DreamDistribution: Prompt Distribution Learning for Text-to-Image Diffusion Models

Brian Nlong Zhao[1], Yuhang Xiao[1*], Jiashu Xu[2*], Xinyang Jiang[3], Yifan Yang[3], Dongsheng Li[3], Laurent Itti[1], Vibhav Vineet[4†], Yunhao Ge[1†]

[1]University of Southern California [2]Harvard University
[3]Microsoft Research Asia [4]Microsoft Research Redmond

[*]=equal contribution as second author [†]=equal contribution

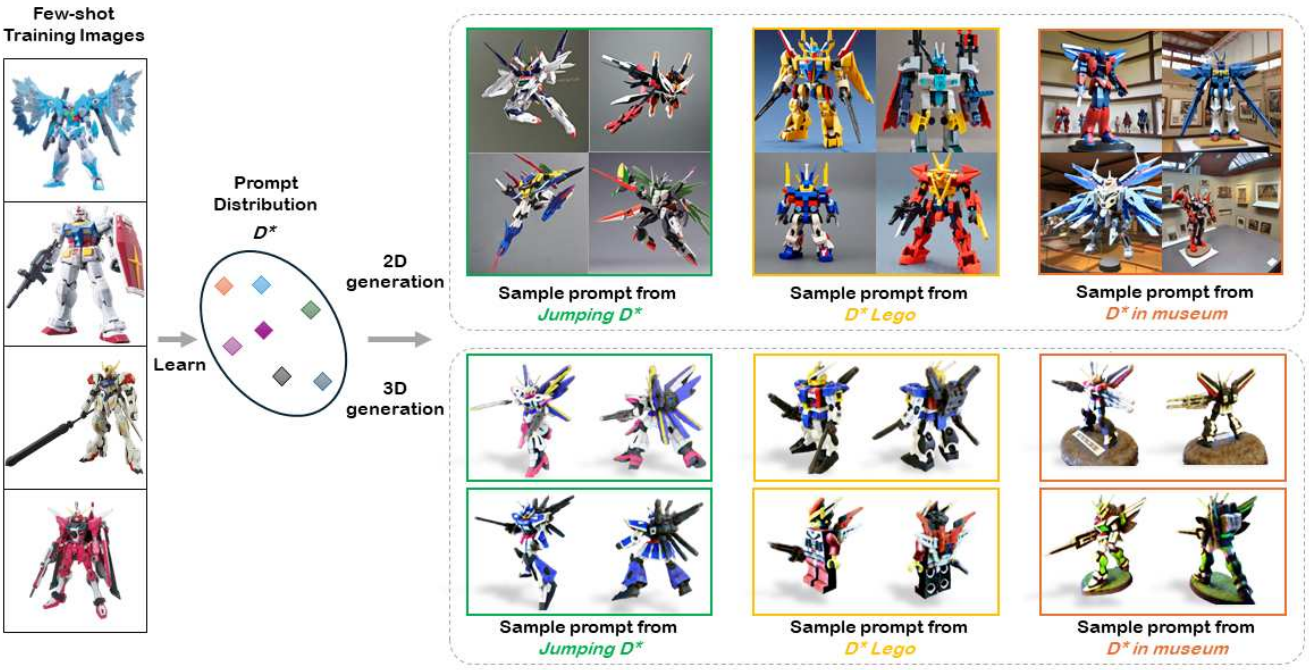

Figure 1. DreamDistribution learns a prompt distribution D^* that represents a distribution of descriptions corresponding to a set of reference images. We can sample new prompts from D^* or modified D^* by text-guided editing to generate images of diverse new instance that follows the visual attributes of reference training images (top). We can also apply a learned distribution flexibly to, for example, a pretrained text-to-3D model, and generate diverse new 3D assets following the reference images (bottom).

Abstract

The popularization of Text-to-Image (T2I) diffusion models enables the generation of high-quality images from text descriptions. However, generating diverse customized images with reference visual attributes remains challenging. This work focuses on personalizing T2I diffusion models at a more abstract concept or category level, adapting commonalities from a set of reference images while creating new instances with sufficient variations. We introduce a solution that allows a pretrained T2I diffusion model to learn a set of soft prompts, enabling the generation of novel images by sampling prompts from the learned distribution. These prompts offer text-guided editing capabilities and additional flexibility in controlling variation and mixing between multiple distributions. We also show the adaptability of the learned prompt distribution to other tasks, such as text-to-3D. Finally we demonstrate effectiveness of our approach through quantitative analysis including automatic evaluation and human assessment. Project website https://briannlongzhao.github.io/DreamDistribution

1. Introduction

Dreams have long been a source of inspiration and novel insights for many individuals [5, 6, 47]. These mysterious subconscious experiences often reflect our daily work and life [6]. However, these reflections are not mere replicas; they often recombine elements of our reality in innovative ways, leading to fresh perspectives and ideas. We aim to emulate this fascinating mechanism in the realm of text-to-image generation.

Text-to-image (T2I) generation has recently been popularized due to the astonishing performance of state-of-the-art diffusion models such as Stable Diffusion [37] and DALL·E 2 [35]. Variations of the T2I models have enabled several fascinating applications that allow user to control the generation, such as conditioned generation based on other input modalities [23, 53, 55], inpainting [27, 52], image editing [1, 29]. One such interesting application is personalization of T2I models, where user provides some reference images of the same instance (*e.g.* their pet dog), and the personalized model can generate images based on the references, with the flexibility of text-guided editing for new context. This is generally achieved by associating a token with the personalized concept through fine-tuning the model parameters [20, 38] or newly added learnable token embeddings [7, 48].

In many cases, however, user may want to personalize T2I generation over a more abstract visual attribute instead of a specific instance-level personalization. For example, a designer may seek inspiration by generating a variety of novel cartoon characters or scenery images following similar visual attributes presented in their previous works. In this case, trying over text prompts is not scalable and hard to get desired result that follows the desired visual attributes. On the other hand, using the existing personalization methods aforementioned is likely to fail when training images when the training images do not represent the same instance, but rather encompass a distribution sharing certain, yet challenging-to-articulate, commonalities. Additionally, existing personalization methods often result in limited diversity and variation during generation (Fig. 3). Since the associated token is fixed, these methods will typically learn a token that is either overfitted to a combination of visual features, or learn a token that is overly generalized, which introduces more randomness into the uncontrollable diffusion process, thereby failing to follow desired visual attributes in generated images.

In this work, we propose DreamDistribution, a prompt distribution learning approach on T2I diffusion model for various downstream tasks (Fig. 1). Our proposed solution has three key components (Fig. 2). First, to adapt a pre-trained fixed T2I model, instead of fine-tuning diffusion model parameters, our method builds on prompt tuning [58, 59], where we use soft learnable prompt embeddings with the flexibility to concatenate with text, to associate with the training image set. This design have several advantages: (1) It prevents catastrophic forgetting of the pre-trained model, enabling it to learn an almost infinite variety of target prompt distributions using the same T2I diffusion model. (2) It is highly efficient in terms of parameters, requiring only the prompt itself as the learnable element. (3) The learned prompts remain within the semantic space of natural language, offering text-guided editing capabilities and generalizing to other pre-trained diffusion models, such as text-to-3D. (4) The learned distribution increased flexibility in managing variations. Second, we introduce a distribution of prompts to model various attributes described by reference images at a broader level. The prompt distribution is modeled by a set of learnable prompt embeddings to associate with the training image set as a whole. The learned prompt distribution can be treated as a distribution of learned "descriptions" of the reference images and should be able to model the commonalities and variations of visual attributes, *e.g.*, foreground, style, background, texture, pose. During inference, we sample from the prompt distribution, which should have a similar semantic meaning, understood by the downstream denoising network, to produce in-distribution outputs with appropriate variations. Lastly, to effectively optimize the set of soft prompts that models the distribution, we apply a simple reparameterization trick [19] and an orthogonal loss to update the prompts at token embedding space simultaneously and orthogonally.

We first demonstrate the effectiveness of our approach in customizing image generation tasks (§4). By taking a small set of images of interest as training images, we demonstrate that our approach can generate diverse in-distribution images where baseline methods fail to generate desired output. The diversity and the quality of our synthetic images are verified via automatic and human evaluation (Section 4.2). We show that the learned distribution holds the capability of text-guided editing, as well as further controllability such as scaling the variance and composition of distributions (Section 4.3). Next we highlight that the learned prompt distribution can be easily applied to other text-guided generation tasks such as pretrained text-to-3D models (Section 4.4). Lastly we show the effectiveness of our method on personalized distribution generation through classification task with synthetic training data as a proxy (Section 4.5).

In summary, our contributions are:

- We propose a distribution based prompt tuning methods for personalized distribution generation by learning soft prompt distribution using T2I diffusion model.
- Using a public available pretrained T2I diffusion model, we experiment our approach on customization T2I generation tasks and show that our approach can capture visual attributes into prompt distribution and can generate diverse in-distribution images that follows text-guided edit-

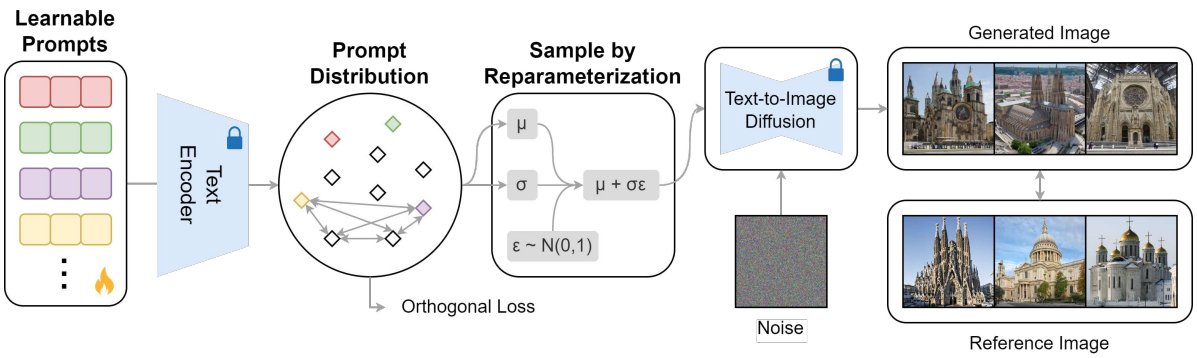

Figure 2. Overview of DreamDistribution for learning a prompt distribution. We keep a set of K learnable soft prompts and model a distribution of them at the CLIP text encoder feature space. Only prompts are learnable, CLIP encoder and the T2I diffusion model are all fixed. We use a reparameterization trick to sample from the prompt distribution and update the learnable prompts through backpropagation. The training objective is to make the generated images aligns with the reference image. An additional orthogonal loss is incorporated to promote differentiation among learnable prompts. For inference, we similarly sample from the prompt distribution at text feature space to guide the pretrained T2I generation.

ing.

