Training on Thin Air: Improve Image Classification with Generated Data

Yongchao Zhou [1 2] **Hshmat Sahak** [1 2] **Jimmy Ba** [1 2]

Abstract

Acquiring high-quality data for training discriminative models is a crucial yet challenging aspect of building effective predictive systems. In this paper, we present Diffusion Inversion, a simple yet effective method that leverages the pre-trained generative model, Stable Diffusion, to generate diverse, high-quality training data for image classification. Our approach captures the original data distribution and ensures data coverage by inverting images to the latent space of Stable Diffusion, and generates diverse novel training images by conditioning the generative model on noisy versions of these vectors. We identify three key components that allow our generated images to successfully supplant the original dataset, leading to a 2-3x enhancement in sample complexity and a 6.5x decrease in sampling time. Furthermore, our approach consistently outperforms generic prompt-based steering methods and KNN retrieval baseline across a wide range of datasets, exhibiting especially remarkable results in specialized fields like medical imaging. Furthermore, we demonstrate the compatibility of our approach with widely-used data augmentation techniques, as well as the reliability of the generated data in supporting various neural architectures and enhancing few-shot learning performance.

1. Introduction

Collecting data from the real world can be complex, costly, and time-consuming. Traditional machine learning datasets are often not curated, noisy, or hand-curated but lacking size. Consequently, obtaining high-quality data remains a critical yet challenging aspect of developing effective predictive systems. Recently, large-scale machine learning models such as GPT-3 (Brown et al., 2020), DALL-E (Ramesh et al., 2022), Imagen (Saharia et al., 2022), and Stable Diffusion (Rombach et al., 2022), which are trained on vast amounts of noisy internet data, have emerged as successful "foundation models" (Bommasani et al., 2021) demonstrating strong generative capabilities, such as co-authoring code, creating art, and writing text. Given their extensive world knowledge, a natural question arises: can large-scale pre-trained generative models help generate high-quality training data for discriminative models?

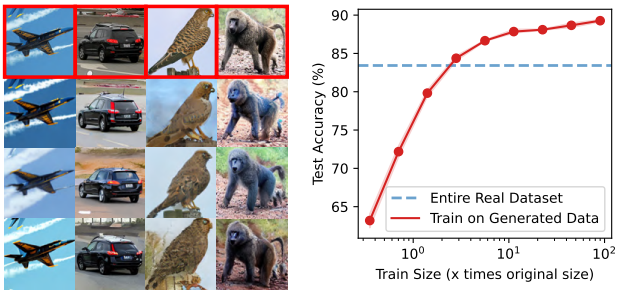

Figure 1. (Left) Synthetic images generated by our method on STL-10. The original real images are highlighted in red. (Right) The test accuracy of ResNet18 trained on synthetic images keeps improving as more data are generated and eventually surpasses the model trained on the entire real STL-10 dataset.

In computer vision, generative models have long been considered for data augmentation. Previous work has explored using VAEs (Shrivastava et al., 2017; Viazovetskyi et al., 2020), GANs (Antoniou et al., 2017; Chai et al., 2021), and Diffusion Models (Antoniou et al., 2017) to enhance model performance in data-scarce settings, such as zero-shot or few-shot learning (Antoniou et al., 2017; He et al., 2022), or to enhance robustness against adversarial attacks or natural distribution shifts (Gowal et al., 2021; Bansal & Grover, 2023). However, due to the limited diversity in samples generated by previous approaches, it has been widely believed and empirically observed that these samples cannot be utilized to train classifiers with higher absolute accuracy compared to those trained on the original datasets (Ravuri & Vinyals, 2019a;b; Gowal et al., 2021; Zhao & Bilen, 2022). Nevertheless, the issue of generator quality may no longer be a hindrance, as state-of-the-art diffusion-based text-to-image models demonstrate remarkable capabilities in synthesizing diverse images with high visual fidelity.

A natural method for using these models to augment the original dataset involves human language intervention. By employing prompt engineering, domain expert knowledge

[1]University of Toronto [2]Vector Institute. Correspondence to: Yongchao Zhou <yczhou@cs.toronto.edu>.

about the target domain can be distilled into a few sets of prompts. Combined with language enhancement techniques (He et al., 2022; Yuan et al., 2022), a diverse array of high-fidelity images can be generated. However, despite their diversity, prompt-based generation often yields off-topic and irrelevant images for the target domain, resulting in low-quality datasets (Bansal & Grover, 2023). To eliminate low-quality image-generation prompts, CLIP filtering (He et al., 2022) has been introduced, enabling a more favorable balance between prompt diversity and quality. Nevertheless, the generation process still disregards the distribution of the training dataset, leading to the creation of distributionally dissimilar images from the original data and a significant gap between real and synthetic datasets (Borji, 2022). Moreover, although in-distribution examples can be generated infinitely, the generated data must still provide adequate coverage of the original dataset to perform well.

To address these challenges and close the performance gap between generated and real data, we present Diffusion Inversion, a simple yet effective method that leverages the general-purpose pre-trained image generator, Stable Diffusion (Rombach et al., 2022). To capture the original data distribution and ensure data coverage, we first obtain a set of embedding vectors by inverting each training image to the output space of the text encoder. Next, we condition Stable Diffusion on a noisy version of these vectors, enabling sampling of a diverse array of novel training images extending beyond the initial dataset. As a result, the final generated images retain semantic meaning while incorporating variability stemming from the rich knowledge embedded within the pre-trained image generator (see examples in Figure 1 and 9). Furthermore, we enhance sampling efficiency by learning condition vectors to generate low-resolution images directly rather than producing them at high resolution and subsequently downsampling. This strategy increases the generation speed of the diffusion model by 6.5 times, rendering it more suitable as a data augmentation tool. To assess our method, we compare it against generic prompt-based steering methods, widely-used data augmentation techniques, and original real data across various datasets. Our primary contributions include:

- We propose Diffusion Inversion, a simple yet effective method that utilizes pre-trained generative models to assist with discriminative learning, bridging the gap between real and synthetic data.
- We pinpoint three vital components that allow models trained on generated data to surpass those trained on real data: 1) a high-quality generative model, 2) a sufficiently large dataset size, and 3) a steering method that considers distribution shift and data coverage.
- Our method outperforms generic prompt-based steering methods and widely-used data augmentation techniques, especially in the realm of specialized datasets such as medical imaging, exhibiting significant data distribution shifts. Additionally, our generated data can enhance various neural architectures and boost few-shot learning performance.

2. Related Work

Generative Models for Image Recognition Generative models, such as VAEs (Kingma & Welling, 2013), GANs (Goodfellow et al., 2020), and Diffusion Models (Dhariwal & Nichol, 2021), have exhibited exceptional capabilities in synthesizing realistic images. Due to their potential to generate an infinite amount of high-quality data, numerous researchers have investigated their application as data augmentation techniques. For example, Shrivastava et al. (2017); Zhu et al. (2017); Viazovetskyi et al. (2020) formulate data augmentation as an image translation task, training an autoencoder-style network to produce multiple variations of input images for downstream prediction models. Some studies have concentrated on data augmentation using GANs, either training them from scratch for few-shot learning (Antoniou et al., 2017) or utilizing pre-trained GANs for self-supervised learning (Chai et al., 2021). Despite their effectiveness in various domains, research has shown that training off-the-shelf convolutional networks, such as ResNet50 (He et al., 2016), on BigGAN (Brock et al., 2018) synthesized images yields inferior results compared to training them on original real training images due to the lack of diversity and the potential domain gap between generated samples and real images (Gowal et al., 2021; Ravuri & Vinyals, 2019a; Bansal & Grover, 2023; Zhao & Bilen, 2022).

