Difficulty Controlled Diffusion Model for Synthesizing Effective Training Data

Zerun Wang[1,2*], **Jiafeng Mao**[1], **Xueting Wang**[1†], **Toshihiko Yamasaki**[2]

[1]CyberAgent
[2]The University of Tokyo
{ze_wang, yamasaki}@cvm.t.u-tokyo.ac.jp, {jiafeng_mao, wang_xueting}@cyberagent.co.jp

Abstract

Generative models have become a powerful tool for synthesizing training data in computer vision tasks. Current approaches solely focus on aligning generated images with the target dataset distribution. As a result, they capture only the common features in the real dataset and mostly generate "easy samples", which are already well learned by models trained on real data. In contrast, those rare "hard samples", with atypical features but crucial for enhancing performance, cannot be effectively generated. Consequently, these approaches must synthesize large volumes of data to yield appreciable performance gains, yet the improvement remains limited. To overcome this limitation, we present a novel method that can learn to control the learning difficulty of samples during generation while also achieving domain alignment. Thus, it can efficiently generate valuable "hard samples" that yield significant performance improvements for target tasks. This is achieved by incorporating learning difficulty as an additional conditioning signal in generative models, together with a designed encoder structure and training–generation strategy. Experimental results across multiple datasets show that our method can achieve **higher performance** with **lower generation cost**. Specifically, we obtain the best performance with only 10% additional synthetic data, saving 63.4 GPU hours of generation time compared to the previous SOTA on ImageNet. Moreover, our method provides insightful visualizations of category-specific hard factors, serving as a tool for analyzing datasets.

Code — https://github.com/komejisatori/Difficulty-Aware-Synthesis

Introduction

Manually collecting and annotating a large number of images for training visual task models is time-consuming and labor-intensive. Recently, the rapid advancement of image generation models (Dhariwal and Nichol 2021; Ho, Jain, and Abbeel 2020) offers a promising way to synthesize new training data automatically.

Training data synthesis methods (Sarıyıldız et al. 2023; Vendrow et al. 2023; Zhou, Sahak, and Ba 2023) have successfully enhanced model performance by augmenting the original datasets with images synthesized by generation models. A general pipeline is to use text-to-image generation models with text prompts related to target class names to generate training data.

Images generated by off-the-shelf diffusion models, however, often suffer from a distribution mismatch with the tar-

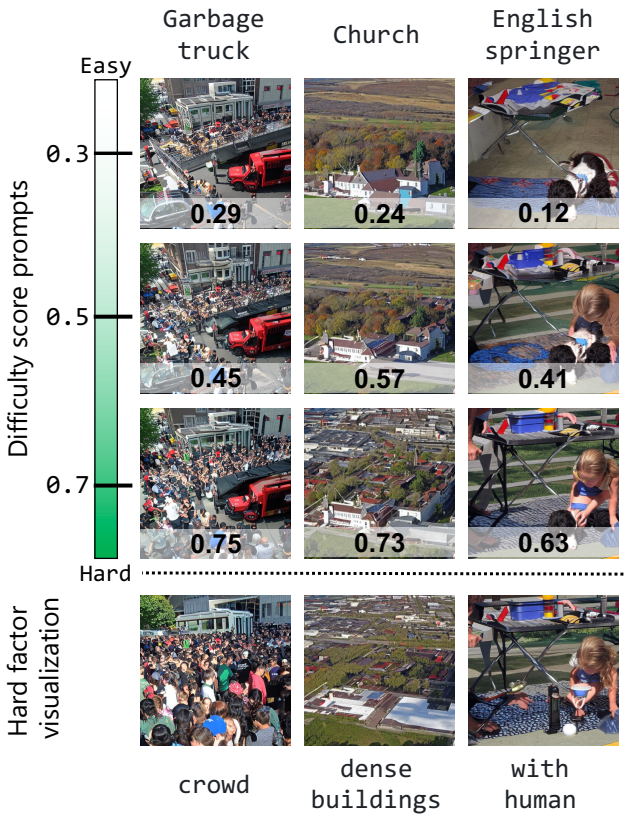

Figure 1: Our method generates images with controllable learning difficulty that align with specified difficulty score prompts. The score on each image is computed by a pre-trained classifier. Additionally, our method reveals and visualizes the factors that contribute to sample difficulty.

*Work done during the internship at CyberAgent.
†Corresponding Author.

get dataset, reducing their effectiveness for training. Recent work (Yuan et al. 2024) addressed this issue by fine-tuning generation models on target real datasets to align distributions. However, this approach introduces new limitations: (1) The fine-tuned model tends to generate easy samples that reflect only the dominant features of the target dataset. These synthetic samples have low learning difficulty, as similar real images are already abundant in the dataset. Consequently, they contribute less to improving the target task. (2) The generation of hard samples is extremely limited. However, these samples, which have higher learning difficulty, are often more effective for enhancing target task performance because they contain minority or atypical features of the target dataset. Thus, a dilemma arises in current methods: without fine-tuning, generated images suffer from distribution mismatch, while aligning domains through fine-tuning leads to primarily generating samples with low learning difficulty.

