Customization Assistant for Text-to-image Generation

Yufan Zhou, Ruiyi Zhang, Jiuxiang Gu, Tong Sun
Adobe Research
{yufzhou, ruizhang, jigu, tsun}@adobe.com

Abstract

Customizing pre-trained text-to-image generation model has attracted massive research interest recently, due to its huge potential in real-world applications. Although existing methods are able to generate creative content for a novel concept contained in single user-input image, their capability are still far from perfection. Specifically, most existing methods require fine-tuning the generative model on testing images. Some existing methods do not require fine-tuning, while their performance are unsatisfactory. Furthermore, the interaction between users and models are still limited to directive and descriptive prompts such as instructions and captions. In this work, we build a customization assistant based on pre-trained large language model and diffusion model, which can not only perform customized generation in a tuning-free manner, but also enable more user-friendly interactions: users can chat with the assistant and input either ambiguous text or clear instruction. Specifically, we propose a new framework consists of a new model design and a novel training strategy. The resulting assistant can perform customized generation in 2-5 seconds without any test time fine-tuning. Extensive experiments are conducted, competitive results have been obtained across different domains, illustrating the effectiveness of the proposed method.

1. Introduction

Customizing pre-trained text-to-image models has drawn significant interest in the research community, due to its potential in real-world applications. Customized generation aims at generating creative images for a specific concept contained in user provided images. Despite of the impressive progress that large-scale text-to-image generation models have made in recent years [1, 5, 24–26, 29, 35, 36], they fail to perform customized generation for novel concept, such as a specific animal or object which only appear in single testing image.

Various approaches have been proposed to tackle this task. Some methods focus on fine-tuning the pre-trained generation model [13, 22, 27] on testing images, so that the model can learn fine-grained details about the novel concept; Some methods aim to represent the concept by embeddings [6, 7, 15, 33, 39], which can be obtained either by optimization method or through a learned encoder. The embeddings are then injected into pre-trained text-to-image generation models to perform customized generation. The test time required for running these methods varies, spanning from few seconds to up to thirty minutes. We will discuss more details about related works in later section.

Figure 1. Generated example from the proposed CAFE. CAFE can perform customized generation based on the user provided image in a tuning-free manner. It outputs creative images along with text explanation and elaboration.

Although existing methods are capable of generating creative contents for the target novel concept, they still have drawbacks and limitations. For instance, all these existing methods are not user-friendly enough: they can only handle prompts that are directive or descriptive in nature, such as a caption "A picture of the dog in comic book style" or an instruction "Generate an image of the dog in comic book style". They can not handle ambiguous input such as "I want to generate something creative, can you help me?", which could be important in applications because the user may only have a vague target instead of precise require-

ments in mind.

In this work, we propose CAFE which is a ***C**ustomization **A**ssistant **F**or text-to-imag**E** generation*. CAFE is able to perform tuning-free customization within 2-5 seconds on testing images from arbitrary domain. Thus it is one of the most efficient customization method. Different from existing methods which take instructions or captions as text input, CAFE can handle both declarative and interrogative sentences because it is built upon a large language model (LLM) Llama-2 [32]. Furthermore, CAFE is user-friendly than any existing methods as it can even infer user's intention when the prompt is ambiguous, and output explanation and elaboration for the generation as shown in Figure 1.

Our contributions can be summarized as following:

- We propose CAFE, a novel method which can perform tuning-free customized generation in 2-5 seconds. Different from previous methods, CAFE is built upon large language models, thus can handle ambiguous text input. Furthermore, CAFE can take extra images as additional semantic condition. It also possesses the unique capability of providing text explanation and elaboration for the generated content, which none of existing customization method can achieve;
- We propose a novel training strategy, which can efficiently construct large-scale high-quality dataset for training CAFE without human supervision thus saves huge amount of cost;
- Extensive experiments are conducted, where CAFE achieves promising quantitative and qualitative results across different domains. We also conduct several ablation studies, which verify the underlying rationale of the proposed method.

2. Related Works

Text-to-image Generation The field of text-to-image generation has been a subject of research for years and has recently seen remarkable advancements. Previous methods which are based on Generative Adversarial Networks [31, 34, 37, 38] lack the ability of generating open-domain images with arbitrary text input. Starting from DALL-E [24], researchers are able to perform impressive zero-shot text-to-image generation with good fidelity and image-text alignment, after training the model on large-scale dataset. Specifically, DALL-E [24] and CogView [5] propose to use transformer model to infer image tokens from text, which will be further transformed into images through an auto-encoder. GLIDE [19] adopts a hierarchical architecture which consists of diffusion models at different resolution, leading to impressive generation quality. The idea of using hierarchical design is also adopted by some follow-up works [1, 25] and proven to be effective in both diffusion models and auto-regressive models. LDM [26] proposes to train a diffusion model inside the lower-dimensional latent space of auto-encoder, leading to better generation efficiency. Base on the research detailed above, numerous efforts have been undertaken to further enhance the proficiency of text-to-image generation models, including introducing better semantic understanding through a pre-trained large-scale text encoder [29] and scaling up the model towards better generalization ability and more promising results [35, 36]. In this work, we chose Stable Diffusion [26] as the foundation for our CAFE due to its open-source availability.

