Using Synthetic Data for Data Augmentation to Improve Classification Accuracy

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

Abstract

Obtaining high quality data for training classification models is challenging when sufficient data covering the real manifold is difficult to find in the wild. In this paper, we present Diffusion Inversion, a dataset-agnostic augmentation strategy for training classification models. Diffusion Inversion is a simple yet effective method that leverages the powerful pretrained Stable Diffusion model to generate synthetic datasets that ensure coverage of the original data manifold while also generating novel samples that extrapolate the training domain to allow for better generalization. We ensure data coverage by inverting each image in the original set to its condition vector in the latent space of Stable Diffusion. We ensure sample diversity by adding noise to the learned embeddings or performing interpolation in the latent space, and using the new vector as the conditioning signal. The method produces high-quality and diverse samples, consistently outperforming generic prompt-based steering methods and KNN retrieval baselines across a wide range of common and specialized datasets. Furthermore, we demonstrate the compatibility of our approach with widely-used data augmentation techniques, and assess the reliability of the generated data in both 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 models such as GPT-3 (Brown et al., 2020), DALL-E (Ramesh et al., 2022),

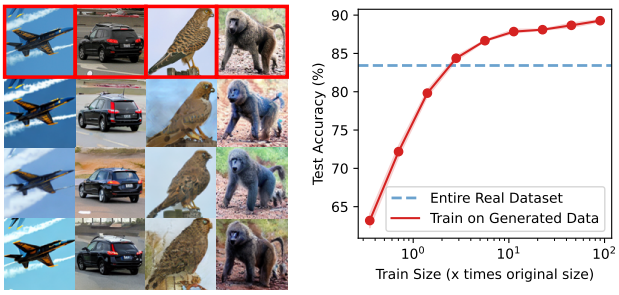

Figure 1. (Left) Top row: 4 real images from STL10 dataset. Bottom rows: For each real image, 3 images generated from Stable Diffusion using learned inverted embedding and 3 random initial noises. (Right) The test accuracy of ResNet18 trained on synthetic images improves as more data is generated and eventually surpasses the model trained on the real STL-10 dataset.

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. 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? Due to the limited diversity in samples generated by previous approaches (Antoniou et al., 2017; He et al., 2022; Gowal et al., 2021; Bansal & Grover, 2023), 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. State-of-the-art diffusion-based text-to-image models like Stable Diffusion demonstrate remarkable capabilities in synthesizing images with high visual fidelity, while maintaining good diversity as they prove to not suffer from mode collapse.

A natural method for using these models to augment the original dataset involves prompt-based generation, which allows a combination of domain expert knowledge and language enhancement techniques (He et al., 2022; Yuan et al., 2022) to produce a diverse array of high-fidelity images from diversified text prompts. 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), even if CLIP filtering (He et al., 2022) is

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

Workshop on Challenges in Deployable Generative AI at International Conference on Machine Learning (ICML), Honolulu, Hawaii, USA. 2023.

used. Specifically, these methods disregard the distribution of the train set, leading to the creation of distributionally dissimilar images from the original data and a significant gap between real and synthetic datasets (Borji, 2022).

To address the challenges in deploying generative models for real-world classification, 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 5). 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 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. Our method offers sample diversity and 6.5x reduction in sampling time
- 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 data distribution shifts from Stable Diffusion training data. Additionally, our generated data can enhance various neural architectures and boost few-shot learning performance

2. Method

Stable Diffusion (Rombach et al., 2022), a model trained on billions of image-text pairs, boasts a wealth of generalizable knowledge. To harness this knowledge for specific classification tasks, we propose a two-stage method that

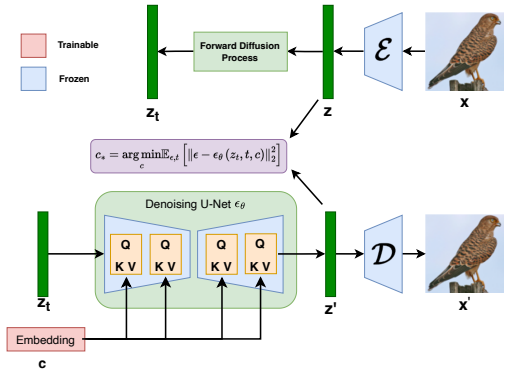

Figure 2. Our method optimizes the standard denoising objective to learn a set of embedding vectors while keeping the model parameters fixed.

guides 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. Then, we produce novel variants by running the inverse diffusion process conditioned on perturbed versions of these vectors. We illustrate our approach in Figure 2.

2.1. Stage 1 - Embedding Learning

Stable Diffusion Stable Diffusion is a type of Latent Diffusion Model (LDM), which is a class of Denoising Diffusion Probabilistic Models. LDMs operate in an autoencoder's latent space and have two main components. First, an autoencoder is pre-trained on a large image dataset to minimize reconstruction loss, using regularization from either KL-divergence loss or vector quantization (Van Den Oord et al., 2017; Agustsson et al., 2017). This 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 converts these latents back into images, such that $D(\mathcal{E}(x)) \approx x$. Next, a diffusion model is trained to minimize the denoising objective in the derived latent space, incorporating conditioning optionally on 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 sampled from the standard gaussian, ϵ_θ denotes the denoising network, and $c_\theta(y)$ is a model mapping conditioning input y to a conditioning vector. During inference, a new image latent z_0 is generated by iteratively denoising a random noise vector with a conditioning vector, and the latent code is transformed into an image using the pre-trained decoder $x' = D\left(z_0\right)$.