Recently, there has been a growing interest in leveraging the power of internet-scale pre-trained diffusion-based models (Nichol et al., 2021; Rombach et al., 2022) for data generation. He et al. (2022) demonstrates that synthetic data from GLIDE (Nichol et al., 2021) can enhance classification models in data-scarce settings or pre-training. Meanwhile, Bansal & Grover (2023) and Yuan et al. (2022) illustrate that Stable Diffusion (Rombach et al., 2022) can serve as a data augmentation tool to improve the robustness of image classifiers under natural distribution shifts. However, the effectiveness of these approaches largely relies on the quality and diversity of language prompts, necessitating extensive manual prompt engineering. Furthermore, the domain gap and data coverage between synthetic data and downstream task real data can still impede the enhancement of synthetic data's effectiveness on classifier learning (Bansal & Grover, 2023; He et al., 2022). In contrast, our method tackles these challenges by learning the conditioning vector of each target image directly, eliminating the need for human intervention in prompt engineering.

Inversion of Generative Models Inverting generative models plays a crucial role in image editing and manipulation tasks (Zhu et al., 2016; Xia et al., 2022). Particularly, given an input image, inversion algorithms (Creswell & Bharath, 2018; Xia et al., 2022) strive to determine the latent representation within the generator that reconstructs the original input. For diffusion models, inversion can be accomplished by adding noise to an image and subsequently denoising it through the network. However, this may result in significant content alterations due to the asymmetry between backward and forward diffusion steps. Choi et al. (2021) address inversion by conditioning the denoising process on noisy, low-pass filtered data from the target image. More recently, inverting text-to-image diffusion models in the context of personalized image generation has gained traction. Gal et al. (2022) propose a textual inversion method that learns to represent visual concepts, such as objects or styles, through new pseudo-words in the embedding space of a frozen text-to-image model. Ruiz et al. (2022) fine-tune the entire network on 3-5 images, which may be susceptible to overfitting. Custom Diffusion (Kumari et al., 2022) mitigates overfitting by fine-tuning only a small subset of model parameters, resulting in improved performance with reduced training time. These works employ inversion as a tool for image editing and have only assessed qualitative human preferences. In contrast, our work seeks to explore how generated images can enhance downstream image classification tasks and proposes using diffusion inversion to address the distribution shift and data coverage problem in synthetic dataset generation.

3. Method

Stable Diffusion (Rombach et al., 2022), which has been trained on billions of image-text pairs, possesses extensive generalizable knowledge. In order to utilize this knowledge for specific classification tasks, we propose a two-stage method that steers a pre-trained generator, $\mathcal{G}$, towards the target domain dataset. In the first stage, we map each image to the model's latent space, generating a dataset of latent embedding vectors. Subsequently, we create new image variants by conditioning on perturbed versions of these vectors. To improve sampling efficiency, we directly learn the embedding for generating target-resolution images, thereby eliminating the need for high-resolution generation and down-sampling. Our method is illustrated in Figure 2.

3.1. Stage 1 - Embedding Learning

Latent Diffusion Model Stable Diffusion belongs to a particular class of Denoising Diffusion Probabilistic Models, known as Latent Diffusion Models (LDM). These models function in the autoencoder's latent space and consist of two primary components. First, an autoencoder is pre-trained on an extensive dataset of images to minimize reconstruction loss, employing regularization from either KL-divergence loss or vector quantization (Van Den Oord et al., 2017; Agustsson et al., 2017). This step allows the encoder $\mathcal{E}$ to map images $x \in \mathcal{D}_x$ to a spatial latent code $z = \mathcal{E}(x)$, while the decoder D can transform these latents back into images, such that $D(\mathcal{E}(x)) \approx x$. Subsequently, a diffusion model is trained to minimize the denoising objective in the obtained latent space, integrating optional conditional information from class labels, segmentation masks, or text tokens.

$$L_{LDM} := \mathbb{E}_{z\sim\mathcal{E}(x),y,\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta\left(z_t, t, c_\theta(y)\right)\|_2^2\right],$$

where t represents the time step, z_t denotes the latent noise at time t, ϵ is the unscaled noise sample, ϵ_θ signifies the denoising network, and $c_\theta(y)$ is a model that maps a conditioning input y to a conditioning vector. During the inference stage, a new image latent z_0 can be generated by iteratively denoising a random noise vector, given a conditioning vector. Lastly, this latent code is transformed into an image using the pre-trained decoder $x' = D\left(z_0\right)$.

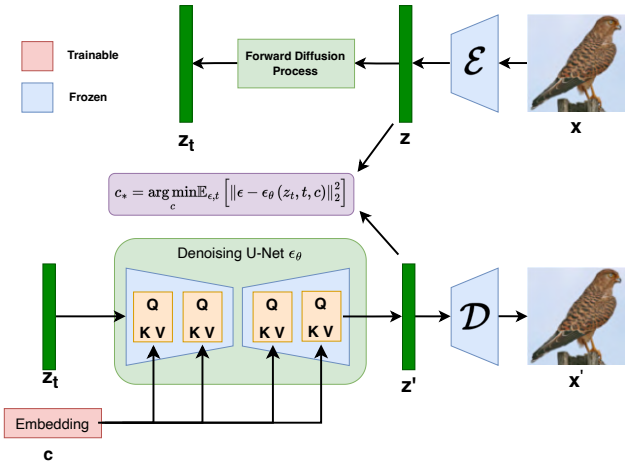

Figure 2. Our method optimizes the standard denoising objective to learn a set of embedding vectors while keeping the model parameters fixed.

Diffusion Inversion Prior research has attempted to invert images back to the input tokens of a text encoder c_θ (Gal et al., 2022). However, this approach is restricted by the expressiveness of the textual modality and constrained to the original output domain of the model. To overcome this limitation, we assume c_θ as an identity mapping and directly optimize the conditioning vector c for each image latent z in the real dataset by minimizing the LDM loss.

$$c_* = \arg\min_c \mathbb{E}_{\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta\left(z_t, t, c\right)\|_2^2\right], \quad (1)$$

Throughout the optimization process, we maintain the original LDM model's training scheme and keep the denoising

model ϵ_θ unchanged to optimally preserve the knowledge acquired during pre-training. Additionally, we boost sampling efficiency by learning condition vectors designed to generate target-resolution images directly, instead of creating high-resolution images and subsequently downsampling.

3.2. Stage 2 - Sampling

Classifier-free Guidance Classifier-free guidance incorporates a guidance weight parameter $w \in \mathcal{R}$ to balance sample quality and diversity in class-conditioned diffusion models. This approach has been extensively employed in large-scale diffusion models such as Stable Diffusion (Rombach et al., 2022), GLIDE (Nichol et al., 2021), and Imagen (Saharia et al., 2022). During sample generation, classifier-free guidance assesses both the conditional diffusion model $\epsilon_\theta(z_t, t, c)$ and the unconditional model $\epsilon_\theta(z_t, t)$. In the case of the Stable Diffusion model, the conditioning vector is calculated as the output of an empty string from the text encoder. At each denoising step, the model output is given by $\hat{\epsilon} = (1+w)\epsilon_\theta(z_t, t, c) - w\epsilon_\theta(z_t, t)$. However, we notice that using an empty string as the conditioning input for the default unconditional embedding is ineffective for the target domain when the data distribution significantly deviates from the Stable Diffusion's training distribution, particularly when the image resolution differs. To address this distribution shift, we utilize the average embedding of all learned vectors as the class-conditioning input for the unconditional models. We demonstrate the efficacy of this approach in Section 5.2.