Customized Generation To enable customized generation for the specific concept presented in few-shot or single image, many customization methods proposed to fine-tune the pre-trained text-to-image generation model. DreamBooth [27] propose to fine-tuned the entire diffusion model, while Custom Diffusion [14] only fine-tuned the cross attention module inside the UNet of diffusion model. LoRA [10] is often used to reduce the number of parameters to be tuned and improve the efficiency of fine-tuning. Recently, OFT [22] is proposed, which can stabilize the fine-tuning process by preserving pairwise neuron relationship of pre-trained diffusion model.

Other works focus on representing target concept by learned embeddings. Textual Inversion [6] propose to encode the concept by embedding vectors inside the input space of text encoder of the diffusion model. Optimization method is utilized to obtain the embedding, which may take up to 30 minutes. To reduce the time cost, different works [2, 7, 15, 39] have been proposed, focusing on pre-training encoders which can directly map the testing images into the target embeddings. Image patch features from pre-trained image encoders are often used to enhance the performance because some detailed information might be challenging to be captured inside the input embedding space of pre-trained text encoder [4, 18, 30, 33].

There are also some other investigations: SuTI [3] proposes to employ apprenticeship learning to obtain one single apprentice model to imitate half a million subject-specific experts; Kosmos-G [20] tries to utilize large language model so that interleaved vision-language prompt can be handled; HyperDreamBooth [28] directly obtains a customized model by training a hyper-network to generate the weights for target models.

3. Method

Our goal is to design an assistant which can generate creative images for a target object or person provided by single testing image in a tuning-free manner, the generated results should be aligned with arbitrary user-input text. Different from existing works, we expect our assistant to be more user-friendly: the assistant should be able to handle both

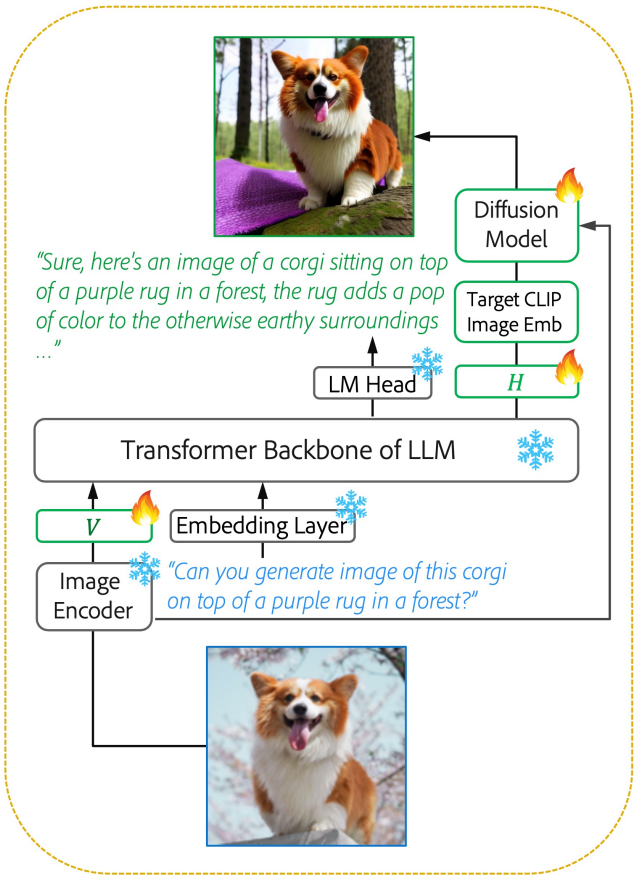


Figure 2. Illustration of our model architecture, where the modules to be fine-tuned are indicated by flame icons.

declarative and interrogative sentences; when the user-input is ambiguous, it should have the ability to infer user's intention and generate corresponding images; As an assistant, it is also expected to explain reason and insight behind its generated content through natural language;

To this end, we propose to utilize the capability of large language models (LLMs). Our model architecture is presented in Figure 2, with more details discussed as follows.

3.1. Customization Assistant

Let $\mathcal{E}$ be a pre-trained image encoder, $\mathcal{M}$ be our multimodal large language model (MLLM), $\mathbf{x}$ be an image containing the object we want to generate images for. $\mathcal{E}(\mathbf{x})$ is injected into MLLM through a vision projection layer V following LLaVA [16].

Based on user-input image $\mathbf{x}$ and text $\mathbf{y}$, $\mathcal{M}$ will infer user's intention and outputs two sequences of embeddings: $\{\mathbf{e}_i\}$ and $\{\mathbf{s}_j\}$. $\{\mathbf{s}_j\}$ will be mapped into special embeddings $\{\mathrm{H}(\mathbf{s}_j)\}$ through a newly introduced projection layer H. $\{\mathrm{H}(\mathbf{s}_j)\}$ will be injected into a diffusion model $\mathcal{G}$, to guide the generation process. Meanwhile, $\{\mathbf{e}_i\}$ will be mapped into natural language through a language modelling (LM) head, which will provide additional information or explanation of the generated results.

