This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

FaceChain-SuDe: Building Derived Class to Inherit Category Attributes for One-shot Subject-Driven Generation

Pengchong Qiao[1,2*] Lei Shang[2*] Chang Liu[3†] Baigui Sun[2] Xiangyang Ji[3] Jie Chen[1,4]
[1]School of Electronic and Computer Engineering, Peking University, Shenzhen, China [2]Alibaba Group, Hangzhou, China
[3]Department of Automation and BNRist, Tsinghua University, Beijing, China [4] Peng Cheng Laboratory, Shenzhen, China
pcqiao@stu.pku.edu.cn {sl172005, baigui.sbg}@alibaba-inc.com
{liuchang2022, xyji}@tsinghua.edu.cn chenj@pcl.ac.cn

Abstract

Recently, subject-driven generation has garnered significant interest due to its ability to personalize text-to-image generation. Typical works focus on learning the new subject's private attributes. However, an important fact has not been taken seriously that a subject is not an isolated new concept but should be a specialization of a certain category in the pre-trained model. This results in the subject failing to comprehensively inherit the attributes in its category, causing poor attribute-related generations. In this paper, motivated by object-oriented programming, we model the subject as a derived class whose base class is its semantic category. This modeling enables the subject to inherit public attributes from its category while learning its private attributes from the user-provided example. Specifically, we propose a plug-and-play method, Subject-Derived regularization (SuDe). It constructs the base-derived class modeling by constraining the subject-driven generated images to semantically belong to the subject's category. Extensive experiments under three baselines and two backbones on various subjects show that our SuDe enables imaginative attribute-related generations while maintaining subject fidelity. For the codes, please refer to FaceChain.

1. Introduction

Recently, with the fast development of text-to-image diffusion models [21, 26, 29, 32], people can easily use text prompts to generate high-quality, photorealistic, and imaginative images. This gives people an outlook on AI painting in various fields such as game design, film shooting, etc.

Among them, subject-driven generation is an interesting application that aims at customizing generation for a specific subject. For example, something that interests you like pets, pendants, anime characters, etc. These subjects are

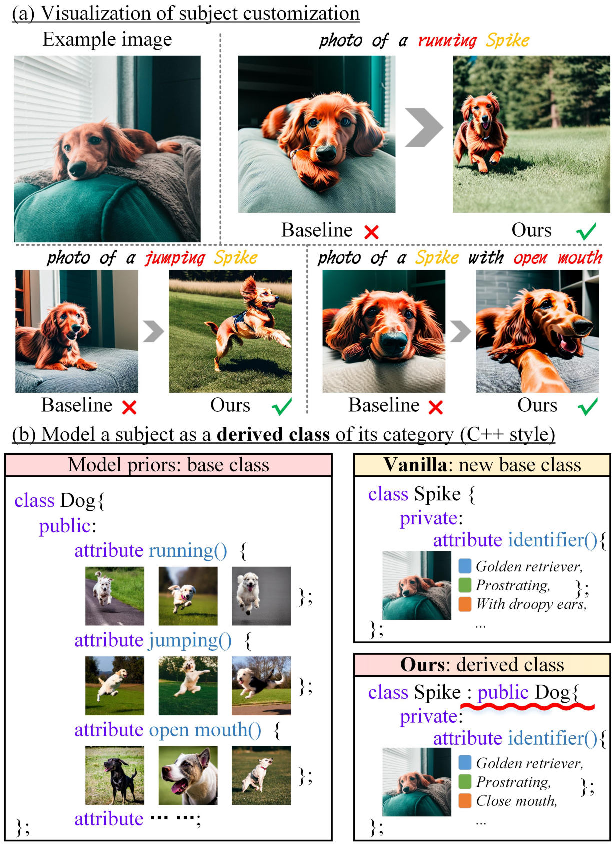

Figure 1. (a) The subject is a golden retriever 'Spike', and the baseline is DreamBooth [30]. The baseline's failure is because the example image cannot provide the needed attributes like 'running'. Our method tackles it by inheriting these attributes from the 'Dog' category to 'Spike'. (b) We build 'Spike' as a derived class of the base class 'Dog'. In this paper, we record the general properties of the base class from the pre-trained model as *public attributes*, while subject-specific properties as *private attributes*. The part marked with a red wavy line is the 'Inherit' syntax in C++ [37].

*Equal contribution.
†Corresponding author.

specific to each natural person (user) and do not exist in the large-scale training of pre-trained diffusion models. To achieve this application, users need to provide a few example images to bind the subject with a special token ($\{S^*\}$), which could then be used to guide further customizations.

Existing methods can be classified into two types: offline ones and online ones. The former [31, 41] employs an offline trained encoder to directly encode the subject examples into text embedding, achieving high testing efficiency. But the training of their encoders depends on an additional large-scale image dataset, and even the pixel-level annotations are also needed for better performances [41]. The latter [12, 13, 17, 30] adopts a test-time fine-tuning strategy to obtain the text embedding representing a specific subject. Despite sacrificing testing efficiency, this kind of method eliminates reliance on additional data and is more convenient for application deployment. Due to its flexibility, we focus on improving the online methods in this paper.

In deployment, the most user-friendly manner only requires users to upload one example image, called *one-shot* subject-driven generation. However, we find existing methods do not always perform satisfactorily in this challenging but valuable scene, especially for attribute-related prompts. As shown in Fig. 1 (a), the baseline method fails to make the 'Spike' run, jump, or open its mouth, which are natural attributes of dogs. Interestingly, the pre-trained model can generate these attributes for non-customized 'Dogs' [21, 26, 29, 32]. From this, we infer that the failure in Fig. 1 is because the single example image is not enough to provide the attributes required for customizing the subject, and these attributes cannot be automatically completed by the pre-trained model. With the above considerations, we propose to tackle this problem by making the subject ('Spike') explicitly inherit these attributes from its semantic category ('Dog'). Specifically, motivated by the definitions in Object-Oriented Programming (OOP), we model the subject as a derived class of its category. As shown in Fig. 1 (b), the semantic category ('Dog') is viewed as a base class, containing public attributes provided by the pre-trained model. The subject ('Spike') is modeled as a derived class of 'Dog' to inherit its public attributes while learning private attributes from the user-provided example. From the visualization in Fig. 1 (a), our modeling significantly improves the baseline for attribute-related generations.

From the perspective of human understanding, the above modeling, i.e., subject ('Spike') is a derived class of its category ('Dog'), is a natural fact. But it is unnatural for the generative model (e.g., diffusion model) since it has no prior concept of the subject 'Spike'. Therefore, to achieve this modeling, we propose a **Subject Derivation regularization (SuDe)** to constrain that the generations of a subject could be classified into its corresponding semantic category. Using the example above, generated images of 'photo of a Spike' should have a high probability of belonging to 'photo of a Dog'. This regularization cannot be easily realized by adding a classifier since its semantics may misalign with that in the pre-trained diffusion model. Thus, we propose to explicitly reveal the implicit classifier in the diffusion model to regularize the above classification.