Sample Diversity Sample diversity is essential for training a downstream classifier on synthetic data (Ravuri & Vinyals, 2019a). In addition to employing various classifier-free guidance strengths, we can initiate the denoising process with different random noises to generate distinct image variants. Furthermore, we investigate two methods of perturbing the conditioning vector: Gaussian noise perturbation and latent interpolation. In the Gaussian noise perturbation approach, we add isotropic Gaussian noise to the conditioning vector, resulting in a new vector $\hat{c} = c + \lambda\epsilon$, where $\epsilon \sim \mathcal{N}(0, 1)$ and λ represents the perturbation strength. For latent interpolation, given two conditioning vectors c_1 and c_2, we create a new vector by linearly interpolating between them: $\hat{c} = \alpha c_1 + (1-\alpha)c_2$. We examine the impact of each component in Section 5.3.

4. Experimental Results

We first investigate CIFAR10/100 (Krizhevsky et al., 2009), STL10 (Coates et al., 2011), and ImageNette (Howard, 2019), focusing on two of the three crucial aspects that enable models trained on generated data to outperform those trained on real data: 1) a high-quality generative model and 2) an adequately large dataset size. We then compare our method with generic prompt-based steering techniques (He et al., 2022) and KNN retrieval from LAION-5B (Schuhmann et al., 2022), emphasizing the importance of a steering method that addresses distribution shift and data coverage when generating data for discriminative downstream tasks. Simultaneously, we showcase our method's effectiveness in few-shot learning scenarios and specialized datasets such as EuroSAT (Helber et al., 2019) and three datasets in MedMNISTv2 (Yang et al., 2023). Moreover, we demonstrate that our approach is compatible with various standard data augmentation strategies, further improving model performance across a wide array of commonly used neural architectures.

We employ the Stable Diffusion model with a default resolution of 512x512 [1]. To enhance learning and sampling efficiency, we directly learn the embedding to generate images with a target resolution of 128x128 for low-resolution datasets like CIFAR10/100 and MedMNIST and a target resolution of 256x256 for other datasets. This modification significantly reduces the image generation time by 27x and 6.5x for 128x128 and 256x256 settings, respectively, making our method a more suitable tool for data augmentation. We provide a detailed runtime analysis in Appendix C.1. For evaluation, we resize all generated images to match the resolution of the original real images, ensuring a fair comparison. Additional details are provided in Appendix B for conciseness.

4.1. Generator Quality and Data Size Matter

Generator Quality In order to examine the impact of generator quality on creating high-quality datasets for downstream classifier training, we initially contrast our approach with the GAN Inversion method (Abdal et al., 2019) on CIFAR10/100. We employ a pre-trained BigGAN model provided by Zhao & Bilen (2022), trained using the state-of-the-art strategy (Zhao et al., 2020b). We generate three synthetic datasets, each containing 50K examples, which is equivalent to the original dataset size. These datasets are produced using randomly sampled latent vectors, GAN Inversion, and our method with classifier-free guidance of 2 and embedding checkpoints at 3K steps. To assess the quality of these datasets, we train a ResNet18 on each dataset and report the mean and standard deviation of the test accuracy using five random seeds.

As depicted in Figure 3, our method exhibits notably superior performance in comparison to GAN approaches, indicating that our technique retains more information from the original dataset, and the quality of the pre-trained generator is crucial for generating high-quality datasets for discriminative models. However, a discrepancy persists between

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

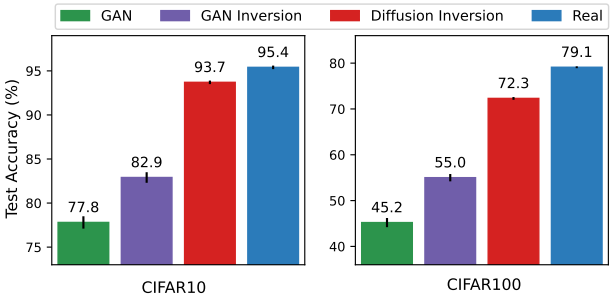

Figure 3. Our method, Diffusion Inversion, dramatically surpasses both GAN and GAN Inversion methods when trained on datasets of equivalent size to the original real dataset, underscoring the importance of a high-quality pre-trained generator.

Table 1. Test accuracy of ResNet18 trained on the VAE-Processed data. Autoencoding results in a substantial loss of information, making it difficult to surpass the performance of the real dataset.

	CIFAR10	CIFAR100
Real (Original)	95.1 ± 0.0	77.9 ± 0.4
Real ($32 \rightarrow 64$)	91.4 ± 0.3	65.5 ± 0.6
Real ($32 \rightarrow 128$)	92.5 ± 0.2	66.2 ± 0.4
Real ($32 \rightarrow 256$)	93.4 ± 0.1	69.8 ± 0.3
Real ($32 \rightarrow 512$)	93.5 ± 0.2	71.1 ± 0.3
Diffusion Inversion	$\mathbf{94.6 \pm 0.1}$	$\mathbf{74.4 \pm 0.3}$

our method and the original real dataset when considering a fixed, equal-size dataset. This infers that the information is more condensed in the real dataset than in the synthetic one.

Scaling in Relation to Real Data Size Next, we investigate the scaling behavior of our approach using four datasets to assess its potential advantages for downstream classifier training by producing an adequate amount of synthetic images. Specifically, we learn embeddings from groups of real datasets comprising varying numbers of training examples. For every embedding, we create 45 distinct variants. The generated examples corresponding to each dataset are illustrated in Figure 5.

As illustrated in Figure 4, for low-resolution datasets such as CIFAR10/100, our method surpasses the real data only in low data regimes (2K for CIFAR10 and 4K for CIFAR100), while exhibiting lower performance as more real training data becomes available. In contrast, for high-resolution datasets like STL-10 and ImageNette, our method consistently outperforms the real data by a significant margin. For instance, considering the entire real dataset, we can enhance the test accuracy on STL-10 from 83.3 to 89.0, and on Imagenette from 93.8 to 95.4. Moreover, to achieve the same level of accuracy, our method can utilize 2-3x less real data. Additionally, Table 6 shows that reconstructing the test data using the autoencoder of the Stable Diffusion model can often improve the test accuracy of the model trained on synthetic data, which is also observed in Razavi et al. (2019).

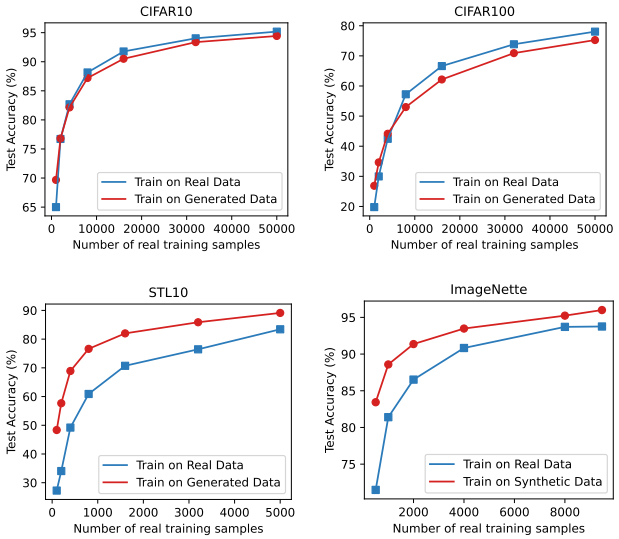

Figure 4. Performance in Relation to Number of Real Data Points. Our approach demonstrates substantially improved performance in low-data scenarios across all datasets. In high-data scenarios, it exhibits comparable performance for low-resolution datasets and superior performance for high-resolution datasets.