The output embeddings $\{\mathrm{H}(\mathbf{s}_j)\}$ are capable of capturing most of the semantic information for the target generation, while it may lose fine-grained details of the original image, due to the difficulty of aligning different modalities within single output space. Thus we also introduce $\mathcal{E}(\mathbf{x})$ into the diffusion model $\mathcal{G}$. Specifically, both $\{\mathrm{H}(\mathbf{s}_j)\}$ and $\mathcal{E}(\mathbf{x})$ will be injected into the diffusion model through cross-attention layers:

$$\mathbf{z}' = \mathrm{Softmax}(\frac{Q_{\mathbf{z}}K_{\mathrm{H}}^T}{\sqrt{d}})V_{\mathrm{H}} + \mathrm{Softmax}(\frac{Q_{\mathbf{z}}K_{\mathcal{E}}^T}{\sqrt{d}})V_{\mathcal{E}}, \quad (1)$$

where d is the scaling factor in attention mechanism, $\mathbf{z}, \mathbf{z}'$ are intermediate features inside the UNet of diffusion model, $Q_{\mathbf{z}}$ is the query value calculated based on $\mathbf{z}$, $K_{\mathrm{H}}, V_{\mathrm{H}}$ denote key and value calculated based on $\{\mathrm{H}(\mathbf{s}_j)\}$, $K_{\mathcal{E}}, V_{\mathcal{E}}$ denote key and value corresponding to $\mathcal{E}(\mathbf{x})$.

Let $\tilde{\mathbf{x}}$ be the target generation based on input image $\mathbf{x}$ and input text $\mathbf{y}$, $\tilde{\mathbf{y}}$ denotes the target response from the language model, our MLLM $\mathcal{M}$ is trained with the loss

$$\mathcal{L}_{\mathcal{M}} = \mathcal{L}_{\mathrm{LM}}(\mathcal{M}(\mathbf{y}), \tilde{\mathbf{y}}) + \lambda d(\mathrm{H}(\mathbf{s}), \mathcal{F}(\tilde{\mathbf{x}})), \quad (2)$$

where $\mathcal{L}_{\mathrm{LM}}$ is the language modeling loss, $\mathcal{M}(\mathbf{y})$ denotes response generated by the language model, $\mathcal{F}(\tilde{\mathbf{x}})$ denotes the CLIP image global embedding of $\tilde{\mathbf{x}}$, $d(\cdot,\cdot)$ measures the difference between two vectors, $\lambda > 0$ is a hyper-parameter. In practice, we set $d(\cdot,\cdot)$ to be mean squared error, which works well than negative cosine similarity according to our experiments. We set $\lambda = 0.2$ which leads to the best results in our implementation.

In Equation (2), $\mathrm{H}(\mathbf{s})$ is designed to be a CLIP [23] image embedding. This design leads to two major benefits. The first benefit is that we can train MLLM $\mathcal{M}$ and diffusion model $\mathcal{G}$ separately because diffusion model $\mathcal{G}$ does not appear in (2). This is important especially considering the fact that both models can have billions of parameters, training MLLM and diffusion model together could be computationally prohibitive. The other important benefit is that it enables more flexible generations: when we meet difficulty in describing our target in words, we can directly use image embedding of another image $\mathbf{w}$ as semantic to guide the generation via

$$\tilde{\mathbf{x}} = \mathcal{G}(\mathcal{E}(\mathbf{x}), \mathcal{F}(\mathbf{w})),$$

which leads to impressive results as shown in Figure 3.

Our diffusion model is trained with

$$\mathcal{L}_{\mathcal{G}} = \mathbb{E}\left[\|\epsilon - \epsilon_\theta(\mathcal{F}(\tilde{\mathbf{x}}), \mathcal{E}(\mathbf{x}), \tilde{\mathbf{x}}_t)\|^2\right] \quad (3)$$

where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ denotes randomly sampled noise, $\tilde{\mathbf{x}}_t$ denotes noised sample [9].

From the above objective functions, the readers may notice that we need samples in the format of quadruplet $(\mathbf{x}, \tilde{\mathbf{x}}, \mathbf{y}, \tilde{\mathbf{y}})$ to train our model. We next present how to construct such samples for training.

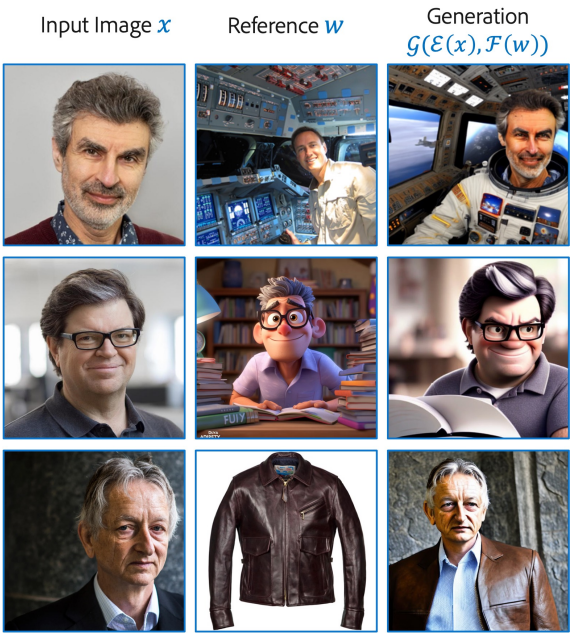


Figure 3. Our method enables tuning-free generation conditioned on multiple images. Details captured by $\mathcal{E}(\mathbf{x})$ can be seamlessly combined with semantic $\mathcal{F}(\mathbf{w})$.

3.2. Dataset Construction

To begin with, we prepare a collection of images $\{\mathbf{x}_i\}$ and some manually designed instructions. Then we generate target images $\{\tilde{\mathbf{x}}_j\}$ by randomly selecting input image and instruction and applying another customization method ProFusion [39], whose implementation is publicly available and leads to promising results efficiently.