We address this dilemma by introducing learning difficulty as an additional conditioning signal during fine-tuning and generation. The model can thus capture the features of samples with different learning difficulties while still maintaining domain alignment. We fine-tune the pretrained text-to-image model using both difficulty score prompts and text prompts as conditioning inputs. Then, the model can generate new samples at specified difficulty levels by adjusting the input difficulty score prompt. Our approach can disentangle learning difficulty from the domain alignment process, enabling the generation of samples with varying difficulties while maintaining domain consistency.

Extensive experiments across multiple image classification datasets demonstrate the effectiveness of our proposed method in generating samples with controllable learning difficulty, thereby improving the performance of the target task more efficiently. Furthermore, our method enables the visual analysis of class-wise hard factors, providing insights into what makes certain class samples difficult in the target dataset. Our contribution can be summarized as follows,

- We show that domain alignment alone causes models to generate mostly easy samples, which provide only limited performance gains.
- We propose a difficulty-controlled generation framework that can be used to synthesize samples with targeted learning difficulty, thus improving performance on target tasks more efficiently.
- We validate our method across multiple image classification tasks, demonstrating its effectiveness in both providing valuable training samples and capturing factors that affect sample difficulty for visualization.

Related Work

Conditioned Image Generation

Diffusion models are one of the mainstream tools for generating images (Dhariwal and Nichol 2021; Ho, Jain, and Abbeel 2020; Ho and Salimans 2021; Nichol and Dhariwal 2021; Song, Meng, and Ermon 2021; Liu et al. 2022a). Starting from Gaussian noise, these models iteratively predict the noise to be removed at each step, gradually denoising samples to obtain high-fidelity outputs. Many contemporary methods (Kim, Kwon, and Ye 2022; Ramesh et al. 2021; Ding et al. 2021; Gafni et al. 2022) leverage text prompts encoded by CLIP (Radford et al. 2021) to guide the denoising process. Notably, Latent Diffusion (Rombach et al. 2022) conducts denoising in a latent space before decoding the denoised latents into pixel space. Various approaches have been explored to enhance control over the generation process through different conditioning signals, such as image-based guidance (Mou et al. 2023; Kawar et al. 2023; Ruiz et al. 2023; Brooks, Holynski, and Efros 2023; Avrahami, Lischinski, and Fried 2022; Wang et al. 2022), compositional conditioning (Liu et al. 2022b; Park et al. 2021; Huang et al. 2023), and layout guidance (Mao, Wang, and Aizawa 2023; Zhang, Rao, and Agrawala 2023; Li et al. 2023b; Voynov, Aberman, and Cohen-Or 2023). In contrast to these works, we propose a novel form of generation guidance using a score reflecting the learning difficulty of training samples.

Training Data Synthesis

Synthetic images have proven effective for serving as new training data in deep learning vision tasks, thereby saving the cost of collecting and labeling real-world data. Early works (Zhang et al. 2021; Besnier et al. 2020) utilize generative adversarial network (GAN)-based models. Recent works apply more powerful diffusion models for data synthesis. Several approaches directly utilize off-the-shelf pretrained diffusion models: Sariyildiz *et al.* (Sarıyıldız et al. 2023) applied text prompt engineering strategies to improve the diversity of generated results. Huang *et al.* (Huang et al. 2024) augment misclassified real data by using them as image guidance for the diffusion model. Meanwhile, some works (Vendrow et al. 2023; Zhou, Sahak, and Ba 2023) leverage textual inversion techniques to encode class-specific characteristics from real data into new tokens. Recent methods point out the importance of aligning the distribution between synthetic and real data. This is achieved by fine-tuning the diffusion models using real data. Azizi *et al.* (Azizi et al. 2023) finetuned the Imagen (Saharia et al. 2022) model for data generation. Real-Fake (Yuan et al. 2024) theoretically shows that fine-tuning achieves domain alignment. They apply the more effective Low-Rank Adaptation (LoRA) (Hu et al. 2022) approach to fine-tune the Stable Diffusion (Rombach et al. 2022) model. These fine-tuning methods, however, tend to reproduce dominant features of the target dataset and consequently generate mostly easy samples. In contrast, our method can generate samples with appropriate difficulty in the target domain, thus further improving the performance.

Preliminary

In this section, we first introduce the setting of training data synthesis. Then, we introduce our investigation of the dilemma of previous training data synthesis methods.

Task Setting

Following previous methods (Azizi et al. 2023; Yuan et al. 2024), we evaluate our method on basic image classification

Data	Real only	+ Synthetic data		
		Easy	Medium	Extremely hard
Acc.	95.0	95.2 (+0.2)	**95.8 (+0.8)**	94.6 (−0.4)

Table 1: Classification accuracy when real data is augmented with synthetic data from different difficulty score ranges.

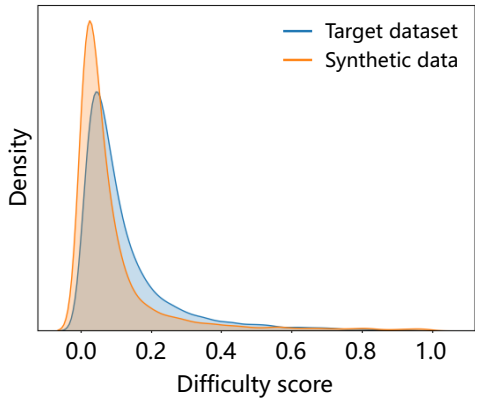

Figure 2: KDE distribution curve of difficulty scores. Simply fine-tuning on the whole target dataset biases the model toward generating easy images.

tasks. Training data synthesis leverages generative models, typically text-to-image diffusion models, to augment a target dataset $\mathcal{D}_t = \{\boldsymbol{x_i}, y_i\}_{i=1}^{n_t}$ with synthetic images and labels $\mathcal{D}_s = \{\boldsymbol{x'_i}, y'_i\}_{i=1}^{n_s}$, where $\boldsymbol{x_i}$ and $\boldsymbol{x'_i}$ denote the real and synthetic images. y_i, y'_i denote the labels for the classification task. The combined dataset $\mathcal{D} = \mathcal{D}_t \cup \mathcal{D}_s$ can be used to train a target classification model, which typically outperforms models trained solely on $\mathcal{D}_t$.