Scaling in Relation to Synthetic Data Size Conversely, we also explore the case where we learn embeddings for every data point in the dataset and continue generating more data. As demonstrated in Figure 1 (Right), more data can consistently enhance the performance of the downstream classifier. It surpasses the real dataset performance when approximately three times more data are generated compared to the original dataset size. The scaling trend suggests that it is possible to further boost the model performance by training the model for a longer duration and generating more data points, ideally in an online manner.

Loss of Information Caused by Autoencoding To comprehend the extent of information loss during the autoencoding process, we create four CIFAR10 variants with images autoencoded at different resolutions. We commence by resizing the images to a resolution of 64x64, which is the minimum requirement for the Stable Diffusion model. Table 1 reveals that although performance continually improves as images are resized to higher resolutions and autoencoded, the best-performing setting, with a resolution of 512, still underperforms compared to training on the original images. This indicates that a significant amount of information is lost during the autoencoding process, or there exists a distribution shift between the reconstructed images and real images. In comparison to the 128-resolution setting where our method is trained, our method substantially enhances test accuracy on CIFAR10 and CIFAR100 from 92.5 and 66.2 to 94.6 and 74.4, respectively.

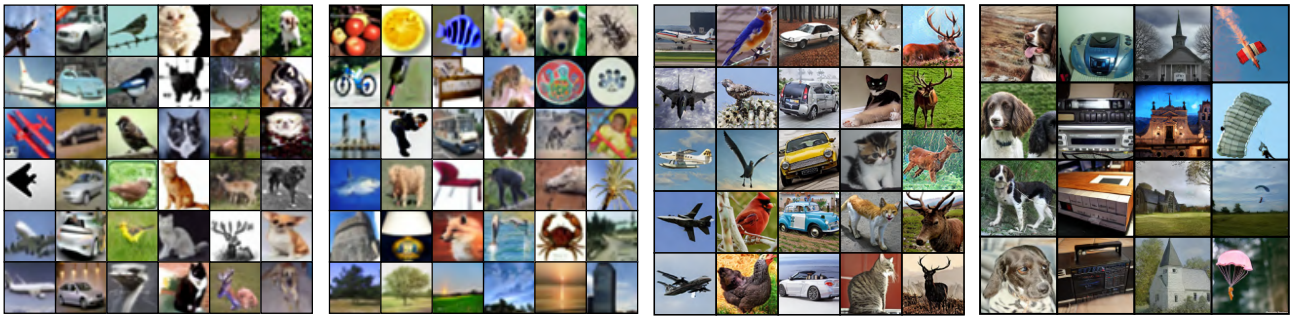

Figure 5. Synthetic images generated by our method. Left to Right: CIFAR10, CIFAR100, STL-10, ImageNette.

Table 2. Our approach significantly improves few-shot learning performance by generating high-quality data on EuroSAT.

	1	2	4	8	16
CoOp	53.3	61.4	70.9	78.4	84.8
Tip Adapter	59.5	66.2	74.1	78.0	84.6
CT w. Real	54.9	63.2	74.4	79.8	85.9
CT w. LECF	64.4	73.7	**80.0**	84.6	**88.6**
CT w. KNN10	40.8	47.2	50.7	56.2	58.2
CT w. KNN25	47.4	51.7	57.4	64.3	67.7
CT w. KNN50	53.1	58.5	64.2	71.4	74.4
CT w. DI (Ours)	**67.2**	**74.9**	79.9	**84.7**	88.0

4.2. Data Distribution and Data Coverage Matter

Comparison with Generic Prompt-Based Steering Methods The recent study, Language Enhancement with Clip Filtering (LECF) by He et al. (2022), utilizes Stable Diffusion to generate data for discriminative models, showcasing state-of-the-art performance in few-shot learning scenarios. We contrasted our approach with LECF in two unique settings: few-shot learning on EuroSAT (where their method achieved the most substantial performance improvement) and standard training on STL10.

Concerning EuroSAT, we also evaluated our method against CoOP (Zhou et al., 2022), Tip Adapter (Zhang et al., 2022), and Classifier Tuning (CT) with Real Data (He et al., 2022). As depicted in Table 2, our method, akin to LECF, enhances the performance of few-shot learning, achieving performance comparable to LECF. Regarding the STL10 dataset, we analyzed the progression of test accuracy concerning the number of generated data points. By training a ResNet18 solely on the generated data, we adjusted the Clip Filtering strength of LECF across [0.0, 0.5, 0.7, 0.9, 0.95, 0.97], determining that 0.95 produced the optimal performance. As illustrated in Table 5, our approach exhibits superior scaling capabilities compared to LECF. This advantage can be attributed to our method's consideration of domain shifts and its improved coverage relative to LECF.

Comparison with KNN Retrieval on LAION Dataset Stable Diffusion was trained on the publicly available LAION dataset (Schuhmann et al., 2022), raising the question of whether generating a synthetic dataset is necessary, or if retrieving the most similar images for data augmentation would suffice. To assess this, we first applied KNN retrieval using clip retrieval on the STL10 dataset. With k=10, 25, and 50, we achieved test accuracies of 85.4%, 88.4%, and 90.9%, respectively. In comparison, our method generated a test accuracy of 88.7% when producing 45 data points per embedding, slightly outperforming the retrieval of 25 real images per training image but falling short of the retrieval of 50 real images per training image. This indicates that KNN retrieval can serve as a solid baseline when target classes like airplanes, cars, and dogs are well-represented in the Stable Diffusion training distribution.

Table 3. Comparison against KNN retrieval on LAION-5B. Our method consistently outperforms KNN retrieval across three specialized medical imaging datasets, highlighting its effectiveness in handling distribution shifts and data coverage.

	K=10	K=25	K=50	DI (Ours)
PathMNIST	22.5	29.9	23.4	**81.0**
DermaMNIST	23.0	27.8	22.1	**66.4**
BloodMNIST	21.7	27.7	25.8	**93.0**

Nonetheless, we contend that this approach is inadequate when substantial distribution shifts exist between the target and source domains, particularly in specialized areas such as medical imaging. To illustrate this, we examined three distinct MedMNISTv2 datasets (Yang et al., 2023): PathMNIST, DermaMNIST, and BloodMNIST. As shown in Table 3, our method consistently outperforms the KNN retrieval baseline. It is crucial to note that LECF would also fail in this situation due to significant distribution shifts and the challenges of devising effective prompts. Moreover, KNN retrieval fails to improve the few-shot learning performance on EuroSAT, as demonstrated in Table 2.

4.3. Comparison against Data Augmentation Methods

We evaluate our approach against widely-used data augmentation techniques for image classification on STL10. These techniques include standard image augmentation methods such as AutoAugment (Cubuk et al., 2018), RandAugment (Cubuk et al., 2020), and CutOut (De-

Table 4. Comparison to Standard Data Augmentation Techniques on STL10: Our approach, in conjunction with default data augmentation, consistently surpasses alternative methods. Moreover, merging the generated data with other techniques can enhance performance further.