- Further experiments show that our learned distribution is controllable and flexible and easy to be adapted to other generation tasks that requires text as input.
- We further quantitatively demonstrate the effectiveness of our approach using synthetic image dataset generation tasks as a proxy and also through automatic evaluation metrics and human evaluation.

2. Related Works

2.1. Text-to-image Diffusion Models

Diffusion models [4, 16, 44] have achieved great success in various image generation tasks. State-of-the-art T2I models such as Imagen [40] and DALL·E 2 [35] trained on large scale data demonstrate remarkable synthesis quality and controllability. Latent Diffusion Models [37] and its open-source implementation, Stable Diffusion [37], have also become a prevailing family of generative models. In these T2I diffusion models, text is encoded into latent vectors by pre-trained language encoders such as CLIP [33], and the denoising process is conditioned on latent vectors to achieve text-to-image synthesis. However, such models trained on large scale text-image pairs are not designed to generate personalized images such as images of one's pet dog, therefore only the text conditioning cannot provide fine-grained control over the generated images.

2.2. Personalized text-to-image Generation

Various approaches are proposed to better control the text-guided diffusion models and achieve personalization. Textual Inversion [7] proposed to search for a new token in the embedding space representing a visual concept via optimizing a word embedding vector. DreamBooth [38] fine-tunes all parameters of the model to associate a personalized subject into an rarely used token. Custom Diffusion [20] employs that fine-tuning method but only fine-tune cross-attention layers to reduce the training time, with the ability to learn multiple concepts jointly. Subsequent works [42, 48, 50] mainly borrow the ideas from these works and focus on solving their drawbacks.

2.3. Prompt Learning

Prompt learning is a popular method in natural language processing (NLP). The main idea is to transfer various downstream NLP tasks to masked language modeling problems via adopting proper prompt templates [2, 21, 22, 32] instead of fine-tuning the pretrained language model. Searching for the appropriate prompts is the key of this method. Prompt engineering [2, 32] adopts carefully-designed discrete (hard) prompts crafting by human, while prompt tuning [21, 22] automatically searches for the desired prompts in the embedding space via learning continuous (soft) prompts. The great success of NLP inspires computer vision researchers and prompt engineering is explored in pretrained vision-language models such as CLIP [33] and ALIGN [17]. CoOp [59] applies the idea of prompt tuning in vision-language tasks, which learns a continuous prompt via minimizing the classification loss of the downstream tasks. ProDA [26] learns a distribution of diverse prompts to capture various representations of a visual concept instead of a single prompt in CoOp [59], which achieves better generalization.

Most relevant to our work are Textual Inversion [7] and ProDA [26]. Textual Inversion learns a fixed token embedding associated with a pseudo-word. Ours learns a distribution of prompts in the CLIP feature space like ProDA [26], allowing for learning the visual concept with diverse visual

representations and capturing the details for reconstructions and plausible synthesis.

3. Method

Given a set of images with some common visual attributes (*e.g*. same category, similar style), our goal is to capture the visual commonalities and variations and model by a prompt distribution in the text feature space, which could be compatible with natural language. The commonalities among reference images may be challenging to articulate with natural language prompts. We can thus sample prompts from the distribution to guide T2I diffusion model to generate diverse unseen images while at the same time following the common traits distribution. The inherent characteristics of the learned prompts are compatible with natural language instructions and other pretrained text-guided generation models.

3.1. Text-to-Image Diffusion

Text-to-image diffusion models are a class of generative models that learns image or image latent distribution by gradually denoising a noise sampled from Gaussian distribution. Specifically, given a natural language text prompt, a tokenizer followed by a text embedding layer map the input text to a sequence of embedding vectors $\mathbf{p}$. A text encoder converts the text embedding into text features $\mathbf{c} = \mathcal{E}(\mathbf{p})$ used for conditioning the generation process. An initial noise $\boldsymbol{\epsilon}$ is sampled from $\mathcal{N}(\mathbf{0}, \mathbf{I})$, and the denoising model $\boldsymbol{\epsilon}_\theta$ predicts the noise added to a noisy version of image of image latent $\mathbf{x}$. The denoising model $\boldsymbol{\epsilon}_\theta$ is optimized using the objective:

$$\mathcal{L} = \mathbb{E}_{\mathbf{x},\mathbf{c},\boldsymbol{\epsilon},t}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\mathbf{x}_t, \mathbf{c}, t\right)\right\|_2^2\right] \tag{1}$$

where $\mathbf{x}$ is the ground-truth image or image latent obtained from a learned autoencoder, $\mathbf{x}_t$ is the noisy version of $\mathbf{x}$ at time-step t, and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

3.2. Prompt Tuning

Our proposed method is grounded in the notion of prompt tuning, which aims to learn a soft continuous prompt on target task and is widely used in fine-tuning NLP models. [11, 21, 22, 24, 25] Specifically, for a pretrained model that takes natural language prompt as input, we can formulate a prompt with continuous learnable token embeddings $\mathcal{P} = [\text{PREFIX}]\,\mathbf{V}\,[\text{SUFFIX}] \in \mathbb{R}^{L\times d}$, where [PREFIX] and [SUFFIX] are word embeddings of natural language prefix and suffix if needed, and L represents the prompt length or the total number of tokens, and d represent the dimension of word embeddings. $\mathbf{V} = [\mathbf{v}]_1 \ldots [\mathbf{v}]_M \in \mathbb{R}^{M\times d}$ represents a sequence of M learnable token embedding vectors with same dimension as word embeddings. During fine-tuning, the parameters of the pretrained generation model remain fixed, and only the learnable token embeddings $\mathbf{V}$ are updated through direct optimization employing the corresponding loss function backpropagated through generator $\boldsymbol{\epsilon}_\theta$ and text encoder $\mathcal{E}$. Formally, prompt tuning aims to find optimized embedding vectors $\mathbf{V}^* = \arg\max_{\mathbf{V}} P(Y \mid \mathcal{P}, X)$, where X and Y are input data and output label, respectively.

Prior works have shown the efficacy of adopting prompt tuning techniques on vision-language models for image classification tasks [18, 58, 59]. Gal *et al*. [7] adopts similar prompt tuning methods that enable personalized generation. However, the limitation of this approach lies in its constraint to personalize only one particular concept, such as a specific dog, as it employs a fixed token embedding for concept encoding.

3.3. Learning Prompt Distribution

We aim to model more general commonalities and variations presented in the reference image set and generate diverse images of new instances that visually align, therefore we propose to model a learnable distribution of prompts for the reference images. Inspired by Lu *et al*. [26], which proposed to estimate a distribution of prompt for image classification tasks, we propose to model a distribution of learnable prompts over a sequence of M token embeddings to capture the distribution of visual attributes on T2I generation task leveraging diffusion model.

Our methods builds on Stable Diffusion [37], where a pretraind CLIP [33] text encoder is used for obtaining text feature of the prompt. Due to the contrastive training objective of CLIP, features of texts that have similar semantic meaning have high cosine similarity and therefore close to each other in CLIP feature space [33]. Lu *et al*. [26] have also shown that for text prompts that describe images of the same category, the CLIP text feature $\mathbf{c}$ output from pretrained CLIP text encoder are adjacent to each other in a cluster. Therefore, it is natural to model a Gaussian distribution of $\mathbf{c}$ that describes images of same category or with shared attributes. To do so, instead of keeping one learnable soft prompt to optimize during training, we maintain a set of K learnable prompts $\mathcal{P}^K = \{\mathcal{P}_k = [\text{PREFIX}]\,\mathbf{V}_k\,[\text{SUFFIX}]\}_{k=1}^K$ corresponds to a set of similar reference images. Our goal is to optimize the set of learnable token embeddings $\{\mathbf{V}_k\}_{k=1}^K$. With K learnable prompts, we can estimate the mean $\boldsymbol{\mu}_\mathbf{c} = \boldsymbol{\mu}\left(\mathcal{E}\left(\mathcal{P}^K\right)\right) \in \mathbb{R}^{L\times d_\mathcal{E}}$ and standard deviation $\boldsymbol{\sigma}_\mathbf{c} = \boldsymbol{\sigma}\left(\mathcal{E}\left(\mathcal{P}^K\right)\right) \in \mathbb{R}^{L\times d_\mathcal{E}}$ at $\mathcal{E}$ text encoder space, where $d_\mathcal{E}$ is the feature dimension of text encoder space.

