Yo'Chameleon: Personalized Vision and Language Generation

Thao Nguyen[1,2] Krishna Kumar Singh[2] Jing Shi[2] Trung Bui[2] Yong Jae Lee[1,¶] Yuheng Li[2,¶]
[1]University of Wisconsin–Madison [2]Adobe Research
https://thaoshibe.github.io/YoChameleon

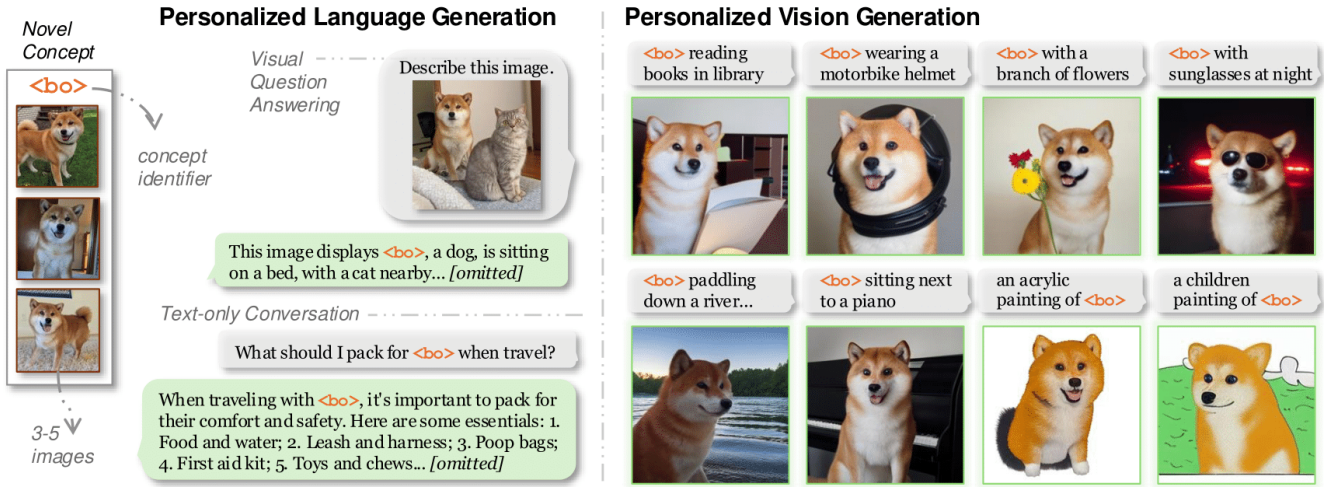

Figure 1. Using only 3-5 images of a novel concept/subject, we personalize Large Multimodal Models (e.g., Chameleon [1]) so that they retain their original capabilities while enabling tailored language and vision generation for the novel concept.

Abstract

Large Multimodal Models (e.g., GPT-4, Gemini, Chameleon) have evolved into powerful tools with millions of users. However, they remain generic models and lack personalized knowledge of specific user concepts. Previous work has explored personalization for text generation, yet it remains unclear how these methods can be adapted to new modalities, such as image generation. In this paper, we introduce Yo'Chameleon, the first attempt to study personalization for large multimodal models. Given 3-5 images of a particular concept, Yo'Chameleon leverages soft-prompt tuning to embed subject-specific information to (i) answer questions about the subject and (ii) recreate pixel-level details to produce images of the subject in new contexts. Yo'Chameleon is trained with (i) a self-prompting optimization mechanism to balance performance across multiple modalities, and (ii) a "soft-positive" image generation approach to enhance image quality in a few-shot setting. Our qualitative and quantitative analyses reveal that Yo'Chameleon can learn concepts more efficiently using fewer tokens and effectively encode visual attributes, outperforming prompting baselines.

¶ denotes equal advising

1. Introduction

Recent advances in Large Multimodal Models (LMMs) have transformed them into versatile, general-purpose AI assistants [1–6]. These models are increasingly being integrated into everyday applications, offering enhanced performance, improved efficiency, and support for multiple modes of communication. The ability to process both visual and textual information within a single system—as demonstrated by models like GPT-4o [2] and Gemini [4]—has streamlined user interactions and improved query comprehension.

Modern LMMs enable seamless two-way communication through both text and images. Users can input queries combining natural language and visual elements, and the models can respond with both textual descriptions and generated images. For instance, when asked to "Describe a Shiba Inu dog and generate a photo of it", these AI assistants can now provide comprehensive responses that combine detailed descriptions with visual representations.

While LMMs excel at general tasks, they face limitations when handling personalized queries. For example, if asked "Can you describe <bo> and generate a photo of <bo> reading books in library?" these models cannot provide accurate

responses without prior knowledge of the specific pet (e.g., a dog named <bo>). This highlights a crucial gap in their capabilities, as human interaction with the world is inherently personal — we engage with our own devices, pets, friends, and environments. To create more meaningful AI interactions, LMMs need mechanisms to learn, understand, and generate user-specific concepts, enabling them to evolve from general-purpose tools into personalized assistants (Fig. 1).

Personalization techniques have been extensively studied for LLMs [7–11] and image generation models [12–18], demonstrating significant progress in these individual domains. Recent works [19–21] have begun exploring personalization for vision-language models like LLaVA, which can take both image and text as inputs but only generate textual outputs. Despite this progress, the challenge of personalizing LMMs — which require both personalized text/image understanding and generation capabilities remains largely unexplored. In this paper, we identify two key challenges in extending personalization to these more comprehensive multimodal systems. To be specific, we focus on Large Multimodal Models that capable of understanding and generating images and text (e.g., Chameleon [1]).

