Dis^{2}Booth: Learning Image Distribution with Disentangled Features for Text-to-Image Diffusion Models

Guanqi Ding[1,3], Chengyu Yang[2], Shuhui Wang[3,5*], Xincheng Li[4], Jinzhe Zhang[4], Xin Jin[4], Qingming Huang[1,3,5]

[1]University of Chinese Academy of Sciences, Beijing, China
[2]Beijing Institute of Technology, Beijing, China
[3]Key Lab of Intell. Info. Process., Inst. of Comput. Tech., CAS, Beijing, China
[4]Huawei Cloud EI Innovation Lab, China
[5]Peng Cheng Laboratory, Shenzhen, China
dingguanqi19@mails.ucas.ac.cn, 1120210671@bit.edu.cn, wangshuhui@ict.ac.cn,
{lixincheng1, zhangjinzhe, jinxin11}@huawei.com, qmhuang@ucas.ac.cn

Abstract

Personalized image generation enables customized content creation based on the text-to-image diffusion models. However, existing personalization methods focus on fine-tuning generative models to generate a specific instance or concept, like a specific Corgi, but cannot handle multiple instances or concepts with commonalities, such as images of multiple different Corgis. In this work, we focus on personalizing a diffusion model to generated **varied data** usually containing multiple instances, which has a more diverse and complex data distribution. Our basic assumption is that the varied data distribution is composed of the common features shared among all samples, as well as the reasonable variations within it. Accordingly, we are capable to decompose the learning process of complex data distributions into two simpler sub-tasks, employing a divide-and-conquer approach. To this end, we propose **Dis^{2}Booth**, a framework that can learn complex image **Dis**tribution by **Dis**entangling data distribution in an unsupervised manner. Specifically, Dis^{2}Booth contains two modules, Anchor LoRA and Delta LoRA, tasked with learning the common features and variational features constrained by Contextual Loss and Delta Loss unsupervisedly. Besides, the Asynchronous Optimization Strategy is proposed to ensure the collaborative training of the two modules. Extensive experiments suggest that Dis^{2}Booth can learn the data distribution with higher diversity and complexity and be flexibly applied across various tasks.

Introduction

The emergence of diffusion models (Ho, Jain, and Abbeel 2020; Song, Meng, and Ermon 2020; Dhariwal and Nichol 2021) such as Stable Diffusion (Rombach et al. 2022) and DALL-E (Ramesh et al. 2022) have pushed up the ceiling of what is possible in terms of generating realistic and creative images.

Personalized image generation methods (Ruiz et al. 2023; Gal et al. 2022) use a small set of reference images to fine-tune network parameters or token embeddings, enabling the diffusion models to generate specific instance or concept in those images. These methods enable customized generation and editing of specific instances.

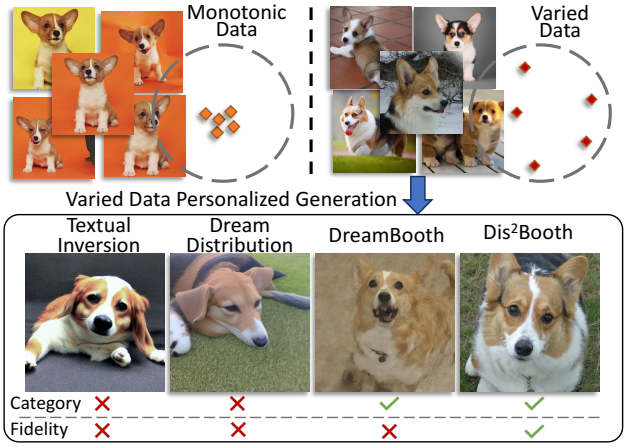

Figure 1: Illustration of distributions of monotonic data (a specific Corgi dog) and varied data (multiple different Corgi dogs) and comparison of different methods for personalized generation of varied data. Existing personalized generation methods for complex data distributions tend to exhibit drift in discriminative semantics (category change), overfitting and significant degradation in generation quality.

Previous personalization methods (Ruiz et al. 2023; Gal et al. 2022) have primarily focused on modifying the pre-trained generation model (Rombach et al. 2022; Podell et al. 2023) to generate a specific instance or concept based on a few given individual reference images. However in many cases, user may want to personalize the generation model over more abstract visual attributes from varied data, instead of instance-level personalization. For example, in art design, automotive designers may generate design sketches of different car models follow a consistent design language, and in industry, images of different kinds of workpieces with specific defects can be created as augmented data for training defect detection models. In these cases, it is more critical to generate different individuals with certain common characteristics, rather than a single individual. To make the issue

*Corresponding author.

clearer, as shown in Figure 1, we define **monotonic data** as data that only contains a specific individual or concept, such as images of a particular Corgi dog. On the contrary, **varied data** is defined as data containing multiple individuals or concepts with shared traits, such as images of multiple different Corgi dogs. Compared to monotonic data, varied data exhibits more complex inter-sample variations and a broader data distribution.

Existing personalization methods (Gal et al. 2022; Zhao et al. 2023; Chen et al. 2023) learn to generate specific instances. However, they cannot accurately generate varied data. As shown in Figure 1, there mainly exist two problems, *i.e.*, the discriminative information drift (category change) and the quality collapse. A subset of these methods (Gal et al. 2022; Zhao et al. 2023) optimize text embeddings to represent the reference distribution. They leverage the priors contained in the pretrained models. Therefore, when reference instances are rare in pretraining datasets or highly diverse, these methods struggle to extract accurate prior information from the pretrained model. In Figure 1, images generated by these methods show severe drift in discriminative information, causing category changes in the generated dogs. Another category of personalization methods (Ruiz et al. 2023; Kumari et al. 2023) involve directly fine-tuning the weights of the network to learn the reference data distribution. As the diversity of the reference images increases, these methods fall to naive solution and tend to present fused samples, which harms the generative diversity greatly. As shown in Figure 1, the image generated by Dreambooth contains many artifacts. In summary, existing methods directly learn the reference data distribution, which leads to semantic drifts and fidelity degradation when the distribution becomes complex.

