

K-LoRA: Unlocking Training-Free Fusion of Any Subject and Style LoRAs

Ziheng Ouyang Zhen Li[†] Qibin Hou
 VCIP, School of Computer Science, Nankai University
 {zihengouyang666, zhenli1031}@gmail.com

Project page: <https://k-lora.github.io/K-LoRA.io/>

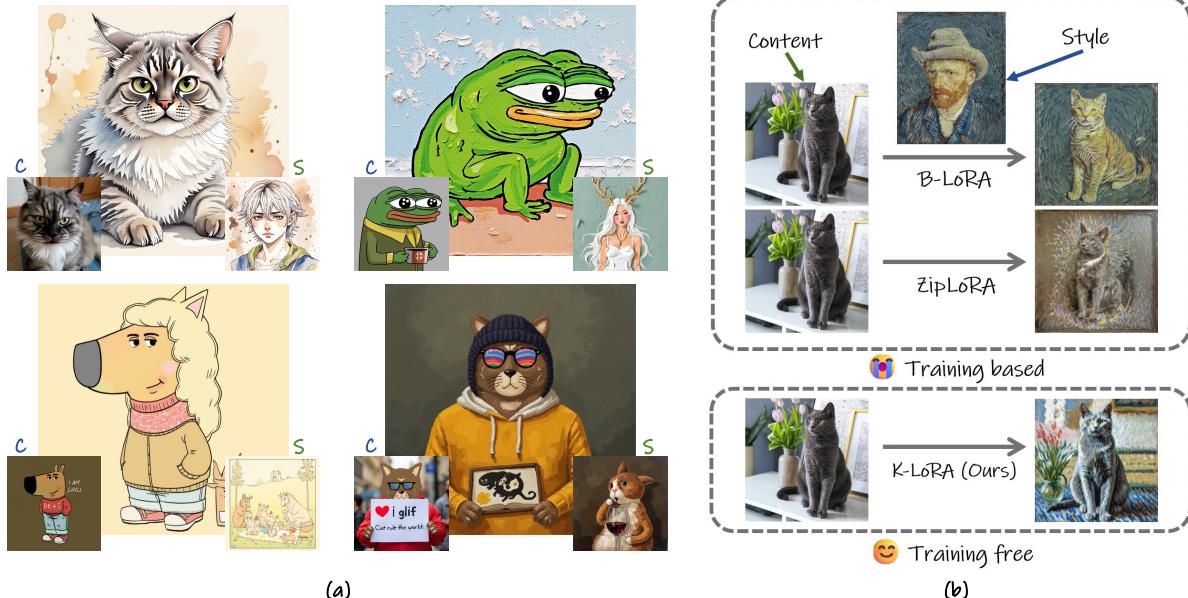


Figure 1. **Visual illustrations.** (a) demonstrates the superior generative performance of our proposed K-LoRA using FLUX [3], where the object reference is presented on the left, the style reference on the right, and the generated image is shown in the center. In contrast, (b) compares our method with existing state-of-the-art methods, B-LoRA [8] and ZipLoRA [26], which tend to lose style or content information due to alterations in the original weight matrix or underutilization of the network structure. Our approach enhances the information captured by each LoRA matrix, thereby achieving superior fusion effects *without requiring additional training*.

Abstract

Recent studies have explored the combination of different LoRAs to jointly generate learned style and content. However, existing methods either fail to effectively preserve both the original subject and style simultaneously or require additional training. In this paper, we argue that the intrinsic properties of LoRA can effectively guide diffusion models in merging learned subject and style. Based on this insight, we propose K-LoRA, a simple yet effective training-free LoRA fusion approach. In each attention layer, K-LoRA compares the Top-K elements in each LoRA to be fused, determining which LoRA to select for optimal fusion. This selection mechanism ensures that the most representative features of both subject and style are retained during the fusion process, effectively balancing their contributions. Experiments

demonstrate that K-LoRA can effectively integrate the subject and style information learned by the original LoRAs, outperforming state-of-the-art training-based approaches in both qualitative and quantitative results.

1. Introduction

Personalization and stylization are two well-established tasks in computer vision and have been active research fields for many years [4, 6, 9, 13, 17, 24, 28, 34, 37, 42]. The primary challenge in these tasks is preserving distinct content or modifying the style of an image, typically guided by textual or visual inputs. In this context, “content” refers

[†] Corresponding authors.