The first challenge is catastrophic forgetting. Image generation tasks require granular information of new concepts, typically necessitating part/full model fine-tuning to achieve satisfactory results (e.g., [12]). However, LMMs store both visual and textual information, and our empirical studies show that fine-tuning for image generation (e.g., similar to [12]) causes the model to rapidly lose its world knowledge, compromising its functionality as a general-purpose AI assistant. Conversely, soft prompt learning [22, 23], which introduce learnable tokens to encode new concepts while keeping the model frozen, is effective for personalized image understanding tasks [19]. Although, our experiments reveal that soft prompt learning with only 3-5 user images fails to produce high-quality image generation results.

To address this challenge, we first identify that the limited number of training images is a key factor preventing soft prompt learning from matching full model fine-tuning's performance. Our study demonstrates that with $\sim$300 real images of a concept, soft prompt learning can achieve comparable performance to full-model fine-tuning while preserving the LMM's pretrained knowledge. However, since users typically only provide 3-5 images for a new concept (positive images), this is not a practical solution. Drawing from this analysis, we propose leveraging "soft-positive" images that share significant visual similarities with positive samples to enrich the training data. To effectively utilize these "soft positive", we implement an adaptive prompt length strategy where the prompt length varies based on the visual similarity between "soft-positive" and positive samples. For instance, when training the model to recognize a user's Shiba Inu, we utilize images of similar-looking Shiba Inu with adaptive prompt lengths to augment the limited training data. The more similar the "soft-positive" image is to the real positive images, we will use longer soft prompt to describe it.

The second challenge is the incompatibility between image generation and understanding capabilities within LMMs. Our experiments reveal that soft prompt optimized for one task cannot effectively transfer to the other. Specifically, when soft prompt trained for image understanding (text generation) are used to for image generation, the LMM produces irrelevant visual content. This phenomenon aligns with prior work [23–25] suggesting that optimized textual representations for one task might not be interpretable. Jointly training the soft prompt on both tasks might seem like an intuitive solution, however, our empirical results show this approach leads to suboptimal performance for both tasks (Fig. 4).

To enable effective personalization across both tasks, we propose using dual soft prompts — one specialized for text generation and another for image generation. This approach demonstrates superior performance compared to using a single set of prompts. Additionally, we introduce a self-prompting mechanism where the model first determines the task type (i.e., understanding or generation) before responding to queries, allowing it to better utilize the appropriate set of prefix tokens for each task.

In summary, our contributions are:

- We introduce the first attempt of personalization with Large Multimodal Models (i.e., models that capable of understanding and generating images and text).
- We present a novel "soft-positive" concept with dynamic prompt length to enhance the image generation quality.
- We propose a novel approach, in which use two set of soft prompts and self-prompting optimization techniques to balance the performance across the modality.

2. Related Work

Personalization for Large Multimodal Models. Large Language Models (LLMs) [26–29] and text-to-image models [30–34] have made tremendous progress recently, demonstrating extensive knowledge and excelling at text and image generation, respectively. Vision-language models [35–40] have emerged as a bridge between these modalities, capable of processing image-text inputs and generating textual outputs. Building upon this, researchers have developed unified Large Multimodal Models (LMMs) [1–5] that capable of understanding and generating both images and text.

However, these foundational models typically possess generic knowledge, making personalization a crucial and active research area. For LLMs, personalization often involves storing descriptions of personalized subjects as prompts in databases for reference during user interactions [7, 9, 41]. In image generation, researchers typically fine-tune either the entire model or specific components to incorporate visual knowledge [12, 14–16, 24, 42, 43]. Recent work by [19, 20]

proposes personalizing vision-language models through soft prompts to enable recognition and discussion of user-specific objects. Despite these advances, personalization of unified image/text generation models remains unexplored. Our work addresses this gap by investigating the challenges and potential solutions in this emerging area.

Parameter-Efficient Fine-Tuning (PEFT). Fine-tuning large pretrained models is often suboptimal due to computational costs and the risk of catastrophic forgetting. Consequently, numerous PEFT methods have been introduced to optimize a small subset of parameters (or introduce extra parameters) for downstream tasks [22, 44, 45]. In the domain of LLMs, prompt tuning (or soft-prompts) has emerged as an effective approach to adapt pretrained language models for various tasks, such as tool utilization [23] and text classification [22]. This approach has recently been extended to personalize vision-language models [19]. However, existing vision-language model approaches (e.g., [19]) primarily focus on text generation objectives. Our experiments reveal that naively extending their soft-prompting approach to encompass both text and image generation yields suboptimal results, as these tasks are not naturally complementary. To this end, we propose a self-prompting technique where the model first predicts the task type before generating the response. This approach effectively resolves the challenges of personalizing models with multi-modal outputs.

Hard negative image mining. Negative images have been widely used in the computer vision community [46–49]. In vision-language model personalization, [19] employs this technique to enhance personal object recognition. For image generation personalization, [12, 18] utilize negative examples as regularization to prevent model forgetting of class-level information. SuTI [50] and COTI [51] leverage negative images that are visually similar to personalized objects to establish a better initialization that facilitates easier adaptation to the target personalized object. However, unlike them which treats all negative images equally, we pursue a more nuanced approach. We propose an adaptive soft prompt length mechanism based on the visual similarity between negative images and positive examples. Specifically, we treat these negative images as "soft-positive" examples, allocate more prompt length to "soft-positive" images that exhibit higher visual similarity to the positive examples, allowing for more fine-grained representation learning.