In this work, we focus on generating semantically accurate and high-quality varied data by fine-tuning a diffusion model on reference images. Existing methods are far from meeting the needs. Following the divide and conquer strategy, we propose **Dis^{2}Booth**, a framework that can learn complex image **Dis**tribution by **Dis**entangling data distribution in an unsupervised manner. As illustrated in the right part of Figure 2, the varied data distribution is explicitly modeled as the combination of common features and reasonable variations. In this way, the learning process is decomposed into two sub-tasks with lower difficulty. To learn them seperately, we introduce two independent modules: Anchor LoRA and Delta LoRA. The former is designed to learn the common features of a category, which ensures consistent discriminative information between generated and reference images. The latter focuses on capturing the reasonable variations within the category, which enhances the diversity of the generated images. Since it's not possible to explicitly label and separate common features and variations, an unsupervised approach is needed to disentangle and constrain the two modules. Inspired by Contrastive Learning (He et al. 2020), we introduce Contextual Loss and Delta Loss, utilizing momentum contrast to achieve unsupervised constraint. Contextual Loss identifies semantic correspondence between feature maps outputted by Anchor LoRA from different samples and constrains their consistency through minimizing Multi-instance Relaxed Earth Mover's Distance (MREM), thereby forcing the Anchor LoRA to learn shared features across samples. Delta Loss constrains the distance between the Delta LoRA features of different samples to learn variations among them. The unsupervised collaborative optimization is unstable, so we propose an asymmetric strategy that limits Delta LoRA's optimization to adapt to Anchor LoRA's, ensuring collaborative optimization and stable convergence.

We have conducted quantitative experiments, qualitative experiments, and downstream classification experiments to validate the effectiveness and superiority of Dis^{2}Booth. Additionally, Dis^{2}Booth possesses high flexibility to combine with other customization methods (Shah et al. 2023; Shi et al. 2023) and a wide range of applications. It is demonstrated in settings of 3D generation and the fusion of LoRAs, showcasing the versatility and applicability of Dis^{2}Booth in diverse scenarios. Our contributions can be summarized as follows:

- We propose **Dis^{2}Booth**, a framework that can learn to generate **varied data distribution** by disentangling data distribution in an unsupervised manner.
- Extensive experiments suggest that Dis^{2}Booth can accurately learn varied data distributions in few-shot image generation task and general object scenes.
- Extensive experiments suggest that Dis^{2}Booth possesses a wide range of applications and high flexibility to combine with other customization methods.

Related Work

Text-to-image Diffusion Models

In recent years, diffusion models (Ho, Jain, and Abbeel 2020; Song, Meng, and Ermon 2020; Dhariwal and Nichol 2021; Rombach et al. 2022) have made significant advancements in the field of image generation. Text-to-image diffusion models employ text as an additional guiding condition for image generation. By utilizing pretrained language models like CLIP (Radford et al. 2021), text is encoded into latent vectors, enabling exceptional performance in generating diverse and high-fidelity images. Drawing on the capabilities of large-scale language models, models like Imagen (Saharia et al. 2022) and DALLE-2 (Ramesh et al. 2022) excel at generating highly realistic images that faithfully align with the provided textual descriptions. The Latent Diffusion Models perform diffusion steps in the latent image space, reducing computational costs. Stable Diffusion (Rombach et al. 2022) and its successor, Stable Diffusion XL (Podell et al. 2023), are image generation models that leverage autoencoders to encode images into latent representations. These models then perform diffusion on the latent space to generate high-quality images.

Personalized text-to-image Generation

In the realm of image generation, text-guided models have gained significant attention for their ability to generate images based on textual descriptions. However, these models are not specifically designed to generate personalized images, such as images of specific faces. Personalized concepts

are often abstract and challenging to comprehend. Text-to-image personalization aims to learn abstract concepts from a small set of sample images and apply them to new scenarios and provide fine-grained control over the generated images. Textual Inversion (Gal et al. 2022) synthesize novel images of a instance by fine-tuning the text embeddings with few images and target texts. DreamBooth (Ruiz et al. 2023) is a naive fine-tuning method that directly fine-tunes the original text-to-image models without modifying its architecture. In addition, methods like CustomDiffusion (Kumari et al. 2023) and SVDiff (Han et al. 2023) have been developed to decrease the number of parameters that require fine-tuning, thereby reducing computational demands. DisenBooth (Chen et al. 2023) fine-tunes a pretrained diffusion model using disentangled embeddings to preserve subject identity and separate identity-relevant and identity-irrelevant information in text-to-image generation. However, for object generation, existing personalization methods predominantly concentrate on generating one specific instance or concept. We constructed a framework based on a divide-and-conquer strategy for disentangled learning of data distributions to achieve accurate generation of varied data.

Few-shot image generation

Existing few-shot image generation methods can be divided into optimization-based, fusion-based, transformation-based and editing-based methods. Optimization-based methods (Clouâtre and Demers 2019; Liang, Liu, and Liu 2020; Phaphuangwittayakul, Guo, and Ying 2021) combine meta-learning and adversarial learning to generate images by fine-tuning the model. However, the images generated by such methods have poor authenticity. Fusion-based methods fuse the features by matching the random vector with the conditional images (Hong et al. 2020b) or interpolate high-level features of conditional images by filling in low-level details (Hong et al. 2020c; Gu et al. 2021; Yang et al. 2022). Simple content fusion limits the diversity of generated images. Transformation-based methods (Antoniou, Storkey, and Edwards 2017; Hong et al. 2020a) capture the cross-category or intra-category transformations to generate novel data. These works capture the transformations from the image differences and may corrupt due to the complex transformations between intra- and inter-category pairs. For Editing-based methods (Ding et al. 2022, 2023; Li, Zhang, and Wang 2023), the intra-category transformation can be alternatively modeled as category-irrelevant image editing based on one sample instead of pairs of samples.