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 treat 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.

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

Throughout the optimization process, we maintain the original LDM model's training scheme and keep the denoising model ϵ_θ unchanged to optimally maintain pre-training knowledge. Furthermore, we improve sampling efficiency by learning condition vectors tailored to generate target-resolution images, instead of creating high-resolution images and subsequently downsampling, thereby considerably reducing the overall generation time (see Figure 8).

2.2. Stage 2 - Sampling

Classifier-free Guidance Classifier-free guidance employs a weight parameter $w \in \mathcal{R}$ to balance sample quality and diversity in class-conditioned diffusion models, commonly used in large-scale models such as Stable Diffusion (Rombach et al., 2022), GLIDE (Nichol et al., 2021), and Imagen (Saharia et al., 2022). During sample generation, both the conditional diffusion model $\epsilon_\theta\left(z_t, t, c\right)$ and the unconditional model $\epsilon_\theta\left(z_t, t\right)$ are evaluated. In Stable Diffusion, the conditioning vector for unconditional image generation is determined by the text encoder's output for an empty string, with the model output at each denoising step given by $\hat{\epsilon} = (1+w)\epsilon_\theta\left(z_t, t, c\right) - w\epsilon_\theta\left(z_t, t\right)$. However, we find that using an empty string as conditioning input is ineffective for the target domain when the data distribution deviates significantly from the training distribution, particularly when image resolution varies. To address this distribution shift, we instead utilize the average embedding of all learned vectors as the class-conditioning input for unconditional models, with the effectiveness of this design demonstrated in Section D.8.2.

Sample Diversity Sample diversity is crucial for training downstream classifiers on synthetic data (Ravuri & Vinyals, 2019a). To achieve this, we employ various classifier-free guidance strengths and initiate the denoising process with different random noises, generating distinct image variants. We also explore two conditioning vector perturbation methods, namely Gaussian noise perturbation and latent interpolation. In the Gaussian approach, we add isotropic Gaussian noise to the conditioning vector, yielding a new vector $\hat{c} = c + \lambda\epsilon$, where $\epsilon \sim \mathcal{N}(0,1)$ and λ indicates the perturbation strength. For latent interpolation, we linearly interpolate between two conditioning vectors c_1 and c_2 to create a new vector: $\hat{c} = \alpha c_1 + (1-\alpha)c_2$, where higher values of α push $\hat{c}$ towards c_1, lower values push towards c_2, and $0 \leq \alpha \leq 1$ typically. We assess each component's impact in Section D.8.3.

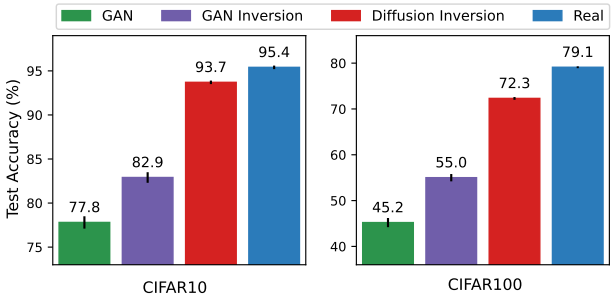

Figure 3. Our method, Diffusion Inversion, dramatically surpasses both GAN and GAN Inversion methods when trained on datasets of equivalent size to the original real dataset, underscoring the importance of a high-quality pre-trained generator.

3. Experimental Results

We employ the Stable Diffusion model with a default resolution of 512x512 [1]. To optimize learning and sampling efficiency, we directly learn the embedding to generate images at target resolutions of 128x128 for low-resolution datasets (e.g., CIFAR10/100, MedM- NISTv2) and 256x256 for other datasets. This modification significantly reduces image generation time by 27x and 6.5x for 128x128 and 256x256 settings, respectively, making our method more suit- able for data augmentation We provide a detailed runtime analysis in Appendix D.1. For evaluation, we resize all generated images to match the resolution of the original real images, ensuring a fair comparison between models trained on real and synthetic data. Experimental details are provided in Appendix C. Training and Sampling design choices are described in Appendix D.8.

3.1. Generator Quality and Data Size Matter

Generator Quality To investigate the influence of generator quality on producing high-quality datasets for downstream classifier training, we initially compare our approach to the GAN Inversion method (Abdal et al., 2019) on CIFAR10/100. Using a pre-trained BigGAN model from (Zhao & Bilen, 2022), we generate three synthetic datasets, each containing 50K examples, equivalent to the original dataset size. The datasets are created using random latent vectors, GAN Inversion, and our method. To evaluate the datasets' quality, we train a ResNet18 on each and report the mean and standard deviation of the test accuracy using five random seeds.