Our SuDe is a plug-and-play method that can combine with existing subject-driven methods conveniently. We evaluate this on three well-designed baselines, DreamBooth [30], Custom Diffusion [17], and ViCo [13]. Results show that our method can significantly improve attributes-related generations while maintaining subject fidelity.

Our main contributions are as follows:

- We provide a new perspective for subject-driven generation, that is, modeling a subject as a derived class of its semantic category, the base class.
- We propose a subject-derived regularization (SuDe) to build the base-derived class relationship between a subject and its category with the implicit diffusion classifier.
- Our SuDe can be conveniently combined with existing baselines and significantly improve attributes-related generations while keeping fidelity in a plug-and-play manner.

2. Related Work

2.1. Object-Oriented Programming

Object-Oriented Programming (OOP) is a programming paradigm with the concept of objects [23, 28, 40], including four important definitions: class, attribute, derivation, and inheritance. A *class* is a template for creating objects containing some *attributes*, which include public and private ones. The former can be accessed outside the class, while the latter cannot. *Derivation* is to define a new class that belongs to an existing class, e.g., a new 'Golden Retriever' class could be derived from the 'Dog' class, where the former is called derived class and the latter is called base class. *Inheritance* means that the derived class should inherit some attributes of the base class, e.g., 'Golden Retriever' should inherit attributes like 'running' and 'jumping' from 'Dog'.

In this paper, we model the subject-driven generation as class derivation, where the subject is a derived class and its semantic category is the corresponding base class. To adapt to this task, we use *public attributes* to represent general properties like 'running', and *private attributes* to represent specific properties like the subject identity. The base class (category) contains public attributes provided by the pre-trained diffusion model and the derived class (subject) learns private attributes from the example image while inheriting its category's public attributes.

2.2. Text-to-image generation

Text-to-image generation aims to generate high-quality images with the guidance of the input text, which is realized by combining generative models with pre-trained vision-language models, e.g., CLIP [24]. From the perspective of generators, they can be roughly categorized into three groups: GAN-based, VAE-based, and Diffusion-based methods. The GAN-based methods [8, 27, 38, 42, 44] employ the Generative Adversarial Network as the generator and perform well on structural images like human faces. But they struggle in complex scenes with varied components. The VAE-based methods [5, 9, 11, 25] generate images with Variational Auto-encoder, which can synthesize diverse images but sometimes cannot match the texts well. Recently, Diffusion-based methods [3, 10, 21, 26, 29, 32] obtain SOTA performances and can generate photo-realistic images according to the text prompts. In this paper, we focus on deploying the pre-trained text-to-image diffusion models into the application of subject-customization.

2.3. Subject-driven generation

Given a specific subject, subject-driven generation aims to generate new images of this subject with text guidance. Pioneer works can be divided into two types according to training strategies, the offline and the online ones. Offline methods [6, 7, 31, 41] directly encode the example image of the subject into text embeddings, for which they need to train an additional encoder. Though high testing efficiency, they are of high cost since a large-scale dataset is needed for offline training. Online methods [12, 13, 17, 30, 39] learn a new subject in a test-time tuning manner. They represent the subject with a specific token '$\{\text{S}^*\}$' by fine-tuning the pre-trained model in several epochs. Despite sacrificing some test efficiency, they don't need additional datasets and networks. But for the most user-friendly one-shot scene, these methods cannot customize attribute-related generations well. To this end, we propose to build the subject as a derived class of its category to inherit public attributes while learning private attributes. Some previous works [17, 30] partly consider this problem by prompt engineering, but not as satisfactory as our SuDe, as discussed in the appendix.

3. Method

3.1. Preliminaries

3.1.1 Text-to-image diffusion models

Diffusion models [14, 34] approximate real data distribution by restoring images from Gaussian noise. They use a forward process gradually adding noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ on the clear image (or its latent code) $\boldsymbol{x}_0$ to obtain a series of noisy variables $\boldsymbol{x}_1$ to $\boldsymbol{x}_T$, where T usually equals 1000, as:

$$\boldsymbol{x}_t = \sqrt{\alpha_t}\boldsymbol{x}_0 + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}, \tag{1}$$

where α_t is a t-related variable that controls the noise schedule. In text-to-image generation, a generated image is guided by a text description $\boldsymbol{P}$. Given a noisy variable $\boldsymbol{x}_t$ at step t, the model is trained to denoise the $\boldsymbol{x}_t$ gradually as:

$$\mathbb{E}_{\boldsymbol{x},\boldsymbol{c},\boldsymbol{\epsilon},t}[w_t||\boldsymbol{x}_{t-1} - x_\theta(\boldsymbol{x}_t, \boldsymbol{c}, t)||^2], \tag{2}$$

where x_θ is the model prediction, w_t is the loss weight at step t, $\boldsymbol{c} = \Gamma(\boldsymbol{P})$ is the embedding of text prompt, and the $\Gamma(\cdot)$ is a pre-trained text encoder, such as BERT [16]. In our experiments, we use Stable Diffusion [2] built on LDM [29] with the CLIP [24] text encoder as our backbone model.

3.1.2 Subject-driven finetuning

Overview: The core of the subject-driven generation is to implant the new concept of a subject into the pre-trained diffusion model. Existing works [12, 13, 17, 30, 43] realize this via finetuning partial or all parameters of the diffusion model, or text embeddings, or adapters, by:

$$\mathcal{L}_{sub} = ||\boldsymbol{x}_{t-1} - x_\theta(\boldsymbol{x}_t, \boldsymbol{c}_{sub}, t)||^2, \tag{3}$$

where the $\boldsymbol{x}_{t-1}$ here is the noised user-provided example at step $t-1$, $\boldsymbol{c}_{sub}$ is the embedding of subject prompt (e.g., 'photo of a $\{\text{S}^*\}$'). The '$\{\text{S}^*\}$' represents the subject token.

Motivation: With Eq. 3 above, existing methods can learn the specific attributes of a subject. However, the attributes in the user-provided single example are not enough for imaginative customizations. Existing methods haven't made designs to address this issue, only relying on the pre-trained diffusion model to fill in the missing attributes automatically. But we find this is not satisfactory enough, e.g., in Fig. 1, baselines fail to customize the subject 'Spike' dog to 'running' and 'jumping'. To this end, we propose to model a subject as a derived class of its semantic category, the base class. This helps the subject inherit the public attributes of its category while learning its private attributes and thus improves attribute-related generation while keeping subject fidelity. Specifically, as shown in Fig. 2 (a), the private attributes are captured by reconstructing the subject example. And the public attributes are inherited via encouraging the subject prompt ($\{S^*\}$) guided $\boldsymbol{x}_{t-1}$ to semantically belong to its category (e.g., 'Dog'), as Fig. 2 (b).