Applying to the training objective of T2I diffusion model, Eq. (1) becomes:

$$\mathcal{L}\left(\mathcal{P}^K\right) = \mathbb{E}_{\mathbf{x},\tilde{\mathbf{c}},\boldsymbol{\epsilon}\,t}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\mathbf{x}_t, \tilde{\mathbf{c}}, t\right)\right\|_2^2\right] \tag{2}$$

where $\tilde{\mathbf{c}} \sim \mathcal{N}\left(\boldsymbol{\mu}_{\mathbf{c}}, \boldsymbol{\sigma}_{\mathbf{c}}^2\right)$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is the sampled Gaussian noise added to the image or image latent. However, sampling $\tilde{\mathbf{c}}$ from a distribution makes it not differentiable for optimization, therefore we apply the reparameterization trick similar to that used in VAE [19]. Formally, since $\tilde{\mathbf{c}} \sim \mathcal{N}\left(\boldsymbol{\mu}_{\mathbf{c}}, \boldsymbol{\sigma}_{\mathbf{c}}^2\right)$, we can rewrite the optimization objective Eq. (2) as:

$$\mathcal{L}\left(\mathcal{P}^K\right) = \mathbb{E}_{\mathbf{x}, \boldsymbol{\omega}, \boldsymbol{\epsilon}, t}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\mathbf{x}_t, \boldsymbol{\mu}_{\mathbf{c}} + \boldsymbol{\omega}\boldsymbol{\sigma}_{\mathbf{c}}, t\right)\right\|_2^2\right] \quad (3)$$

where $\boldsymbol{\omega} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ has the same dimension as $\boldsymbol{\mu}_{\mathbf{c}}$ and $\boldsymbol{\sigma}_{\mathbf{c}}$. Since the exact computation of $\mathcal{L}\left(\mathcal{P}^K\right)$ is intractable, we use a Monte Carlo approach to sample $\boldsymbol{\omega}$ for S times to approximate the expected value for optimization:

$$\mathcal{L}\left(\mathcal{P}^K\right) = \frac{1}{S}\sum_{s=1}^{S}\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\mathbf{x}_t, \boldsymbol{\mu}_{\mathbf{c}} + \boldsymbol{\omega}_s\boldsymbol{\sigma}_{\mathbf{c}}, t\right)\right\|_2^2 \quad (4)$$

In order to avoid the scenario wherein multiple prompt features converge to a same vector, which will result in a non-representative low-variance distribution, we apply a similar orthogonal loss proposed in [26] to penalize on the cosine similarity and encourage orthogonality between each pair of prompts:

$$\mathcal{L}_{\text{ortho}} = \frac{1}{K(K-1)}\sum_{i=1}^{K}\sum_{j=i+1}^{K}\left|\langle\mathcal{E}(\mathcal{P}_i), \mathcal{E}(\mathcal{P}_j)\rangle\right| \quad (5)$$

where $\langle\cdot,\cdot\rangle$ is cosine similarity between a pair of vectors. The total loss is therefore:

$$\mathcal{L} = \mathcal{L}(\mathcal{P}^K) + \lambda\mathcal{L}_{\text{ortho}} \quad (6)$$

where λ is a hyperparameter.

Implementation Details In all experiments, we use Stable Diffusion 2.1 [37] and keep all the default hyperparameters. We use $S = 4$ and $\lambda = 5 \times 10^{-3}$. We use $K = 32$ prompts in all personalized generation experiments, and $K = 10$ prompts to reduce computation in synthetic dataset experiments. We use 1,500 steps with constant learning rate of 10^{-3}.

4. Experiments

In this section, we demonstrate several experiments and applications of our approach and show visual results of generated images. We show the ability of our approach to capture a distribution of reference images and generate in-distribution novel images in Sec. 4.1. We present additional quantitative results including automatic evaluation and user studies in Sec. 4.2. We also show the flexibility and effects of manipulating and text-guide editing learned prompt distribution in Sec. 4.3. We further highlight easy application of our learned prompt distribution to other text-based generation tasks using text-to-3D as an example in Sec. 4.4. Finally in Sec. 4.5 we present experiments that show the effectiveness of our approach in generating synthetic training dataset.

4.1. Diverse Personalized Generation

We first demonstrate the ability of our approach to generate images that preserve general visual features shown in training set and at the same time generate new images with high diversity. Given a diverse set of few training images (typically 5-20) that are not easily describable in texts and at the same time share some similar visual attributes, we can generate diverse in-distribution images by simply sampling from the learned distribution as the input prompt text embedding to T2I diffusion model. Our learned prompt distribution can be therefore treated as a distribution of descriptions corresponding to the set of training images.

Baselines. We compare with popular instance-level personalization methods including **Textual Inversion** [7], **DreamBooth** [38], **Custom Diffusion** [20]. We also evaluate against **Short Caption** that uses a short description as text prompt, and **Long Caption** that uses a longer text caption with detailed descriptions. These comparisons emphasize our method's ability to take care of both similarity and diversity referencing the training images. We use the same pretrained Stable Diffusion version 2.1 with default hyperparameters provided in baseline works. We use $M = 8$ context vectors without adding any prefix or suffix texts in either training or inference process for DreamDistribution.

Results Fig. 3 shows visualized comparison with baselines. In general, both short and long text prompting methods fail to generate results that visually follow the reference images since there is no training involved and the image details are hard to describe in language. Images generated using baseline methods generally show limited variation or inconsistent visual attributes in all examples. All these methods try to associate different visual concepts with a fixed token, which does not provide any semantic variations itself. Although the denoising process enables some randomness, the training objective of associating various concepts with a fixed token will either fail to capture a distribution due to non-convergence, leading to underfitting to generic image category information, or overfits to a visual combination of the training images. By modeling multiple concepts using multiple prompts and optimizing the prompt distribution, our proposed method is able to produce substantial variations of style and view points, for example, following the reference images in the cathedral example (first column). Ours method can also model the texture and background information and generate new instance with significant vari-

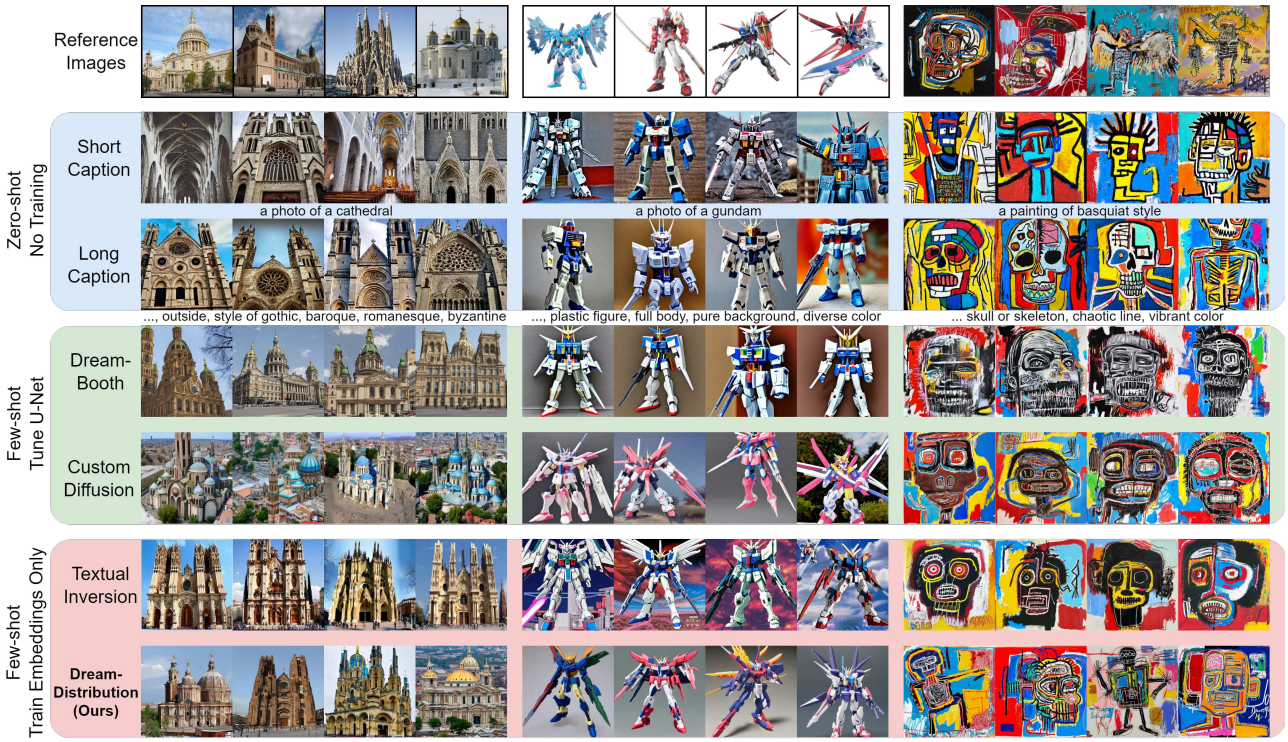

Figure 3. Comparison of results with existing methods. Given a set of training images (typically 5-20, we only show 4 here), we compare generation results with other existing methods. We use Stable Diffusion version 2.1 for all methods. As can be seen on the bottom row, our method is able to generate more diverse and coherent images (also quantitatively analyzed by automatic and human evaluation in Section 4.2).

ations in color and pose following the reference images of the Gundam example (second column), as well as patterns, lines, style as a whole and generate novel artistic creations as shown in the Basquiat's painting example (third column). In all, DreamDistribution is able to produce substantial variations on style, viewpoints, pose, layout, etc., with appropriate visual attributes following the reference images.

4.2. Generation Quality and Diversity Evaluation

Model	FID↓	CLIP-I↑	DINO↑	Density↑	Coverage↑
DreamBooth [38]	$234.90_{71.87}$	$0.79_{0.06}$	$0.46_{0.10}$	$0.91_{0.52}$	$0.74_{0.32}$
Textual Inversion [7]	$224.23_{75.49}$	$0.83_{0.04}$	$0.48_{0.10}$	$1.28_{0.44}$	$0.82_{0.17}$
Custom Diffusion [20]	$236.61_{72.76}$	$0.80_{0.05}$	$0.46_{0.07}$	$1.45_{0.79}$	$0.87_{0.18}$
Ours	$\mathbf{215.15}_{72.65}$	$\mathbf{0.84}_{0.03}$	$\mathbf{0.50}_{0.09}$	$\mathbf{1.59}_{0.47}$	$\mathbf{0.93}_{0.09}$

Table 1. Our method achieves the best quality and diversity automatic metrics across 12 scenarios. Mean metrics are reported with standard deviations shown in subscript.

We quantitatively assess our methods in terms of diversity and quality, and further use synthetic ImageNet classification performance as a proxy in Section 4.5. We train DreamBooth, Textual Inversion, Custom Diffusion and DreamDistribution on 12 diverse image scenarios including photos of real objects in large and small scales, works of famous artists, as well as illustrations of cartoon characters and scenery images with prominent styles, sourced from illustrators from online communities. For our approach we use $M = 4$ learnable context with no prefix and suffix in both training and generating stages.

Automatic Metrics We evaluate the generative images on established automatic evaluation metrics that measure the diversity of synthetic images and the similarity between real and synthetic images. Following prior works [15, 30, 38, 56], in Tab. 1 we evaluate image quality using **FID** [15] that measures the distance between the distribution of generated images and the distribution of real images via InceptionV3 [45]; **CLIP-I** and **DINO** [38] that measures average pairwise cosine similarity between CLIP [33] and DINOv1 [3] embeddings. Our method achieves the best quality across all three quality measurements, suggesting that our method is capable of creating more high-quality images that fulfill the prompt requirement. Additionally, we report **Density** and **Coverage** [30] in Tab. 1. Density measures samples in regions where real samples are densely packed, while coverage calculates fraction of real samples whose neighbourhoods contain at least one generated sam-

ple. Both metrics are calculated with DINOv2 [31]. Our method achieves the best coverage and diversity across the board.