	Default	AutoAug	RandAug	CutOut	MixUp	CutMix	AugMix	ME-ADA
Original Dataset	83.2	87.0	86.3	84.3	89.4	88.1	83.8	83.4
Synthetic (Ours)	**89.5**	**91.5**	**91.0**	**89.5**	**91.5**	**92.6**	**89.2**	**89.1**

Table 5. Scaling Capabilities of Diffusion Inversion vs LECF

Number of Generated Data	1750	3500	7K	14K	28K	56K	112K	224K
LECF (threshold=0.95)	52.4	55.7	63.9	70.3	73.3	77.2	78.4	80.7
Diffusion Inversion (Ours)	63.2	72.2	79.8	84.4	86.7	87.9	88.1	88.7

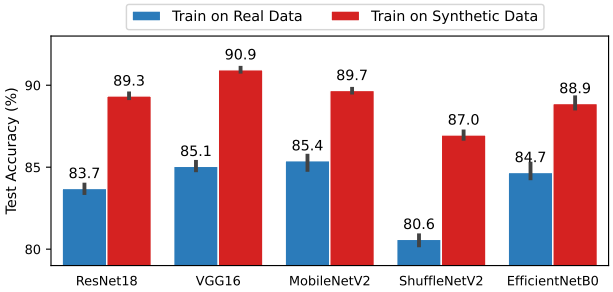

Figure 6. Our method's synthetic dataset significantly enhances the performance of various neural architectures on the STL10 dataset.

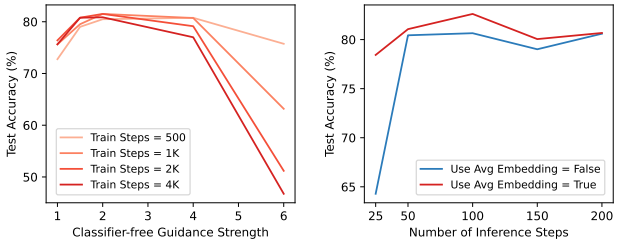

Figure 7. The effect of the number training steps & Classifier-free guidance strength (Left) and inference steps & Unconditional embedding (Right) on model performance.

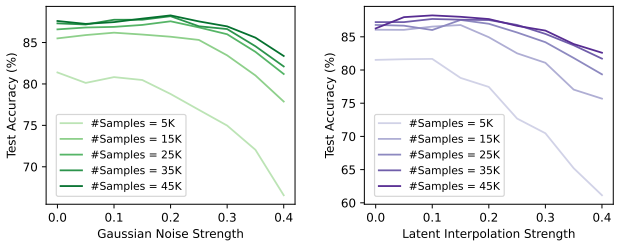

Figure 8. The effect of Gaussian Noise (Left) and Latent Interpolation (Right) on the performance as we generate more data.

Vries & Taylor, 2017); interpolation-based methods like MixUp (Zhang et al., 2017), CutMix (Yun et al., 2019), and AugMix (Hendrycks et al., 2019); and the adversarial domain augmentation (ADA) method ME-ADA (Zhao et al., 2020a). A comprehensive description of each technique is provided in Appendix B.2.3.

As shown in Table 4, our method (89.5%) combined with default data augmentation (i.e., random crop and flip) outperforms all the aforementioned techniques (indicated by the first row). Moreover, our approach complements other data augmentation techniques, and their integration can result in even higher performance.

4.4. Evaluation on Various Architectures

The usefulness of synthetic data is greatly amplified when it is compatible with a wide range of neural architectures. To evaluate the efficacy of our generated data, we examine its performance across a diverse selection of popular neural network architectures, such as ResNet18 (He et al., 2016), VGG16 (Simonyan & Zisserman, 2014), MobileNetV2 (Sandler et al., 2018), ShuffleNetV2 (Ma et al., 2018), and EfficientNetB0 (Tan & Le, 2019), using the STL10 dataset. As illustrated in Figure 6, synthetic images significantly improve performance across all tested architectures. This demonstrates that our method effectively extracts generalizable knowledge from the pre-trained Stable Diffusion and incorporates it into the generated dataset.

5. Quantitative Analysis

We conduct numerous quantitative evaluations on STL10 to comprehend the impact of certain design choices and the influence of each hyperparameter.

5.1. Training Steps and Classifier-free Guidance Strength

Figure 7 (Left) illustrates the performance variation with increasing training steps for embedding vectors and classifier-free guidance strength. It suggests that extending the embedding vector training beyond 1K steps yields minimal performance improvement. However, as training becomes more extensive, the optimal classifier-free guidance strength decreases. A high guidance strength leads to a significant performance drop. In practice, initiating with a classifier-free guidance strength between 2 and 4 proves effective.

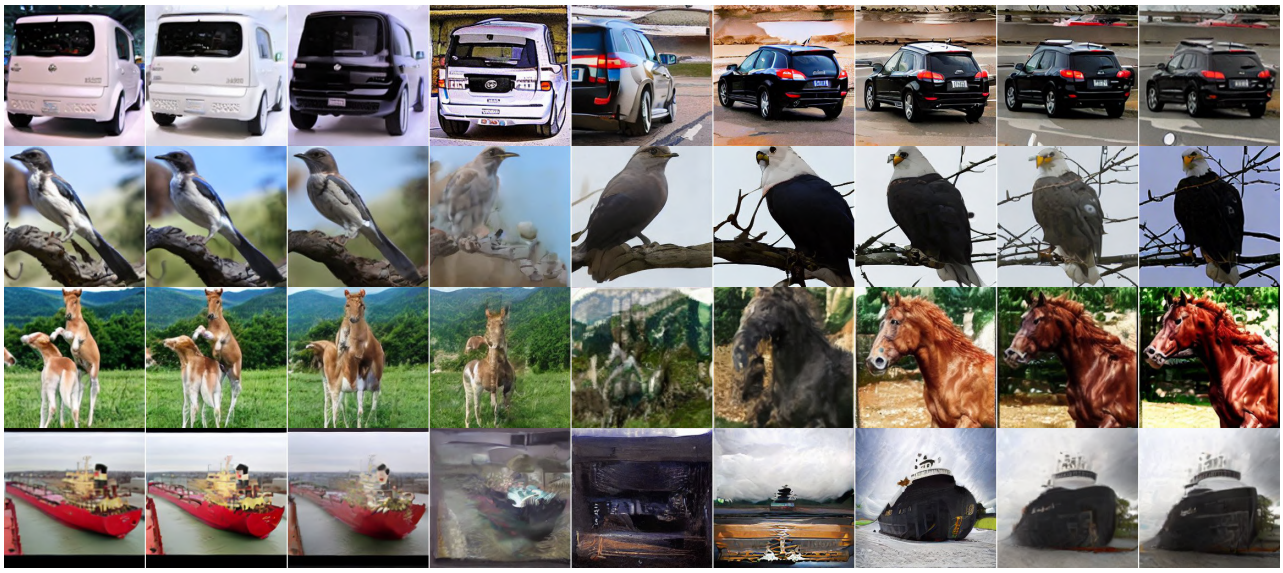

Figure 9. Generate image variants by interpolating two embedding vectors. From left to right, interpolation strength α: 0.0, 0.1, 0.2, 0.3, 0.5, 0.7, 0.8, 0.9, 1.0. Some pairs of the embedding vectors can generate novel and natural images regardless of the chosen interpolation strength, while others only work when the interpolation strength is small.

5.2. Inference Steps and Unconditional Embedding

Figure 7 (Right) illustrates that using the mean embedding of all learned vectors as the class-conditioning input for unconditional models consistently outperforms the text encoder's output with an empty string. However, the learned embedding does not effectively generate images at varying resolutions. For instance, a learned vector from a 512-resolution image struggles to create a 128-resolution image. This highlights the suboptimal performance of the empty string embedding, as the initial text encoder is co-trained with the denoising model on higher-resolution images (512x512). Regarding inference steps, we determine that 100 steps provide a suitable balance between performance and computational cost, leading us to adopt 100 inference steps as the default setting.