As depicted in Figure 3, our method exhibits superior performance in comparison to GAN approaches, indicating that the quality of the pre-trained generator is crucial for generating high-quality datasets for discriminative models. How-

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

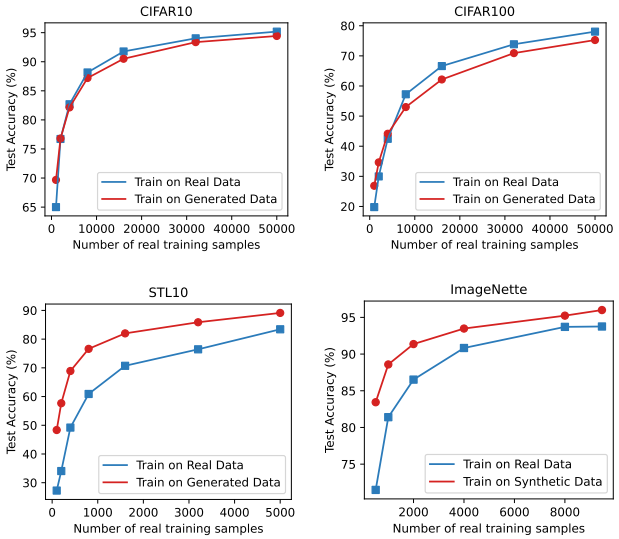

Figure 4. Performance in Relation to Number of Real Data Points. Our approach demonstrates substantially improved performance in low-data scenarios across all datasets. In high-data scenarios, it exhibits comparable performance for low-resolution datasets and superior performance for high-resolution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

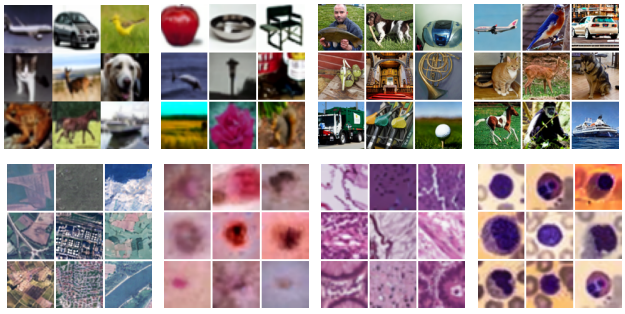

Figure 5. Synthetic images generated by our method. Left to Right (top): CIFAR10, CIFAR100, ImageNette, STL-10. Left to Right (bottom): EuroSAT, DermaMNIST, PathMNIST, BloodMNIST.

ever, the information is more condensed in the real dataset as it achieves higher accuracy with equal-size datasets.

Scaling in Number of Real Data Next, we assess the scalability of our approach by evaluating its benefits for downstream classifier training using four datasets. We generate a sufficient number of synthetic images and learn embeddings from real datasets with varying numbers of real training examples. For each embedding, we create 45 unique variants and train a ResNet18 on derived datasets.

Figure 4 demonstrates that our method outperforms real data in low data regimes (2K for CIFAR10 and 4K for CIFAR100) for low-resolution datasets like CIFAR10/100 but is slightly worse when more real training data is available. Conversely, for high-resolution datasets such as STL10 and ImageNette, our method consistently surpasses real data by a significant margin. For example, it improves test accuracy on STL10 from 83.3 to 89.0 and on Imagenette from 93.8 to 95.4, using 2-3x less real data.

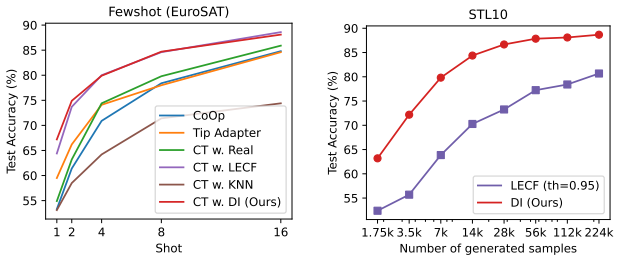

Figure 6. (Left) Our method improves few-shot learning performance, yielding results similar to LECF. (Right) Our method demonstrates significantly better scalability than LECF on STL10.

Scaling in Number of Synthetic Data 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), increased data consistently improves downstream classifier performance, surpassing the real dataset when roughly 3x more data is generated. This scaling trend indicates that extended training time and online data generation could further enhance model performance.

3.2. 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), employs Stable Diffusion to generate data for discriminative models, demonstrating cutting-edge performance in few-shot learning. We compared our approach to LECF in two distinct settings: few-shot learning on EuroSAT (Helber et al., 2019) and standard training on STL10.

We 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) on the EuroSAT dataset. As depicted in Figure 6 (Left), our approach enhances few-shot learning performance, achieving results comparable to LECF. For the STL10 dataset, we analyzed test accuracy progression concerning the number of generated data points. Training a ResNet18 exclusively on generated data and adjusting the Clip Filtering strength of LECF within [0.0, 0.1, 0.3, 0.5, 0.7, 0.9, 0.95, 0.97], we determined that 0.95 yielded optimal performance. Figure 6 (Right) highlights our method's superior scaling capabilities compared to LECF, owing to its consideration of domain shifts and improved coverage. This is further evidenced in Table 8, comparing FID, precision, recall, and coverage between our method and LECF at the different Clip Filtering strengths.