3. Yo'Chameleon

Given a handful of images of a concept that we want to learn $I^1, I^2, ..., I^n$ (typically 3-5 images), our goal is to enable LMMs (i.e., Chameleon [1]) to embed the concept into a special token (e.g., <sks>) and to perform: (1) Personalized language generation (e.g., "Describe <sks>"; or given an image, "Where is <sks> in this image?"); and (2) Personalized vision generation (e.g., "Generate a photo of <sks>").

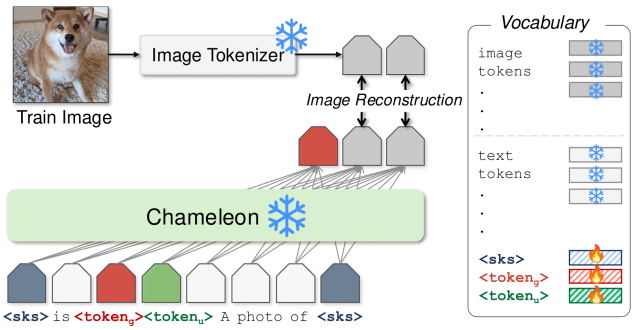

Figure 2. Image Reconstruction. The generated image, conditioned on a personalized prompt, is compared with the ground truth image to calculate the image reconstruction loss.

We first present how to present novel concept for as learnable prompt for LMMs (Sec. 3.1). Subsequently, we outline how to achieve personalized image generation (Sec. 3.2). Finally, we discuss how to unify both image generation and understanding capabilities within a single model (Sec. 3.3). As we chose Chameleon [1] as our base model, we named our method Yo'Chameleon, with *Yo* (short for *Your*) adopted from Yo'LLaVA's [19] personalization of LLaVA [37].

3.1. Representing a Concept as a Learnable Prompt

In image generation, prior work demonstrated that prompt tuning can effectively encode visual concepts for personalization [13, 24, 25]. This success has extended to vision-language models, where studies like [19, 23] show that prompt tuning can effectively encode novel visual attributes for text-only generation and image understanding. Building on this paradigm, we propose to represent personalized subjects as learnable prompts for LMMs:

"<sks> is <token$_1$><token$_2$>...<token$_k$>."

where <sks> is a learnable unique identifier for this new concept, and <token$_i$> are the learnable tokens which should encode visual information of that concept. This approach offers computational efficiency by only requiring updates to a small subset of parameters (i.e., tokens) while preserving the original core model weights.

In the context of Chameleon [1], a model that we choose to build upon in this work, an image is broken down into a series of image tokens, wrapped by special tokens which indicate the start-of-image <soi> and the end-of-image <eoi>. The training objective for both image and text remains consistent with standard autoregressive modeling, where the model learns to predict the next token in the sequence conditioned on the previous tokens. Thus, training for personalization follows an instruction-tuning paradigm, where the loss computation is specifically focused on the response portion of the instruction-response pairs. Given

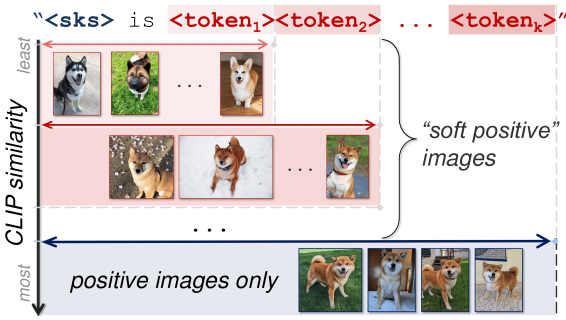

Figure 3. "Soft positive" images. Retrieved images are ranked according to their similarity to positive images using CLIP image similarity scores. Images that are more similar to the actual positive images are described with more latent tokens (i.e., more details).

the conversation pair $(\mathbf{X}_{\text{q}}^i, \mathbf{X}_{\text{a}}^i)$, where $\mathbf{X}_{\text{q}}^i$ is the question, and $\mathbf{X}_{\text{a}}^i$ is the corresponding answer, the masked language modeling loss for each conversation of length L by:

$$p(\mathbf{X}_{\text{a}}) = \prod_{j=1}^{L} p_{\boldsymbol{\theta}}(\boldsymbol{x}_j | \mathbf{X}_{\text{a},<j}), \tag{1}$$

where $\mathbf{X}_{\text{a},<j}$ are the instruction and answer tokens in all turns before the current prediction token $\boldsymbol{x}_j$. $\boldsymbol{\theta}$ is the trainable parameters, in this case, including the concept identifier <sks>, k latent tokens, and the final classifier head matrix W of the language model that associated with these tokens.

3.2. Personalizing Image Generation

A straightforward approach to personalization would be to directly train soft prompt on a limited set of n positive images. However, optimization with such limited data often yields suboptimal results. To this end, researchers have explored two primary approaches to expand the training samples: (1) data augmentation (e.g., background inpainting) to treat augmented images as additional positive training samples [17, 52, 53], and (2) leveraging hard-negative samples as an initialization, in which we first train on these negative images, then add an additional step to fine-tune the results with a limited number of positive examples to enhance personalization [50, 51, 54]. Empirical evidence suggests that utilizing real negative examples produces superior results compared to synthetic data augmentation approaches.