Investigation

Previous methods aligned the domain of $\mathcal{D}_s$ with $\mathcal{D}_t$ by fine-tuning the generative model. However, we investigated their generated data from the learning difficulty perspective and found that they are dominated by "easy samples".

Difficulty Score. We define a difficulty score s to represent the learning difficulty of a sample. Given $c \in (0, 1)$ as the predicted probability of the ground-truth class produced by a classifier after the softmax activation, the score is computed as

$$s = 1 - c. \tag{1}$$

Thus, a higher difficulty score leads to a sample with higher learning difficulty for the model and vice versa.

Dilemma of Current Methods. We then analyzed the difficulty score distribution and its impact on target-task performance using the recent Real-Fake method, and found that the generated samples are dominated by easy examples. We used Imagenette (Howard 2019) as the target classification dataset and applied a ResNet-50 (He et al. 2016) model pretrained on the training split to compute the difficulty score. Based on our experiments, we found that:

(1) Current methods mainly generate easy samples. We first compared the difficulty score distribution of $\mathcal{D}_t$ and $\mathcal{D}_s$ generated by Real-Fake. We generated the same number of images as the real training dataset and assessed the difficulty score. The Kernel Density Estimation (KDE) distributions in Fig. 2 show that the synthetic data distribution is more extreme and dominated by easy samples.

(2) Samples with appropriate difficulty are more effective. We further analyzed the relationship between the difficulty score and the improvement effect on the target task. We divided the difficulty score $s \in (0.0, 1.0)$ into three levels: Easy $(0, 0.33)$, Medium $(0.33, 0.66)$, and Extremely Hard$(0.66, 1)$. Then we generated and selected the same amount of data (25% of the real training set) for each difficulty level. The results in Table 1 show that synthetic data with a medium-level difficulty score yields the most significant training improvements. However, Real-Fake is **highly inefficient** at generating such samples as the medium-difficulty examples constitute only about 1% of all generated images (Fig. 2). As a result, a large amount of additional data needs to be generated to filter out a sufficient number of medium-difficulty samples. Meanwhile, using only extremely hard samples degrades training performance, highlighting the importance of controllable sample difficulty.

Difficulty Controlled Dataset Synthesis

Previous methods suffer from the above dilemma because their strategies, while achieving domain alignment, captured only the dominant features of the target dataset. Motivated by this observation, we propose a method that disentangles (1) domain alignment and (2) difficulty-aware feature modeling. We achieve this by introducing learning difficulty as an explicit conditioning to control the image generation process. Fig. 3 illustrates the structure of our model and the fine-tuning pipeline. For the model structure, we incorporate a difficulty encoder into a standard text-to-image diffusion model. During fine-tuning, the model is trained on data annotated with difficulty scores and text prompts. With our method, the learning difficulty of generated images can be controlled by adjusting the input difficulty score prompt.

Model Structure

We adopt the text-to-image Stable Diffusion (Rombach et al. 2022) model for image generation. The model consists of a CLIP text encoder and a denoising U-Net. Additionally, we introduce a difficulty encoder to condition the model on the input difficulty score.

Difficulty Encoder. To establish the mapping relationship between difficulty scores and the characteristics of samples, we construct a difficulty encoder, $\mathcal{E}_d$, which is a multilayer perceptron (MLP) model that projects the difficulty score of the ith sample into a latent embedding $\boldsymbol{h_i}$ for controlling the generation.

Although samples may share the same difficulty score, their visual characteristics can differ substantially across categories. Thus, the difficulty encoder must produce distinct embeddings for different categories. Therefore, our difficulty encoder takes as input the concatenation of the cate-