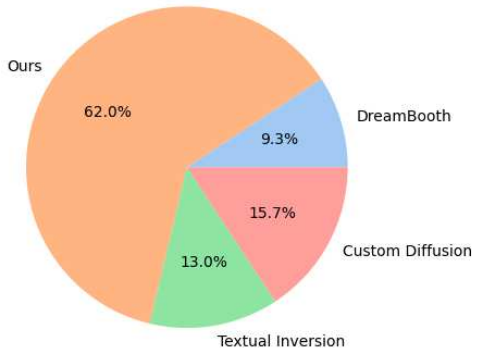

Figure 4. Human Evaluation on image diversity (Section 4.2) aligns with automatic evaluation (Tab. 1). Our method shows significantly greater diversity, which may explain why it was able to better train image classifiers in Tab. 2.

Human Evaluation Admittedly, automatic evaluation does not fully capture the richness perceived by human observers. We further investigate if Tab. 1 correlates with human perception via conducting human evaluation based on those 12 sets of reference images. For each reference image set, we generate images using DreamBooth, Textual Inversion, Custom Diffusion, and our method, with 40 images per method, resulting in a total of 1,920 generated images in the evaluation set. We assign 10 independent annotators. For each of the 12 reference sets, annotators are asked to choose the most preferable set of generated images based on their perceived similarity with the reference set and the diversity within the generated set. The methods are anonymized so annotators are unaware of which generated set corresponds to which method. We collect a total of 120 samples and count the frequency of preferences. Fig. 4 demonstrates that our generated images exhibit superior diversity compared to three baseline models, reinforcing our intuition that by learning distribution we are able to generate diverse images with coherent content and visual attributes presented in the reference image.

4.3. Controllability of Prompt Distribution

Since our learned prompt distribution is in the CLIP text feature space, it is natural to manipulate the learned distribution based on the property of CLIP text feature space. We show several interesting distribution manipulation methods, including text-guided editing, scaling the variance for diversity control, interpolation between multiple distributions.

Figure 5. Effect of scaling the variance of a learned prompt distribution. Image diversity increases as the scaling factor γ increases.

Figure 6. Composition of prompt distributions using linear interpolation between Chinese painting and Van Gogh. Mixing ratio changes linearly from left to right. The middle columns show mixtures of two styles.

Text-guide Editing Similar to existing personalization methods [7, 20, 38], our learned distribution preserves the flexibility of text-guided editing . As shown in Fig. 1 and Fig. 7, we are able to generate diverse in-distribution Gundam figures that follows the style of reference images but with different pose, style, context, using user provided text-guidance at inference time. With a set of learned prompt, we concatenate them with the same text prefix and/or suffix to fit a new distribution at the CLIP text feature space to enable text-guided editing of a prompt distribution. Application includes but not limited to, generating objects of interests in a different background or context, transferring style using text, and controlling the pose, viewpoints, layout, of objects of interests.

Scaling Variance for Diversity Control Once a prompt distribution is learned, we can easily control the diversity of generated images by changing the variance or standard

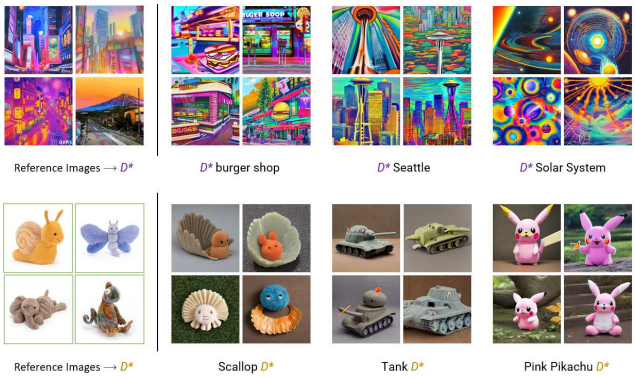

Figure 7. Results on text-editability of our methods. Left column shows samples of reference images, right columns are generated results with corresponding prompts.

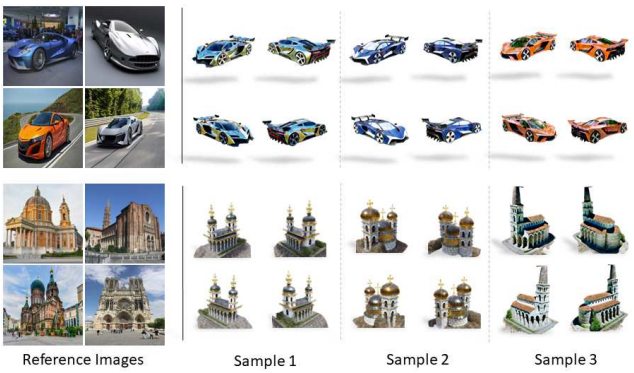

Figure 8. 3D generation results by learning a prompt distribution over the reference images and then inference using MVDream [43] (without extra texts).

Figure 9. 3D generation results by learning a prompt distribution over the reference images and then inference with text-guided editing using MVDream [43].

deviation of the learned distribution. We show an example of the effect of multiplying different scale factors γ to the variance of a learned prompt distribution in Fig. 5. When $\gamma = 0$, the generated images show very similar patterns following some of the reference images. As γ increases, more different layouts emerge, and when we further scale the variance for $\gamma = 2$, the generated images become more diverse with significant randomness.

Composition of Distributions Given multiple prompt distributions in CLIP feature space, we can composite distributions by finding a linearly interpolated distribution between them. This distribution in the CLIP feature space should represent a text with semantic meaning that is a weighted mixture of the given prompt distributions, thereby showing a mixture of visual attributes in the generated images. We naively use a weighted sum of the distributions to interpolate between distributions:

$$\boldsymbol{\mu}_{\mathbf{c}}^{*} = \sum_{i=1}^{N} \alpha_i \boldsymbol{\mu}_{\mathbf{c}_i}, \quad \boldsymbol{\sigma}_{\mathbf{c}}^{*} = \sum_{i=1}^{N} \sqrt{\alpha_i} \boldsymbol{\sigma}_{\mathbf{c}_i} \tag{7}$$

where $\boldsymbol{\mu}_{\mathbf{c}}^{*}$ and $\boldsymbol{\sigma}_{\mathbf{c}}^{*}$ are mean and standard deviations of the interpolated distribution, and α_i is the weight of i-th prompt distribution with mean and standard deviation $\boldsymbol{\mu}_{\mathbf{c}_i}$ and $\boldsymbol{\sigma}_{\mathbf{c}_i}$ respectively, and $\sum_{i=1}^{N} \alpha_i = 1$ are mixing weight parameters.

We show an example of mixing distributions of Chinese paintings and Van Gogh paintings in Fig. 6. From the left column to right, we adjust the mixing ratio to increase the weight of Van Gogh and decrease the weight of Chinese painting.

4.4. Applying to Text-to-3D Generation

Our learned distribution can be flexibly applied to other text-driven tasks, as long as the generation pipeline uses the same pretrained text encoder as the text feature extractor. In this section, we highlight and demonstrate the flexibility of our method by using a prompt distribution trained on T2I diffusion for text-to-3D task. We use MVDream [43], a state-of-the-art text-to-3D model that train a NeRF [28] and render a 3D asset following a text prompt, which in our case is a prompt sampled from prompt distribution. As shown in Fig. 1 and Fig. 8, although MVDream incorporates some extra prior in its modified multi-view diffusion model that leads to reduced diversity, our prompt distribution can still generate 3D assets with significant variation in design details. Moreover, as shown in Fig. 9, the pipeline possesses text-guided editing capabilities akin to those of DreamBooth3D [34], yet it can generate instances that exhibit more diverse appearances.

4.5. Applying to Synthetic Dataset Generation

Our proposed method can also be effectively used in generating synthetic image classification datasets. By giving

Training Dataset	IN [46]		IN-V2 [36]		IN-Sketch [49]		IN-R [13]		IN-A [14]	
	Top-1	Top5	Top-1	Top5	Top-1	Top5	Top-1	Top5	Top-1	Top5
Real	88.0	96.7	85.1	94.9	45.1	63.9	66.1	85.2	26.7	65.8
Class Names	45.5	70.0	46.2	72.5	24.1	43.3	53.6	75.8	8.1	38.8
CLIP Prompts [33]	45.6	69.2	46.1	69.6	**36.2**	**60.1**	58.8	81.1	12.2	45.7
ImageNet-SD [41]	55.4	77.5	55.8	77.5	29.4	49.0	**59.8**	**80.0**	**15.9**	49.4
DreamDistribution (Ours)	**64.3**	**84.0**	**61.7**	**81.6**	25.2	45.8	53.0	74.8	15.7	**50.4**

Table 2. Classification accuracy on different real test sets after training a classifier on synthetic ImageNet (IN) generated by a given method. When training on images from our method, the resulting classifier performs better on the respective test sets, indicating that the images synthesized by our method allowed the classifier to learn those object categories better.

several dozens to hundreds of images that correspond to a class in a classification dataset, our method can capture and encode distributions of the dataset images into the learnable prompt distributions, and thereby generate diverse training images with similar distribution as the training set.

We generate "synthetic copy" [8–10, 41] of ImageNet [39] via DreamDistribution using Stable Diffusion version 2.1 with default hyperparameters. Due to the large size of ImageNet-1K, we follow previous works [41] to mainly experiment on **ImageNet-100** [46], a 100-class subset. For each class, we generate 2,000 synthetic images and use CLIP [33] to select top 1,300 images with highest cosine similarity to the embedding vector of the corresponding class name, resulting the same total number of images as real ImageNet training set. We also compare with four baselines: **Real** uses the real ImageNet training set, **Class Names** and **CLIP Prompts** generate images by feeding Stable Diffusion class name of each class or 80 diverse text prompts from CLIP. [1] **ImageNet-SD** [41] generates images using prompts in the form of "c, h_c inside b", where c represents the class name, h_c represents the hypernym (WordNet parent class name) of the class, and b is a random background description from the class names from Places365 dataset [57].

We train a ResNet-50 [12] classifier on *synthetic images only* for 300 epochs using 0.2 alpha for mixup augmentation [54] and auto augment policy v0 via `timm` [51].

