confirm that our approach can generate samples with more distribution-consistent patterns.

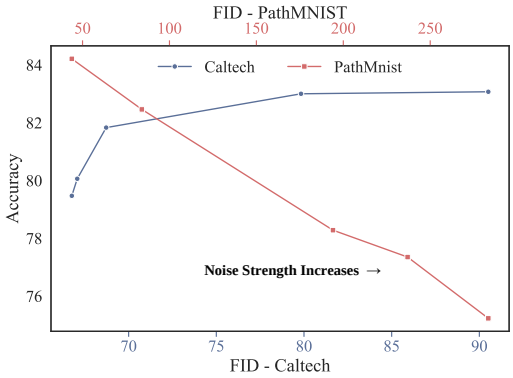


Figure 7: Comparison with FID and accuracy across varying noise strengths.

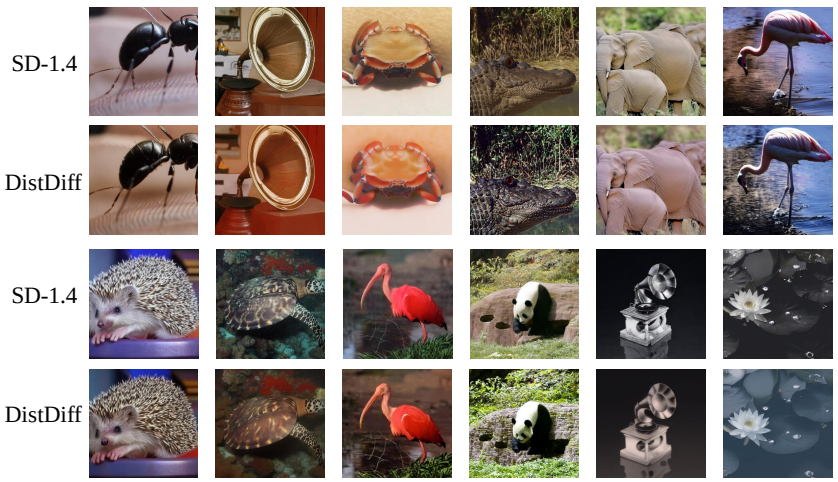


Figure 8: Comparison of visualizations between original Stable Diffusion 1.4 and our DistDiff.

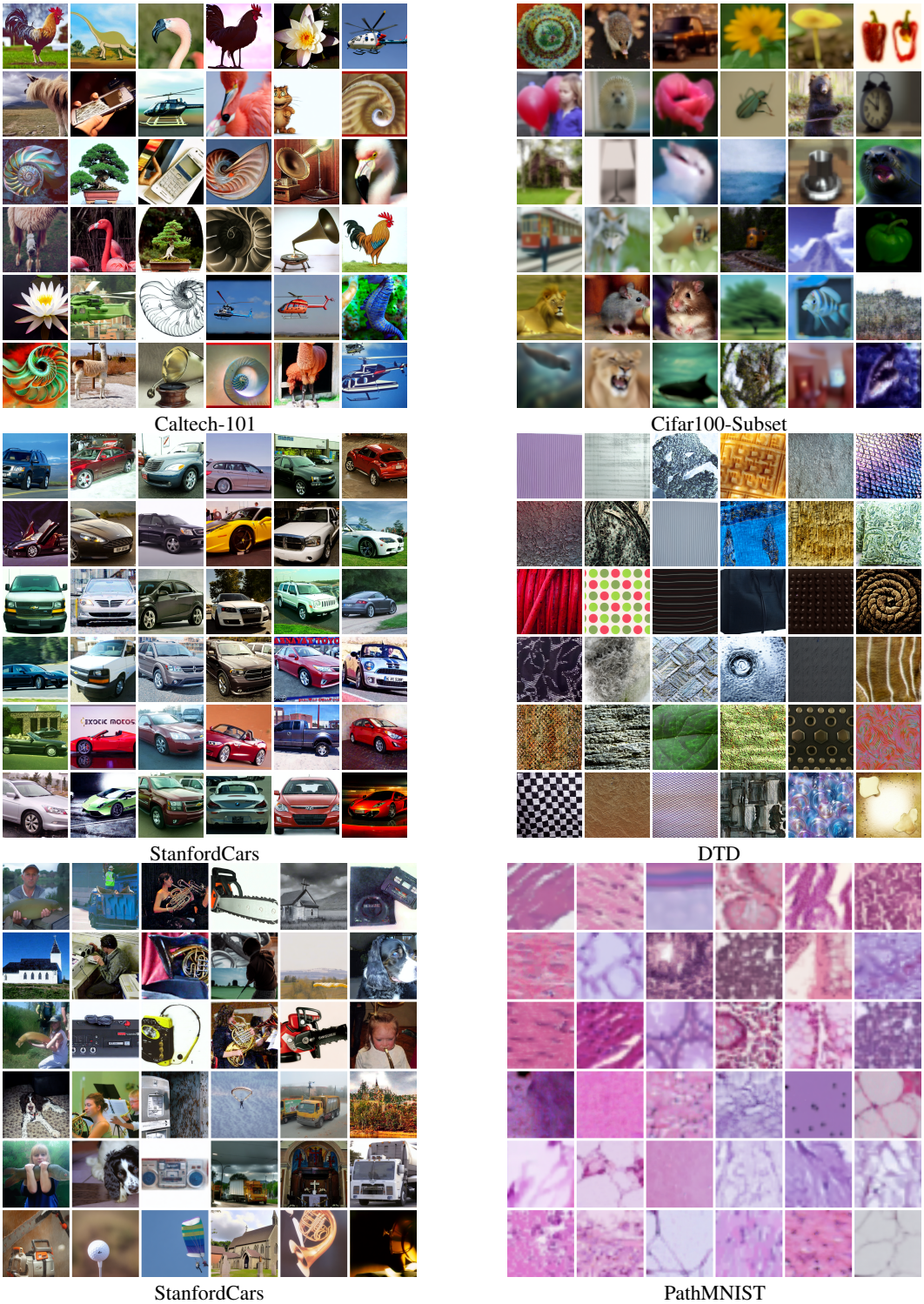


Figure 9: Visualization of synthetic images produced by our method.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract provides a concise summary of the main findings and contributions of the paper, while the introduction elaborates on the problem statement and research objectives, thereby clarifying the contributions.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: In Appendix A, we expound upon the limitations of the work conducted and provide a brief discussion thereof.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: In Section 3 and Appendix C, we provide detailed mathematical derivations for all the formulas appearing in the paper.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: In Section 4.1, we introduced the details of experimental setup and model training to ensure reproducibility.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: In Section 4.1, we introduced the details of experimental setup and model training, and conducted ablation experiments in Section 4.4 to elucidate the selection of hyperparameters.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report error bars of our experimental results in Section 4.3 and 4.4.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: For the experiments, we furnished detailed specifications of the GPU models used along with their corresponding tasks. Furthermore, we included specific information regarding the model training batch size and the number of training epochs.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have carefully reviewed the NeurIPS Code of Ethics and adhere to its principles.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss both potential positive societal impacts and negative societal impacts of the work performed in Appendix A.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The creators or original owners of assets, such as code, data, or models, used in the paper, are properly credited. Additionally, the license and terms of use associated with these assets are explicitly mentioned and respected in accordance with ethical and legal standards.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: Our paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.