Motivated by this, in our approach, we retrieve hard-negative images, but unlike prior work [50, 51], we use them as "soft positive" images. The key insight is that hard negative images can share varying degrees of similar characteristics with the positive samples, and thus should contribute differently to the learning process. Taking the same example of an user's pet (a Shiba Inu) again, in this scenario, each negative image can function as a "soft positive" to varying extents. The similarity ranges from less to more similar negative images: for example, "A dog" (least similar), followed by "A dog with a yellow coat" (more similar), and so on.

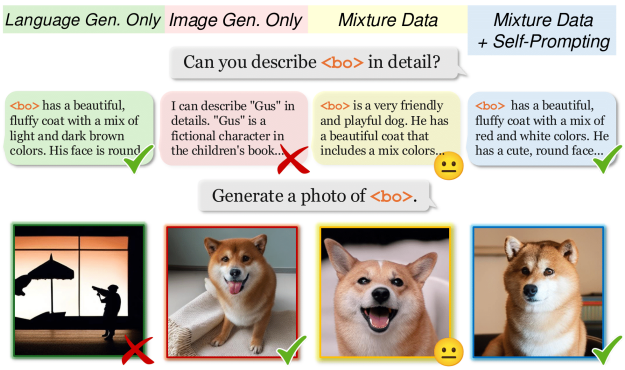

Figure 4. Optimized tokens for one task cannot effectively perform another, and simply training on a mixture of data yields suboptimal performance across tasks. We propose a self-prompting approach, where the model predicts which task to perform first, achieving the best of both worlds. (Input images are given in Fig. 1).

Specifically, given N retrieved negative images, we rank them from most to least similar to the average feature of the positive samples (based on CLIP image similarity score [55]); Then, we divide them into $k-1$ groups, each containing roughly $N/(k-1)$ images, according to their ranking. During training, we implement an adaptive token allocation strategy: negative images with higher similarity scores are assigned more learnable tokens, allowing for more detailed representation of relevant features. The complete set of tokens is reserved exclusively for the true positive images <sks>, ensuring that the model maintains the ability to distinguish the target concept while leveraging relevant features from similar soft positive examples. Fig. 3 illustrates this hierarchical token allocation strategy.

3.3. Personalizing Image and Language Generation

For text generation and image understanding tasks, we follow the approach in [19] to create a training dataset, which comprises two primary components: recognition data and question-answering data. For recognition data, we construct a balanced dataset containing the handful of positive examples alongside both 100 easy negative examples and 100 hard negative examples. For question-answering, we adopt the template in [19], which includes 10 questions (e.g., "What type of object is <sks>?"). We use GPT-4o [2] to generate answers for these questions using the positive images.

To achieve our goal of personalizing LMMs for both text and image generation capabilities, a straightforward approach would be to simultaneously train soft prompt using both the understanding data (recognition and question-answering) with image generation data (mentioned in the Sec. 3.2). However, our experiments reveal that naive joint training with mixed data leads to degraded performance compared to task-specific training.

As shown in Fig. 4, when tokens are trained exclusively for language generation, their application to image gener-

Figure 5. Self-prompting mechanism. When multiple tasks are presented, the model first predicts which information (latent tokens) should be used for this task first, and then performs the task.

ation tasks results in outputs that fail to capture the target concepts (1st column). Conversely, tokens optimized solely for image generation prove inadequate for text generation tasks (2nd column). Furthermore, we find that joint training yields a compromised solution that underperforms in both domains, suggesting that the model struggles to learn representations that effectively serve both objectives simultaneously (3rd column). This observation aligns with previous work [23–25] which suggests that tokens optimized for one specific task may lack semantic relevance for other tasks.

To overcome this limitation, we propose using two series of learnable tokens, each dedicated to a specific task. Specifically, the personalized concepts are represented as:

"<sks> is <g-tokens><u-tokens>."

where <g-tokens> and <u-tokens> represent k and h learnable tokens for image generation and understanding.

During training, to force the model to learn the distinct roles of the two sets of tokens, we create the training data such that the model first predicts which set of tokens (<g-tokens> vs. <u-tokens>) will be used for the task. We refer to this as "self-prompting" as the model needs to prompt itself first; Fig. 5 shows examples. For instance, for text understanding tasks (e.g., "What kind of object is <sks>?"), the target output first includes <u-tokens>, followed by the actual answer. The same technique is applied for image generation tasks. By requiring the model to first predict the appropriate token set, we force it to align the corresponding tokens with each task.

This is partially inspired by [23], where multiple tokens are used for calling different tools/tasks (e.g., mathematics, robot actions, etc). However, the key difference is that the task token in [23] is solely used for tool calling. In our approach, these tokens not only serve as task-mode calling tokens (i.e., for image or text generation) but also function as latent tokens, which contain the information needed to perform the task. This approach is flexible and could be adopted to other modalities as well (e.g., audio), and self-prompt tokens could be designed in a different way. We leave these possibilities for future work.

4. Experiments

Training. Unless otherwise stated, we use $n = 4$ input images per concept and $k = h = 16$ tokens to form a learnable prompt for each task, resulting in a total of 32 latent tokens for personalized concepts. For optimization, we employ AdamW [56] with a learning rate of 1×10^{-4}. Each concept is trained for 15 epochs, with the best checkpoint selected based on a composite score averaging recognition accuracy and generation quality (measured by CLIP image similarity with training examples). All experiments are conducted on an A100 GPUs with a batch size of 4.