To analyze generalizability, we also evaluate the trained model on validation set of ImageNet variants including **ImageNetV2** [36], **ImageNet-Sketch** [49], **ImageNet-R** [13], and **ImageNet-A** [14]. Top-1 and top-5 accuracy is reported in Tab. 2. In all settings, the classifier is exclusively exposed to synthetic images, but images generated using our method shows the highest classification accuracy on ImageNet validation set. This is because DreamDistribution can generate a diverse set of high-quality images following training set distribution, while other prompt engineering methods cannot follow the real image distribution and tend to show limited diversity within classes, therefore resulting in performance degradation. We also achieve the best results on ImageNet-V2 and comparable results on ImageNet-A. For the Sketch and Rendition variant, in contrast to our method, CLIP Prompts and ImageNet-SD offer specific prompts to generate images of other domains, which may account for our comparatively lower performance.

[1] *e.g.* "a photo of c", "a drawing of c", where c is the class name.

5. Limitations

Despite the ability of our method to generate diverse novel in-distribution images, it does have certain limitations. Specifically, our method may struggle to capture visual features when the number of training images is limited and very diverse. Moreover, the Gaussian distribution assumption could be overly restrictive depending on the training images and the text encoder's latent space. In the future, we hope to find a more robust approach to learning distributions from a few, highly diverse images, with more accurate assumptions and resilient distribution forms.

6. Conclusion

We introduced DreamDistribution, a distribution based prompt tuning method for personalizing T2I diffusion models to generate diverse in-distribution images following a small set of reference images. The key idea of our methods lies in modeling the commonalities and variations of visual attributes using a prompt distribution at text feature space. We show a variety of experiments and application that is enabled by our method.

7. Acknowledgment

This work was supported by the National Science Foundation (award 2318101), C-BRIC (one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA), the Army Research Office (W911NF2020053), and Amazon ML Fellowship. The authors affirm that the views expressed herein are solely their own, and do not represent the views of the United States government or any agency thereof.

References

[1] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023. 2

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 3

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 6

[4] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 3

[5] Christopher L Edwards, Perrine M Ruby, Josie E Malinowski, Paul D Bennett, and Mark T Blagrove. Dreaming and insight. *Frontiers in Psychology*, 4:979, 2013. 2

[6] Sigmund Freud. *Die traumdeutung*. F. Deuticke, 1921. 2

[7] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022. 2, 3, 4, 5, 6, 7

[8] Yunhao Ge, Harkirat Behl, Jiashu Xu, Suriya Gunasekar, Neel Joshi, Yale Song, Xin Wang, Laurent Itti, and Vibhav Vineet. Neural-sim: Learning to generate training data with nerf. In *European Conference on Computer Vision*, pages 477–493. Springer, 2022. 9

[9] Yunhao Ge, Jiashu Xu, Brian Nlong Zhao, Laurent Itti, and Vibhav Vineet. Dall-e for detection: Language-driven context image synthesis for object detection. *arXiv preprint arXiv:2206.09592*, 2022.

[10] Yunhao Ge, Jiashu Xu, Brian Nlong Zhao, Neel Joshi, Laurent Itti, and Vibhav Vineet. Beyond generation: Harnessing text to image models for object detection and segmentation, 2023. 9

[11] Karen Hambardzumyan, Hrant Khachatrian, and Jonathan May. WARP: Word-level Adversarial ReProgramming. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4921–4933, Online, 2021. Association for Computational Linguistics. 4

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 9

[13] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8349, 2021. 9

[14] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15262–15271, 2021. 9

[15] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 6

[16] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 3

[17] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR, 2021. 3

[18] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European Conference on Computer Vision*, pages 709–727. Springer, 2022. 4

[19] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2, 5

[20] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2, 3, 5, 6, 7

[21] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021. 3, 4

[22] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021. 3, 4

[23] Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. Gligen: Open-set grounded text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22511–22521, 2023. 2

[24] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*, 2021. 4

[25] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. Gpt understands, too. *AI Open*, 2023. 4

[26] Yuning Lu, Jianzhuang Liu, Yonggang Zhang, Yajing Liu, and Xinmei Tian. Prompt distribution learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5206–5215, 2022. 3, 4, 5

[27] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11461–11471, 2022. 2

[28] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 8

[29] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6038–6047, 2023. 2

[30] Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable fidelity and diversity metrics for generative models. In *International Conference on Machine Learning*, pages 7176–7185. PMLR, 2020. 6

[31] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 7

[32] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 3

[33] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 3, 4, 6, 9

[34] Amit Raj, Srinivas Kaza, Ben Poole, Michael Niemeyer, Nataniel Ruiz, Ben Mildenhall, Shiran Zada, Kfir Aberman, Michael Rubinstein, Jonathan Barron, et al. Dreambooth3d: Subject-driven text-to-3d generation. *arXiv preprint arXiv:2303.13508*, 2023. 8

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

[36] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, pages 5389–5400. PMLR, 2019. 9

[37] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 2, 3, 4, 5

[38] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. *arXiv preprint arxiv:2208.12242*, 2022. 2, 3, 5, 6, 7

[39] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115 (3):211–252, 2015. 9

[40] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. 3

[41] Mert Bulent Sariyildiz, Karteek Alahari, Diane Larlus, and Yannis Kalantidis. Fake it till you make it: Learning (s) from a synthetic imagenet clone. *arXiv preprint arXiv:2212.08420*, 2022. 9

[42] Jing Shi, Wei Xiong, Zhe Lin, and Hyun Joon Jung. Instantbooth: Personalized text-to-image generation without test-time finetuning. *arXiv preprint arXiv:2304.03411*, 2023. 3

[43] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*, 2023. 8

[44] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015. 3

[45] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 6

[46] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 776–794. Springer, 2020. 9

[47] Gustave E Von Grunebaum and Roger Caillois. *The dream and human societies*. Univ of California Press, 2023. 2

[48] Andrey Voynov, Qinghao Chu, Daniel Cohen-Or, and Kfir Aberman. $p+$: Extended textual conditioning in text-to-image generation. *arXiv preprint arXiv:2303.09522*, 2023. 2, 3

[49] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. *Advances in Neural Information Processing Systems*, 32, 2019. 9

[50] Yuxiang Wei, Yabo Zhang, Zhilong Ji, Jinfeng Bai, Lei Zhang, and Wangmeng Zuo. Elite: Encoding visual concepts into textual embeddings for customized text-to-image generation. *arXiv preprint arXiv:2302.13848*, 2023. 3

[51] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019. 9

[52] Shaoan Xie, Zhifei Zhang, Zhe Lin, Tobias Hinz, and Kun Zhang. Smartbrush: Text and shape guided object inpainting with diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22428–22437, 2023. 2

[53] Zhengyuan Yang, Jianfeng Wang, Zhe Gan, Linjie Li, Kevin Lin, Chenfei Wu, Nan Duan, Zicheng Liu, Ce Liu, Michael Zeng, et al. Reco: Region-controlled text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14246–14255, 2023. 2

[54] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 9

[55] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 2

[56] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 6

[57] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 9

[58] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16816–16825, 2022. 2, 4

[59] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022. 2, 3, 4

DreamDistribution: Prompt Distribution Learning for Text-to-Image Diffusion Models

Supplementary Material

8. More Implementation Details

In all experiments, we use Stable Diffusion 2.1, which is the latest version. We use the default parameters, including 7.5 guidance scale and 50 denoising steps. For all visual results, we generate images in 768×768 resolution, and for synthetic dataset experiments, we generate 256×256 images to save time and resources. We provide a pseudocode for learning a prompt distribution in Algorithm 1.

Algorithm 1 Training prompt distribution

Require: Set of reference images $\mathcal{I} = \{\mathbf{x}_i\}_{i=1}^{N}$
Require: Set of learnable prompts $\mathcal{P}^K = \{\mathcal{P}_i\}_{i=1}^{K}$
Require: Text encoder $\mathcal{E}$, noise predictor $\boldsymbol{\epsilon}_\theta$, hyperparameter λ
Random initialize all learnable embeddings $\mathbf{V}$ in $\mathcal{P}^K$
for image $\mathbf{x} \in \mathcal{I}$ **do**
  Sample time step t
  Sample noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$
  $\mathbf{x}_t \leftarrow \mathbf{x}$ with noise added based on $\boldsymbol{\epsilon}$ and t
  Compute $\boldsymbol{\mu_c}$ of $\mathcal{E}(\mathcal{P}^K)$
  Compute $\boldsymbol{\sigma_c}$ of $\mathcal{E}(\mathcal{P}^K)$
  for $s \in [S]$ **do**
    Sample $\boldsymbol{\omega}_s \sim \mathcal{N}(0, \mathbf{I})$
    $\mathcal{L}_s\left(\mathcal{P}^K\right) = \left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\mathbf{x}_t, \boldsymbol{\mu_c} + \boldsymbol{\omega}_s \boldsymbol{\sigma_c}, t\right)\right\|_2^2$
  end for
  $\mathcal{L}\left(\mathcal{P}^K\right) = \frac{1}{S}\sum_{s=1}^{S} \mathcal{L}_s(\mathcal{P}^K)$
  $\mathcal{L}_{ortho} = \frac{1}{K(K-1)} \sum_{i=1}^{K} \sum_{j=i+1}^{K} |\langle \mathcal{E}(\mathcal{P}_i), \mathcal{E}(\mathcal{P}_j) \rangle|$
  $\mathcal{L} = \mathcal{L}\left(\mathcal{P}^K\right) + \lambda \mathcal{L}_{ortho}$
  Update learnable embeddings $\mathbf{V}$ in $\mathcal{P}^K$ based on $\mathcal{L}$
end for

9. Evaluation Set

We show samples of reference images from our evaluation set in Fig. 11. Each row shows 8 samples reference images from the same set.

10. Additional Result

10.1. Diverse Image Instance Generation

We show additional image generation results of our method in Fig. 12 using the reference images from our evaluation set. Each row is generated using reference images of the corresponding row in Fig. 11.

10.2. Text-guided Editing

We show more results on the ability of our method to generate diverse images with text-guided editing in Fig. 13 using different sets of reference images.

10.3. Scaling Variance for Diversity Control

We show more results on generating images from prompt distribution with scaled standard deviations in Fig. 14

10.4. Composition of Distribution

In Fig. 15 we show more results on composition of two different learned prompt distributions with various weights.