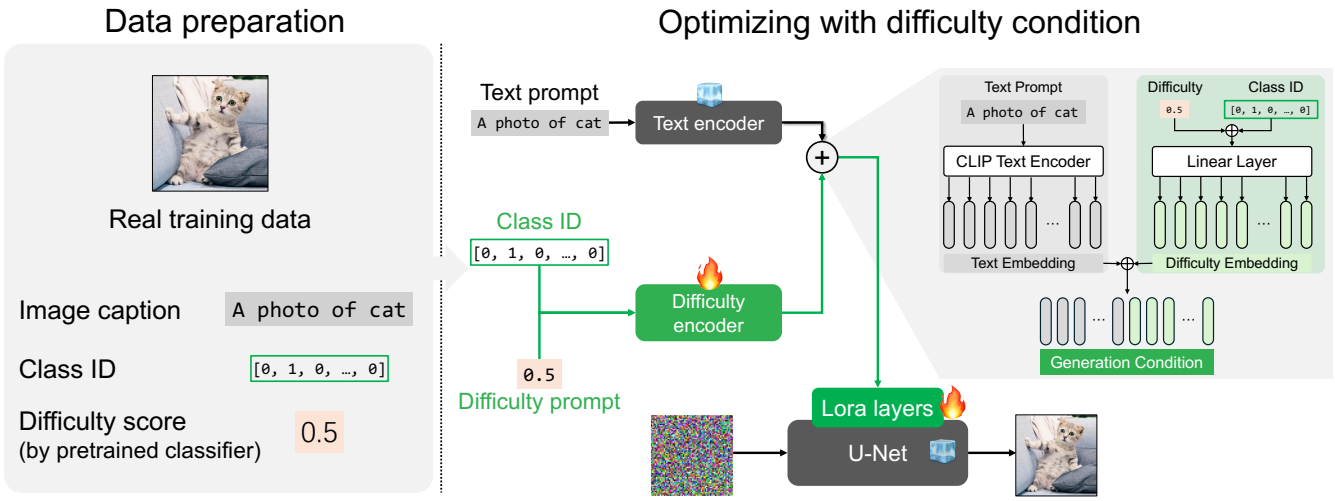

Figure 3: Overview of our method. **Left**: Real training images are annotated with a text caption and a difficulty score assessed by a pretrained classifier. **Right**: A difficulty encoder is integrated into the text-to-image diffusion model. The model is fine-tuned to incorporate the difficulty score as an additional condition.

gory labels and the difficulty score, *i.e.*, $\boldsymbol{h_i} = \mathcal{E}_d([y_i] \oplus [s_i])$, where $\oplus$ indicates a concatenation operation.

$\boldsymbol{h_i}$ is concatenated with the CLIP text-prompt embedding produced by the pretrained CLIP encoder $\mathcal{E}_{\text{text}}$ in the Stable Diffusion model, as illustrated in Fig. 3. We use the concatenated embeddings, $\boldsymbol{\tau}_i = \mathcal{E}_{\text{text}}(p_i) \oplus \boldsymbol{h_i}$, to guide the generation process, enabling control based on both the text description p and the difficulty score.

Difficulty Controlled Fine-Tuning

Data Preparation. To fine-tune the text-to-image model on the target dataset $\mathcal{D}_t$ using our method, we generate difficulty scores and text prompts for each training sample.

For difficulty score, we use a classifier pretrained on $\mathcal{D}_t$ to compute each sample's score s, as defined in the *Investigation section*. Note that we do not depend on any specific classifier architecture for scoring. For text prompts, each image is paired with a text caption p in the form of "`a photo of [CLS]`", where `[CLS]` denotes the category name in the target dataset. In contrast to previous works using complex prompts generated by pretrained captioning models, we adopt a simple template during training, enabling difficulty control to be handled by the difficulty encoder rather than by prompt complexity.

Finally, the original target dataset $\mathcal{D}_t$ is extended to $\mathcal{D}_t' = \{\boldsymbol{x}_i, y_i, s_i, p_i\}_{i=1}^{n_t}$, where s_i and p_i denote the assigned difficulty score and text prompt for the image.

Optimization with Difficulty Condition. The difficulty encoder is trained from scratch. And we employ Low-Rank Adaptation (LoRA) (Hu et al. 2022) to efficiently fine-tune the diffusion model on our prepared dataset. The model is trained to predict the added noise ϵ given a noised latent $\boldsymbol{z}_t$ at timestep $t \in \{1, \ldots, T\}$, using the conditioning input. The denoising loss is formulated as:

$$\mathcal{L} = \mathbb{E}_{\mathcal{E}(\boldsymbol{x}), \boldsymbol{\tau}, \epsilon \sim \mathcal{N}(0,1), t} \left[\| \epsilon - \epsilon_{(\theta,\delta)}(\boldsymbol{z}_t, t, \boldsymbol{\tau}) \|_2^2 \right], \quad (2)$$

where δ denotes the parameters of LoRA layers.

During the training process, LoRA enables efficient fine-tuning of the diffusion model to denoise perturbed images, ensuring the generated outputs align with the target dataset distribution. Simultaneously, the difficulty encoder learns to project difficulty scores into a latent space to control the generation process. The distinct roles of these two learnable components effectively disentangle the domain alignment and difficulty control.

Difficulty Controlled Data Synthesis

After optimizing on the target dataset, we employ the model for training data synthesis. For each class in the target classification task, we use class-specific text prompts and sample difficulty scores from a predefined distribution.

For text prompts, unlike during training, where simple templates are used, we adopt more diverse text descriptions, following Real-Fake, to enhance generation diversity. For the difficulty score distribution, we sample the difficulty score s from a Gaussian distribution $s \sim \mathcal{N}(\mu, \sigma)$, where μ is centered around a mid-level difficulty, and the standard deviation σ controls the degree of diversity. We experimentally found the suitable μ, σ during generation.

The generated images are then used to augment the target dataset and enhance classification performance. We experimentally demonstrate that our method effectively controls difficulty while remaining compatible with diverse text prompts.