Comparison with KNN Retrieval on LAION Dataset Stable Diffusion, trained on the LAION dataset (Schuhmann et al., 2022), prompted the question of synthetic dataset necessity versus similar image retrieval for data augmentation. We assessed this using KNN retrieval from LAION with clip retrieval on the STL10 dataset. Test accuracies achieved

Table 1. 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 2. 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**

were 85.4%, 88.4%, and 90.9% for k=10, 25, and 50, respectively. Our method generated 88.7% test accuracy with 45 data points per embedding, slightly surpassing 25-image retrieval but not 50-image retrieval. This suggests KNN retrieval as a strong baseline for target classes like airplanes, cars, and dogs in Stable Diffusion training distribution.

However, we argue that this method falls short when significant distribution shifts occur between target and source domains, especially in specialized fields like medical imaging. To demonstrate this, we analyzed three distinct MedMNISTv2 (Yang et al., 2023) datasets: PathMNIST, DermaMNIST, and BloodMNIST. As depicted in Table 2, our approach consistently surpasses the KNN retrieval baseline. It is crucial to acknowledge that LECF would falter in this scenario due to significant distribution shifts and challenges in creating effective prompts. Additionally, KNN retrieval does not improve few-shot learning performance on EuroSAT, as demonstrated in Figure 6 (Left). The high-quality generated images for these specialized domain datasets, depicted in Figure 5, closely resemble the original dataset, underscoring the importance of a steering method that addresses distribution shift and data coverage.

3.3. Comparison against Data Augmentation Methods

We evaluate our approach against widely-used data augmentation techniques for image classification on STL10. These encompass standard methods such as AutoAugment (Cubuk et al., 2018), RandAugment (Cubuk et al., 2020), and CutOut (DeVries & Taylor, 2017); interpolation-based techniques like MixUp (Zhang et al., 2017), CutMix (Yun et al., 2019), and AugMix (Hendrycks et al., 2019); as well as the adversarial domain augmentation (ADA) method ME-ADA (Zhao et al., 2020a). A description of each technique is provided in Appendix C.2.3. As shown in Table 1, 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, combining our approach with other augmentation techniques can further improve performance.

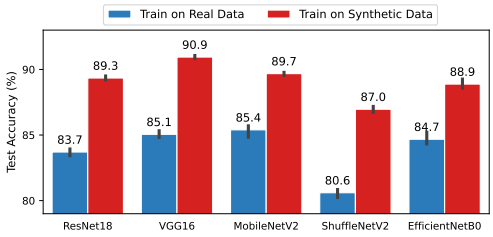

Figure 7. Our method's synthetic dataset significantly enhances the performance of various neural architectures on the STL10 dataset.

3.4. Evaluation on various Neural Architectures

To evaluate the efficacy of our generated data, we examine its performance across several neural network architectures, namely 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 7, synthetic images significantly improve performance across all tested architectures, indicating that our method successfully extracts generalizable pre-trained knowledge.

4. Conclusion

Our method, Diffusion Inversion, generates high-quality synthetic data that boosts image classification performance on several real datasets by leveraging the Stable Diffusion model. Our method effectively addresses the challenges of data distribution shift and data coverage, surpassing conventional prompt-based steering approaches and prevalent data augmentation techniques. Impressively, our synthesized images can supplant original datasets, resulting in sample complexity and sampling time improvements. Our study highlights the potential of utilizing pre-trained generative models for data augmentation, especially in domains where data acquisition and curation are costly and labor-intensive.

Limitations Although our method significantly reduces total generation time (Figure 8), scaling it to large-scale datasets like ImageNet (Russakovsky et al., 2015) presents challenges due to storage requirements and inefficient sampling of Stable Diffusion. Incorporating fast sampling techniques (Meng et al., 2022) represents a promising direction for maximizing the impact of diffusion-based data generation for discriminative models. We discuss the societal impact of using such models in the real-world in Appendix A.

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.

Azizi, S., Kornblith, S., Saharia, C., Norouzi, M., and Fleet, D. J. Synthetic data from diffusion models improves imagenet classification. *arXiv preprint arXiv:2304.08466*, 2023.

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.

Meng, C., Gao, R., Kingma, D. P., Ermon, S., Ho, J., and Salimans, T. On distillation of guided diffusion models. *arXiv preprint arXiv:2210.03142*, 2022.

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.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 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.

Sariyildiz, M. B., Alahari, K., Larlus, D., and Kalantidis, Y. Fake it till you make it: Learning transferable representations from synthetic imagenet clones. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

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. Related Work

Utilizing Generative Models for Image Data Augmentation 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, several works (Shrivastava et al., 2017; Viazovetskyi et al., 2020; Zhu et al., 2017) 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 (Bansal & Grover, 2023; Gowal et al., 2021; Ravuri & Vinyals, 2019a; Zhao & Bilen, 2022).