5.3. Gaussian Noise and Latent Interpolation

Gaussian Noise We investigate the influence of Gaussian noise on model performance by adjusting the noise strength and setting the latent interpolation strength α to 0. Figure 8 (Left) demonstrates the relationship between Gaussian noise strength and model test accuracy. Our findings indicate that the optimal performance is achieved when generating a dataset of equal size to the original without noise perturbation. However, when sampling additional data, it is advantageous to increase the noise strength accordingly, with a noise strength of $\lambda = 0.2$ as a suitable starting point.

Latent Interpolation In this study, we examine the impact of latent interpolation on model performance by adjusting the interpolation strength and setting the Gaussian noise strength (λ) to 0. Figure 8 (Right) demonstrates the relationship between interpolation strength and performance, revealing that a high strength significantly reduces performance. Notably, unlike the Gaussian noise strength, increasing the sample size does not benefit high strength. The optimal value resides between 0.1 and 0.15. Figure 9 displays the samples, suggesting that novel and realistic images can be generated with any interpolation strength, provided the two embedding vectors are highly compatible. However, if the embeddings are not carefully chosen, the interpolated image at $\alpha = 0.3$ appears quite perplexing. In our experiment, we randomly select two embedding vectors for generating new images, resulting in a small optimal interpolation strength.

6. Conclusion

We introduce Diffusion Inversion, a simple yet effective method for generating high-quality synthetic data that boosts image classification performance by leveraging pre-trained generative models, specifically Stable Diffusion. Our method effectively addresses the challenges of data distribution shift and data coverage, surpassing conventional prompt-based steering approaches and prevalent data augmentation techniques. Furthermore, it proves advantageous for various neural architectures and substantially enhances few-shot learning performance. Impressively, our synthesized images can successfully supplant original datasets, resulting in significant improvements in sample complexity and sampling time. Our study highlights the promising potential of utilizing pre-trained generative models for data augmentation, especially in niche domains where data acquisition and curation are both costly and labor-intensive.

References

Abdal, R., Qin, Y., and Wonka, P. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4432–4441, 2019.

Agustsson, E., Mentzer, F., Tschannen, M., Cavigelli, L., Timofte, R., Benini, L., and Van Gool, L. Soft-to-hard vector quantization for end-to-end learned compression of images and neural networks. *arXiv preprint arXiv:1704.00648*, 3, 2017.

Antoniou, A., Storkey, A., and Edwards, H. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017.

Bansal, H. and Grover, A. Leaving reality to imagination: Robust classification via generated datasets. *arXiv preprint arXiv:2302.02503*, 2023.

Bender, E. M., Gebru, T., McMillan-Major, A., and Shmitchell, S. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pp. 610–623, 2021.

Birhane, A., Prabhu, V. U., and Kahembwe, E. Multimodal datasets: misogyny, pornography, and malignant stereotypes. *arXiv preprint arXiv:2110.01963*, 2021.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Borji, A. How good are deep models in understanding generated images? *arXiv preprint arXiv:2208.10760*, 2022.

Brock, A., Donahue, J., and Simonyan, K. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Chai, L., Zhu, J.-Y., Shechtman, E., Isola, P., and Zhang, R. Ensembling with deep generative views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14997–15007, 2021.

Cho, J., Zala, A., and Bansal, M. Dall-eval: Probing the reasoning skills and social biases of text-to-image generative transformers. *arXiv preprint arXiv:2202.04053*, 2022.

Choi, J., Kim, S., Jeong, Y., Gwon, Y., and Yoon, S. Ilvr: Conditioning method for denoising diffusion probabilistic models. *arXiv preprint arXiv:2108.02938*, 2021.

Coates, A., Ng, A., and Lee, H. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215–223. JMLR Workshop and Conference Proceedings, 2011.

Creswell, A. and Bharath, A. A. Inverting the generator of a generative adversarial network. *IEEE transactions on neural networks and learning systems*, 30(7):1967–1974, 2018.

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

Cubuk, E. D., Zoph, B., Shlens, J., and Le, Q. V. 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.

DeVries, T. and Taylor, G. W. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

Gal, R., Alaluf, Y., Atzmon, Y., Patashnik, O., Bermano, A. H., Chechik, G., and Cohen-Or, D. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

Gowal, S., Rebuffi, S.-A., Wiles, O., Stimberg, F., Calian, D. A., and Mann, T. A. Improving robustness using generated data. *Advances in Neural Information Processing Systems*, 34:4218–4233, 2021.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

He, R., Sun, S., Yu, X., Xue, C., Zhang, W., Torr, P., Bai, S., and Qi, X. Is synthetic data from generative models ready for image recognition? *arXiv preprint arXiv:2210.07574*, 2022.

Helber, P., Bischke, B., Dengel, A., and Borth, D. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019.

Hendrycks, D., Mu, N., Cubuk, E. D., Zoph, B., Gilmer, J., and Lakshminarayanan, B. Augmix: A simple data processing method to improve robustness and uncertainty. *arXiv preprint arXiv:1912.02781*, 2019.

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

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images, 2009.

Kumari, N., Zhang, B., Zhang, R., Shechtman, E., and Zhu, J.-Y. Multi-concept customization of text-to-image diffusion. *ArXiv*, abs/2212.04488, 2022.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Lyu, S. Deepfake detection: Current challenges and next steps. In *2020 IEEE international conference on multimedia & expo workshops (ICMEW)*, pp. 1–6. IEEE, 2020.

Ma, N., Zhang, X., Zheng, H.-T., and Sun, J. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 116–131, 2018.

Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., and Chen, M. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

Ravuri, S. and Vinyals, O. Classification accuracy score for conditional generative models. *Advances in neural information processing systems*, 32, 2019a.

Ravuri, S. and Vinyals, O. Seeing is not necessarily believing: Limitations of biggans for data augmentation, 2019b.

Razavi, A., Van den Oord, A., and Vinyals, O. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. 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.

Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., and Aberman, K. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. *arXiv preprint arXiv:2208.12242*, 2022.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3): 211–252, 2015.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S. K. S., Ayan, B. K., Mahdavi, S. S., Lopes, R. G., et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

Scheuerman, M. K., Hanna, A., and Denton, E. Do datasets have politics? disciplinary values in computer vision dataset development. *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW2):1–37, 2021.

Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402*, 2022.

Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., and Webb, R. Learning from simulated and unsupervised images through adversarial training. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2107–2116, 2017.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Tan, M. and Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pp. 6105–6114. PMLR, 2019.

Van Den Oord, A., Vinyals, O., et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

Viazovetskyi, Y., Ivashkin, V., and Kashin, E. Stylegan2 distillation for feed-forward image manipulation. In *European conference on computer vision*, pp. 170–186. Springer, 2020.

Xia, W., Zhang, Y., Yang, Y., Xue, J.-H., Zhou, B., and Yang, M.-H. Gan inversion: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Yang, J., Shi, R., Wei, D., Liu, Z., Zhao, L., Ke, B., Pfister, H., and Ni, B. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10(1):41, 2023.

Yuan, J., Pinto, F., Davies, A., Gupta, A., and Torr, P. Not just pretty pictures: Text-to-image generators enable interpretable interventions for robust representations. *arXiv preprint arXiv:2212.11237*, 2022.

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., and Yoo, Y. 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.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Zhang, R., Zhang, W., Fang, R., Gao, P., Li, K., Dai, J., Qiao, Y., and Li, H. Tip-adapter: Training-free adaption of clip for few-shot classification. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXV*, pp. 493–510. Springer, 2022.