Hard Factors Extraction

Our difficulty encoder generates latent embeddings corresponding to specific levels of learning difficulty. This allows our method to analyze difficulty-inducing visual factors inherent in the target dataset. To achieve this, we generate images conditioned solely on difficulty scores, without using

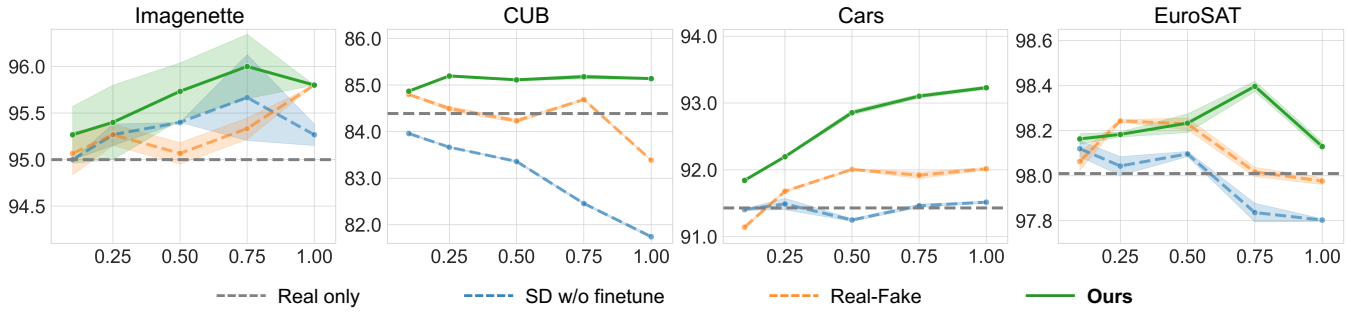

Figure 4: Top-1 classification accuracy (%) on various tasks with ResNet-50 model. The x-axis denotes the ratio of additional synthetic images. All results are averaged over three runs, and shaded regions represent the standard deviation. The detailed numerical results are provided in the appendix.

Method	Synthetic data ratio						
	Real only	5%	10%	25%	50%	75%	100%
Generation time (GPU hours)	0	7.9	15.9	39.6	79.3	118.9	158.5
SD w/o finetune		77.96	77.90	77.80	77.91	77.73	77.64
Real-Fake (Yuan et al. 2024)	78.21	78.32	78.61	78.68	78.73	78.70	78.62
Ours		**78.47**	**78.74**	**78.76**	**78.73**	**78.71**	**78.63**

Table 2: Top-1 classification accuracy (%) on ImageNet with ResNet-50. GPU hours are measured on a single NVIDIA A100 GPU with a generation batch size of 128. The difficulty encoder introduces an additional latency of only 8%. Therefore, all reported timings are based on the original Stable Diffusion model.

	Standard deviation $\sigma = 0.1$			
Mean value (μ)	0.3	0.5	0.7	0.9
Acc.	95.8	**96.4**	96.0	95.2
	Mean value $\mu = 0.5$			
Standard deviation (σ)	0.00	0.01	0.10	0.50
Acc.	95.8	95.8	**96.4**	96.0

Table 3: Classification accuracy on Imagenette under different difficulty score distributions. A synthetic data ratio of 75% is used.

any text prompts, thereby isolating the visual features associated with different levels of difficulty. We demonstrate this analysis in our experiments.

Experiments

Implementation Details

Target Tasks and Models. We use Imagenette (Howard 2019), CUB (Wah et al. 2023), Cars (Krause et al. 2013), and ImageNet (Deng et al. 2009) as target taskss. For the target classifiers, we use ResNet-50, ResNet-101 (He et al. 2016), and ViT-Small (Dosovitskiy et al. 2020). The training settings follow Real-Fake (Yuan et al. 2024) for a fair comparison.

Model	ViT-small	ResNet-50	ResNet-101
Real only	82.6	95.0	95.6
Real-Fake	84.8	95.4	95.8
Ours	**86.0**	**96.4**	**96.8**

Table 4: Classification accuracy of different model structures on Imagenette. A synthetic data ratio of 75% is used.

Diffusion Model. We use the pretrained Stable Diffusion v1.5 (Rombach et al. 2022) with an image resolution of 512×512, following the same configuration as Real-Fake. Our implementation is based on the Diffusers library (von Platen et al. 2022) library, and the LoRA (Hu et al. 2022) fine-tuning schedule follows its default configuration.

Training Data Synthesis. We augment the real training data for each classification task by generating synthetic images at various ratios. Generation is performed using the default PLMS sampler with 30 steps and a guidance scale of 2.0. Following Real-Fake, we use captions generated by the pretrained BLIP-2 model (Li et al. 2023a) for Imagenette and ImageNet, and simple prompts of the form "`a photo of [CLS]`" for the other datasets. Difficulty scores are sampled from a Gaussian distribution with mean $\mu = 0.5$ and standard deviation $\sigma = 0.1$. Other detailed settings are reported in the appendix.