Method

Preliminaries

Diffusion models (Ho, Jain, and Abbeel 2020; Song, Meng, and Ermon 2020; Dhariwal and Nichol 2021) are a class of generative models that learn to generate high-quality samples by denoising of a variable sampled from a Gaussian distribution step by step. In practice, at every timestep t, the objective of diffusion models is simplified to predicting the true noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ added to the image $\mathbf{x}$:

$$\nabla_\theta \left\| \epsilon - \epsilon_\theta(\mathbf{x}_t, \mathbf{c}, t) \right\|^2, \tag{1}$$

where $\mathbf{x}_t$ is the noisy version of $\mathbf{x}$ at timestep t and $\epsilon_\theta(\mathbf{x}_t, \mathbf{c}, t)$ is the noise predicted by the diffusion model conditioned on $\mathbf{c}$ at timestep t. Latent diffusion models (Rombach et al. 2022) learn the diffusion process in the latent space, resulting in higher computational efficiency and the ability to handle larger training scales.

Low-Rank Adaptation (LoRA) refers to a highly efficient technique used in the context of fine-tuning large pre-trained models. It involves decomposing the weight update $\Delta\mathbf{W}^i$ of the weight of the i-th layer in the pre-trained model $\mathbf{W}_0^i \in \mathbb{R}^{m\times n}$ into low-rank components:

$$\Delta\mathbf{W}^i = \mathbf{B}^i\mathbf{A}^i, \tag{2}$$

where $\mathbf{B}^i \in \mathbb{R}^{m\times r}$ and $\mathbf{A}^i \in \mathbb{R}^{r\times n}$ with $r \ll \min(m, n)$. During the fine-tuning process, only the weights B^i and A^i are updated, while the weights $\mathbf{W}_0^i$ and $\Delta\mathbf{W}^i$ are fused together to form the updated weights during inference:

$$\mathbf{W}^i = \mathbf{W}_0^i + \mathbf{B}^i\mathbf{A}^i. \tag{3}$$

We drop the superscript i for simplicity since the operations are applied over all the LoRA-enabled weights.

Dis^{2}Booth

Dis^{2}Booth is a framework that can learn to generate varied data distribution by disentangling data distribution into two components: the shared common features and the reasonable variations among samples. We employ two modules, Anchor LoRA and Delta LoRA, to model the aforementioned two components separately. Anchor LoRA $\Delta\mathbf{W}_a$ is used to learn the common features, while Delta LoRA $\Delta\mathbf{W}_d$ is employed to capture the reasonable variations. The benefit of using LoRAs is that it does not require explicit modeling of the distributions for the two components, but rather allows the model to spontaneously learn a complex data distribution. Finally, the weights of these two modules are summed together to form the weight updates of the pretrained model W_0:

$$\mathbf{W} = \mathbf{W}_0 + \alpha\Delta\mathbf{W}_a + \beta\Delta\mathbf{W}_d, \tag{4}$$

where α and β are the coefficients of the two weights.

As shown in Figure 2, to separately constrain the learning of Anchor LoRA and Delta LoRA, we draw the idea from Contrastive Learning (He et al. 2020) and first construct two queues Anchor Queue $Q_a = \{\mathbf{F}_a^1, \mathbf{F}_a^2, ..., \mathbf{F}_a^l\}$ and Delta Queue $Q_d = \{\mathbf{F}_d^1, \mathbf{F}_d^2, ..., \mathbf{F}_d^l\}$ to store the feature maps $\mathbf{F}_a^i, \mathbf{F}_d^i \in \mathbb{R}^{h\times w\times c}$ outputted by each of them. These two queues serve as the dictionaries of features and allow us to reuse the encoded features for the computation of momentum contrast. The queues are dynamically updated throughout the training process to ensure the features in the queue are consistent to the updated module. At the end of each iteration, the latest obtained features are inserted at the rear of the queues, while an equal number of features are popped out from the front of the queues. Based on the two queues, we construct Contextual Loss and Delta Loss to respectively optimize Anchor LoRA and Delta LoRA in an unsupervised manner.