Zhao, B. and Bilen, H. Synthesizing informative training samples with gan. *arXiv preprint arXiv:2204.07513*, 2022.

Zhao, L., Liu, T., Peng, X., and Metaxas, D. Maximum-entropy adversarial data augmentation for improved generalization and robustness. *Advances in Neural Information Processing Systems*, 33:14435–14447, 2020a.

Zhao, S., Liu, Z., Lin, J., Zhu, J.-Y., and Han, S. Differentiable augmentation for data-efficient gan training. *Advances in Neural Information Processing Systems*, 33: 7559–7570, 2020b.

Zhou, K., Yang, J., Loy, C. C., and Liu, Z. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022.

Zhu, J.-Y., Krähenbühl, P., Shechtman, E., and Efros, A. A. Generative visual manipulation on the natural image manifold. In *European conference on computer vision*, pp. 597–613. Springer, 2016.

Zhu, X., Liu, Y., Qin, Z., and Li, J. Data augmentation in emotion classification using generative adversarial networks. *arXiv preprint arXiv:1711.00648*, 2017.

A. Societal Impact

As generative models advance, harnessing them for high-quality training data can substantially cut time and resources spent on data collection and annotation. Our method offers a streamlined, efficient means of utilizing these powerful models, potentially allowing smaller organizations and researchers with limited resources to develop effective machine learning models more feasibly.

However, implementing this approach in real-life applications requires caution due to concerns about bias and fairness (Scheuerman et al., 2021). Generative models, such as Stable Diffusion (Rombach et al., 2022), are trained on extensive, diverse, and uncurated internet data that may contain harmful biases and stereotypes (Bender et al., 2021; Birhane et al., 2021). These biases can worsen during generation (Cho et al., 2022), leading to discriminatory AI decision-making. However, our method can be utilized to generate diverse, high-quality data for underrepresented groups, fostering fairer and less biased AI systems.

Another potential drawback is the misuse of generated data. High-quality generated data could be exploited for malicious purposes, such as deepfakes (Lyu, 2020), leading to the proliferation of misinformation and manipulation in various domains, including politics, social media, and entertainment.

To counter these negative societal impacts, it is vital to ensure responsible development and deployment of the Diffusion Inversion method and related technologies. This entails incorporating mechanisms to detect and mitigate biases, exploring ethical policies and regulations for synthetic data use, and conducting further research to curate generated data and create fairer multimodal representations of the real world. Establishing responsible practices and guidelines for such methods is crucial for promoting their positive societal impact.

B. Experimental Details

B.1. Implementation Details

Datasets We evaluate our methods on the following datasets: i) **CIFAR** (Krizhevsky et al., 2009): A standard image dataset with two tasks, CIFAR10 (10 classes) and CIFAR100 (100 classes), each containing 50,000 training examples and 10,000 test examples at a 32x32 resolution. ii) **STL10** (Coates et al., 2011): An image dataset of 113,000 color images at a 96x96 resolution, designed for semi-supervised learning. It has 5,000 labeled training images and 8,000 labeled test images across ten classes. We use only the labeled portion to test our method's performance on higher-resolution, low-data settings. iii) **ImageNette** (Howard, 2019): A 10-class subset of ILSVRC2012 (Russakovsky et al., 2015) containing 9,469 training and 3,925 testing examples, resized to a 256x256 resolution. iv) **EuroSAT** (Helber et al., 2019): A dataset based on Sentinel-2 satellite images, covering 13 spectral bands and consisting of 27,000 labeled and geo-referenced samples across ten classes. v) **MedMNISTv2** (Yang et al., 2023): A large-scale collection of standardized biomedical images, including 12 datasets pre-processed into 28x28 resolution. We use three datasets focused on multi-class image classification: PathMNIST, DermaMNIST, and BloodMNIST.

Training We utilize the publicly accessible 1.4 billion-parameter text-to-image model by Rombach et al. (2022), pretrained on the LAION-400M dataset[2]. The model's default image resolution is 512x512, with a minimum functional requirement of 64x64. However, some datasets have a 32x32 resolution. To accommodate this and our training budget, we resize CIFAR10 and CIFAR100 images to 128x128 and STL-10 and ImageNette images to 256x256. We optimize Eq. 1 using AdamW (Loshchilov & Hutter, 2017) with a constant learning rate of 0.03 for up to 3K steps to learn the conditioning vector for each real dataset image, without data augmentation.

Sampling Although on-the-fly data generation is ideal, it is computationally costly. We pre-generate a fixed-size dataset and train models on it. Unless specified, we generate each new image in 100 denoising steps using 3K-step checkpoints with classifier-free guidance strength of 2, Gaussian noise strength of 0.1, and embedding interpolation strength of 0.1.

Evaluation We train a ResNet18 (He et al., 2016) on real and generated data at the default resolution. The ResNet is trained using SGD with momentum, a batch size of 128, a cosine learning rate schedule with an initial learning rate of 0.1, and a standard data augmentation scheme, including random horizontal flips and random crops after zero-padding.

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

B.2. Experimental Setups

B.2.1. Generator Quality

To emphasize the significance of generator quality in producing high-quality datasets for discriminative model training, we first compare our approach with the GAN Inversion method (using a pre-trained BigGAN by Abdal et al. (2019)) on CIFAR10 and CIFAR100. We learn a latent vector $z \in \mathbf{R}^{d_z}$ for each image $x \in \mathbf{R}^{d_i}$ in the real dataset by minimizing the weighted sum of feature and pixel distances between synthetic and real images, with a pre-trained feature extractor ψ_ϑ, feature dimension d_f, and default $\lambda_{\text{pixel}} = 1$.

$$\arg\min_{\boldsymbol{z}} \frac{1}{d_f} \|\psi_\vartheta(G(\boldsymbol{z})) - \psi_\vartheta(\boldsymbol{x})\|^2 + \frac{\lambda_{\text{pixel}}}{d_I} \|G(\boldsymbol{z}) - \boldsymbol{x}\|^2$$

Using the pre-trained BigGAN provided by Zhao & Bilen (2022) and trained with a state-of-the-art strategy (Zhao et al., 2020b), we create three synthetic datasets equivalent in size to the original dataset. These datasets are generated using random latent vectors, GAN Inversion, and our method with classifier-free guidance of 2 and checkpoints at 3K steps. To evaluate dataset quality, we train a ResNet18 on each dataset and report the mean and standard deviation of five random seeds.

B.2.2. Scaling in Relation to Real Data Size

In Figure 4, we obtain an embedding for each data point and generate 45 samples per embedding over 100 denoising steps. We use checkpoints at 1K, 2K, and 3K steps, a classifier-free guidance strength sampled from [2, 3, 4], and Gaussian noise and embedding interpolation strengths of 0.1.

B.2.3. Comparison against Image Data Augmentation Methods

i) **AutoAugment:** We utilize torchvision.transforms.AutoAugment, PyTorch's built-in implementation of AutoAugment, with the ImageNet policy comprising 25 transforms. During training, one transform is randomly chosen and applied with a specified probability and magnitude. ii) **RandAugment:** Similar to AutoAugment, we randomly select two operations from a list of 14 and apply them with certainty. iii) **CutOut:** Our CutOut implementation masks out a random square region, sized at 1/8 of the input image. **MixUp:** We use interpolated images as new inputs for network training by combining a permuted batch of inputs with the original batch, sampling interpolation strength from the beta distribution (beta = 1). The loss function is adapted accordingly. iv) **CutMix:** We replace a region of each input with a corresponding region from another input by permuting each batch and sampling a region size from the beta distribution. The modified loss function from MixUp is used, with lam representing the area ratio of the selected region to the image. v) **AugMix:** Images are augmented and mixed with the original image by sampling and composing operations. One chain is randomly applied to obtain the augmented image, which is then combined with the original image using an interpolation weight sampled from the beta distribution (alpha=1). Our implementation uses PyTorch's torchvision.transforms.AugMix method with default parameters. vi) **ME-ADA:** In ME-ADA, an adversarial data augmentation method, a minimax procedure runs K times. Each cycle consists of a minimization stage (T_min steps of network training) and a maximization stage (converting input-label pairs to adversarial examples by nudging inputs towards the loss function gradient).