We choose Chameleon [1] as our base model due to its simplicity in objective function (autoregressive for both text and image generation) and its unified LMM architecture. It is worth noting that our method generalizes to other LMMs, as it relies solely on token-level optimizations rather than model-specific architectures. While Chameleon was not originally published with image generation capabilities, we use the checkpoint from Anole [57], which recovered these capabilities through fine-tuning on an image generation dataset.

Baselines. The most straightforward baseline is using the base model (Chameleon [1]) with personalized text and image prompting. For personalized text prompting, we first obtain detailed captions of each concept by providing reference images to GPT-4o [2]. These captions are then human-audited, and appended to Chameleon's system prompt (e.g., "<sks> is a cinnamon-colored Shiba Inu with..."). For personalized image prompting, we append the reference image(s) (e.g., "This is a photo of <sks><image>"). Additionally, we compare our approach with GPT-4o [2], a proprietary multimodal chatbot, using the same two types of personalized text and image prompts.

Dataset. We utilize the Yo'LLaVA dataset [19], which consists of 40 subjects (10 humans and 30 non-human concepts). For negative images, we retrieve them from LAION-5B [58] based on the average CLIP Image Similarity [55] score between retrieved images and the mean feature representation of positive examples. After filtering NSFW content, we obtain approximately 1,000 negative images per concept. These images serve as "soft-positive" examples, with the top 100 most similar images designated as hard-examples for recognition. Additionally, we randomly sample 100 easy-negative examples from LAION-5B [58], which remain consistent across all concepts. In total, approximately 1,100 negative images are used for training each concept.

Metrics. To evaluate image understanding and text generation, we assess the model's recognition accuracy and question-answering ability. In total, there are 333 positive and 13,000 negative images for recognition. During testing, we present a photo and ask the model "Is <sks> in this photo?" The ground-truth answer is either "Yes" or "No". We use a weighted accuracy metric to balance the positive and negative classes, following the protocol in [19]. For question-answering, we provide multiple-choice questions (A or B) with 100 visual and 400 text-based questions.

For image generation, we produce 100 images per concept

Figure 6. Qualitative comparison with Chameleon [1] and GPT-4o [2] on image prompting. Yo'Chameleon (Ours) demonstrates more precise and personalized image generation.

using the prompt "A photo of <sks>" and compute the CLIP Image Similarity Score [55] between the generated images and positive examples. Additionally, in ablation studies, we further extend our analysis by reporting the Facial Similarity Score between the generated and positive images for 10 human faces (where applicable) using the off-the-shelf facial feature extractor ArcFace [49].

4.1. Personalized Language Generation

Tab. 1 shows the recognition and question-answering abilities of the evaluated models. The vanilla Chameleon model, lacking personalized concept information, performs essentially at random (0.474–0.500) on both tasks. With the addition of personalized text prompts, Chameleon's performance improves (0.523–0.727). Image prompting shows mixed results. When given a single example (∼1k column), it improves question-answering but does not enhance recognition accuracy. Providing multiple images (∼4k column) generally leads to a drop in performance on both tasks.

Notably, our approach outperforms Chameleon across all language generation tasks, with the recognition accuracy increases significantly (0.727 to 0.845). We achieve these improvements using fewer tokens (32) compared to the detailed text (∼64) or image (∼1k) prompting of Chameleon.

	Ours	Chameleon	Chameleon [1] + Prompt			GPT-4o [2] + Prompt		
Type	Learnable	$\emptyset$	Text	Image		Text	Image	
# tokens	32	0	∼64	∼1k	∼4k	∼64	∼1k	∼4k
Recognition Accuracy	**0.845**	0.500	0.727	0.361	0.327	0.841	0.902	0.915
Question Answering								
Visual	**0.604**	0.474	0.523	0.580	0.547	0.923	0.867	0.887
Text	**0.721**	0.405	0.716	0.573	0.231	0.798	0.982	0.978
Image Generation								
CLIP-I	**0.783**	0.425	0.566	0.487	0.589	0.636	0.657*	0.680*
Facial Sim	**0.212**	0.009	0.012	0.013	0.059	0.028	0.036*	0.063*

Table 1. Comparisons with Chameleon [1] and GPT-4o [2] using personalized image/text prompts. Our approach achieves significantly improved personalized image generation capabilities.

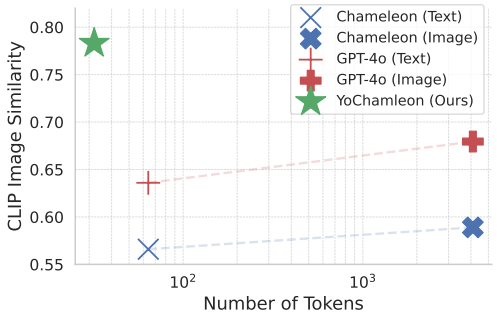

Figure 7. Number of Prompting Tokens vs. Personalized Image Generation Quality.

We also present GPT-4o's results for reference. GPT-4o performs well with both text and image prompts. For recognition tasks, our approach achieves comparable results (0.845 vs. 0.902) while requiring significantly fewer tokens (32 vs. ∼1k). For question answering, GPT-4o demonstrates better performance. This discrepancy can be attributed to two factors: (1) our use of a less powerful base model (i.e., Chameleon), and (2) the question data from [19] being relatively simple and generic (e.g., "What is the color of this subject?", "What material is this subject made of?"), where text descriptions as prompt are often sufficient. This explains why we achieve comparable results in recognition tasks, which require more fine-grained visual details. Therefore, we believe our approach offers value in terms of token efficiency while maintaining competitive performance.