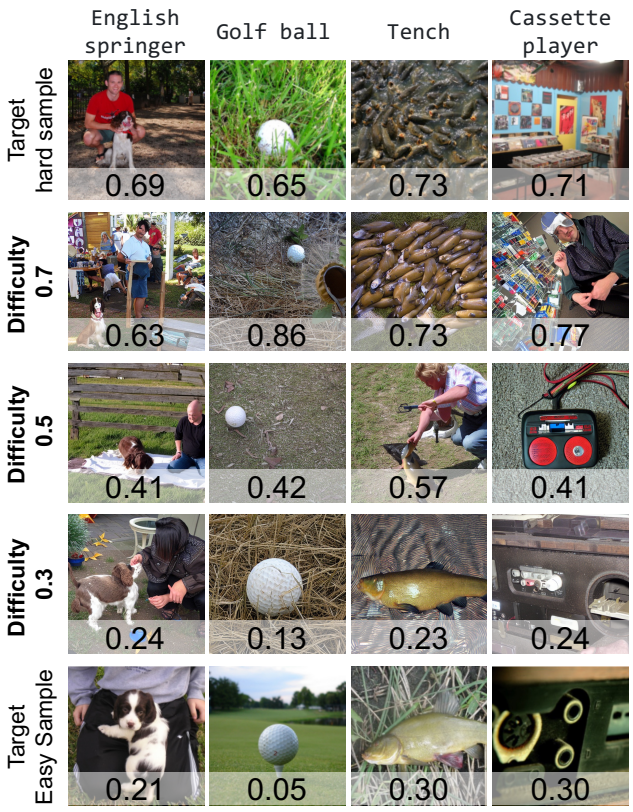

Figure 5: Visualization of synthetic images from four classes in Imagenette, with different difficulty score inputs shown on the left. Easy and hard samples from the target dataset are also shown for comparison. The numbers on the images are difficulty scores assessed by a pretrained ResNet-50 model.

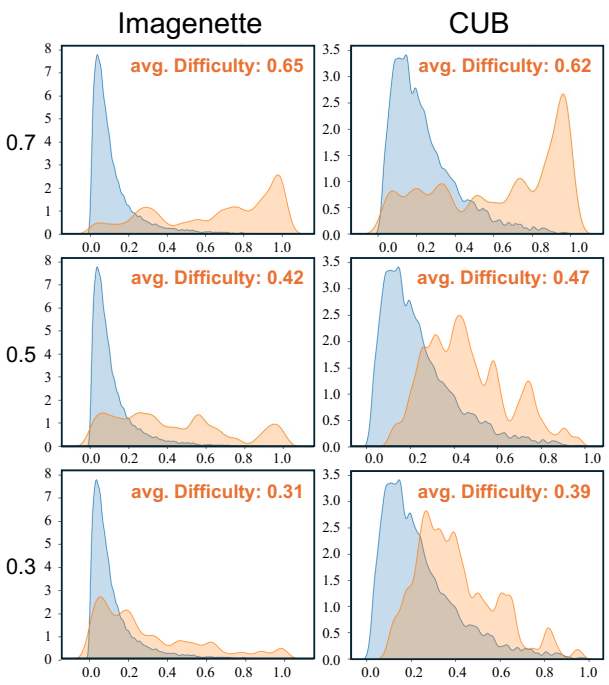

Figure 6: Difficulty score distributions of the real datasets (blue) and 100 randomly generated images (orange) using our method, shown for two datasets. X-axis: difficulty score; Y-axis: KDE density. Each row corresponds to a different difficulty score input, as indicated on the left. All difficulty scores are assessed by a pretrained ResNet-50 model.

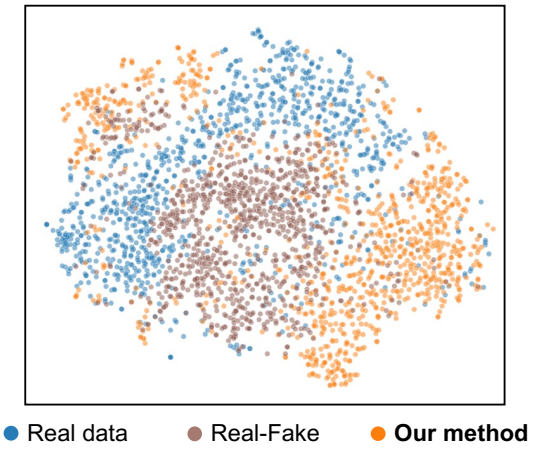

Figure 7: T-SNE visualization of features extracted by a pretrained ResNet-50 for samples from the same class in Imagenette.

Training Performance of Synthetic Data

Classification Accuracy. To evaluate the effectiveness of our method, we augment the real training split in each task with varying amounts of synthetic data generated by three methods: (1) our proposed approach, (2) Real-Fake, and (3) a Stable Diffusion model without fine-tuning. Fig. 4 presents results on Imagenette, CUB, Cars, and EuroSAT. The results indicate that **our method achieves higher accuracy using fewer synthetic samples.** For example, on the Cars dataset, our method improves accuracy by over 1.2% compared to Real-Fake. Table 2 reports the results on ImageNet, showing that generating only an additional 10% of data with our method is sufficient to surpass Real-Fake's best result, while significantly reducing generation cost by approximately 63.4 GPU-hours of generation time. Moreover, compared to other methods, our method achieves more stable improvements over training with real data only.

Parameter Optimization. We further analyze the hyperparameters used in our data synthesis strategy. Table 3 shows the effect of the difficulty score distribution parameters (μ, σ) on classification accuracy. The results show that our method enables effective tailoring of the synthetic data to each target dataset by simply tuning the difficulty score distribution. Moreover, our method exhibits robustness to parameter selection, consistently outperforming Real-Fake across a broad range of μ and σ values.

Figure 8: Hard factor visualization for the "garbage truck" category in Imagenette. Our method captures both hard and easy factors associated with the sample's learning difficulty.