3.2. Subject Derivation Regularization

Derived class is a definition in object-oriented programming, not a proposition. Hence there is no sufficient condition that can be directly used to constrain a subject to be a derived class of its category. However, according to the definition of derivation, there is naturally a necessary condition: a derived class should be a subclass of its base class. We find that constraining this necessary condition is very effective for helping a subject to inherit the attributes of its category. Specifically, we regularize the subject-driven generated images to belong to the subject's category as:

$$\mathcal{L}_{sude} = -\log[p(\boldsymbol{c}_{cate}|x_\theta(\boldsymbol{x}_t, \boldsymbol{c}_{sub}, t))], \tag{4}$$

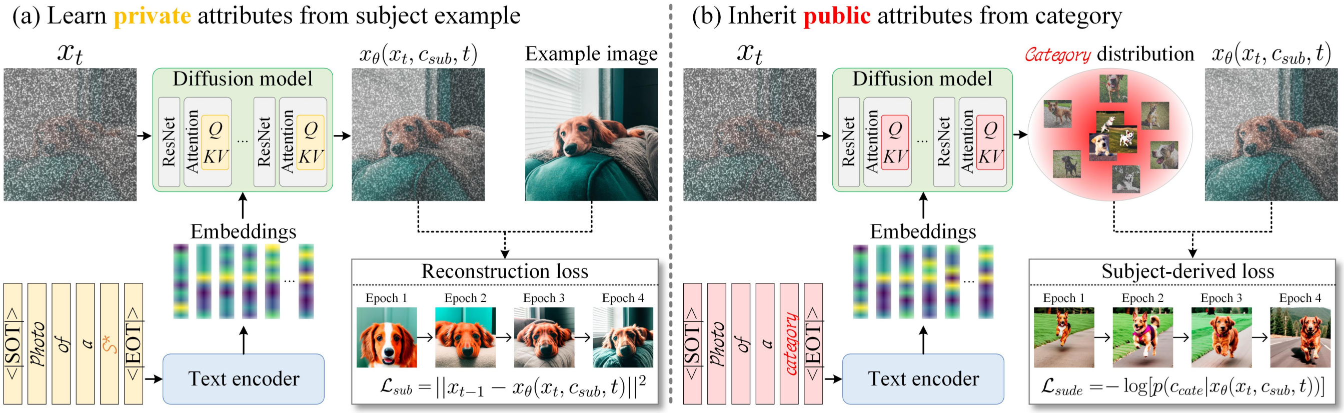

Figure 2. **The pipeline of SuDe.** (a) Learn private attributes by reconstructing the subject example with the $\mathcal{L}_{sub}$ in Eq. 3. (b) Inherit public attributes by constraining the subject-driven $\boldsymbol{x}_{t-1}$ semantically belongs to its category (e.g., dog), with the $\mathcal{L}_{sude}$ in Eq. 4.

where $\boldsymbol{c}_{cate}$ and $\boldsymbol{c}_{sub}$ are conditions of category and subject. The Eq. 4 builds a subject as a derived class well for two reasons: (1) The attributes of a category are reflected in its embedding $\boldsymbol{c}_{cate}$, most of which are public ones that should be inherited. This is because the embedding is obtained by a pre-trained large language model (LLM) [16], which mainly involves general attributes in its training. (2) As analyzed in Sec. 4, optimizing $\mathcal{L}_{sude}$ combined with the Eq. 3 is equivalent to increasing $p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{c}_{sub}, \boldsymbol{c}_{cate})$, which means generating a sample with the conditions of both $\boldsymbol{c}_{sub}$ (private attributes) and $\boldsymbol{c}_{cate}$ (public attributes). Though the form is simple, Eq. 4 cannot be directly optimized. In the following, we describe how to compute it in Sec. 3.2.1, and a necessary strategy to prevent training crashes in Sec. 3.2.2.

3.2.1 Subject Derivation Loss

The probability in Eq. 4 cannot be easily obtained by an additional classifier since its semantics may misalign with that in the pre-trained diffusion model. To ensure semantics alignment, we propose to reveal the implicit classifier in the diffusion model itself. With the Bayes' theorem [15]:

$$p(\boldsymbol{c}_{cate}|x_\theta(\boldsymbol{x}_t, \boldsymbol{c}_{sub}, t)) = C_t \cdot \frac{p(x_\theta(\boldsymbol{x}_t, \boldsymbol{c}_{sub}, t)|\boldsymbol{x}_t, \boldsymbol{c}_{cate})}{p(x_\theta(\boldsymbol{x}_t, \boldsymbol{c}_{sub}, t)|\boldsymbol{x}_t)}, \tag{5}$$

where the $C_t = p(\boldsymbol{c}_{cate}|\boldsymbol{x}_t)$ is unrelated to $t-1$, thus can be ignored in backpropagation. In the Stable Diffusion [2], predictions of adjacent steps (i.e., $t-1$ and t) are designed as a conditional Gaussian distribution:

$$\begin{aligned} &p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{c}) \sim \mathcal{N}(\boldsymbol{x}_{t-1}; x_\theta(\boldsymbol{x}_t, \boldsymbol{c}, t), \sigma_t^2\mathbf{I}) \\ &\propto exp(-||\boldsymbol{x}_{t-1} - x_\theta(\boldsymbol{x}_t, \boldsymbol{c}, t)||^2/2\boldsymbol{\sigma}_t^2), \end{aligned} \tag{6}$$

where the mean value is the prediction at step t and the standard deviation is a function of t. From Eq. 5 and 6, we can convert Eq. 4 into a computable form:

$$\begin{aligned} \mathcal{L}_{sude} = \frac{1}{2\boldsymbol{\sigma}_t^2}[&||x_\theta(\boldsymbol{x}_t, \boldsymbol{c}_{sub}, t) - x_{\bar{\theta}}(\boldsymbol{x}_t, \boldsymbol{c}_{cate}, t)||^2 \\ &- ||x_\theta(\boldsymbol{x}_t, \boldsymbol{c}_{sub}, t) - x_{\bar{\theta}}(\boldsymbol{x}_t, t)||^2], \end{aligned} \tag{7}$$

where the $x_{\bar{\theta}}(\boldsymbol{x}_t, \boldsymbol{c}_{cate}, t)$ is the prediction conditioned on $\boldsymbol{c}_{cate}$, the $x_{\bar{\theta}}(\boldsymbol{x}_t, t)$ is the unconditioned prediction. The $\bar{\theta}$ means detached in training, indicating that only the $x_\theta(\boldsymbol{x}_t, \boldsymbol{c}_{sub}, t)$ is gradient passable, and the $x_{\bar{\theta}}(\boldsymbol{x}_t, \boldsymbol{c}_{cate}, t)$ and $x_{\bar{\theta}}(\boldsymbol{x}_t, t)$ are gradient truncated. This is because they are priors in the pre-trained model that we want to reserve.