4.2. Personalized Image Generation

Personalized image generation is generally a more challenging task than language generation. This is because recreating novel concepts with pixel-level detail is much more complex than simply answering questions based on existing references. In these cases, our learnable prompts with Chameleon clearly show advantages, outperforming all other methods by a significant margin. Specifically, Tab. 1 clearly shows that Yo'Chameleon achieves the highest CLIP Image Similarity Score (0.783), significantly surpassing the scores for Chameleon with either Image/Text Prompts (0.566–0.487).

When compared with GPT-4o [2], we find that GPT-4o generally captures high-level semantic details of personalized concepts reasonably well with both image and text prompts (i.e., 0.636–0.657). However, it struggles to capture the nuanced details of personalized subjects (see Fig. 6). This limitation is evident in the Facial Similarity Score, where we compare generated images to real images of 10 human faces. GPT-4o's generated images show low similarity to the actual person (e.g., 0.028–0.036), while Yo'Chameleon generated images more accurately capture facial details, making it far more suitable for personalization.

5. Ablation Studies

In all subsequent studies, we use 10 human faces from the Yo'LLaVA [19] dataset, and evaluate the Facial Similarity Score between the generated and positive images using ArcFace [49], which is specifically trained to distinguish nuanced differences between faces, providing a reliable metric for personalization generation. We ablate the: (1) importance of "soft positive" images, (2) number of "soft positive" images, (3) number of learnable tokens for the image generation task, and (4) different training strategies. As the focus of the first three experiments is on image generation, we train only with image generation data for these experiments, while the last one is trained with a mixture of data. The number of trainable tokens for each task are set to $k = 16$.

Importance of "soft positive" images. We compare our gradually added negative images with three main baselines: (1) Positive only (2–3 images), (2) Data augmentation via inpainting (1000 images), and (3) Soft Positive Images (Ours). For data augmentation, we first use SAM [59] to extract the foreground of the subjects, then randomly resize the subject to 30–70% of the 512x512 image size, and inpaint the background using Stable Diffusion-XL [60]. We generate 100 background captions with GPT-4o [2], which are then human-audited. Results are presented in Fig. 8 (first plot). As shown, while data augmentation improves the results compared to training with positive images only, it still falls short of the performance achieved by "soft positive" images.

Number of augmented/ "soft-positive" images. We next investigate the impact of varying the number of soft positive images (including both hard-negative and inpainting-augmented samples) used during training. Results demonstrate that using soft positive images consistently outperforms augmented images (Fig. 8, second plot). This superiority likely stems from the inherent limitations of segmentation and inpainting models for augmented data.

Number of learnable tokens. With the number of "soft-positive" images fixed at 1,000, we vary the number of trainable tokens k from 0 to 64. $k = 0$ means no latent tokens are trained for this task. Overall, increasing the number of train-

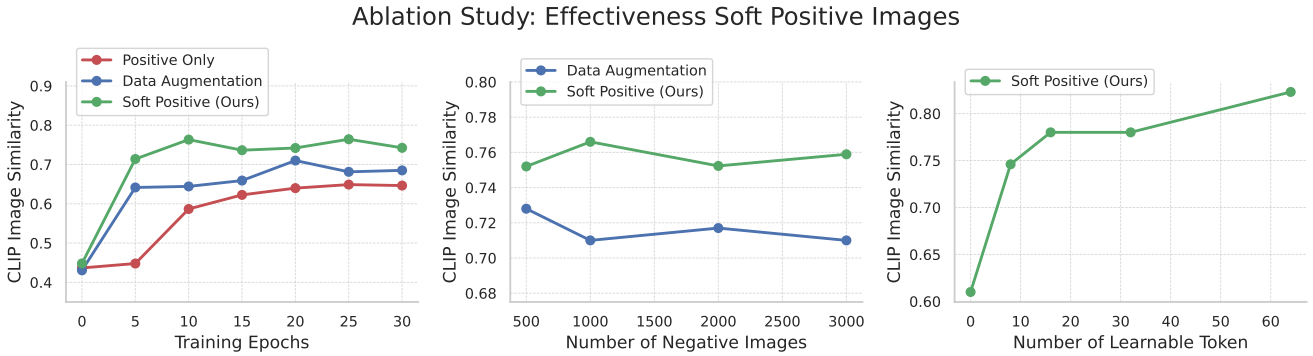

Figure 8. Ablation studies for image generation tasks. Overall, using "soft positive" and increasing the latent tokens boost performance.

able tokens improves the quality of image generation (Fig. 8, third plot). Quantitatively and qualitatively, we find that 16 tokens achieve reasonable results for most concepts, making it an effective and compact choice. For higher-quality image generation, one may increase the number of latent tokens. Empirically, we also note that, although the generated subjects appear visually similar, there is still room for improvement in the accuracy of generated human faces (e.g., current facial similarity is 0.212, while the threshold for a good human facial similarity would be 0.4 or higher).