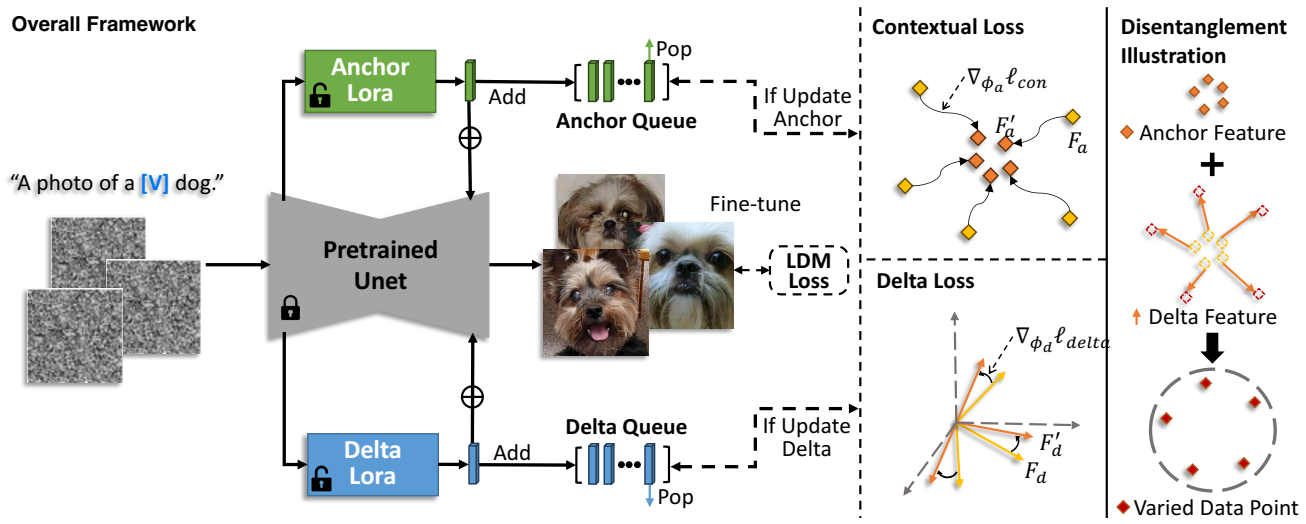

Figure 2: The left part is the overview of Dis^{2}Booth. Dis^{2}Booth disentangles the learning of varied data distribution with two modules: Anchor LoRA and Delta LoRA. Anchor LoRA learns the common feature among the data and Delta LoRA learns the reasonable variations between the samples. Contextual Loss and Delta Loss constrain the optimizations of the two modules in an unsupervised manner. The right part is the illustration of disentanglement of varied data distribution.

Contextual Loss Since Anchor LoRA learns the common features among different reference samples, Contextual Loss is utilized to constrain the consistency of the features outputted by Anchor LoRA for different samples. Due to the complex variations among samples, including significant differences in spatial distribution, directly computing the distances between their feature maps and imposing point-to-point spatial consistency constraints is not applicable. Therefore, we aim to overcome this spatial limitation and constrain the consistency of Anchor Features based on their semantic correspondence. We are specifically inspired by previous methods that utilized Earth Mover's Distance (EMD). Given two sets of features $\mathbf{F}_x = \{\mathbf{f}_x^1, \mathbf{f}_x^2, ..., \mathbf{f}_x^m\}$ and $\mathbf{F}_y = \{\mathbf{f}_y^1, \mathbf{f}_y^2, ..., \mathbf{f}_y^n\}$, EMD is formulated as:

$$\begin{aligned} \mathrm{EMD}(\mathbf{F}_x, \mathbf{F}_y) &= \min_{\mathbf{T}\geq 0} \sum_{ij} \mathbf{T}_{ij}\mathbf{C}_{ij}, \\ \text{s.t. } \sum_i \mathbf{T}_{ij} &= 1/n, \ \sum_j \mathbf{T}_{ij} = 1/m, \end{aligned} \tag{5}$$

where $\mathbf{T}$ is the transport matrix which defines partial pairwise assignments, and $\mathbf{C}$ is the cost matrix which defines the distances between elements in $\mathbf{F}_x$ and $\mathbf{F}_y$. EMD breaks the point-to-point spatial constraints and freely constrains the distance relationships between elements within the feature map.

In Dis^{2}Booth, the distances between the current iteration's Anchor Feature $\mathbf{F}_a^c$ and the features in the Anchor Queue $Q_a = \{\mathbf{F}_a^1, ..., \mathbf{F}_a^k, ..., \mathbf{F}_a^l\}$ need to be constrained. Therefore, according to the setting and to reduce computational complexity, we relax the EMD constraint and propose the Multi-instance Relaxed Earth Mover's Distance (MREMD):

$$\begin{aligned} \mathrm{MREMD}(\mathbf{F}_c, Q_a) &= \frac{1}{l}\sum_{k=0}^{l}(\min_{\mathbf{T}\geq 0}\sum_{ij}\mathbf{T}_{ij}^k\mathbf{C}_{ij}^k), \\ \text{s.t. } \sum_j \mathbf{T}_{ij}^k &= 1/(h\times w), \end{aligned} \tag{6}$$

where $\mathbf{T}^k$ and $\mathbf{C}^k$ are the transport matrix and cost matrix between $\mathbf{F}_a^c$ and $\mathbf{F}_a^k$.

The semantic similarity between different positions in the feature maps can be obtained by calculating the element-wise distances, which represents the local pattern consistency. For every element in $\mathbf{F}_a^c$, we find the most similar element in $\mathbf{F}_a^k$ and minimize their distance. The contextual loss can be defined as:

$$\ell_{con} = \frac{1}{l}\sum_{k=0}^{l}(\frac{1}{h\times w}\sum_i \min_j \mathbf{C}_{ij}^k). \tag{7}$$

Specifically, we define the cost $\mathbf{C}_{ij}^k$ as the cosine distance between the elements in $\mathbf{F}_a^c$ and $\mathbf{F}_a^k$:

$$\mathbf{C}_{ij}^k = 1 - \mathrm{D}_{cos}(\mathbf{f}_{a^c}^i, \mathbf{f}_{a^k}^j), \tag{8}$$

$$\mathrm{D}_{cos}(\mathbf{X}, \mathbf{Y}) = \left| \frac{\mathbf{X}\cdot\mathbf{Y}}{\|\mathbf{X}\|\,\|\mathbf{Y}\|} \right|. \tag{9}$$

Delta Loss Delta LoRA learns the differences between reference samples. To accommodate unique variations in each sample, we constrain the Delta Features $\mathbf{F}_d$ of different samples to have a certain angle between them, allowing for some overlap without strict orthogonality, reflecting potential shared variations among different samples:

$$\mathrm{D}(\mathbf{X}, \mathbf{Y}) = \begin{cases} \mathrm{D}_{cos}(\mathbf{X}, \mathbf{Y}), & \mathrm{D}_{cos}(\mathbf{X}, \mathbf{Y}) \geq m \\ 0, & \mathrm{D}_{cos}(\mathbf{X}, \mathbf{Y}) < m \end{cases}, \tag{10}$$

where m represents the threshold that controls the maximum angle constraint between Delta Features. In Delta Loss, a certain angle is constrained between the Delta Feature of the current iteration and the features in the Delta Queue:

$$\ell_{delta} = \frac{1}{l}\sum_{k=1}^{l}\mathrm{D}(\mathbf{F}_d^c, \mathbf{F}_d^k), \tag{11}$$

where $\mathbf{F}_d^c$ is the Delta Feature of the current iteration and $\mathbf{F}_d^k$ is the k-th featrue in the Delta Queue.