11. Ablation study

We additionally ablate K, the number of prompts in personalized generation. We randomly select 4 sets of reference images from our evaluation set and compute the average performance based on automatic quality and diversity metrics introduced in main Section 4.1. In Fig. 10, we show the effect of K in terms of both generation quality and diversity. We observe a positive correlation between the performance (in terms of both quality and diversity) and the number of prompts. More prompts offer more flexibility for our methods to model a better distribution of prompts, thus enabling the model to encapsulate content better (quality) and adapt to various nuances of the training images (diversity).

12. Synthetic dataset

12.1. More Analysis on Synthetic Dataset

Number of training images We experiment with different number of training images. We use randomly selected 10, 100, 500 images per class, as well as all ImageNet training images to train our learnable prompts and generate same size synthetic dataset. From results shown in Fig. 16 column 1 & 2, we found that using about 100-500 images per class (7%-38% of the real training set) would be enough to reach high classification accuracy on real validation set, while using more data would not further improve accuracy. For validation sets of ImageNet variants, less training data would obtain higher accuracy due to the domain gap between the real training set and different validation sets.

Mixing synthetic data with real training data We also experiment with mixing different sizes of synthetic image data with the real training images. We mix additional 20%,

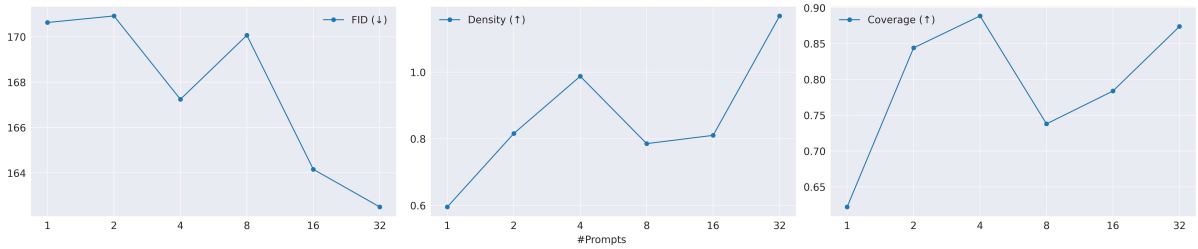

Figure 10. In general, with more prompts, the performance increases in terms of both quality and diversity.

40%, 60%, 80% and 100% synthetic data with real training data, where 100% means the size of the mixed dataset is twice of the size of the ImageNet training set, and the ratio of the number of real images to the number of synthetic images is approximately 1:1. As shown in Fig. 16 column 3 & 4, adding more synthetic data would improve the accuracy on ImageNet validation set. On the validation sets of different domains, however, adding more synthetic data would not show significant improving trends on accuracy.

12.2. Implementation detail

For training on ImageNet dataset, we use the same training hyperparameters except for reducing the number of learnable prompts to 10 per class. We train for 5 epochs for training prompt distribution and 300 epochs for training ResNet-50 using generated or mixed dataset. All results are averaged over 3 runs of training using the generated or mixed dataset. For ImageNet-R and ImageNet-A, we only evaluate on the overlapping classes with ImageNet-100.

12.3. Visual result

We show some generated training images using our method and compare them with generated images using solely class names as text prompts for generation in Fig. 17. Compared to the images generated using class names, our generated images shows more diversity with different real-world contexts.

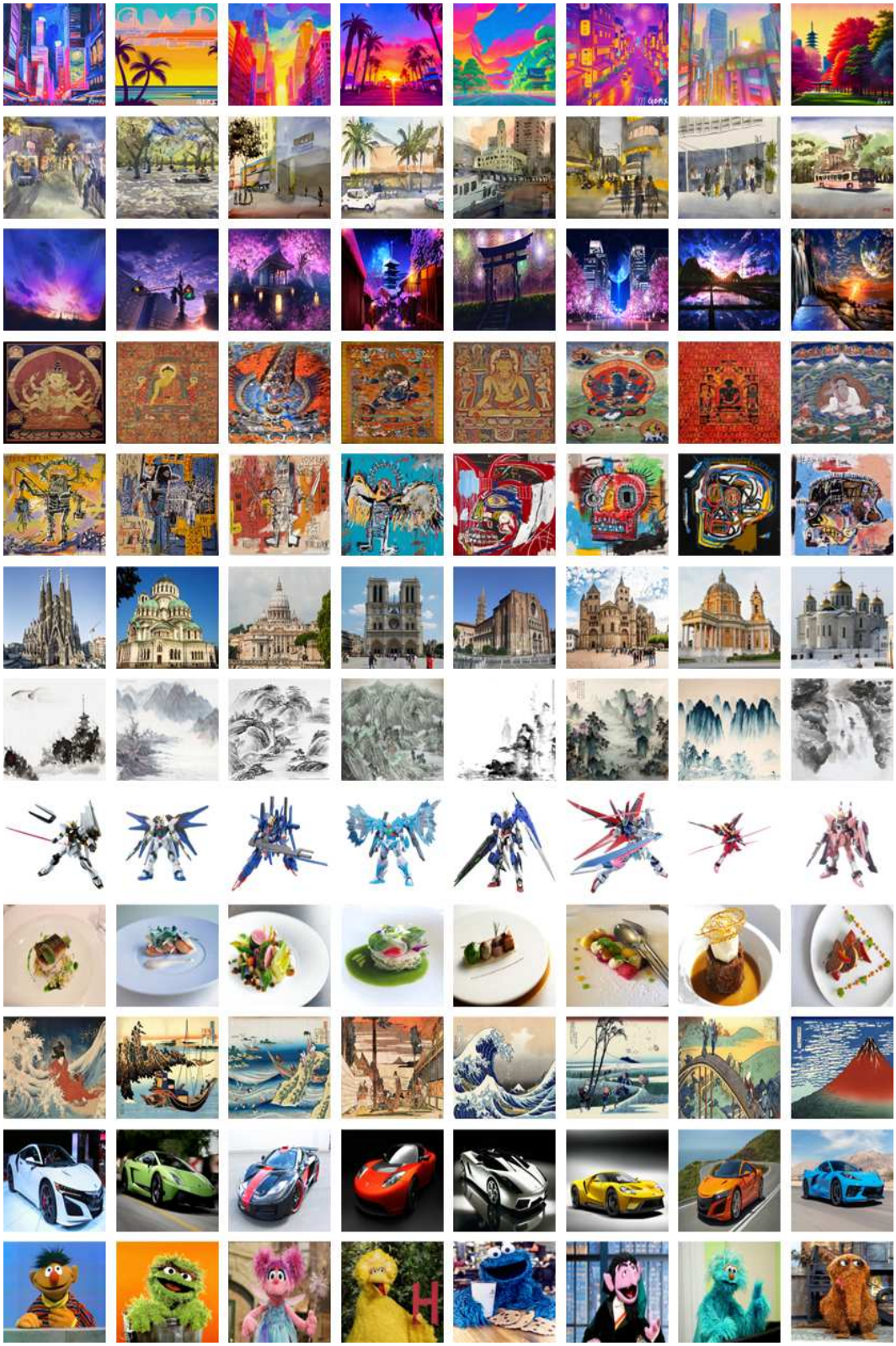

Figure 11. Samples of reference images from our evaluation set

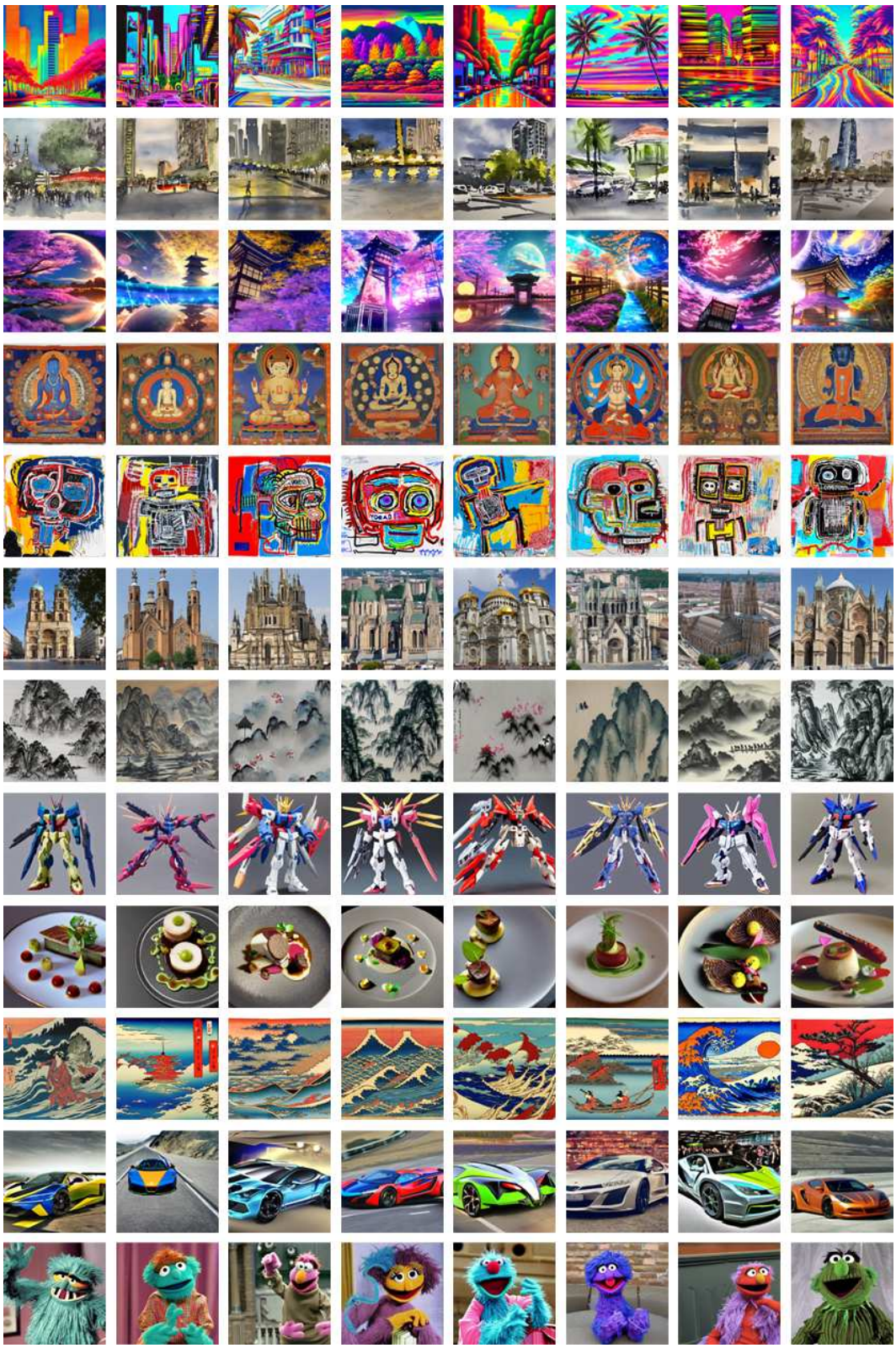

Figure 12. Samples of generated image results using reference images from the evaluation set. Each row is generated using reference images of the corresponding row in Fig. 11.