3.2.2 Loss Truncation

Optimizing Eq. 4 will leads the $p(\boldsymbol{c}_{cate}|x_\theta(\boldsymbol{x}_t, \boldsymbol{c}_{sub}, t))$ to increase until close to 1. However, this term represents the classification probability of a noisy image at step $t-1$. It should not be close to 1 due to the influence of noise. Therefore, we propose to provide a threshold to truncate $\mathcal{L}_{sude}$. Specifically, for generations conditioned on $\boldsymbol{c}_{cate}$, their probability of belonging to $\boldsymbol{c}_{cate}$ can be used as a reference. It represents the proper classification probability of noisy images at step $t-1$. Hence, we use the negative log-likelihood of this probability as the threshold τ, which can be computed by replacing the $\boldsymbol{c}_{sub}$ with $\boldsymbol{c}_{cate}$ in Eq. 7:

$$\begin{aligned} \tau_t &= -\log[p(\boldsymbol{c}_{cate}|x_\theta(\boldsymbol{x}_t, \boldsymbol{c}_{cate}, t))] \\ &= -\frac{1}{2\boldsymbol{\sigma}_t^2}||x_{\bar{\theta}}(\boldsymbol{x}_t, \boldsymbol{c}_{cate}, t) - x_{\bar{\theta}}(\boldsymbol{x}_t, t)||^2. \end{aligned} \tag{8}$$

The Eq. 8 represents the lower bound of $\mathcal{L}_{sude}$ at step t. When the loss value is less than or equal to τ_t, optimization should stop. Thus, we truncate $\mathcal{L}_{sude}$ as:

$$\mathcal{L}_{sude} = \lambda_\tau \cdot \mathcal{L}_{sude}, \quad \lambda_\tau = \begin{cases} 0, & \mathcal{L}_{sude} \leq \tau_t \\ 1, & else. \end{cases} \tag{9}$$

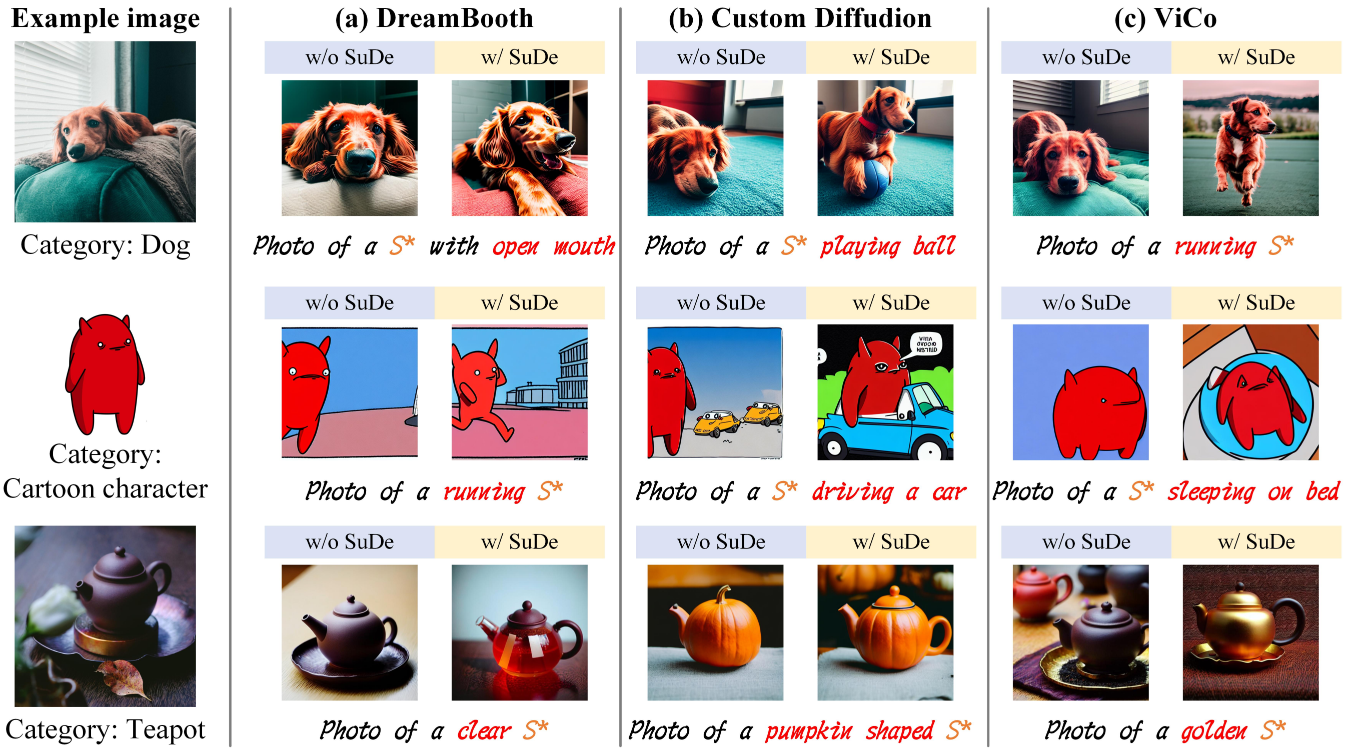

Figure 3. (a), (b), and (c) are generated images using DreamBooth [30], Custom Diffusion [17], and ViCo [13] as the baselines, respectively. Results are obtained using the DDIM [36] sampler with 100 steps. In prompts, we mark the subject token in orange and attributes in red.

In practice, this truncation is important for maintaining training stability. Details are provided in Sec. 5.4.2.

3.3. Overall Optimization Objective

Our method only introduces a new loss function $\mathcal{L}_{sude}$, thus it can be conveniently implanted into existing pipelines in a plug-and-play manner as:

$$\mathcal{L} = \mathbb{E}_{\boldsymbol{x},\boldsymbol{c},\boldsymbol{\epsilon},t}[\mathcal{L}_{sub} + w_s\mathcal{L}_{sude} + w_r\mathcal{L}_{reg}], \tag{10}$$

where $\mathcal{L}_{sub}$ is the reconstruction loss to learn the subject's private attributes as described in Eq. 3. The $\mathcal{L}_{reg}$ is a regularization loss usually used to prevent the model from overfitting to the subject example. Commonly, it is not relevant to $\boldsymbol{c}_{sub}$ and has flexible definitions [13, 30] in various baselines. Using the DreamBooth as an example, we have discussed the difference between $\mathcal{L}_{reg}$ and our $\mathcal{L}_{sude}$ in Sec. 5.4.4. The w_s and w_r are used to control loss weights. In practice, we keep the $\mathcal{L}_{sub}$, $\mathcal{L}_{reg}$ follow baselines, only changing the training process by adding our $\mathcal{L}_{sude}$.