The generated images are then automatically filtered via CLIP and DINO similarities[1]: we compute the image-instruction similarity and filter out image whose similarity is less than 0.3; then we filtered out image whose DINO similarity with original image is less than 0.6. After filtering, the resulting images are expected to contain target object with good identity preservation and instruction-alignment.

The resulting images will be further filtered again by human workers. The workers are provided the original image, generation instruction and generated image. They are then asked to filter out low-quality generations which are not aligned with instruction or the identity is not well preserved by human preference.

At last, we prompt Llama-2-70B-chat [32] model to generate input text $\mathbf{y}$ and target response $\tilde{\mathbf{y}}$, which simulate the interaction between user and assistant. The prompt we used is provided in the Appendix. Examples from our resulting dataset are shown in Figure 4.

[1]We use pre-trained CLIP ViT-B/32 and DINO ViT-S/16 in all the automatic data filtering.

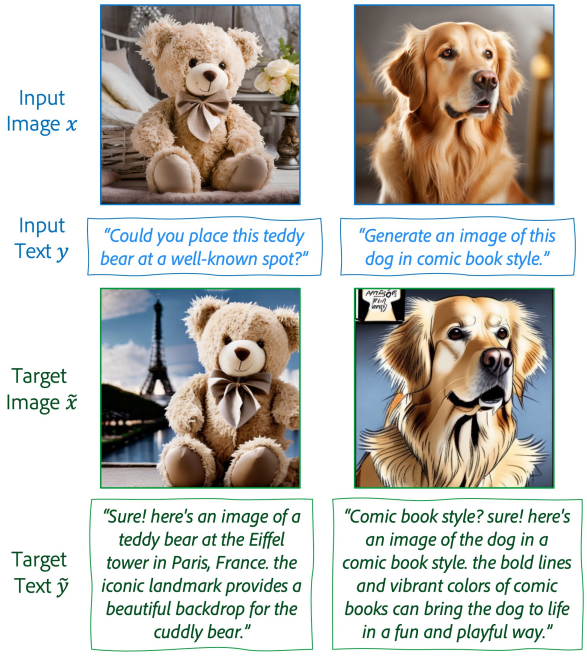


Figure 4. Two examples from our dataset, each sample contains four elements $(\mathbf{x}, \mathbf{y}, \tilde{\mathbf{x}}, \tilde{\mathbf{y}})$.

3.3. Self-improvement via Distillation

Although we are able to construct desired dataset with the above pipeline, it actually requires a massive amount of computation and time cost.

In our initial trial, 10,000 Nvidia A100 GPU hours are spent to generate around 3,000,000 samples. After automatic filtering, around 1,500 worker hours are spent to obtain the resulting dataset, which only consists of 93,000 samples.

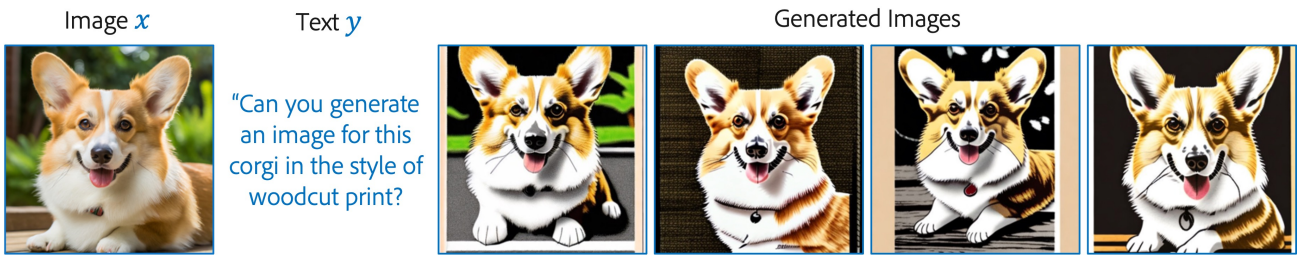
Obviously, constructing a dataset which may cover arbitrary domain is expensive with the above pipeline. A consequential question is, is it possible to obtain a larger and better dataset more efficiently? To this end, we propose a novel strategy, which is totally automatic, and can be easily scaled up because it does not require any human filtering workload. We present the details below.

First of all, we remove the human filtering stage in previous pipeline, and directly train a customization assistant using the automatically filtered data. Then we use the trained model to generate more samples, which will be used to finetune the model itself after automatic filtering, thus our strategy is termed as Self-improvement via Distillation (SID). Specifically, we propose to generate new training images by

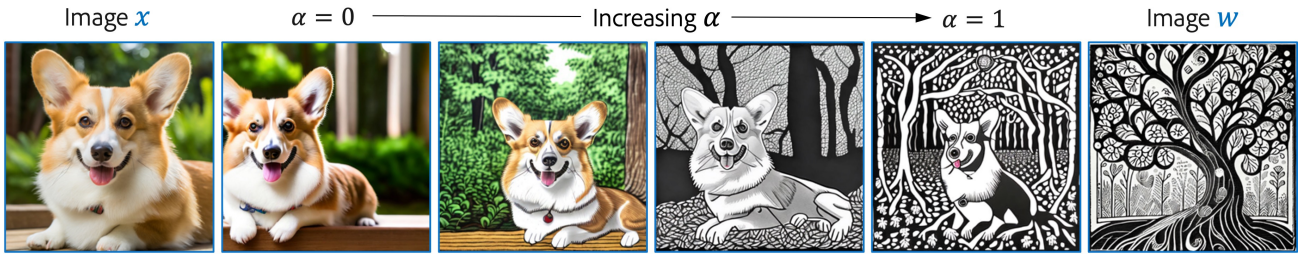
$$\tilde{\mathbf{x}} = \mathcal{G}(\mathcal{E}(\mathbf{x}), \alpha\mathcal{F}(\mathbf{w}) + (1-\alpha)\mathcal{F}(\mathbf{x})) \tag{4}$$

instead of

$$\tilde{\mathbf{x}} = \mathcal{G}(\mathcal{E}(\mathbf{x}), \mathrm{H}(\mathbf{s})),$$