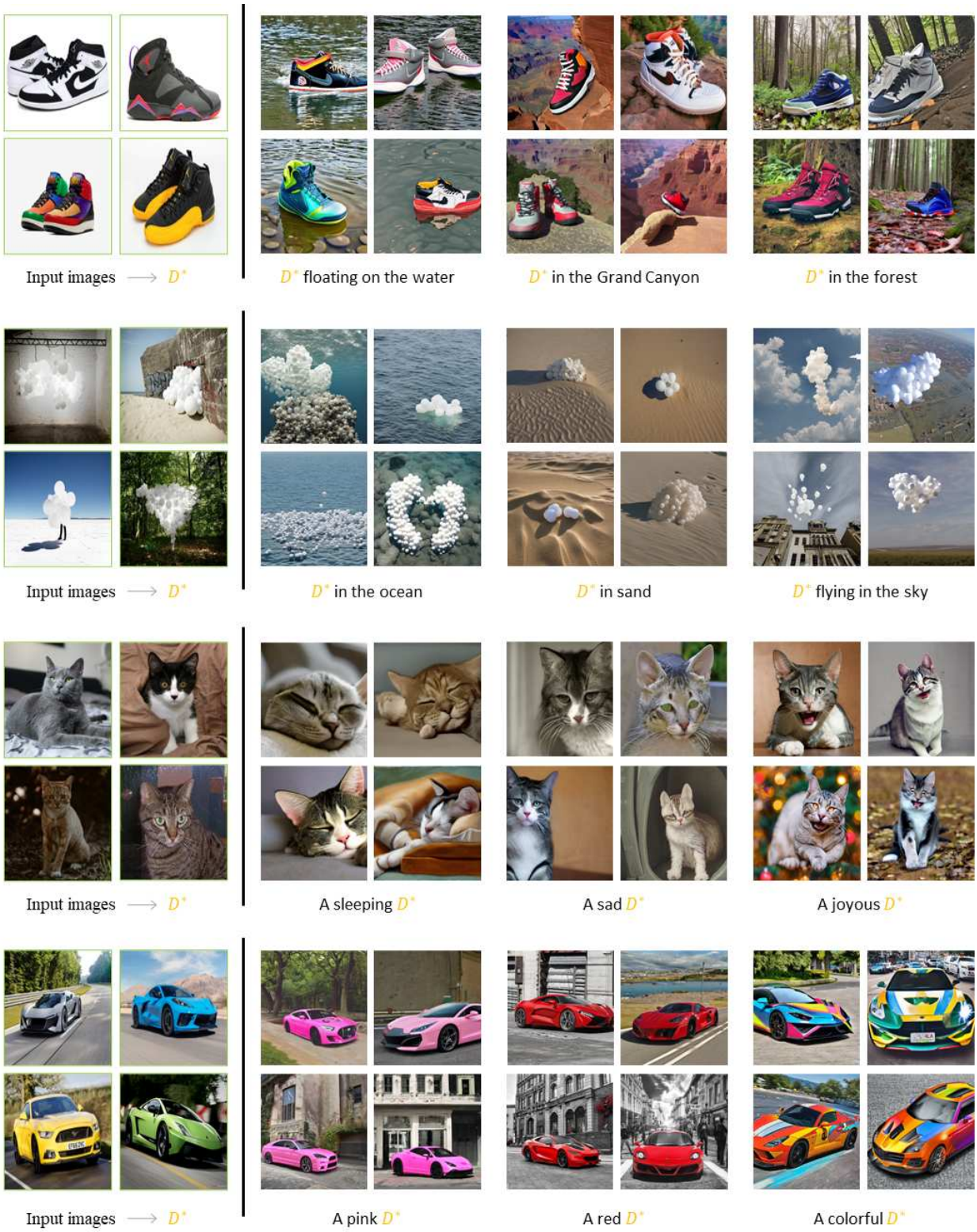

Figure 13. More results on text-editability of our methods. Left column shows samples of reference images, right columns are generated results with corresponding prompts.

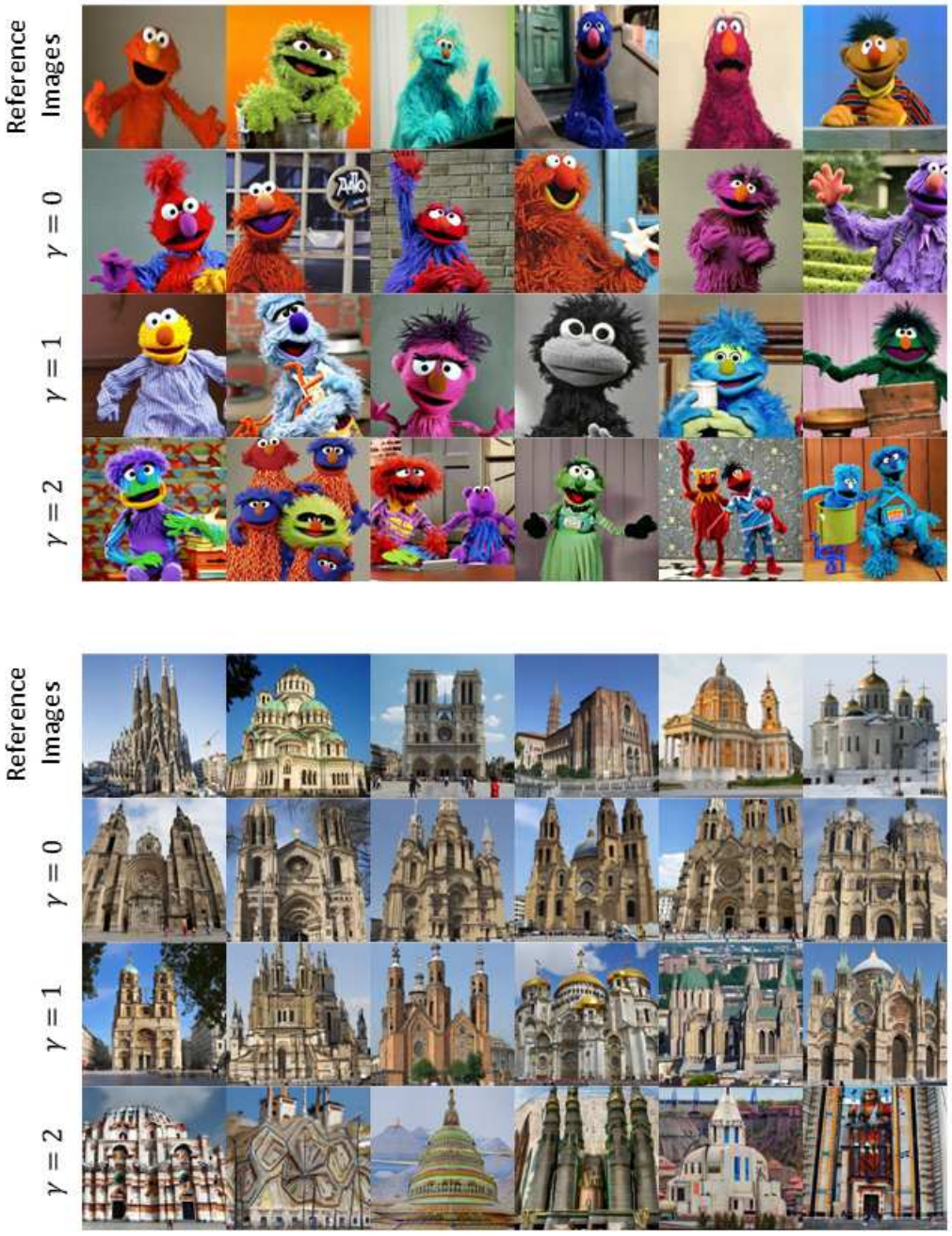

Figure 14. More results on scaling standard deviation of learned prompt distribution.