4. Theoretical Analysis

Here we analyze that SuDe works well since it models the $p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{c}_{sub}, \boldsymbol{c}_{cate})$. According to Eq. 3, 4 and DDPM [14], we can express $\mathcal{L}_{sub}$ and $\mathcal{L}_{sude}$ as:

$$\begin{aligned}\mathcal{L}_{sub} &= -\log[p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{c}_{sub})],\\ \mathcal{L}_{sude} &= -\log[p(\boldsymbol{c}_{cate}|\boldsymbol{x}_{t-1}, \boldsymbol{c}_{sub})].\end{aligned} \tag{11}$$

Here we first simplify the w_s to 1 for easy understanding:

$$\begin{aligned}\mathcal{L}_{sub} + \mathcal{L}_{sude} &= -\log[p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{c}_{sub}) \cdot p(\boldsymbol{c}_{cate}|\boldsymbol{x}_{t-1}, \boldsymbol{c}_{sub})]\\ &= -\log[p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{c}_{sub}, \boldsymbol{c}_{cate}) \cdot p(\boldsymbol{c}_{cate}|\boldsymbol{x}_t, \boldsymbol{c}_{sub})]\\ &= -\log[p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{c}_{sub}, \boldsymbol{c}_{cate})] + S_t,\end{aligned} \tag{12}$$

where $S_t = -\log[p(\boldsymbol{c}_{cate}|\boldsymbol{x}_t, \boldsymbol{c}_{sub})]$ is unrelated to $t-1$. From the Eq. 12, we find that our method models the distribution of $p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{c}_{sub}, \boldsymbol{c}_{cate})$, which takes both $\boldsymbol{c}_{sub}$ and $\boldsymbol{c}_{cate}$ as conditions, thus generates images with both private attributes from $\boldsymbol{c}_{sub}$ and public attributes from $\boldsymbol{c}_{cate}$.

In practice, w_s is a changed hyperparameter on various baselines. This does not change the above conclusion since:

$$\begin{aligned}&w_s \cdot \mathcal{L}_{sude} = -\log[p^{w_s}(\boldsymbol{c}_{cate}|\boldsymbol{x}_{t-1}, \boldsymbol{c}_{sub})],\\ &p^{w_s}(\boldsymbol{c}_{cate}|\boldsymbol{x}_{t-1}, \boldsymbol{c}_{sub}) \propto p(\boldsymbol{c}_{cate}|\boldsymbol{x}_{t-1}, \boldsymbol{c}_{sub}),\end{aligned} \tag{13}$$

where the $a \propto b$ means a is positively related to b. Based on Eq. 13, we can see that the $\mathcal{L}_{sub} + w_s\mathcal{L}_{sude}$ is positively related to $-\log[p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{c}_{sub}, \boldsymbol{c}_{cate})]$. This means that optimizing our $\mathcal{L}_{sude}$ with $\mathcal{L}_{sub}$ can still increase $p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{c}_{sub}, \boldsymbol{c}_{cate})$ when w_s is not equal to 1.

Table 1. **Quantitative results.** These results are average on 4 generated images for each prompt with a DDIM [36] sampler with 50 steps. The † means performances obtained with a flexible w_s. The improvements our SuDe brought on the baseline are marked in red.

Method	Results on Stable diffusion v1.4 (%)				Results on Stable diffusion v1.5 (%)			
	CLIP-I	DINO-I	CLIP-T	BLIP-T	CLIP-I	DINO-I	CLIP-T	BLIP-T
ViCo [13]	75.4	53.5	27.1	39.1	78.5	55.7	28.5	40.7
ViCo w/ SuDe	76.1	56.8	29.7 (+2.6)	43.3 (+4.2)	78.2	59.4	29.6 (+1.1)	43.3 (+2.6)
ViCo w/ SuDe†	75.8	57.5	30.3 (+3.2)	44.4 (+5.3)	77.3	58.4	30.2 (+1.7)	44.6 (+3.9)
Custom Diffusion [17]	76.5	59.6	30.1	45.2	76.5	59.8	30.0	44.6
Custom Diffusion w/ SuDe	76.3	59.1	30.4 (+0.3)	46.1 (+0.9)	76.0	60.0	30.3 (+0.3)	46.6 (+2.0)
Custom Diffusion w/ SuDe†	76.4	59.7	30.5 (+0.4)	**46.3** (+1.1)	76.2	60.3	**30.3** (+0.3)	**46.9** (+2.3)
DreamBooth [30]	77.4	59.7	29.0	42.1	**79.5**	**64.5**	29.0	41.8
DreamBooth w/ SuDe	**77.4**	**59.9**	29.5 (+0.5)	43.3 (+1.2)	78.8	63.3	29.7 (+0.7)	43.3 (+1.5)
DreamBooth w/ SuDe†	77.1	59.7	**30.5** (+1.5)	45.3 (+3.2)	78.8	64.0	29.9 (+0.9)	43.8 (+2.0)

5. Experiments

5.1. Implementation Details

Frameworks: We evaluate that our SuDe works well in a plug-and-play manner on three well-designed frameworks, DreamBooth [30], Custom Diffusion [17], and ViCo [13] under two backbones, Stable-diffusion v1.4 (SD-v1.4) and Stable-diffusion v1.5 (SD-v1.5) [2]. In practice, we keep all designs and hyperparameters of the baseline unchanged and only add our $\mathcal{L}_{sude}$ to the training loss. For the hyperparameter w_s, since these baselines have various training paradigms (e.g., optimizable parameters, learning rates, etc), it's hard to find a fixed w_s for all these baselines. We set it to 0.4 on DreamBooth, 1.5 on ViCo, and 2.0 on Custom Diffusion. A noteworthy point is that users can adjust w_s according to different subjects in practical applications. This comes at a very small cost because our SuDe is a plugin for test-time tuning baselines, which are of high efficiency (e.g., $\sim$ 7 min for ViCo on a single 3090 GPU).

Dataset: For quantitative experiments, we use the DreamBench dataset provided by DreamBooth [30], containing 30 subjects from 15 categories, where each subject has 5 example images. Since we focus on one-shot customization here, we only use one example image (numbered '00.jpg') in all our experiments. In previous works, their most collected prompts are attribute-unrelated, such as 'photo of a $\{S^*\}$ in beach/snow/forest/...', only changing the image background. To better study the effectiveness of our method, we collect 5 attribute-related prompts for each subject. Examples are like 'photo of a *running* $\{S^*\}$' (for dog), 'photo of a *burning* $\{S^*\}$' (for candle). Moreover, various baselines have their unique prompt templates. Specifically, for ViCo, its template is 'photo of a $\{S^*\}$', while for DreamBooth and Custom Diffusion, the template is 'photo of a $\{S^*\}$ [category]'. In practice, we use the default template of various baselines. In this paper, for the convenience of writing, we uniformly record $\{S^*\}$ and $\{S^*\}$ [category] as $\{S^*\}$. Besides, we also show other qualitative examples in appendix, which are collected from Unsplash [1].