(a) Generation with $\mathcal{G}(\mathcal{E}(\mathbf{x}), \mathrm{H}(\mathbf{s}))$



(b) Generation with $\mathcal{G}(\mathcal{E}(\mathbf{x}), \alpha\mathcal{F}(\mathbf{w}) + (1-\alpha)\mathcal{F}(\mathbf{x}))$

Figure 5. We can generate high quality training data efficiently with (4). The semantic and identity can also be easily controlled through single hyper-parameter α.

where $\mathbf{w}$ is a reference image retrieved from database or generated by pre-trained Stable Diffusion with target instruction. Some details are provided in the Appendix. As shown in Figure 5, the initial model may fail to generate target image, when its style or semantic is rare in the training data. (4) provides an efficient way to generate those images, which can be then be used to construct a more comprehensive dataset or balance the training data distribution. Furthermore, by simply using different image $\mathbf{w}$ and hyper-parameter α, we are able to control identity preservation and semantics efficiently.

Recall that ProFusion requires testing time fine-tuning to perform customized generation: given a testing image, around 30 seconds of fine-tuning is needed. On the contrary, our customization assistant is a tuning-free method that generates image in few seconds, thus can save a huge amount of time. As we will show in the experiment, our proposed strategy can generate high-quality data efficiently which leads to better results even than the model trained on human filtered data.

4. Experiment

4.1. Implementation Details

We conduct all the experiments with PyTorch [21] on Nvidia A100 GPUs. Our final dataset consists of around 1 million $(\mathbf{x}_i, \tilde{\mathbf{x}}_i, \mathbf{y}_i, \tilde{\mathbf{y}}_i)$ quadruplet samples, which costs around 20,000 GPU hours. 355K samples are in human image domain, while the rest images focus on open domain objects. The reference images $\{\mathbf{x}_i\}$ contain both public dataset and generated images: we directly use FFHQ [11] dataset to generate $(\mathbf{x}_i, \tilde{\mathbf{x}}_i, \mathbf{y}_i, \tilde{\mathbf{y}}_i)$ samples for human face domain; for object domain, we use pre-trained Stable Diffusion 2 [26] to generate reference images $\{\mathbf{x}_i\}$ for some selected object classes, then use the generated images to construct $(\mathbf{x}_i, \tilde{\mathbf{x}}_i, \mathbf{y}_i, \tilde{\mathbf{y}}_i)$ samples. More details are provided in the Appendix.

Two stages are implemented in training the model: the customization assistant is first trained for 5 epochs on samples generated by ProFusion, then fine-tuned for 5 epochs on the samples generated by the first stage model.

We set $\mathcal{F}$ to be CLIP ViT-L/14@336px model, and use DINOv2-Giant as image encoder $\mathcal{E}$. As a result, $\mathcal{H}(\mathbf{s}) \in \mathbb{R}^{1\times 768}, \mathcal{E}(\mathbf{x}) \in \mathbb{R}^{257\times 1536}$. Our large language model is initialized from Llama-2-13B-chat checkpoint. We introduce a fully-connected layer which projects image embeddings into the input space of Llama-2, and a fully-connected layer which is the projection head for CLIP embedding prediction. A learn-able token is appended at the end of text tokens, which will be used in predicting the CLIP global embedding. AdamW [17] optimizer with learning rate of 2e-3 and batch size of 128 is used. Llama-2 backbone is kept frozen, only newly introduced projection layers are fine-tuned.

Our diffusion model is initialized from pre-trained Stable Diffusion 2. We remove its original text encoder, and introduce new cross attention layers so that the UNet can take global embedding from CLIP ViT-L/14@336px and patch embeddings from DINOv2-Giant. The diffusion model is also trained with AdamW optimizer. The learning rate is set to be 2e-5 and batch size is 64. During training, the DINO and CLIP embeddings are randomly dropped independently with a probability of 0.1 to enable classifier-free guidance [8].

Figure 6. Generated examples from the proposed CAFE.

4.2. Quantitative Results

Following previous works [7, 27], we conduct experiments on object domain and human image domain. Some generated examples are presented in Figure 6 and Figure 7. More examples will be provided in the Appendix, including examples on multi-round generation and image editing task.

Object domain We conduct quantitative evaluation on DreamBench [27], which consists of 30 subjects and 25 prompts for each subjects. 4 images are generated for each of the 750 unique combinations. Following [27], we calculate image similarities with pre-trained DINO ViT-S/16 and CLIP ViT-B/32 models, which evaluate the identity preservation between generated image and original image by computing cosine similarity between their extracted features. The metrics are denoted as DINO and CLIP-I respectively. Image-text similarity between generated image and prompt is calculated using pre-trained CLIP ViT-B/32 model, which is denoted as CLIP-T in the results.