Augmenting Image Data using Text-to-Image Models Recently, there has been a growing interest in leveraging the power of internet-scale pre-trained diffusion-based models (Rombach et al., 2022; Nichol et al., 2021) 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, several works (Bansal & Grover, 2023; Yuan et al., 2022; Sariyildiz et al., 2023) illustrate that Stable Diffusion (Rombach et al., 2022) can serve as a data augmentation tool to learn generalizable features and 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 between synthetic and real data in downstream tasks may continue to hinder the improvement of synthetic data's effectiveness in classifier learning (He et al., 2022; Bansal & Grover, 2023). To enhance the alignment of the text-to-image model with the downstream dataset, Azizi et al. (2023) suggests fine-tuning the model weights and sampling parameters while retaining the text prompts as concise one or two-word class names from (Radford et al., 2021). Nonetheless, generative diversity and data coverage may still present obstacles, resulting in the generation of data that is inferior to real data. In contrast, our approach addresses these issues by directly learning the conditioning vector for each target image and producing new variants by conditioning on noisy versions of these vectors. This method eliminates the need for human prompt engineering, guarantees data coverage, and promotes diversity.

Inversion Techniques in Generative Models Inverting generative models plays a crucial role in image editing and manipulation tasks (Zhu et al., 2016; Xia et al., 2022; Creswell & Bharath, 2018). 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 through new pseudo-words in the embedding space of a frozen text-to-image model. In contrast, 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.

C. Experimental Details

C.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. 2.1 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.

C.2. Experimental Setups

C.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.

C.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.

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

C.2.3. Comparison against Image Data Augmentation Methods

i) **AutoAugment (Cubuk et al., 2018):** 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 (Cubuk et al., 2020):** Similar to AutoAugment, we randomly select two operations from a list of 14 and apply them with certainty.

iii) **CutOut (DeVries & Taylor, 2017):** Our CutOut implementation masks out a random square region, sized at 1/8 of the input image.

iv) **MixUp (Zhang et al., 2017):** 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.

v) **CutMix (Yun et al., 2019):** 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.

vi) **AugMix (Hendrycks et al., 2019):** 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.

vii) **ME-ADA (Zhao et al., 2020a):** 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).

D. Additional Results

D.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.

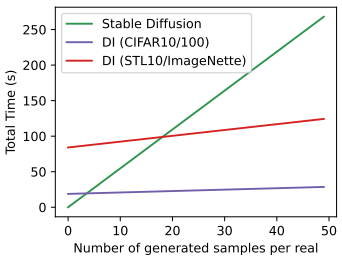

Figure 8. Despite the overhead incurred by embedding learning, our method substantially decreases the overall time required to generate numerous images due to improved sampling.

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. This is shown in Figure 8. Ideally, we want to generate data on-the-fly during training, which further supports the efficiency of our method.

D.2. Combine real and generated data

In our study, we analyze a model trained on a combination of generated data and synthetic data. We employ a straightforward approach to merge the real and synthetic data. Specifically, at each gradient step, we construct a batch using a mixture of synthetic and real data and train the model following the same protocol as when training on either generated data only or real data only (e.g., optimizer, training steps, and batch size). We experiment with varying the real-to-synthetic mixture ratio from [1:7, 1:3, 1:1, 3:1, 7:1] and report the best performance in Table 3. Our observations indicate that although generated data underperforms real data on low-resolution datasets such as CIFAR10, CIFAR100, and PathMNIST, combining both types enhances performance, as also observed by Azizi et al. (2023). However, in some instances, like Imagenette, DermaMNIST, and BloodMNIST, the combination leads to a slight performance decrease compared to using real or generated data alone. A similar observation was made by Ravuri & Vinyals (2019a) (Fig. 5), where they found that mixing generated samples with real data degrades Top-5 classifier accuracy for almost all models tested. Concurrently, Azizi et al. (2023) (Table 4) notes that model performance with higher resolution images does not continue to improve with larger amounts of generative data augmentation after a certain point. This may be attributable to the bias in the generated data inherited from the generative models, suggesting that a more sophisticated method for merging real and generated data is necessary. We leave the comprehensive study on how to combine the real and generated data for future work.

Table 3. Test accuracy of ResNet18 trained on real data only, generated only, and a combination of real and generated data.

	CIFAR-10	CIFAR-100	STL10	Image-nette	Path-MNIST	Derma-MNIST	Blood-MNIST
Real Only	95.1	77.9	83.3	93.8	89.6	**67.5**	**96.4**
Generated Only	94.6	74.4	89.0	**95.4**	82.1	**67.5**	93.7
Real + Generated	**95.2**	**78.0**	**90.0**	95.0	**92.1**	66.3	95.7

D.3. 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 4 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.

Table 4. 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	$\mathbf{94.6 \pm 0.1}$	$\mathbf{74.4 \pm 0.3}$

D.4. 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). This is shown in Table 5.