Algorithm 1: Asynchronous Optimization

ϕ_a, ϕ_d and, ϕ represent the parameters of Anchor LoRA, Delta LoRA, and their combined model, respectively.
for number of training iterations **do**
• Update Anchor LoRA by descending its gradient:

$$\nabla_{\phi_a} \frac{1}{l} \sum_{k=0}^{l} \left(\frac{1}{h \times w} \sum_i \min_j \mathbf{C}_{ij}^k \right) \tag{12}$$

• Update Anchor LoRA and Delta LoRA by descending its gradient:

$$\nabla_{\phi} \mathbb{E}_{t,\mathrm{x}_0,\epsilon} \left[\left\| \epsilon - \epsilon_\phi \left(\mathbf{x}_t, \mathbf{c}, t \right) \right\|^2 \right]$$

.
for k steps **do**
• Update Delta LoRA by descending its gradient:

$$\nabla_{\phi_d} \frac{1}{l} \sum_{i=1}^{l} \mathrm{D}(F_d^{cur}, F_d^i).$$

• Update Anchor LoRA and Delta LoRA by descending its gradient:

$$\nabla_{\phi} \mathbb{E}_{t,\mathrm{x}_0,\epsilon} \left[\left\| \epsilon - \epsilon_\phi \left(\mathbf{x}_t, \mathbf{c}, t \right) \right\|^2 \right].$$

end for
end for

Latent Diffusion Model Loss Lastly, the Latent Diffusion Model Loss is employed to constrain the consistency between the images generated by the model after being updated with Dis^{2}Booth and the real reference images:

$$\ell_{LDM} := \mathbb{E}_{t,\mathrm{x}_0,\epsilon} \left[\left\| \epsilon - \epsilon_\phi \left(\mathbf{x}_t, \mathbf{c}, t \right) \right\|^2 \right], \tag{13}$$

where ϕ represents the parameters of Dis^{2}Booth.

The overall loss function is:

$$\ell = \ell_{LDM} + \lambda \ell_{con} + \gamma \ell_{delta}, \tag{14}$$

where λ and γ are coefficients.

Asynchronous Optimization In our experiments, we observe that synchronously optimizing Anchor LoRA and Delta LoRA without supervised optimization can lead to extreme instability, even resulting in model collapse. In Dis^{2}Booth, Anchor LoRA learns the common features that determines the position of the reference data distribution in the entire data space, while Delta LoRA learns the variations that determine how the reference data distribution unfolds at that position.

Therefore, Delta LoRA's optimization should be adapted to Anchor LoRA's, rather than synchronously. As mentioned in Algorithm 1, for every optimization step of Anchor LoRA, we optimize Delta LoRA k times to adapt to the changes in Anchor LoRA. This strategy enables the cooperative optimization of Anchor LoRA and Delta LoRA, leading to stable convergence of the model.

Experiments

Implementation Details

We conduct experiments on Dis^{2}Booth using Stable Diffusion XL (Podell et al. 2023) as the base model with one NVIDIA A100 GPU, one AMD EPYC 7763 CPU and 100GB Memory. We first fine-tune a single LoRA of rank 64 for initializing Anchor LoRA and Delta LoRA. We keep the text encoders of SDXL frozen during the LoRA fine-tuning. For Dis^{2}Booth, we use $\lambda = 1e^{-5}$ and $\gamma = 1e^{-6}$ in Eq. 14, $\alpha = 0.5$ and $\beta = 0.5$ in Eq. 4, and $k = 5$ in Algorithm 1. The code can be found at https://github.com/UniBester/Dis2Booth.

Datasets

We evaluate our method on Animal Faces (Liu et al. 2019), and Flowers (Nilsback and Zisserman 2008). These datasets consist of images from multiple categories, with each category exhibiting a high level of diversity among its samples.
Animal Faces. 119 categories are selected as seen categories for the few-shot image generation methods and 30 as unseen categories.
Flowers. The dataset is divided into 85 seen categories and 17 unseen categories.

Due to the presence of low-resolution data in the aforementioned dataset, which is not compatible with high-quality image generation models (Rombach et al. 2022; Podell et al. 2023), we have performed filtering on the test set based on the image resolution. For each category, only the top 100 images with the highest resolution were retained.

Quantitative Experiments

We evaluate the quality of the generated images based on commonly used FID (Heusel et al. 2017) and LPIPS (Zhang et al. 2018). FID is used to measure the fidelity of the generated images and LPIPS is used to measure the diversity.

We compare Dis^{2}Booth with three diffusion-based personalization models: Dreambooth (Ruiz et al. 2023) with LoRA, Textual Inversion (Gal et al. 2022), and DreamDistribution(Zhao et al. 2023). Since SDXL includes two text encoders, it is challenging to collaboratively optimize both to convergence for DreamDistribution. Therefore, besides using SD 2.0 for DreamDistribution, we have chosen SDXL as the base model for other diffusion model-based methods. Besides, we also compare our method with two state-of-the-art few-shot image generation models: WaveGAN (Yang et al. 2022) and HAE (Li, Zhang, and Wang 2023).