Various Model Structures. Table 4 shows the effectiveness of our method on various model structures, including CNN and Transformer-based models. All synthetic data are generated using difficulty scores assessed by a ResNet-50 model pretrained on the target dataset. This demonstrates that our method does not rely on using the same model architecture for data preparation and final training.

Controlling Efficacy of Difficulty

Visualization. We randomly generate 100 samples with each of the difficulty scores 0.3, 0.5, and 0.7. Representative examples of generated images are shown in Fig. 5. The generated samples exhibit visual properties consistent with those of real data at corresponding difficulty levels.

Difficulty Score Distribution. Fig. 6 shows the difficulty score distributions of the generated images. This demonstrates that our method effectively controls the learning difficulty of generated data, addressing a key limitation of existing methods that tend to produce mostly easy samples.

Feature Distribution. We compare the feature distributions of images generated by our method and Real-Fake. As shown in Fig. 7, our method generates samples that more effectively delineate the boundaries of the feature distribution of real data, whereas Real-Fake's samples tend to cluster together. This indicates that our method effectively disentangles difficulty control from the domain alignment process.

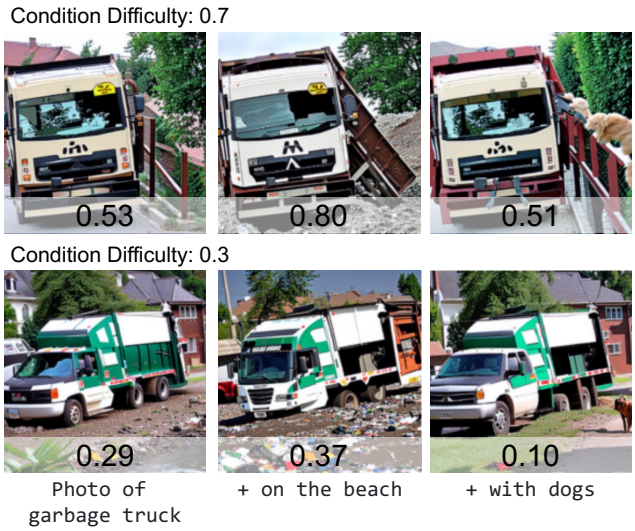

Figure 9: Our method supports diverse text prompts and can reflect both textual semantics and the target difficulty score.

More Analysis

Hard Factor Visualization. Omitting the text prompt during generation allows our method to reveal the visual factors associated with different difficulty levels for the target dataset. As shown in Fig. 8, generated samples in the "garbage truck" category reveal that hard examples typically depict crowded scenes, while easy ones feature clearly visible vehicles and uncluttered backgrounds. Similar analyses for other categories are provided in the appendix.

Compatibility with Diverse Text Prompts. Despite being fine-tuned on simple text prompts, our method generalizes well to more diverse inputs, as illustrated in Fig. 9. The results show that our method successfully aligns with both difficulty scores and diverse text prompts. This enables the integration of our method with captions generated by powerful vision-language models (Lei et al. 2023), enabling the synthesis of more diverse images.

Conclusions

In this paper, we demonstrate the critical role of appropriately difficult samples in model training. We identify a fundamental limitation in existing training data synthesis methods: domain alignment alone causes models to capture only common features and to generate mostly easy samples. To overcome this, we propose a generation method that conditions on learning difficulty to disentangle difficulty-aware feature modeling from domain alignment. We validate the effectiveness of our method through extensive experiments by evaluating classification performance and the difficulty distributions of the generated samples. In addition, our method serves as a useful tool for the visual analysis of hard factors within datasets.

References

Avrahami, O.; Lischinski, D.; and Fried, O. 2022. Blended diffusion for text-driven editing of natural images. In *CVPR*, 18208–18218.

Azizi, S.; Kornblith, S.; Saharia, C.; Norouzi, M.; and Fleet, D. J. 2023. Synthetic Data from Diffusion Models Improves ImageNet Classification. *TMLR*.

Besnier, V.; Jain, H.; Bursuc, A.; Cord, M.; and Pérez, P. 2020. This dataset does not exist: training models from generated images. In *ICASSP*, 1–5.

Brooks, T.; Holynski, A.; and Efros, A. A. 2023. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, 18392–18402.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*, 248–255.

Dhariwal, P.; and Nichol, A. 2021. Diffusion models beat gans on image synthesis. *NeurIPS*, 34: 8780–8794.

Ding, M.; Yang, Z.; Hong, W.; Zheng, W.; Zhou, C.; Yin, D.; Lin, J.; Zou, X.; Shao, Z.; Yang, H.; et al. 2021. Cogview: Mastering text-to-image generation via transformers. *NeurIPS*, 34: 19822–19835.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*.

Gafni, O.; Polyak, A.; Ashual, O.; Sheynin, S.; Parikh, D.; and Taigman, Y. 2022. Make-a-scene: Scene-based text-to-image generation with human priors. In *ECCV*, 89–106.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *NeurIPS*, 33: 6840–6851.

Ho, J.; and Salimans, T. 2021. Classifier-Free Diffusion Guidance. In *NeurIPS Workshop*.

Howard, J. 2019. Imagenette: A smaller subset of 10 easily classified classes from Imagenet.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *ICLR*.

Huang, L.; Chen, D.; Liu, Y.; Shen, Y.; Zhao, D.; and Zhou, J. 2023. Composer: creative and controllable image synthesis with composable conditions. In *ICML*, 13753–13773.