Table 5. 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	$\mathbf{95.1 \pm 0.0}$	94.6 ± 0.1	94.7 ± 0.1
CIFAR100	$\mathbf{77.9 \pm 0.4}$	74.4 ± 0.3	75.2 ± 0.2
STL-10	83.3 ± 0.7	$\mathbf{89.0 \pm 0.2}$	88.8 ± 0.2
ImageNette	93.8 ± 0.2	95.4 ± 0.1	$\mathbf{95.6 \pm 0.1}$

D.5. 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 6 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.

Table 6. 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

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 in individual image quality. However, Figure 11 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 7. 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 8 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 8. 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

D.6. Scaling Capabilities of Diffusion Inversion

As illustrated in Table 9, 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.

Table 9. 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

D.7. 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 11 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 9 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.

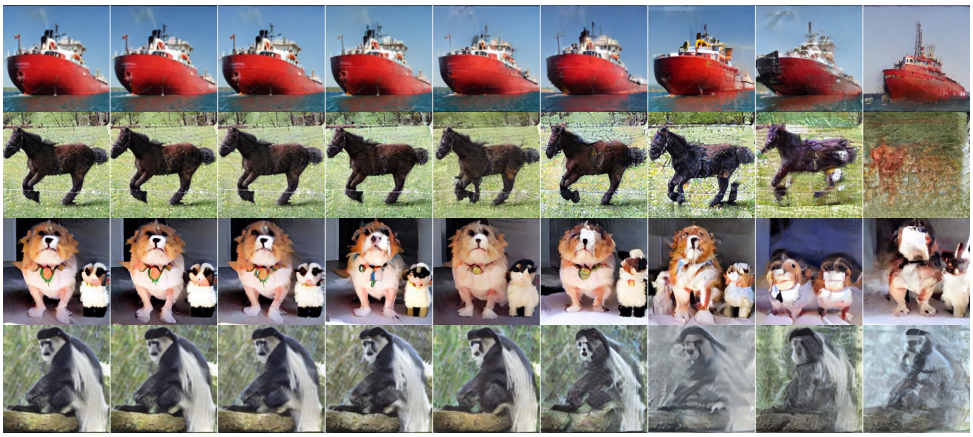

Figure 9. Generate image variants by perturbing the embedding vector using random Gaussian noise. Noise strength λ from left to right: 0.0, 0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40. The generated images of some embeddings are still meaningful under high strength.

D.8. Hyperparameter Settings

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

D.8.1. Training Steps and Classifier-free Guidance Strength

Figure 10 (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.

D.8.2. Inference Steps and Unconditional Embedding

Figure 10 (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.

D.8.3. 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 11 (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 11 (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 14 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.

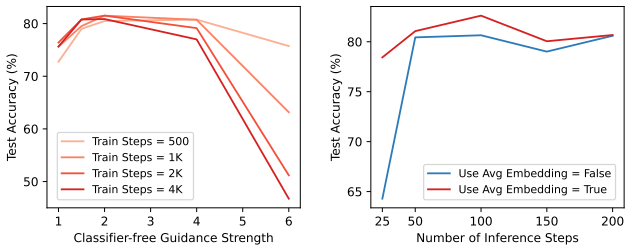

Figure 10. The effect of the number training steps & Classifier-free guidance strength (Left) and inference steps & Unconditional embedding (Right) on model performance.

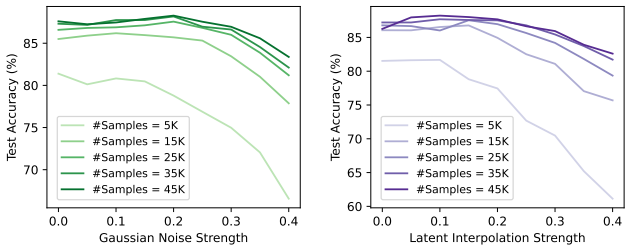

Figure 11. The effect of Gaussian Noise (Left) without any latent interpolation and Latent Interpolation (Right) without any Gaussian noise on the performance as we generate more data.