For the images of the seen categories, they are only used to train the base models for few-shot image generation methods. For the images of the unseen categories, each category's 100 images are divided into $\mathcal{S}_{ref}$ and $\mathcal{S}_{real}$ subsets. Each $\mathcal{S}_{ref}$ subset consists of 30 reference images and each $\mathcal{S}_{real}$ subset consists of 70 images. For few-shot image generation methods, images in $\mathcal{S}_{ref}$ are used as the inputs. For the personalization methods, images in $\mathcal{S}_{ref}$ are used as

Methods	Animal Faces		Flowers	
	FID(↓)	LPIPS(↑)	FID(↓)	LPIPS(↑)
WaveGAN	52.30	0.4501	105.61	0.3839
HAE	55.58	0.5080	65.10	0.3964
Textual Inversion	55.48	0.5323	94.91	0.5186
DreamDistribution	57.63	0.5351	96.77	**0.5207**
DreamBooth	32.50	0.5276	112.27	0.5022
Dis^{2}Booth	**27.63**	**0.5356**	**58.15**	0.5189

Table 1: FID(↓) and LPIPS(↑) of images generated by different methods on Animal Faces and Flowers datasets.

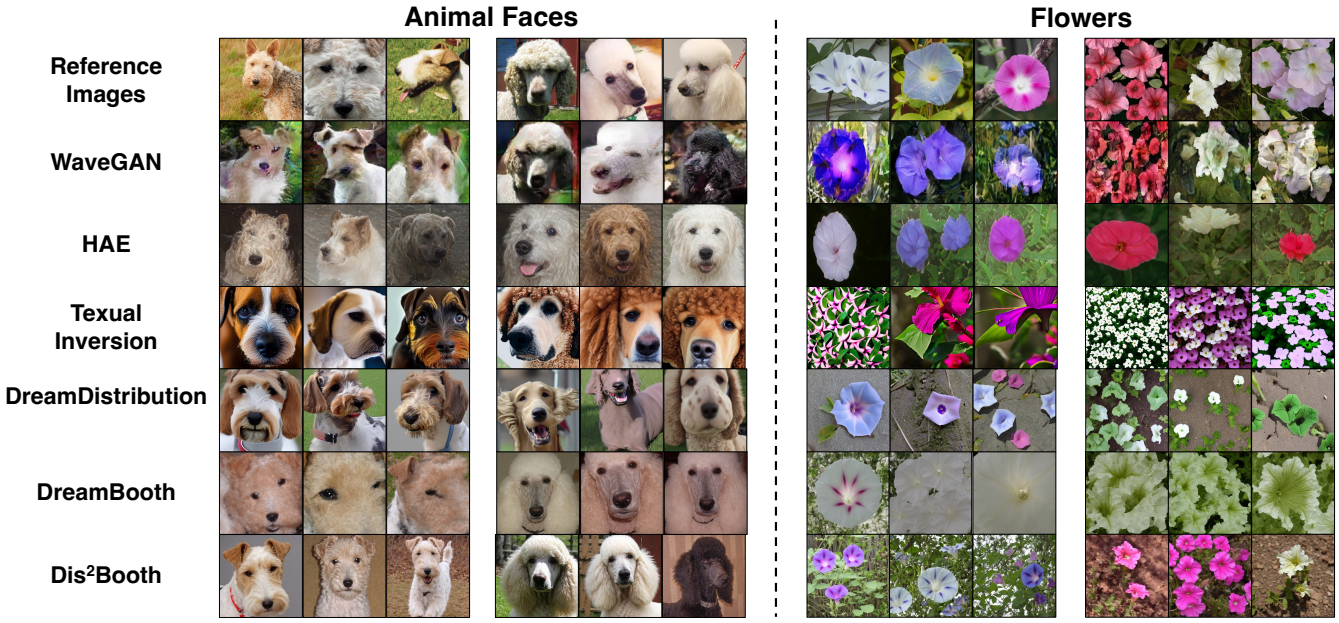

Figure 3: Comparison between images generated by WaveGAN, HAE, Textual Inversion, DreamDistribution, DreamBooth and Dis^{2}Booth on Animal Faces and Flowers.

the fine-tuning data. The images in $\mathcal{S}_{real}$ subset is used as the real comparison images for metric calculation. For every method, 128 images are generated for each unseen category, denoted as $\mathcal{S}_{gen}$. FID and LPIPS are calculated based on the $\mathcal{S}_{real}$ and $\mathcal{S}_{gen}$.

The results of different methods are reported in Table 1. Dis^{2}Booth achieves state-of-the-art performance on the majority of metrics. The methods (Gal et al. 2022; Zhao et al. 2023) based on textual inversion fails to accurately learn the discriminative features of the category, resulting in the incorporation of features from other categories. This leads to an abnormal increase in diversity, an artificially high LPIPS metric, and poor performance in the FID metric. Thanks to the explicit modeling of data distribution and the text control capabilities inherent in pretrained diffusion models, the images generated by Dis^{2}Booth exhibit a high level of diversity while ensuring authenticity.

Qualitative Experiments

We conduct visualizations and qualitative comparisons of images generated by different methods. In Figure 3, all personalization methods utilize the prompt "A close-up photo of [V] XXX". Figure 4 presents visualizations of images generated by the personalization methods on objects with distinctive features, like Lamborghini sports cars and Gundam robots and Vespa motorbike.

Fusion-based methods, like WaveGAN, can only interpolate features from the conditioned images. In many instances, WaveGAN does not well disentangle and fuse the features of different images. This can lead to crashes. For instance, in Figure 3, we observed distortions in the shape of the dogs' faces and the petals of the flowers. HAE is limited by the capabilities of GAN Inversion (Zhu et al. 2016). The generated images may lose detailed information and discriminative features. For example, in the given images, the petals of the flower appear as a single mass, and the dogs' faces appear shortened.