In all the experiments, we use slightly different prompts from other methods. For example, in the case where one want to generate an image for a specific dog on the beach, the prompt for other methods might be "A S^* dog on the beach" where S^* represents the embedding capturing the characteristics of the dog. While prompt of our model is randomly selected from "Can you generate an image for this dog on the beach?" and "Generate an image for this dog on the beach.". Nevertheless, we still use their prompts in computing the CLIP-T similarity for fair comparison.

The main results are reported in Table 1, where we compare our method with Textual Inversion [6], DreamBooth [27], CustomDiffusion [14], BLIP-Diffusion [15], ELITE [33], Subject-Diffusion [18], SuTI [3], Kosmos-G [20]. Results of baseline methods can directly taken from previous papers. Our method achieves competitive results, and is the only model which can generate text explanations and elaborations along with images. Note that SuTI is based on Imagen [29], which is a stronger base model than our Stable Diffusion 2. SuTI requires 15-20 seconds to perform generation, while our method only needs 5 seconds, which can be further reduced to 2 seconds if one directly uses CLIP embedding from an image instead of embedding generated by MLLM.

Human image domain We then conduct experiments on human face domain following [7, 39]. We train a model on human image subset of our dataset, then evaluate it with all the 23 prompts, 7 researcher images provided in [7]. 10 images are generated for each image-prompt combination. The generated images are then evaluated by two metrics

Method	Tuning-free	DINO (↑)	CLIP-I (↑)	CLIP-T (↑)
Real Images	-	0.774	0.885	-
Textual Inversion	✗	0.569	0.780	0.255
DreamBooth	✗	0.668	0.803	0.305
CustomDiffusion	✗	0.643	0.790	0.305
BLIP-Diffusion	✗	0.670	0.805	0.302
BLIP-Diffusion	✓	0.594	0.779	0.300
Re-Imagen	✓	0.600	0.740	0.270
ELITE	✓	0.621	0.771	0.293
Subject-Diffusion	✓	0.711	0.787	0.293
SuTI	✓	0.741	0.819	0.304
Kosmos-G	✓	0.694	0.847	0.287
CAFE (Ours)	✓	0.715	0.827	0.294

Table 1. Quantitative evaluation on DreamBench.

Method	Tuning-free	ID (↑)	CLIP-T (↑)
Textual Inversion	✗	0.210	0.257
Dreambooth	✗	0.307	0.283
E4T	✗	0.426	0.277
ProFusion	✗	0.432	0.293
CAFE (Ours)	✓	0.464	0.297

Table 2. Results evaluated in human face domain.

following [39]: we utilize CLIP ViT-B/32 models to calculate the image-text similarity; we evaluate identity similarity by the cosine similarity between extracted features of generated and original image using pre-trained face recognition model [12]. The main results are presented in Table 2 where the identity similarity is denoted as ID. We compare our method with previous methods including Textual Inversion [6], DreamBooth [27], E4T [7] and ProFusion [39]. Better results are obtained with our method, indicating the effectiveness of the proposed method. Some qualitative comparisons are provided in Figure 7. We also include results from another tuning-free method PhotoVerse [2], which is specifically designed for human face domain. However, the implementation of PhotoVerse is not available, thus only qualitative comparison is provided. We can see that the proposed CAFE leads to better identity preservation and image fidelity.

4.3. Ablation Studies

Effectiveness of SID One important question is, given the same amount of computation resources and time, will the proposed SID training strategy lead to a better model which outperforms the model trained on human filtered data samples?

We conduct an ablation study to answer the above question. Specifically, we start from a dataset generated by ProFusion in human face domain, which cost around 10,000 A100 GPU hours. Then different model variants are trained on the following datasets:

- Dataset $\mathcal{D}_1$, which only contains automatically filtered samples;
- Dataset $\mathcal{D}_2$, which is constructed by asking human workers to filter out low-quality samples in $\mathcal{D}_1$;
- Dataset $\mathcal{D}_3$, which only consists of samples generated by the model trained on $\mathcal{D}_1$. Filtering with CLIP and DINO similarity is also performed;
- Dataset $\mathcal{D}_4$, which is the union $\mathcal{D}_1 \cup \mathcal{D}_3$;

Dataset	Total Cost	# of Sample	ID (↑)	CLIP-T (↑)
$\mathcal{D}_1$	10,000 GPU hour	207K	0.433	0.294
$\mathcal{D}_2$	11,500 hour	93K	0.441	0.288
$\mathcal{D}_3$	11,500 GPU hour	148K	0.465	0.291
$\mathcal{D}_4$	11,500 GPU hour	355K	0.464	0.297

Table 3. Ablation study with different dataset, the proposed can efficiently construct a high-quality dataset, which leads to improved model performance.

Loss Function	ID (↑)	CLIP-T (↑)
Mean Sqaure Error	0.464	0.297
Nagative Cosine Similarity	0.456	0.295

Table 4. Ablation study with different loss functions.

CLIP Model	DINO Model	ID (↑)	CLIP-T (↑)
ViT-B/32	DINOv2-Giant	0.449	0.252
ViT-L/14@336px	DINOv2-Base	0.408	0.295
ViT-L/14@336px	DINOv2-Giant	0.464	0.297

Table 5. Ablation study with different image encoders.

The cost of generating samples for $\mathcal{D}_3$ is set to be 1,500 GPU hours for fair comparison, as 1,500 worker hours are spent in human filtering stage of constructing $\mathcal{D}_2$.