C. Additional Results

C.1. Run Time Analysis

The computation for our method comprises two main components: embedding learning and sampling. For ImageNette and STL-10, we learn an embedding for each image and prompt the Stable Diffusion model to generate an image with a resolution of 256x256. On an A40, training the embedding for 3,000 steps takes an average of 84.1 seconds per embedding. In contrast, for CIFAR10/100, we learn an embedding that enables Stable Diffusion to generate 128x128 images directly, with the embedding learning taking an average of 18.8 seconds per embedding.

Another computational cost arises from sampling using the learned embeddings. Standard Stable Diffusion sampling requires approximately 5.28 seconds to generate a 512x512 image with 100 diffusion steps. However, generating a 256x256 or 128x128 image based on the learned embedding takes only 0.82 seconds (6.44 times faster) or 0.20 seconds (26.4 times faster), respectively. This speedup is due to the absence of a CLIPText encoder for text prompt embedding and the diffusion process running in a lower-dimensional space. The original diffusion's latent space has a dimension of (64, 64, 4), while ImageNette/STL10 and CIFAR10/CIFAR100 in our experiments have dimensions of (32, 32, 4) and (16, 16, 4), respectively. The following are the average times required to generate one image using 100 inference steps. It is important to note that the dimension size plays a more significant role in time reduction than the text encoder.

- (64, 64, 4) with Text Encoder: 5.28s
- (64, 64, 4) without Text Encoder: 5.19s
- (32, 32, 4) without Text Encoder: 0.82s
- (16, 16, 4) without Text Encoder: 0.20s

To generate 45 samples per learned embedding (our default setting), the total time for both embedding learning and sampling in ImageNette and STL10 is approximately 121 seconds, while for CIFAR10/100, it takes only 27.8 seconds. In comparison to the standard Stable Diffusion sampling for 45 images, which takes 237.6 seconds, our method is almost twice as fast for ImageNette/STL10 and 8.5 times faster for CIFAR10/100. Moreover, the amortized cost of learning the embedding decreases when generating more data, making our approach more suitable as a data augmentation tool. Ideally, we want to generate data on-the-fly during training, which further supports the efficiency of our method.

C.2. Model achieves better accuracy on VAE processed test data

We observe that reconstructing test data with the Stable Diffusion model's autoencoder often enhances test accuracy for models trained on synthetic data, as also noted in Razavi et al. (2019).

Table 6. Test accuracy using the entire dataset. Transforming the test data using VAE can often improve the model performance.

		Synthetic Data	
	Real Data	Original	VAE-Processed
CIFAR10	**95.1 ± 0.0**	94.6 ± 0.1	94.7 ± 0.1
CIFAR100	**77.9 ± 0.4**	74.4 ± 0.3	75.2 ± 0.2
STL-10	83.3 ± 0.7	**89.0 ± 0.2**	88.8 ± 0.2
ImageNette	93.8 ± 0.2	95.4 ± 0.1	**95.6 ± 0.1**

C.3. FID, Precision, Recall, Density, and Coverage

We assess the FID, precision, recall, density, and coverage of our generated data on STL10 using the implementation from https://github.com/POSTECH-CVLab/PyTorch-StudioGAN.

Interpolation Strength Table 8 demonstrates the variations in FID, precision, recall, density, and coverage with respect to the interpolation strength α. As indicated, increasing the interpolation strength adversely affects FID, precision, and density, while improving recall. Coverage peaks at an interpolation of 0.1, suggesting a trade-off between generation quality and diversity.

Gaussian Noise Strength Table 7 demonstrates the variations in FID, precision, recall, density, and coverage as the additive noise value increases. The results indicate that higher noise levels adversely impact all metrics, signifying a decline

Table 7. Image Generation Evaluation Metrics vs Interpolation Strength.

alpha	FID	Precision	Recall	Density	Coverage
0.00	17.930	0.894	0.644	0.734	0.753
0.10	17.678	0.831	0.661	0.732	0.787
0.20	26.177	0.635	0.751	0.584	0.739
0.30	43.160	0.448	0.787	0.363	0.605
0.40	62.773	0.328	0.805	0.245	0.500

in individual image quality. However, Figure 9 reveals that incorporating some noise can enhance model accuracy, as the overall information in the dataset may still increase despite the diminished quality of each image.

Table 8. Image Generation Evaluation Metrics vs Noise Value.

Noise Value	FID	Precision	Recall	Density	Coverage
0.00	12.002	0.898	0.984	0.740	0.968
0.10	13.210	0.865	0.979	0.698	0.947
0.20	19.981	0.727	0.949	0.545	0.860
0.30	38.839	0.476	0.893	0.294	0.614
0.40	76.255	0.208	0.851	0.094	0.251

Comparison against LECF Table **??** compares FID, precision, recall, density, and coverage between our method and LECF across various clip filter thresholds. Our approach outperforms LECF in all metrics, indicating that while choosing an optimal threshold improves the baseline LECF results, our method excels at generating high-quality, diverse images.

Table 9. Our method outperforms LECF in all metrics, suggesting that while selecting the optimal threshold enhances baseline LECF outcomes, our approach excels in generating higher quality and more diverse images.

Name	FID	Precision	Recall	Density	Coverage
LECF (threshold=0.0)	40.852	0.552	0.415	0.585	0.431
LECF (threshold=0.1)	40.858	0.552	0.431	0.591	0.432
LECF (threshold=0.3)	38.107	0.576	0.416	0.626	0.445
LECF (threshold=0.5)	37.061	0.589	0.413	0.641	0.449
LECF (threshold=0.7)	35.950	0.602	0.412	0.663	0.464
LECF (threshold=0.9)	34.522	0.631	0.416	0.708	0.477
LECF (threshold=0.95)	33.606	0.648	0.392	0.731	0.486
LECF (threshold=0.97)	33.224	0.664	0.381	0.756	0.490
Diffusion Inversion (Ours)	17.678	0.831	0.661	0.732	0.787

C.4. Gaussian Noise

We investigate the influence of Gaussian noise on model performance by adjusting the noise strength and setting the latent interpolation strength α to 0. Figure 7c demonstrates the relationship between Gaussian noise strength and model test accuracy. Our findings indicate that the optimal performance is achieved when generating a dataset of equal size to the original without noise perturbation. However, when sampling additional data, it is advantageous to increase the noise strength accordingly, with a noise strength of $\lambda = 0.2$ as a suitable starting point.

Figure 10 presents the generated images at varying noise levels, showing minimal differences between perturbed and original images when the noise level is below $\lambda = 0.2$. Nonetheless, significant variations are observed at higher noise levels, such as the ship image remaining discernible at $\lambda = 0.4$, while the horse becomes indistinguishable. Ideally, we may want to employ distinct Gaussian noise strengths for each image rather than using a single fixed value for all.

Figure 10. Generate image variants by perturbing the embedding vector using random Gaussian noise. Noise strength λ from left to right: 0.0, 0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40. The generated images of some embeddings are still meaningful under high strength.