Metrics: For the subject-driven generation task, two important aspects are *subject fidelity* and *text alignment*. For the first aspect, we refer to previous works and use DINO-I and CLIP-I as the metrics. They are the average pairwise cosine similarity between DINO [4] (or CLIP [24]) embeddings of generated and real images. As noted in [13, 30], the DINO-I is better at reflecting fidelity than CLIP-I since DINO can capture differences between subjects of the same category. For the second aspect, we refer to previous works that use CLIP-T as the metric, which is the average cosine similarity between CLIP [24] embeddings of prompts and generated images. Additionally, we propose a new metric to evaluate the text alignment about attributes, abbreviated as *attribute alignment*. This cannot be reflected by CLIP-T since CLIP is only coarsely trained at the classification level, being insensitive to attributes like actions and materials. Specifically, we use BLIP-T, the average cosine similarity between BLIP [18] embeddings of prompts and generated images. It can measure the attribute alignment better since the BLIP is trained to handle the image caption task.

5.2. Qualitative Results

Here, we visualize the generated images on three baselines with and without our method in Fig. 3.

Attribute alignment: Qualitatively, we see that generations with our SuDe align the attribute-related texts better. For example, in the 1st row, Custom Diffusion cannot make the dog ***playing ball***, in the 2nd row, DreamBooth cannot let the cartoon character ***running***, and in the 3rd row, ViCo cannot give the teapot a ***golden material***. In contrast, after combining with our SuDe, their generations can reflect these attributes well. This is because our SuDe helps each subject inherit the public attributes in its semantic category.

Image fidelity: Besides, our method still maintains subject fidelity while generating attribute-rich images. For example, in the 1st row, the dog generated with SuDe is in a very different pose than the example image, but we still

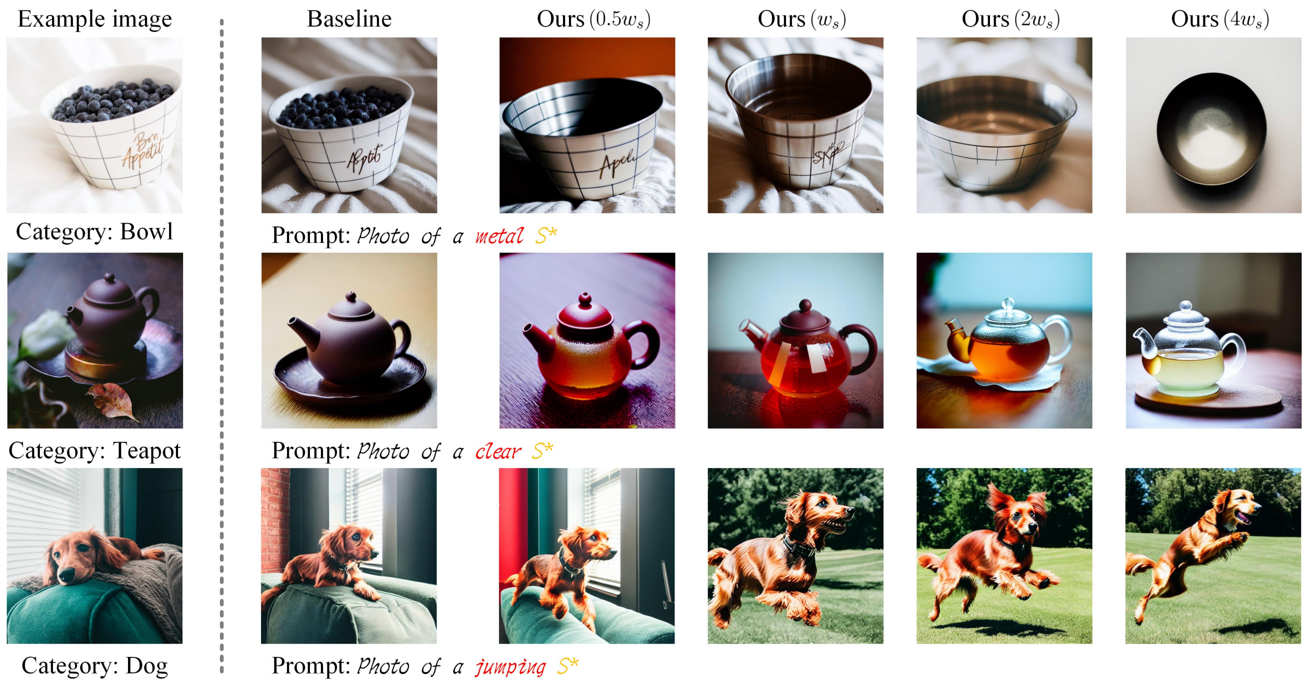

Figure 4. **Visual comparisons by using different values of w_s.** Results are from DreamBooth w/ SuDe, where the default w_s is 0.4.

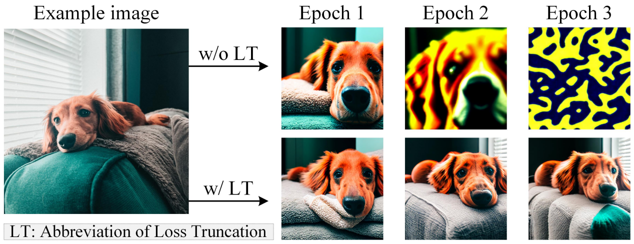

Figure 5. **Loss truncation.** SuDe-generations with and without truncation using Custom Diffusion as the baseline.

can be sure that they are the same dog due to their private attributes, e.g., the golden hair, facial features, etc.

5.3. Quantitative Results

Here we quantitatively verify the conclusion in Sec. 5.2. As shown in Table 1, our SuDe achieves stable improvement on attribute alignment, i.e., BLIP-T under SD-v1.4 and SD-v1.5 of 4.2% and 2.6% on ViCo, 0.9% and 2.0% on Custom Diffusion, and 1.2% and 1.5% on Dreambooth. Besides, we show the performances (marked by †) of a flexible w_s (best results from the [0.5, 1.0, 2.0] $\cdot$ w_s). We see that this low-cost adjustment could further expand the improvements, i.e., BLIP-T under SD-v1.4 and SD-v1.5 of 5.3% and 3.9% on ViCo, 1.1% and 2.3% on Custom Diffusion, and 3.2% and 2.0% on Dreambooth. More analysis about the w_s is in Sec. 5.4.1. For the subject fidelity, SuDe only brings a slight fluctuation to the baseline's DINO-I, indicating that our method will not sacrifice the subject fidelity.