The comparison is provided in Table 3, along with some dataset statistics. All the models use the same architecture. From the results we can conclude the proposed self-distillation strategy does lead to better performance. Although human filtered dataset may have better quality in terms of image fidelity, it may not lead to a model with good generalization ability because the amount of sample is limited. We also notice that the model trained on $\mathcal{D}_3$ leads to good performance, illustrating the effectiveness of constructing dataset with (4).

Different objective functions As mentioned in Section 3, we can choose different $d(\cdot, \cdot)$ in (2). Because the CLIP model is trained with contrastive loss using cosine similarity, thus we conduct ablation study to compare using negative cosine similarity and mean squared error in (2). Hyperparameter λ in (2) is selected from $[0.1, 0.2, 0.5, 1.0, 2.0]$ by their resulting performance. The quantitative evaluation is presented in Table 4, from which we can see that mean square error leads to better performance.

Different image encoders Recall that CLIP ViT-L/14@336px and DINOv2-Giant are used in our experiments, readers may be curious about how will different variants of these pre-trained models influence the model perfor-

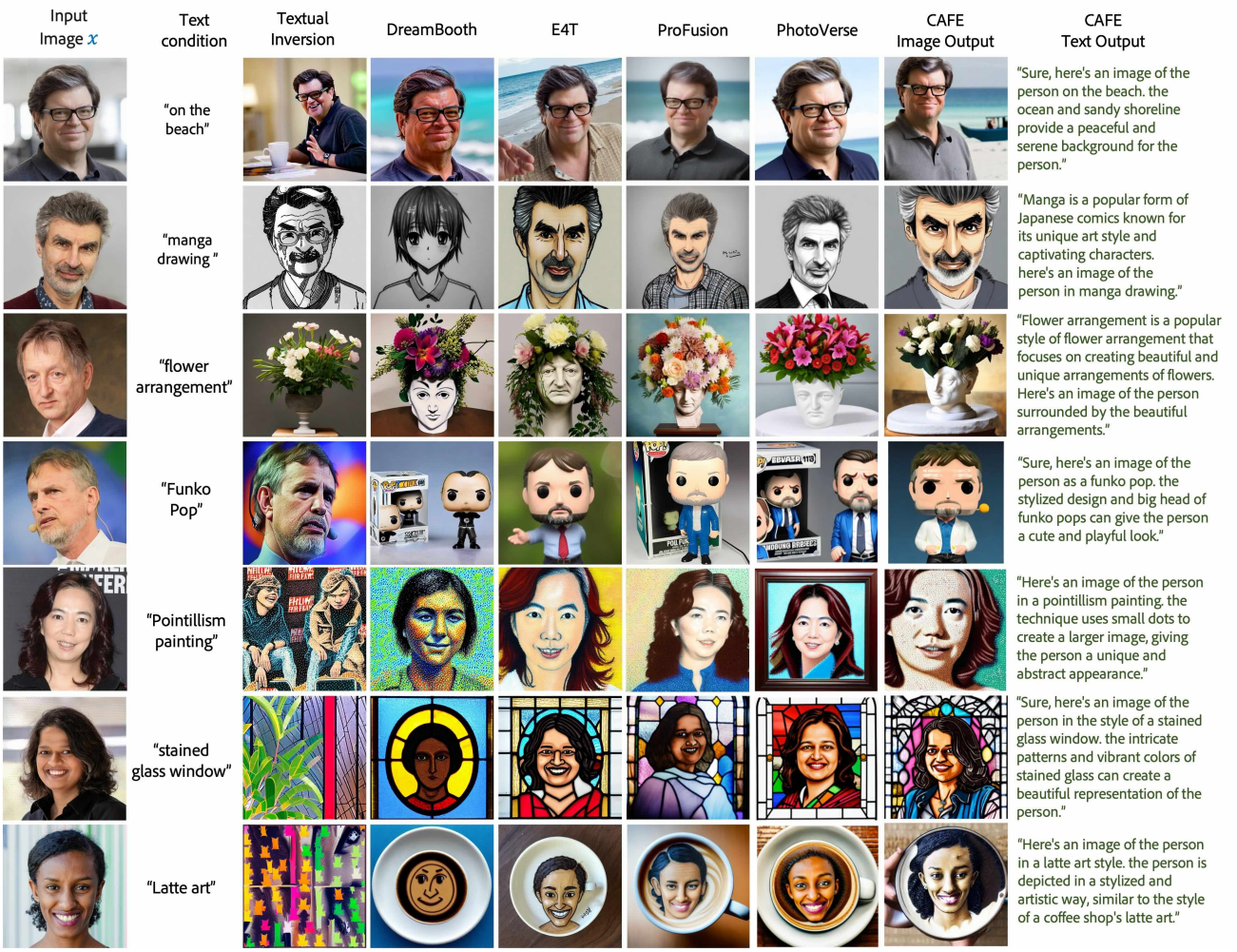


Figure 7. Comparison with related methods on human images. Results of other methods are directly taken from corresponding papers. PhotoVerse and CAFE are tuning-free methods. Our CAFE is able to capture details such as the tiny microphone in the Funko Pop example.

mance. To better understand the impact of different image encoders, we conduct ablation study where the encoders are replaced by smaller variants. The quantitative results are presented in Table 5, which is also evaluated on human faces following previous experiments. As expected, model with CLIP ViT-B/32 encoder obtains much worse results in terms of image-text similarity; while DINOv2-base leads to worse identity similarity than DINOv2-Giant.