D.9. Additional Visualization

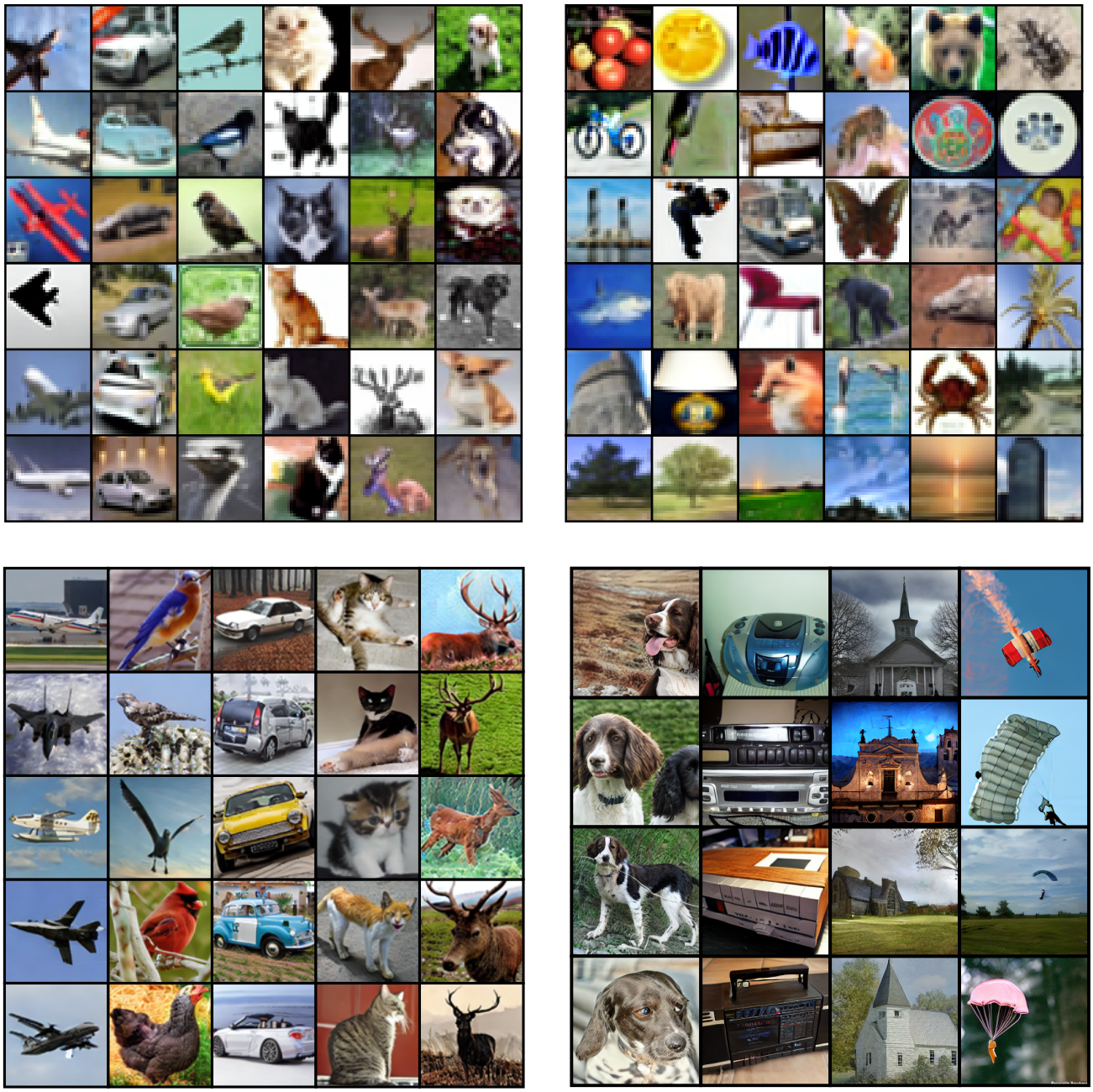

Figure 12. Synthetic images produced by our method: exhibiting diversity, realism, and comprehensive representation of the original dataset, effectively serving as a suitable substitute. From top left, going clockwise: CIFAR10, CIFAR100, Imagenette, STL10