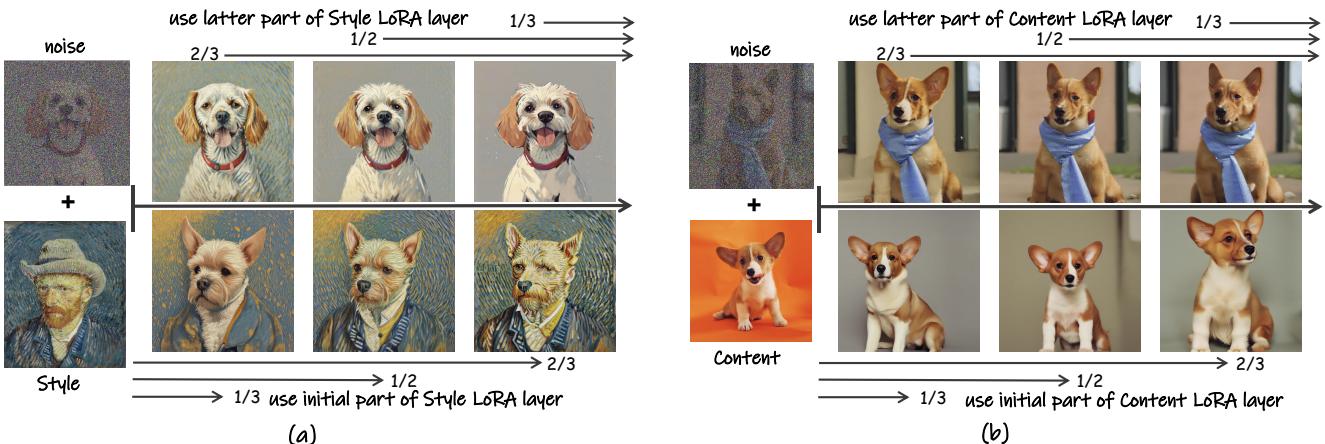


Figure 2. **Visual results of findings.** (a) Results fine-tuned using only content LoRA. (b) Results only using style LoRA. In these experiments, we test the differences between adding the LoRA layers in the initial and latter timesteps.

to the objects and structures within the image, while “style” encompasses visual attributes such as color, texture, and patterns. Manipulating image style is particularly challenging due to the subjective nature of style definitions and the strong interdependence between style and content, which complicates effective decoupling of these elements.

Recent techniques, such as LoRA [12], have gained more and more attention for their ability to achieve efficient fine-tuning in image synthesis. Although styles and objects are trained separately, LoRA provides an effective solution to the problems of decoupling style and content in image generation tasks, which excels in controlling style transfer by training style features independently from the content features. With the growing popularity of personalized applications utilizing LoRA, numerous efforts have been put to fuse objects and styles by merging LoRA weights [25]. These approaches aim to allow users to adjust each LoRA’s contribution ratio through variable coefficients. There are also methods, such as ZipLoRA [26], attempting to train a fusion ratio vector to balance different LoRAs. More recently, some approaches propose the periodic integration of LoRA into models [41]. Additionally, the B-LoRA [8] technique fine-tunes only two attention modules to facilitate style transfer.

In our experiments with these methods, we identify two key issues, as shown in Fig. 1(b). First, *style details often lose in the generated images, and the object characteristics are inconsistently maintained*. Second, *manual tuning of certain hyperparameters and seeds is required, or additional training is necessary*. For the first issue, we conduct extensive experiments and observe that merging the attention layers of two LoRAs at the element level could lead to a smoothing of style details and textures, or even the loss of object characteristics. Given that element-level merging may lead to suboptimal results, we conduct experiments by selectively removing certain elements to keep good per-

formance. For the second issue, inspired by the core ideas in recent studies [20, 29, 35], we incorporate the attention layers of LoRA into the model according to diffusion time steps to assess their effects on performance. Through this approach, we derive key conclusions. (i) Only a restricted number of diffusion prediction steps are sufficient to retain the original effect as illustrated in Fig. 2. (ii) When applying LoRA, the initial diffusion steps are responsible for reconstructing the object and capturing larger texture details, while the subsequent steps focus on enhancing and refining the finer details of the object and the texture in style.