Different training strategies. Our approach employs separate tokens for each task (understanding and generation) with self-prompting prior to prediction. We next validate this design. We begin by training the same set of tokens for two different tasks (Shared learnable prompt, in Table 2). Results indicate that using a single set of tokens and training them specifically for each task achieves optimal performance for that particular task. For example, (1) Language data only: achieves the best recognition accuracy but fails to generate images. For (2) Image generation data only, we explore three variations: (2.1) Positive only: training exclusively with positive images; (2.2) Negative + Finetune: treating all negative samples equally during training across all tokens, followed by fine-tuning with positive images; and (3) Soft-positive (Ours): gradually incorporating more tokens as soft-positive images become more similar to true positives. Our results demonstrate that the soft-positive strategy achieves the best results. (3) Training on mixture data yields intermediate scores across both tasks, suggesting that generation and understanding tasks may not trivially be complementary.

The above findings suggest two key insights: (1) our proposed approach of adaptively setting soft-positive images with varying token lengths is more effective for generation tasks, and (2) shared learnable prompts are suboptimal when handling multiple tasks simultaneously, necessitating separate token sets for different tasks.

For the separate learnable prompts approach, we also ablate different strategies: (1) Concatenate: training two sets of tokens independently for each task and concatenating them at test time; (2) Concatenate + Fine-tune: extending strategy (1) with an additional fine-tuning step post-concatenation; and (3) Self-prompting (Ours): our proposed mechanism that first predicts prompt tokens before making the actual prediction. Results demonstrate that our self-prompting approach achieves optimal performance, matching the effectiveness of task-specific token training.

	Acc. (↑)	CLIP-I (↑)	Face Sim (↑)
Shared learnable prompt *(16 tokens in total)*			
Language data only	**0.784**	0.120	0.032
Image generation data only			
Positive only	0.104	0.678	0.188
Negative + Finetune	0.004	0.711	0.193
Soft positive	0.108	<u>0.742</u>	**0.225**
Mixture data	0.564	0.687	0.193
Mixture data (32 tokens)	0.562	0.684	0.194
Separated learnable prompt *(32 tokens in total)*			
Concatenate	0.502	0.615	0.156
Concatenate + Finetune	0.251	0.648	0.189
Self-Prompting (Ours)	<u>0.747</u>	**0.761**	<u>0.224</u>

Table 2. Ablation studies on different training strategy. We use recognition accuracy to evaluate understanding capability, and CLIP and Face similarities for image generation quality.

6. Conclusion

We presented the first attempt to personalize Large Multimodal Models (LMMs) for both vision and language understanding and generation tasks. We introduced a dual prefix prompt architecture with a self-prompting mechanism to achieve strong performance in both understanding and generation capabilities. We also proposed a novel soft-positive training strategy that leverages hard-negative samples to enhance generation quality in spite of limited user data. Experimental results demonstrated that our approach successfully maintains the model's general knowledge while enabling effective personalization across both tasks, representing a significant step toward making LMMs more personally relevant for real-world applications.

Acknowledgment

This work was supported in part by NSF IIS2404180, Adobe Data Science award, Microsoft Accelerate Foundation Models Research Program, and Institute of Information & communications Technology Planning & Evaluation (IITP) grants funded by the Korea government (MSIT) (No. 2022-0-00871, Development of AI Autonomy and Knowledge Enhancement for AI Agent Collaboration) and (No. RS-2022-00187238, Development of Large Korean Language Model Technology for Efficient Pre-training).

References

[1] Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. In *arXiv*, 2024. 1, 2, 3, 5, 6, 7

[2] OpenAI. Gpt-4o system card. In *arXiv*, 2024. 1, 4, 5, 6, 7

[3] Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. In *arXiv*, 2024.

[4] Gemini Team. Gemini: A family of highly capable multimodal models. In *arXiv*, 2024. 1

[5] Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model. In *arXiv*, 2024. 2

[6] Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, et al. Janus: Decoupling visual encoding for unified multimodal understanding and generation. In *arXiv*, 2024. 1

[7] Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. LaMP: When large language models meet personalization. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *ACL*, 2024. 2

[8] Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. Personalizing dialogue agents: I have a dog, do you have pets too? In Iryna Gurevych and Yusuke Miyao, editors, *ACL*, 2018.

[9] Shuai Liu, Hyundong Cho, Marjorie Freedman, Xuezhe Ma, and Jonathan May. RECAP: Retrieval-enhanced context-aware prefix encoder for personalized dialogue response generation. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *ACL*, 2023. 2

[10] Matej Gjurković, Mladen Karan, Iva Vukojević, Mihaela Bošnjak, and Jan Snajder. PANDORA talks: Personality and demographics on Reddit. In Lun-Wei Ku and Cheng-Te Li, editors, *SocialNLP*, 2021.

[11] Joel Jang, Seungone Kim, Bill Yuchen Lin, Yizhong Wang, Jack Hessel, Luke Zettlemoyer, Hannaneh Hajishirzi, Yejin Choi, and Prithviraj Ammanabrolu. Personalized soups: Personalized large language model alignment via post-hoc parameter merging. In *arXiv*, 2023. 2

[12] 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 *CVPR*, 2023. 2, 3

[13] 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. In *arXiv*, 2022. 3

[14] Jing Shi, Wei Xiong, Zhe Lin, and Hyun Joon Jung. Instantbooth: Personalized text-to-image generation without test-time finetuning. In *CVPR*, 2024. 2

[15] Zecheng He, Bo Sun, Felix Juefei-Xu, Haoyu Ma, Ankit Ramchandani, Vincent Cheung, Siddharth Shah, Anmol Kalia, Harihar Subramanyam, Alireza Zareian, Li Chen, Ankit Jain, Ning Zhang, Peizhao Zhang, Roshan Sumbaly, Peter Vajda, and Animesh Sinha. Imagine yourself: Tuning-free personalized image generation. In *arXiv*, 2024.