Textual Inversion and DreamDistribution face similar issues, *i.e.*, optimizing only text embeddings, they generate images resembling a recombination of priors learned by the base model. While being effective for common data, they struggle with uncommon ones, often failing to select suit-

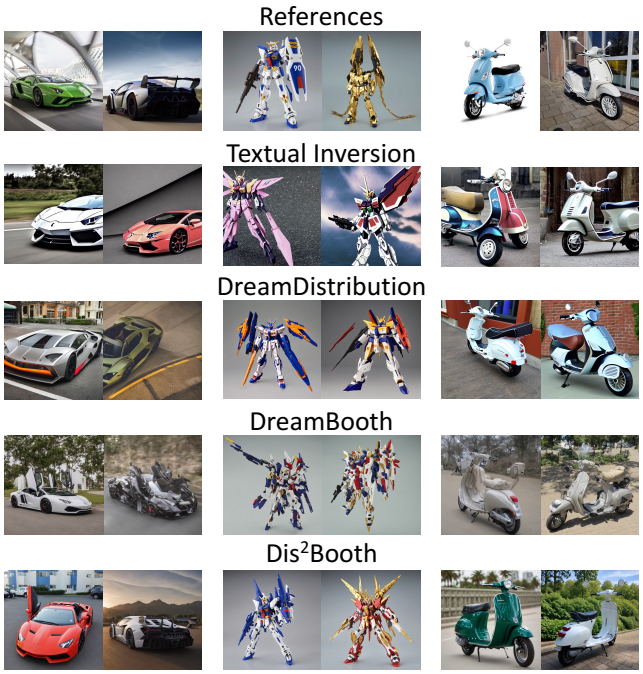

Figure 4: Objects (Lamboghini cars, Gundam models, and Vespa motorbikes) generated by Textual Inversion, DreamDistribution, DreamBooth and Dis^{2}Booth.

able priors for fusion, resulting in drift or loss of discriminative semantics. As shown in Figure 3 the flowers with wrong color are generated by them, the Poodle generated by Textual Inversion has a square mouth, and the Fox Terrier generated by DreamDistribution has long ears. In Figure 4, the Gundams models generated by them are all lack realism and Vespa motorbikes have structural errors. As the diversity of the reference images increases, DreamBooth struggles to accurately learn the data distribution, resulting in a decline in the fidelity of generated images. In Figure 4, it generates vehicles and models with repetitive structures and meaningless artifacts. Images generated by Dis^{2}Booth maintain high fidelity while achieving higher diversity and semantic consistency.

Experiments on Flexibility

In Dis^{2}Booth, the weights of Anchor LoRA and Delta LoRA can be fused into a single LoRA weight during the inference process. Theoretically, Dis^{2}Booth also possesses the flexibility and a wide range of applications that LoRA has. We conduct experiment on two settings, namely, 3D generation and the fusion of LoRAs.

Fusion of LoRAs ZipLoRA (Shah et al. 2023) allows for merging of independently trained style and instance LoRAs to generate any instance in any style. By combining Dis^{2}Booth and ZipLoRA, we have enabled the generated diverse instances to transform into different styles. As shown in Fig. 5 (a), the images generated by Distribution LoRA can seamlessly blend into various styles.

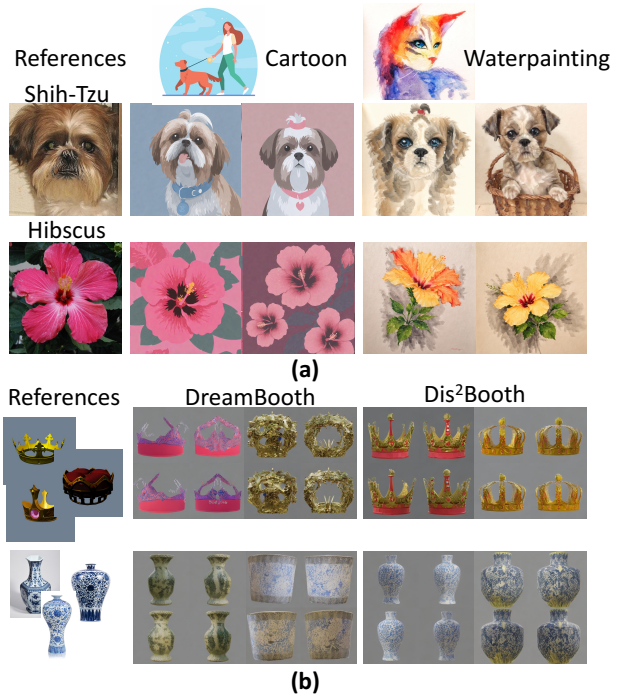

Figure 5: (a) Images generated by fusing Dis^{2}Booth and a style LoRA. (b) Personalized 3D model generation comparison between DreamBooth and Dis^{2}Booth.

3D Generation MVDream (Shi et al. 2023) is a text-to-3D diffusion model. It is capable of generating 3D models or multi-view consistent images based on text descriptions. We combine Dis^{2}Booth with MVDream to enable personalized generation of more complex data distributions. As shown in Fig. 5 (b), the 3D models generated by Dis^{2}Booth geometric regularity and texture realism compared to DreamBooth.

Limitations

Although Dis^{2}Booth is capable of learning higher diversity and complexity in data distribution, it still has some limitations. Firstly, large-scale pretrained diffusion models trained at high resolutions can crash during fine-tuning with low-resolution images. When there are very few reference images and the training iterations are excessive, overfitting can occur. In the future, we will explore effective prior preservation methods to prevent such overfitting phenomena.