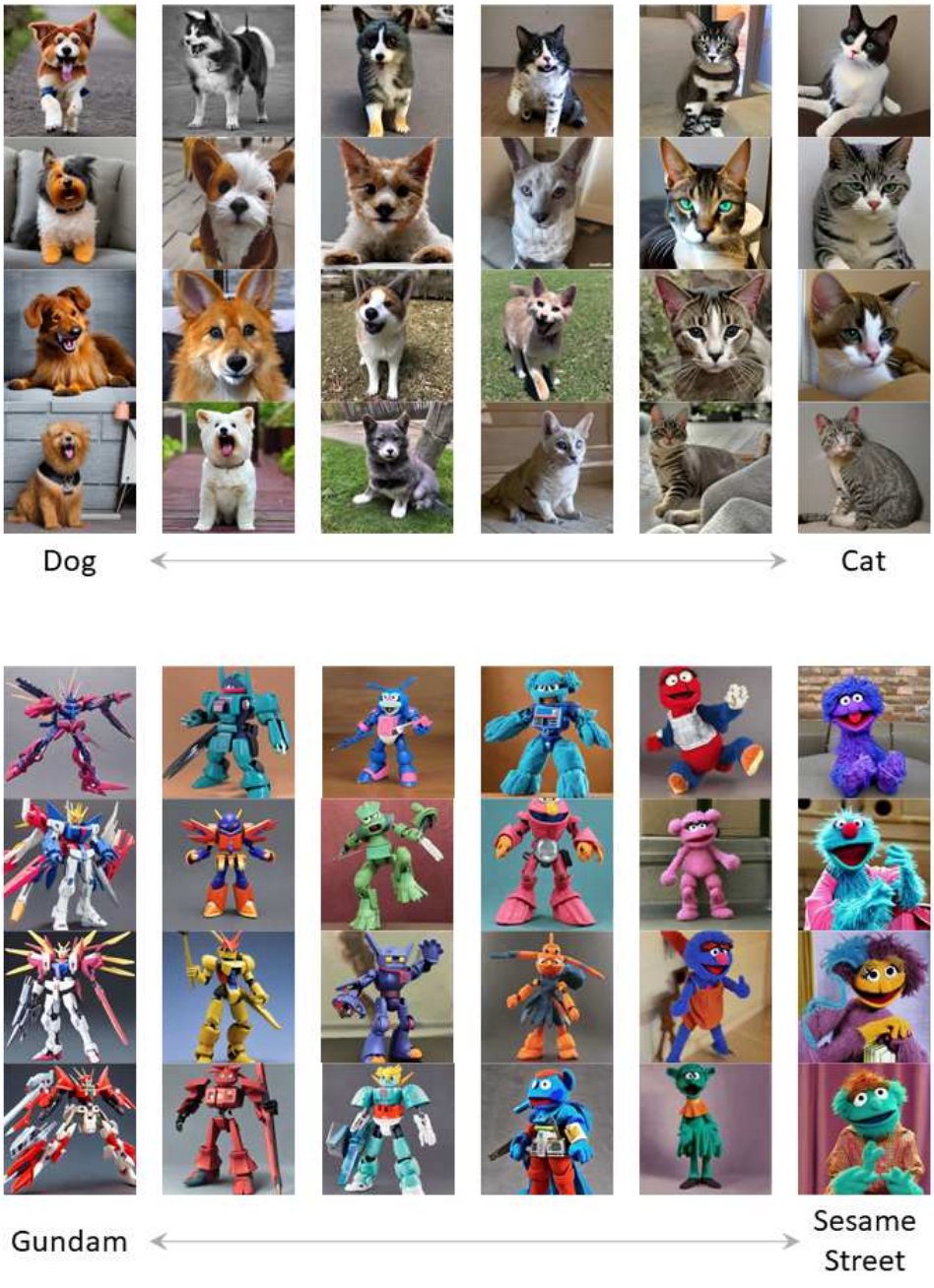

Figure 15. More results on composition of multiple prompt distributions using different weights.

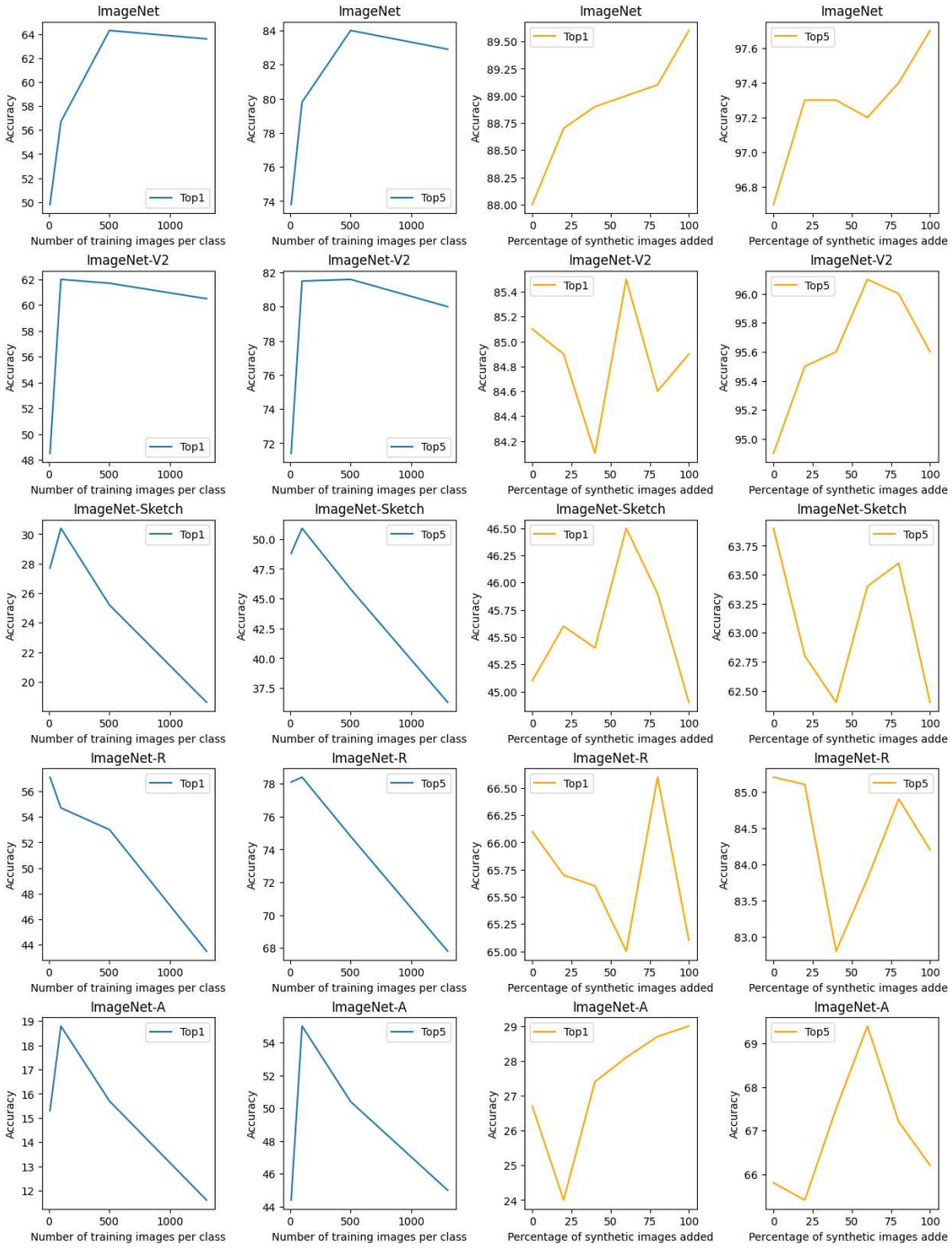

Figure 16. Column 1 & 2: Top-1 and top-5 accuracy on ImageNet validation set versus using different number of training images to train prompt distribution. Column 3 & 4; Top-1 and top-5 accuracy on ImageNet validation set versus percentage of synthetic images added to the real training set.

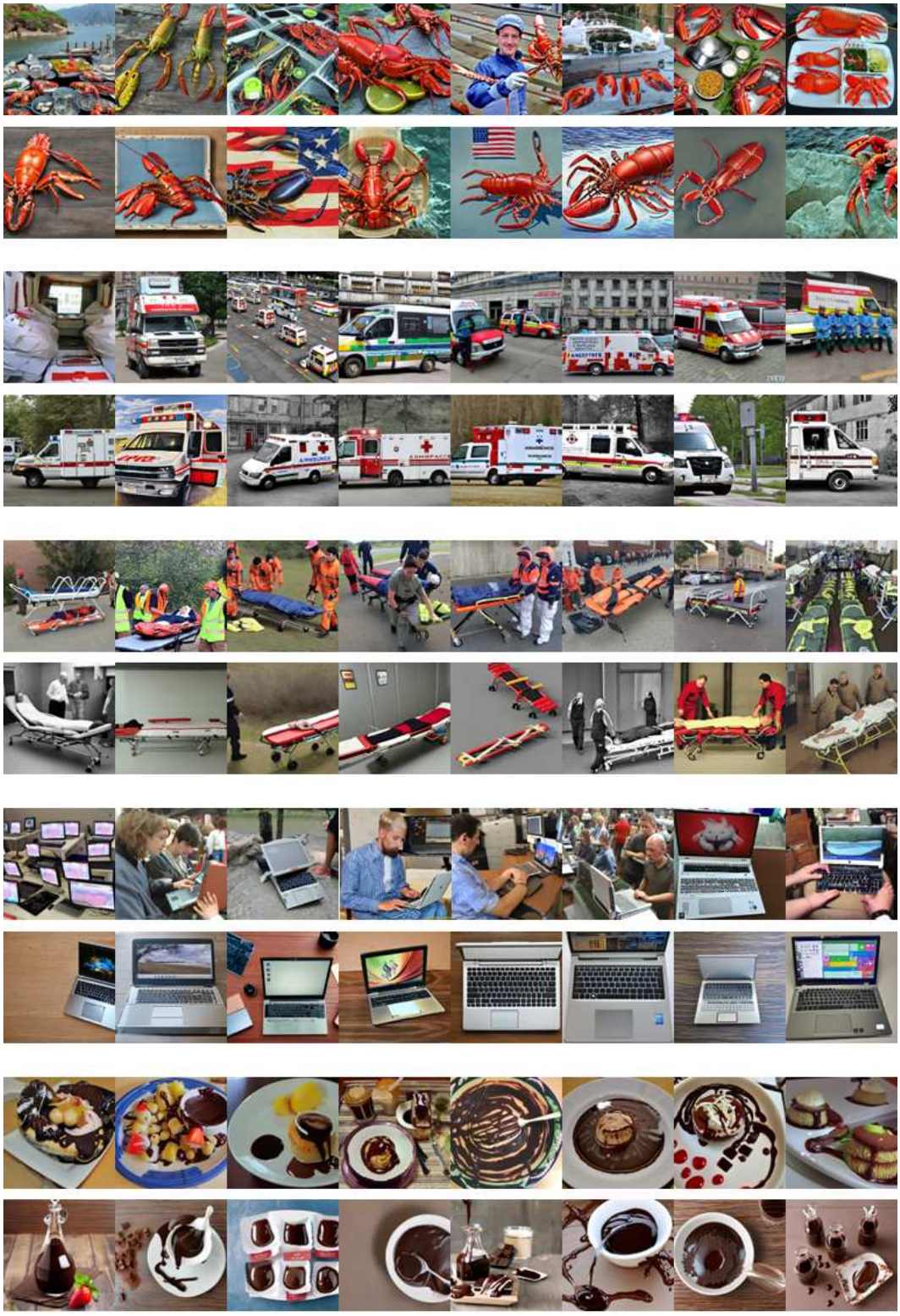

Figure 17. Comparison between training images generated using our method and image generated using solely class names as text prompts. For each group of two rows, the top row shows samples of our generated training images, and the bottom row shows the training images generated using class names as text prompts.