[16] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. In *arXiv*, 2023. 2

[17] Jian Ma, Junhao Liang, Chen Chen, and Haonan Lu. Subject-diffusion:open domain personalized text-to-image generation without test-time fine-tuning. In *arXiv*, 2024. 4

[18] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. In *CVPR*, 2023. 2, 3

[19] Thao Nguyen, Haotian Liu, Yuheng Li, Mu Cai, Utkarsh Ojha, and Yong Jae Lee. Yo'llava: Your personalized language and vision assistant. In *NeurIPS*, 2024. 2, 3, 4, 5, 7

[20] Yuval Alaluf, Elad Richardson, Sergey Tulyakov, Kfir Aberman, and Daniel Cohen-Or. Myvlm: Personalizing vlms for user-specific queries. In *ECCV*, 2024. 2

[21] Personalized Captioning. Remember, retrieve and generate: Understanding infinite visual concepts as your personalized assistant. 2

[22] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *EMNLP*, 2021. 2, 3

[23] Shibo Hao, Tianyang Liu, Zhen Wang, and Zhiting Hu. ToolkenGPT: Augmenting frozen language models with massive tools via tool embeddings. In *NeurIPS*, 2023. 2, 3, 5

[24] Thao Nguyen, Yuheng Li, Utkarsh Ojha, and Yong Jae Lee. Visual instruction inversion: Image editing via image prompting. In *NeurIPS*, 2023. 2, 3

[25] Yuxin Wen, Neel Jain, John Kirchenbauer, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery. In *NeurIPS*, 2023. 2, 3, 5

[26] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. In *arXiv*, 2023. 2

[27] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin

Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. In *arXiv*, 2023.

[28] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. In *arXiv*, 2023.

[29] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *arXiv*, 2020. 2

[30] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *arXiv*, 2021. 2

[31] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, 2021.

[32] Zhan Shi, Xu Zhou, Xipeng Qiu, and Xiaodan Zhu. Improving image captioning with better use of captions. In *arXiv*, 2020.

[33] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *arXiv*, 2022.

[34] 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 *CVPR*, 2023. 2

[35] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge. In *arXiv*, 2024. 2

[36] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *CVPR*, 2023.

[37] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 3

[38] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *arXiv*, 2023.

[39] Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. In *arXiv*, 2024.

[40] Zhangwei Gao, Zhe Chen, Erfei Cui, Yiming Ren, Weiyun Wang, Jinguo Zhu, Hao Tian, Shenglong Ye, Junjun He, Xizhou Zhu, et al. Mini-internvl: A flexible-transfer pocket multimodal model with 5% parameters and 90% performance. In *arXiv*, 2024. 2

[41] Hongjin Qian, Zhicheng Dou, Yutao Zhu, Yueyuan Ma, and Ji-Rong Wen. Learning implicit user profile for personalized retrieval-based chatbot. In *CIKM*. Association for Computing Machinery, 2021. 2

[42] Kuan Heng Lin, Sicheng Mo, Ben Klingher, Fangzhou Mu, and Bolei Zhou. Ctrl-x: Controlling structure and appearance for text-to-image generation without guidance. In *Advances in Neural Information Processing Systems*, 2024. 2

[43] 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*, 2023. 2

[44] 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. In *arXiv*, 2021. 3

[45] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *ACL*, 2021. 3

[46] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *arXiv*, 2016. 3

[47] Hong Xuan, Abby Stylianou, Xiaotong Liu, and Robert Pless. Hard negative examples are hard, but useful. In *arXiv*, 2021.

[48] Shaohua Wan, Zhijun Chen, Tao Zhang, Bo Zhang, and Kong kat Wong. Bootstrapping face detection with hard negative examples. In *arXiv*, 2016.

[49] Jiankang Deng, Jia Guo, Xue Niannan, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, 2019. 3, 6, 7

[50] Wenhu Chen, Hexiang Hu, Yandong Li, Nataniel Ruiz, Xuhui Jia, Ming-Wei Chang, and William W Cohen. Subject-driven text-to-image generation via apprenticeship learning. In *arXiv*, 2023. 3, 4

[51] Jianan Yang, Haobo Wang, Yanming Zhang, Ruixuan Xiao, Sai Wu, Gang Chen, and Junbo Zhao. Controllable textual inversion for personalized text-to-image generation. In *arXiv*, 2023. 3, 4

[52] Yuheng Li, Haotian Liu, Yangming Wen, and Yong Jae Lee. Generate anything anywhere in any scene. In *arXiv*, 2023. 4

[53] Ligong Han, Yinxiao Li, Han Zhang, Peyman Milanfar, Dimitris Metaxas, and Feng Yang. Svdiff: Compact parameter space for diffusion fine-tuning. In *arXiv*, 2023. 4

[54] Ge Yuan, Xiaodong Cun, Yong Zhang, Maomao Li, Chenyang Qi, Xintao Wang, Ying Shan, and Huicheng Zheng. Inserting anybody in diffusion models via celeb basis. 2023. 4

[55] 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 *arXiv*, 2021. 4, 5, 6

[56] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 5

[57] Ethan Chern, Jiadi Su, Yan Ma, and Pengfei Liu. Anole: An open, autoregressive, native large multimodal models for interleaved image-text generation. In *arXiv*, 2024. 5

[58] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models. In *arXiv*, 2022. 5

[59] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. In *arXiv*, 2023. 7

[60] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *arXiv*, 2023. 7