Huang, T.; Liu, J.; You, S.; and Xu, C. 2024. Active generation for image classification. In *ECCV*, 270–286.

Kawar, B.; Zada, S.; Lang, O.; Tov, O.; Chang, H.; Dekel, T.; Mosseri, I.; and Irani, M. 2023. Imagic: Text-based real image editing with diffusion models. In *CVPR*, 6007–6017.

Kim, G.; Kwon, T.; and Ye, J. C. 2022. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *CVPR*, 2426–2435.

Krause, J.; Stark, M.; Deng, J.; and Fei-Fei, L. 2013. 3d object representations for fine-grained categorization. In *ICCV Workshops*, 554–561.

Lei, S.; Chen, H.; Zhang, S.; Zhao, B.; and Tao, D. 2023. Image captions are natural prompts for text-to-image models. *arXiv preprint arXiv:2307.08526*.

Li, J.; Li, D.; Savarese, S.; and Hoi, S. 2023a. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 19730–19742.

Li, Y.; Liu, H.; Wu, Q.; Mu, F.; Yang, J.; Gao, J.; Li, C.; and Lee, Y. J. 2023b. Gligen: Open-set grounded text-to-image generation. In *CVPR*, 22511–22521.

Liu, L.; Ren, Y.; Lin, Z.; and Zhao, Z. 2022a. Pseudo Numerical Methods for Diffusion Models on Manifolds. In *ICLR*.

Liu, N.; Li, S.; Du, Y.; Torralba, A.; and Tenenbaum, J. B. 2022b. Compositional visual generation with composable diffusion models. In *ECCV*, 423–439.

Mao, J.; Wang, X.; and Aizawa, K. 2023. Guided Image Synthesis via Initial Image Editing in Diffusion Model. *ACM MM*.

Mou, C.; Wang, X.; Xie, L.; Zhang, J.; Qi, Z.; Shan, Y.; and Qie, X. 2023. T2I-Adapter: Learning Adapters to Dig out More Controllable Ability for Text-to-Image Diffusion Models. *arXiv e-prints*, arXiv–2302.

Nichol, A. Q.; and Dhariwal, P. 2021. Improved denoising diffusion probabilistic models. In *ICML*, 8162–8171.

Park, D. H.; Azadi, S.; Liu, X.; Darrell, T.; and Rohrbach, A. 2021. Benchmark for compositional text-to-image synthesis. In *NeurIPS Datasets and Benchmarks Track (Round 1)*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *ICML*, 8748–8763.

Ramesh, A.; Pavlov, M.; Goh, G.; Gray, S.; Voss, C.; Radford, A.; Chen, M.; and Sutskever, I. 2021. Zero-shot text-to-image generation. In *ICML*, 8821–8831.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *CVPR*, 10684–10695.

Ruiz, N.; Li, Y.; Jampani, V.; Pritch, Y.; Rubinstein, M.; and Aberman, K. 2023. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *CVPR*, 22500–22510.

Saharia, C.; Chan, W.; Saxena, S.; Li, L.; Whang, J.; Denton, E. L.; Ghasemipour, K.; Gontijo Lopes, R.; Karagol Ayan, B.; Salimans, T.; et al. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *NeurIPS*, 35: 36479–36494.

Sarıyıldız, M. B.; Alahari, K.; Larlus, D.; and Kalantidis, Y. 2023. Fake it till you make it: Learning transferable representations from synthetic imagenet clones. In *CVPR*, 8011–8021.

Song, J.; Meng, C.; and Ermon, S. 2021. Denoising Diffusion Implicit Models. In *ICLR*.

Vendrow, J.; Jain, S.; Engstrom, L.; and Madry, A. 2023. Dataset interfaces: Diagnosing model failures using controllable counterfactual generation. *arXiv preprint arXiv:2302.07865*.

von Platen, P.; Patil, S.; Lozhkov, A.; Cuenca, P.; Lambert, N.; Rasul, K.; Davaadorj, M.; Nair, D.; Paul, S.; Berman, W.; Xu, Y.; Liu, S.; and Wolf, T. 2022. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers.

Voynov, A.; Aberman, K.; and Cohen-Or, D. 2023. Sketch-Guided Text-to-Image Diffusion Models. *SIGGRAPH*.

Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. 2023. The Caltech-UCSD Birds-200-2011 Dataset.

Wang, T.; Zhang, T.; Zhang, B.; Ouyang, H.; Chen, D.; Chen, Q.; and Wen, F. 2022. Pretraining is all you need for image-to-image translation. *arXiv preprint arXiv:2205.12952*.

Yuan, J.; Zhang, J.; Sun, S.; Torr, P.; and Zhao, B. 2024. Real-Fake: Effective Training Data Synthesis Through Distribution Matching. In *ICLR*.

Zhang, L.; Rao, A.; and Agrawala, M. 2023. Adding conditional control to text-to-image diffusion models. In *ICCV*, 3836–3847.

Zhang, Y.; Ling, H.; Gao, J.; Yin, K.; Lafleche, J.-F.; Barriuso, A.; Torralba, A.; and Fidler, S. 2021. Datasetgan: Efficient labeled data factory with minimal human effort. In *CVPR*, 10145–10155.

Zhou, Y.; Sahak, H.; and Ba, J. 2023. Training on thin air: Improve image classification with generated data. *arXiv preprint arXiv:2305.15316*.