5.4. Empirical Study

5.4.1 Training weight w_s

The w_s affects the weight proportion of $\mathcal{L}_{sude}$. We visualize the generated image under different w_s in Fig. 4, by which we can summarize that: **1)** As the w_s increases, the subject (e.g., teapot) can inherit public attributes (e.g., clear) more comprehensively. A w_s within an appropriate range (e.g., $[0.5, 2] \cdot w_s$ for the teapot) could preserve the subject fidelity well. But a too-large w_s causes our model to lose subject fidelity (e.g., 4 $\cdot w_s$ for the bowl) since it dilutes the $\mathcal{L}_{sub}$ for learning private attributes. **2)** A small w_s is more proper for an attribute-simple subject (e.g., bowl), while a large w_s is more proper for an attribute-complex subject (e.g., dog). Another interesting phenomenon in Fig. 4 1st line is that the baseline generates images with berries, but our SuDe does not. This is because though the berry appears in the example, it is not an attribute of the bowl, thus it is not captured by our derived class modeling. Further, in Sec. 5.4.3, we show that our method can also combine attribute-related and attribute-unrelated generations with the help of prompts, where one can make customizations like 'photo of a metal $\{S*\}$ with cherry'.

5.4.2 Ablation of loss truncation

In Sec. 3.2.2, the loss truncation is designed to prevent the $p(\boldsymbol{c}_{cate}|x_{\theta}(\boldsymbol{x}_t, \boldsymbol{c}_{sub}, t))$ from over-optimization. Here we verify that this truncation is important for preventing the

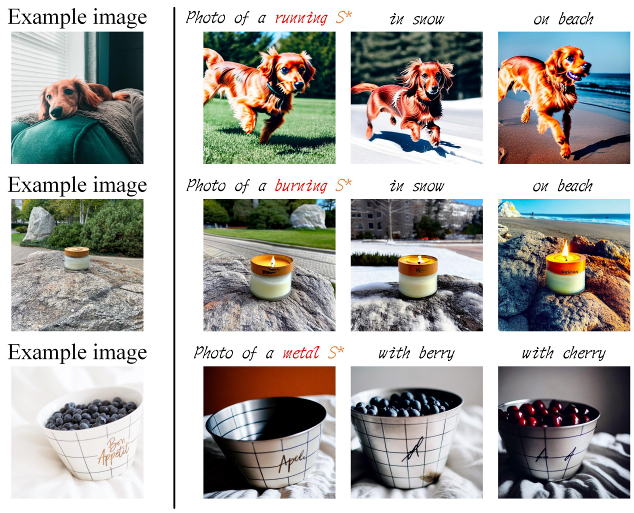

Figure 6. **Combine with attribute-unrelated prompts.** Generations with both attribute-related and attribute-unrelated prompts.

training from collapsing. As Fig. 5 shows, without truncation, the generations exhibit distortion at epoch 2 and completely collapse at epoch 3. This is because over-optimizing $p(\boldsymbol{c}_{cate}|x_\theta(\boldsymbol{x}_t, \boldsymbol{c}_{sub}, t))$ makes a noisy image have an exorbitant classification probability. An extreme example is classifying a pure noise into a certain category with a probability of 1. This damages the semantic space of the pretrained diffusion model, leading to generation collapse.

5.4.3 Combine with attribute-unrelated prompts

In the above sections, we mainly demonstrated the advantages of our SuDe for attribute-related generations. Here we show that our approach's advantage can also be combined with attribute-unrelated prompts for more imaginative customizations. As shown in Fig. 6, our method can generate images harmoniously like, a $\{S^*\}$ (dog) running in various backgrounds, a $\{S^*\}$ (candle) burning in various backgrounds, and a $\{S^*\}$ metal (bowl) with various fruits.

5.4.4 Compare with class image regularization

In existing subject-driven generation methods [13, 17, 30], as mentioned in Eq. 10, a regularization item $\mathcal{L}_{reg}$ is usually used to prevent the model overfitting to the subject example. Here we discuss the difference between the roles of $\mathcal{L}_{reg}$ and our $\mathcal{L}_{sude}$. Using the class image regularization $\mathcal{L}_{reg}$ in DreamBooth as an example, it is defined as:

$$\mathcal{L}_{reg} = ||x_{\bar{\theta}_{pr}}(\boldsymbol{x}_t, \boldsymbol{c}_{cate}, t) - x_\theta(\boldsymbol{x}_t, \boldsymbol{c}_{cate}, t)||^2, \quad (14)$$

where the $x_{\bar{\theta}_{pr}}$ is the frozen pre-trained diffusion model. It can be seen that Eq. 14 enforces the generation conditioned on $\boldsymbol{c}_{cate}$ to keep the same before and after subject-driven finetuning. Visually, based on Fig. 7, we find that the $\mathcal{L}_{reg}$ mainly benefits background editing. But it only uses the 'category prompt' ($\boldsymbol{c}_{cate}$) alone, ignoring modeling the affiliation between $\boldsymbol{c}_{sub}$ and $\boldsymbol{c}_{cate}$. Thus it cannot benefit attribute editing like our SuDe.

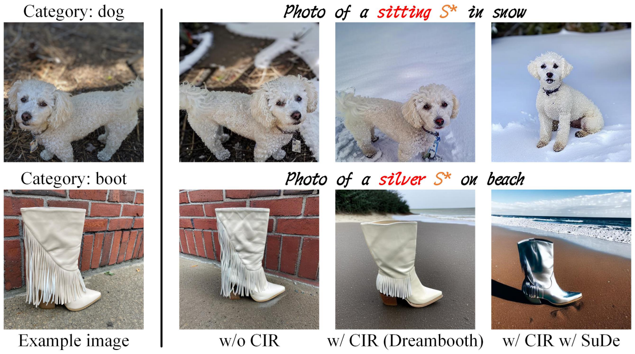

Figure 7. 'CIR' is the abbreviation for class image regularization.

6. Conclusion

In this paper, we creatively model subject-driven generation as building a derived class. Specifically, we propose subject-derived regularization (SuDe) to make a subject inherit public attributes from its semantic category while learning its private attributes from the subject example. As a plugin-and-play method, our SuDe can conveniently combined with existing baselines and improve attribute-related generations. Our SuDe faces the most challenging but valuable one-shot scene and can generate imaginative customizations, showcasing attractive application prospects.