Contribution of CLIP and DINO embedding We also conduct ablation study where only CLIP embeddings or DINO embeddings are used in generation. The results are presented in Table 6, where we can find that using only CLIP embedding leads to good CLIP-T score and bad ID score, while using only DINO embedding leads to good ID score and poor CLIP-T score. The results are aligned with our expectation that semantics are mainly controlled by CLIP embedding, fine-grained details are mainly controlled by DINO embedding.

CLIP embedding	DINO embedding	ID (↑)	CLIP-T (↑)
✓	✗	0.216	0.297
✗	✓	0.418	0.234
✓	✓	0.464	0.297

Table 6. Ablation study where only CLIP embedding or DINO embedding is used in generation.

5. Conclusion

In this work, we propose CAFE which is a tuning-free method for customizing pre-trained text-to-image generation model. Different from existing works, the proposed CAFE is based on large language model thus can handle ambiguous user input and output explanations along with generated images. A novel strategy is proposed, which leads to more efficient and scalable dataset construction for training better CAFE. Competitive results are obtained in experiments across different domains, indicating the effectiveness of the proposed method.

References

[1] Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T Freeman, Michael Rubinstein, et al. Muse: Text-to-image generation via masked generative transformers. *arXiv preprint arXiv:2301.00704*, 2023. 1, 2

[2] Li Chen, Mengyi Zhao, Yiheng Liu, Mingxu Ding, Yangyang Song, Shizun Wang, Xu Wang, Hao Yang, Jing Liu, Kang Du, et al. Photoverse: Tuning-free image customization with text-to-image diffusion models. *arXiv preprint arXiv:2309.05793*, 2023. 2, 7

[3] 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. 2, 6

[4] Xi Chen, Lianghua Huang, Yu Liu, Yujun Shen, Deli Zhao, and Hengshuang Zhao. Anydoor: Zero-shot object-level image customization. *arXiv preprint arXiv:2307.09481*, 2023. 2

[5] 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, 2021. 1, 2

[6] 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 *The Eleventh International Conference on Learning Representations*, 2022. 1, 2, 6, 7

[7] Rinon Gal, Moab Arar, Yuval Atzmon, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. Encoder-based domain tuning for fast personalization of text-to-image models, 2023. 1, 2, 6, 7

[8] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 5

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

[10] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 2

[11] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 5

[12] Minchul Kim, Anil K Jain, and Xiaoming Liu. Adaface: Quality adaptive margin for face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 7

[13] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. *arXiv preprint arXiv:2212.04488*, 2022. 1

[14] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1931–1941, 2023. 2, 6

[15] Dongxu Li, Junnan Li, and Steven CH Hoi. Blip-diffusion: Pre-trained subject representation for controllable text-to-image generation and editing. *arXiv preprint arXiv:2305.14720*, 2023. 1, 2, 6

[16] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023. 3

[17] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018. 5

[18] Jian Ma, Junhao Liang, Chen Chen, and Haonan Lu. Subject-diffusion: Open domain personalized text-to-image generation without test-time fine-tuning. *arXiv preprint arXiv:2307.11410*, 2023. 2, 6

[19] Alex 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. *arXiv preprint arXiv:2112.10741*, 2021. 2

[20] Xichen Pan, Li Dong, Shaohan Huang, Zhiliang Peng, Wenhu Chen, and Furu Wei. Kosmos-g: Generating images in context with multimodal large language models. *arXiv preprint arXiv:2310.02992*, 2023. 2, 6

[21] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 5

[22] Zeju Qiu, Weiyang Liu, Haiwen Feng, Yuxuan Xue, Yao Feng, Zhen Liu, Dan Zhang, Adrian Weller, and Bernhard Schölkopf. Controlling text-to-image diffusion by orthogonal finetuning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 1, 2

[23] 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

[24] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021. 1, 2

[25] 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. 2

[26] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1, 2, 5

[27] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510, 2023. 1, 2, 6, 7

[28] 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

[29] 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. 1, 2, 6

[30] 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. 2

[31] Ming Tao, Hao Tang, Songsong Wu, Nicu Sebe, Xiao-Yuan Jing, Fei Wu, and Bingkun Bao. Df-gan: Deep fusion generative adversarial networks for text-to-image synthesis, 2021. 2

[32] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 2, 4

[33] 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. 1, 2, 6

[34] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1316–1324, 2018. 2

[35] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2022. 1, 2

[36] Lili Yu, Bowen Shi, Ramakanth Pasunuru, Benjamin Muller, Olga Golovneva, Tianlu Wang, Arun Babu, Binh Tang, Brian Karrer, Shelly Sheynin, et al. Scaling autoregressive multi-modal models: Pretraining and instruction tuning. *arXiv preprint arXiv:2309.02591*, 2023. 1, 2

[37] Han Zhang, Jing Yu Koh, Jason Baldridge, Honglak Lee, and Yinfei Yang. Cross-modal contrastive learning for text-to-image generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 833–842, 2021. 2

[38] Yufan Zhou, Ruiyi Zhang, Changyou Chen, Chunyuan Li, Chris Tensmeyer, Tong Yu, Jiuxiang Gu, Jinhui Xu, and Tong Sun. Lafite: Towards language-free training for text-to-image generation. *arXiv preprint arXiv:2111.13792*, 2021. 2

[39] Yufan Zhou, Ruiyi Zhang, Tong Sun, and Jinhui Xu. Enhancing detail preservation for customized text-to-image generation: A regularization-free approach. *arXiv preprint arXiv:2305.13579*, 2023. 1, 2, 4, 6, 7