Conclusion

In this work, we focus on personalizing the diffusion models to learn to generate varied data with higher diversity and complexity. We present a new personalized image generation method, Dis^{2}Booth, which explicitly models the data distribution as the combination of common features and reasonable variations. Extensive experiments suggest that Dis^{2}Booth can acurately generate varied image data while maintaining high flexibility.

Acknowledgments

This work is supported in part by the National Key R&D Program of China under Grant 2023YFC2508704, in part by National Natural Science Foundation of China: 62236008, U21B2038 and 61931008, and in part by the Fundamental Research Funds for the Central Universities.

References

Antoniou, A.; Storkey, A.; and Edwards, H. 2017. Data Augmentation Generative Adversarial Networks. *arXiv preprint arXiv:1711.04340*.

Chen, H.; Zhang, Y.; Wu, S.; Wang, X.; Duan, X.; Zhou, Y.; and Zhu, W. 2023. Disenbooth: Identity-preserving disentangled tuning for subject-driven text-to-image generation. In *The Twelfth International Conference on Learning Representations*.

Clouâtre, L.; and Demers, M. 2019. FIGR: Few-shot Image Generation with Reptile. *arXiv preprint arXiv:1901.02199*.

Dhariwal, P.; and Nichol, A. 2021. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34: 8780–8794.

Ding, G.; Han, X.; Wang, S.; Jin, X.; Tu, D.; and Huang, Q. 2023. Stable Attribute Group Editing for Reliable Few-shot Image Generation. *arXiv preprint arXiv:2302.00179*.

Ding, G.; Han, X.; Wang, S.; Wu, S.; Jin, X.; Tu, D.; and Huang, Q. 2022. Attribute group editing for reliable few-shot image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11194–11203.

Gal, R.; Alaluf, Y.; Atzmon, Y.; Patashnik, O.; Bermano, A. H.; Chechik, G.; and Cohen-or, D. 2022. An Image is Worth One Word: Personalizing Text-to-Image Generation using Textual Inversion. In *The Eleventh International Conference on Learning Representations*.

Gu, Z.; Li, W.; Huo, J.; Wang, L.; and Gao, Y. 2021. LoFGAN: Fusing Local Representations for Few-Shot Image Generation.

Han, L.; Li, Y.; Zhang, H.; Milanfar, P.; Metaxas, D.; and Yang, F. 2023. Svdiff: Compact parameter space for diffusion fine-tuning. *arXiv preprint arXiv:2303.11305*.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9729–9738.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33: 6840–6851.

Hong, Y.; Niu, L.; Zhang, J.; Liang, J.; and Zhang, L. 2020a. DeltaGAN: Towards Diverse Few-shot Image Generation with Sample-Specific Delta.

Hong, Y.; Niu, L.; Zhang, J.; and Zhang, L. 2020b. Matchinggan: Matching-Based Few-Shot Image Generation.

Hong, Y.; Niu, L.; Zhang, J.; Zhao, W.; Fu, C.; and Zhang, L. 2020c. F2gan: Fusing-and-filling gan for few-shot image generation. In *Proceedings of the 28th ACM International Conference on Multimedia*, 2535–2543.

Kumari, N.; Zhang, B.; Zhang, R.; Shechtman, E.; and Zhu, J.-Y. 2023. Multi-concept customization of text-to-image diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1931–1941.

Li, L.; Zhang, Y.; and Wang, S. 2023. The euclidean space is evil: hyperbolic attribute editing for few-shot image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 22714–22724.

Liang, W.; Liu, Z.; and Liu, C. 2020. DAWSON: A Domain Adaptive Few Shot Generation Framework. *arXiv preprint arXiv:2001.00576*.

Liu, M.-Y.; Huang, X.; Mallya, A.; Karras, T.; Aila, T.; Lehtinen, J.; and Kautz., J. 2019. Few-shot Unsueprvised Image-to-Image Translation.

Nilsback, M.-E.; and Zisserman, A. 2008. Automated Flower Classification over a Large Number of Classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*.

Phaphuangwittayakul, A.; Guo, Y.; and Ying, F. 2021. Fast Adaptive Meta-Learning for Few-shot Image Generation. *IEEE Transactions on Multimedia*, 1–1.

Podell, D.; English, Z.; Lacey, K.; Blattmann, A.; Dockhorn, T.; Müller, J.; Penna, J.; and Rombach, R. 2023. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*.

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 *International conference on machine learning*, 8748–8763. PMLR.

Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; and Chen, M. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2): 3.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 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. *Advances in Neural Information Processing Systems*, 35: 36479–36494.

Shah, V.; Ruiz, N.; Cole, F.; Lu, E.; Lazebnik, S.; Li, Y.; and Jampani, V. 2023. Ziplora: Any subject in any style by effectively merging loras. *arXiv preprint arXiv:2311.13600*.

Shi, Y.; Wang, P.; Ye, J.; Mai, L.; Li, K.; and Yang, X. 2023. MVDream: Multi-view Diffusion for 3D Generation. In *The Twelfth International Conference on Learning Representations*.

Song, J.; Meng, C.; and Ermon, S. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Yang, M.; Wang, Z.; Chi, Z.; and Feng, W. 2022. WaveGAN: Frequency-Aware GAN for High-Fidelity Few-Shot Image Generation. In *European Conference on Computer Vision*, 1–17. Springer.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 586–595.

Zhao, B. N.; Xiao, Y.; Xu, J.; Jiang, X.; Yang, Y.; Li, D.; Itti, L.; Vineet, V.; and Ge, Y. 2023. DreamDistribution: Prompt Distribution Learning for Text-to-Image Diffusion Models. *arXiv preprint arXiv:2312.14216*.

Zhu, J.-Y.; Krähenbühl, P.; Shechtman, E.; and Efros, A. 2016. Generative Visual Manipulation on the Natural Image Manifold. *ECCV*.