Broader Impact. Subject-driven generation is a newly emerging application, most works of which currently focus on image customizations with attribute-unrelated prompts. But a foreseeable and valuable scenario is to make more modal customizations with the user-provided image, where attribute-related generation will be widely needed. This paper proposes the modeling that builds a subject as a derived class of its semantic category, enabling good attribute-related generations, and thereby providing a promising solution for future subject-driven applications.

Acknowledgments. This work was supported in part by the National Key R&D Program of China (No.2022ZD0118201), Natural Science Foundation of China (No.61972217, 32071459, 62176249, 62006133, 62271465), and the Shenzhen Medical Research Funds in China (No.B2302037). Also, we extend our gratitude to the FaceChain community for their contributions to this work.

References

[1] Unsplash. In https://unsplash.com/. 6, 14

[2] Stable diffusion. In https://huggingface.co/

CompVis/stable-diffusion-v-1-4-original, 2022. 3, 4, 6

[3] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Qinsheng Zhang, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, Tero Karras, and Ming-Yu Liu. eDiff-I: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022. 3

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *International Conference on Computer Vision*, pages 9650–9660, 2021. 6

[5] Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T Freeman, Michael Rubinstein, Yuanzhen Li, and Dilip Krishnan. Muse: Text-to-image generation via masked generative transformers. *arXiv preprint arXiv:2301.00704*, 2023. 3

[6] Hong Chen, Yipeng Zhang, Xin Wang, Xuguang Duan, Yuwei Zhou, and Wenwu Zhu. DisenBooth: Disentangled parameter-efficient tuning for subject-driven text-to-image generation. *arXiv preprint arXiv:2305.03374*, 2023. 3

[7] Wenhu Chen, Hexiang Hu, Yandong Li, Nataniel Rui, Xuhui Jia, Ming-Wei Chang, and William W Cohen. Subject-driven text-to-image generation via apprenticeship learning. *arXiv preprint arXiv:2304.00186*, 2023. 3

[8] Katherine Crowson, Stella Biderman, Daniel Kornis, Dashiell Stander, Eric Hallahan, Louis Castricato, and Edward Raff. Vqgan-clip: Open domain image generation and editing with natural language guidance. In *European Conference on Computer Vision*, pages 88–105. Springer, 2022. 3

[9] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, and Jie Tang. Cogview: Mastering text-to-image generation via transformers. *Advances in Neural Information Processing Systems*, 34:19822–19835, 2021. 3

[10] Ming Ding, Wendi Zheng, Wenyi Hong, and Jie Tang. Cogview2: Faster and better text-to-image generation via hierarchical transformers. *Advances in Neural Information Processing Systems*, 35:16890–16902, 2022. 3

[11] Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-a-scene: Scene-based text-to-image generation with human priors. In *European Conference on Computer Vision*, pages 89–106. Springer, 2022. 3

[12] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit Haim Bermano, Gal Chechik, and Daniel Cohen-or. An Image is Worth One Word: Personalizing text-to-image generation using textual inversion. In *International Conference on Learning Representations*, 2022. 2, 3

[13] Shaozhe Hao, Kai Han, Shihao Zhao, and Kwan-Yee K Wong. ViCo: Detail-preserving visual condition for personalized text-to-image generation. *arXiv preprint arXiv:2306.00971*, 2023. 2, 3, 5, 6, 8, 12

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 3, 5

[15] James Joyce. Bayes' theorem. *Stanford Encyclopedia of Philosophy*, 2003. 4

[16] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*, pages 4171–4186, 2019. 3, 4

[17] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. In *Computer Vision and Pattern Recognition*, pages 1931–1941, 2023. 2, 3, 5, 6, 8, 12, 15

[18] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022. 6

[19] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9): 1–35, 2023. 12

[20] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 13

[21] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In *International Conference on Machine Learning*, pages 16784–16804. PMLR, 2022. 1, 2, 3

[22] Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. Language models as knowledge bases? Association for Computational Linguistics, 2019. 12

[23] Stephen Prata. *C++ primer plus*. Sams Publishing, 2002. 2

[24] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 3, 6

[25] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021. 3

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

[27] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *International Conference on Machine Learning*, pages 1060–1069. PMLR, 2016. 3

[28] Tim Rentsch. Object oriented programming. *ACM Sigplan Notices*, 17(9):51–57, 1982. 2

[29] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 1, 2, 3

[30] 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. In *Computer Vision and Pattern Recognition*, pages 22500–22510, 2023. 1, 2, 3, 5, 6, 8, 11, 12, 13

[31] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Wei Wei, Tingbo Hou, Yael Pritch, Neal Wadhwa, Michael Rubinstein, and Kfir Aberman. HyperDreamBooth: Hypernetworks for fast personalization of text-to-image models. *arXiv preprint arXiv:2307.06949*, 2023. 2, 3

[32] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. 1, 2, 3

[33] Timo Schick and Hinrich Schütze. Exploiting cloze-questions for few-shot text classification and natural language inference. In *European Chapter of the Association for Computational Linguistics*, pages 255–269, 2021. 12

[34] 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

[35] Chengyu Song, Fei Cai, Jianming Zheng, Xiang Zhao, and Taihua Shao. AugPrompt: Knowledgeable augmented-trigger prompt for few-shot event classification. *Information Processing & Management*, 60(4):103153, 2023. 12

[36] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2020. 5, 6

[37] Bjarne Stroustrup. An overview of c++. In *Proceedings of the 1986 SIGPLAN workshop on Object-oriented programming*, pages 7–18, 1986. 1

[38] Ming Tao, Hao Tang, Fei Wu, Xiao-Yuan Jing, Bing-Kun Bao, and Changsheng Xu. DF-GAN: A simple and effective baseline for text-to-image synthesis. In *Computer Vision and Pattern Recognition*, pages 16515–16525, 2022. 3

[39] Yoad Tewel, Rinon Gal, Gal Chechik, and Yuval Atzmon. Key-locked rank one editing for text-to-image personalization. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–11, 2023. 3

[40] Peter Wegner. Concepts and paradigms of object-oriented programming. *ACM Sigplan Oops Messenger*, 1(1):7–87, 1990. 2

[41] 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. In *European Conference on Computer Vision*, 2023. 2, 3, 13

[42] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks. In *Computer Vision and Pattern Recognition*, pages 1316–1324, 2018. 3

[43] Yuxin Zhang, Weiming Dong, Fan Tang, Nisha Huang, Haibin Huang, Chongyang Ma, Tong-Yee Lee, Oliver Deussen, and Changsheng Xu. Prospect: Expanded conditioning for the personalization of attribute-aware image generation. *arXiv preprint arXiv:2305.16225*, 2023. 3

[44] Minfeng Zhu, Pingbo Pan, Wei Chen, and Yi Yang. DM-GAN: Dynamic memory generative adversarial networks for text-to-image synthesis. In *Computer Vision and Pattern Recognition*, pages 5802–5810, 2019. 3