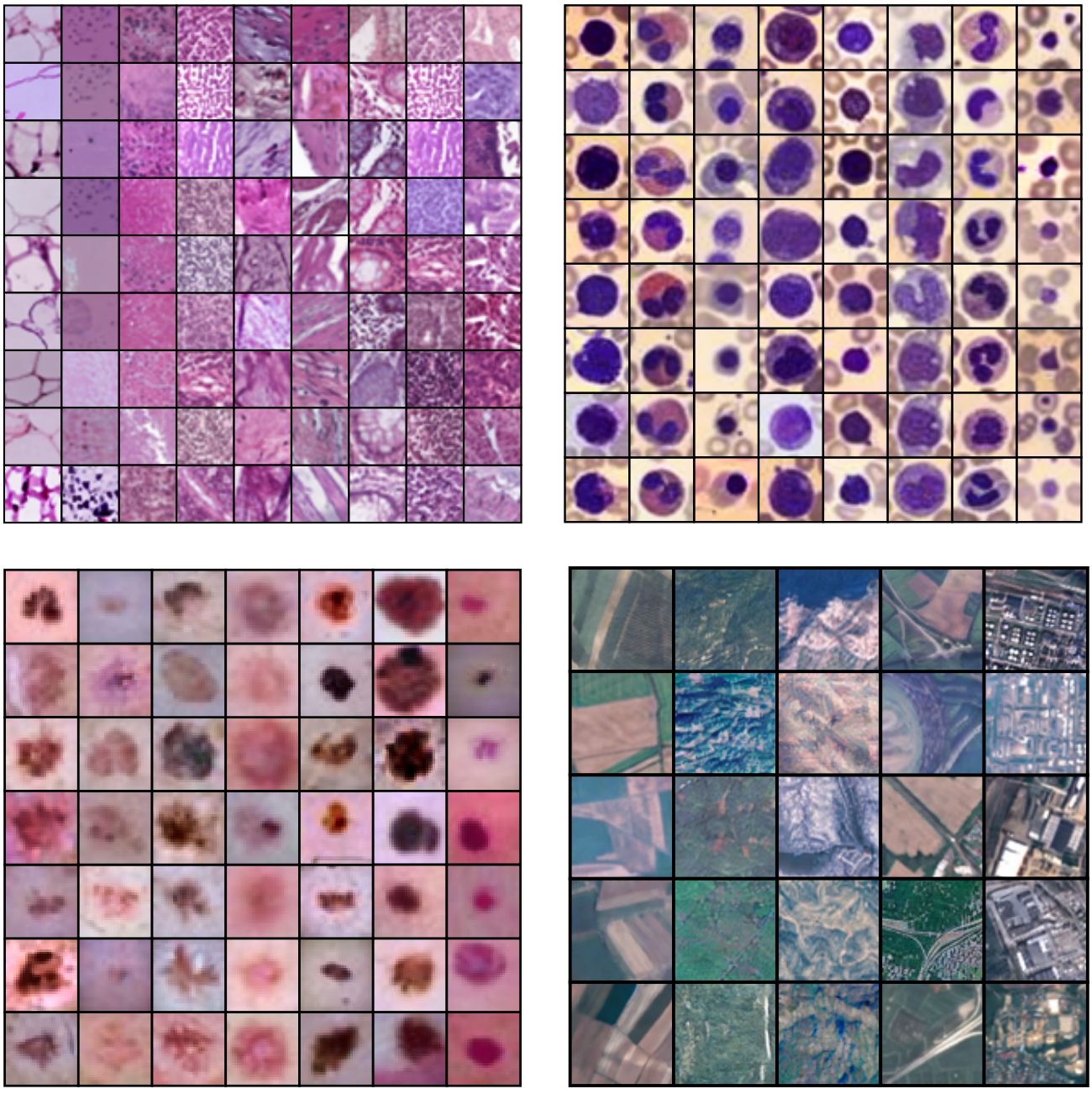

Figure 13. Synthetic images produced by our method: exhibiting diversity, realism, and comprehensive representation of the original dataset, effectively serving as a suitable substitute. From top left, going clockwise: PathMNIST, BloodMNIST, EuroSAT, DermaMNIST

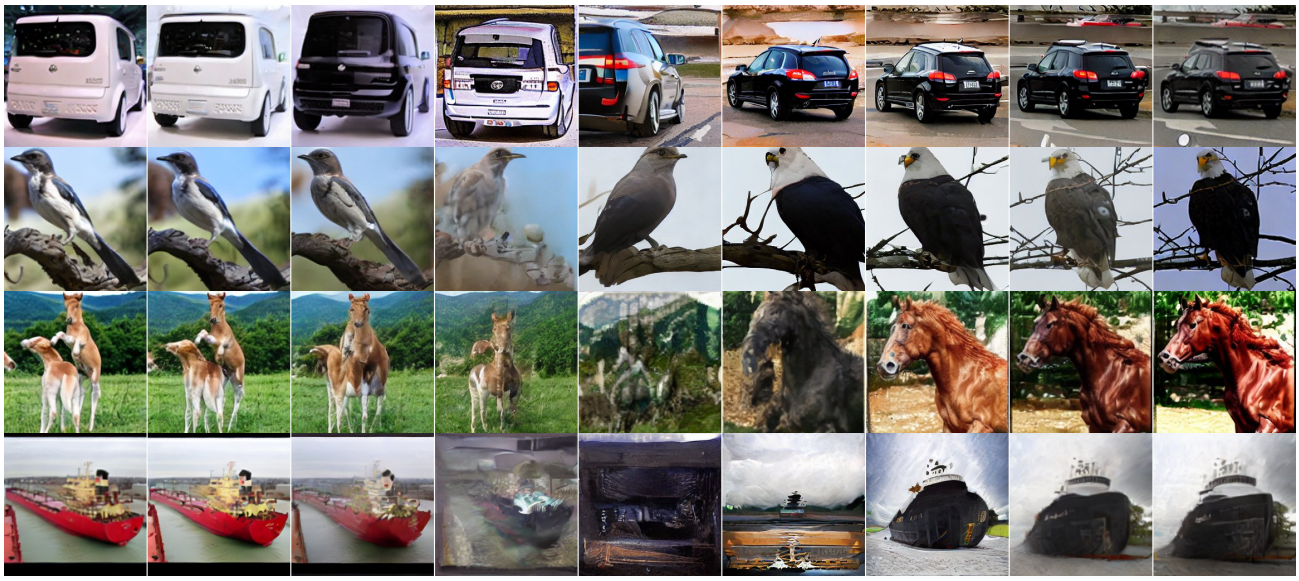

Figure 14. Generate image variants by interpolating two embedding vectors. From left to right, interpolation strength α: 0.0, 0.1, 0.2, 0.3, 0.5, 0.7, 0.8, 0.9, 1.0. Some pairs of the embedding vectors can generate novel and natural images regardless of the chosen interpolation strength, while others only work when the interpolation strength is small.