Based on these findings, we propose K-LoRA, which simultaneously addresses both issues identified in our experiments, as illustrated in Fig. 1(a), leveraging our first insight by incorporating a Top-K selection process within each forward pass of the attention layers to identify the most suitable attention components at each position. Additionally, we apply a scaling factor during the selection process, utilizing our second insight to emphasize the distinct roles that style and content play throughout the diffusion process.

Our method can effectively resolve the aforementioned issues, ensuring that the merged LoRA captures both subject and stylistic features when faced with challenging content and style combinations. This results in stable generative outputs and significantly enhances the performance of merged LoRAs. Furthermore, our approach is user-friendly, as it requires no additional training. We summarize our contributions as follows:

- We propose K-LoRA, a simple yet effective optimization technique that seamlessly merges content and style LoRAs, enabling the generation of any desired style for any theme while preserving intricate details.
- Our method is user-friendly, eliminating the need for re-training and directly applicable to existing LoRA weights. It demonstrates superior performance across diverse image stylization tasks, surpassing existing methods.

2. Related Work

Diffusion models for customization. In the realm of diffusion models [23] for customized tasks, customization refers to the process by which the model learns to interpret new definitions provided by the user. Techniques such as Textual Inversion [1, 29, 40], DreamBooth [24], and Custom Diffusion [16] enable the model to capture target concepts with only a limited number of images through token-based optimization. Specifically, Textual Inversion fine-tunes embeddings to reconstruct the target, DreamBooth uses less common class-specific terms to expand object categories, and Custom Diffusion focuses on fine-tuning the cross-attention layers within the diffusion model to learn new concepts. Additionally, there are methods that do not require training when inferring [2, 27, 31, 32], but their approaches to utilize pre-trained modules may perform sub-optimally for certain specialized tasks. LoRA [12] and its variants [11, 15, 22, 39, 43, 43, 44] are well-known for their ability to fine-tune large models and deliver high-quality results, making them a good choice for practitioners.

LoRA combination in image generation. In the field of image generation, research on LoRA combinations has primarily been advanced in two directions, including the integration of multiple objects and the fusion of contents with styles. For object integration, studies have mainly focused on enabling models to integrate diverse object concepts encapsulated within multiple LoRAs [7, 10, 14, 18, 36]. By fine-tuning the subject LoRAs, these models can assimilate various new concepts and manage object layouts through masking techniques. Regarding content-style fusion, several works, such as MergingLoRA [25], Mixture-of-Subspaces [30], and ZipLoRA [26], have proposed approaches involving hyperparameter tuning and learning fusion matrices to merge pre-trained LoRA weight layers. However, these methods may face challenges, such as concept dilution, blurring of fine details, and specific training requirements. Recently, B-LoRA [8] has identified distinct roles for attention modules in the generative process, thereby achieving object-style decoupling within LoRA by training only two core attention modules. Additionally, LoRA Composition [41] uses a cyclic update of the model’s LoRA modules to allow multiple LoRAs to collaboratively guide the model, allowing a variety of cross-concept fusion. Despite these advancements, existing methods continue to face challenges, including insufficient control precision, loss of object style, and high training requirements.

3. Method

3.1. Preliminaries

LoRA is an effective method initially designed to adapt large-scale language models. The core premise of LoRA is

that, when fine-tuning a large model and comparing it with a baseline model, the parameter update matrix $\Delta W \in \mathbb{R}^{m \times n}$ is typically found to contain small or near-zero elements, exhibiting a low-rank structure. This property allows ΔW to be factorized into two low-rank matrices, $B \in \mathbb{R}^{m \times r}$ and $A \in \mathbb{R}^{r \times n}$, where r represents the intrinsic rank of ΔW , and it is assumed that $r \ll \min(m, n)$. This characteristic enables us to freeze the base weight W_0 and train only the matrices B and A to replace ΔW , thereby achieving an efficient parameterization in the form $\Delta W = BA$. Finally, ΔW is added to the base weight in the original model to perform fine-tuning. The updated weights can be expressed as $W_0 + \Delta W$.

In our work, we adopt the same notation as used in ZipLoRA [26]. Let D be a base diffusion model, and W_0 denote the pre-trained weights that need to be updated with LoRA layer. The base model D can be adapted to a specific concept simply by adding an additional trained LoRA weight set ΔW_x to the model weights, resulting in $D' = W_0 + \Delta W_x$. Given two independently trained LoRA weight sets, ΔW_c and ΔW_s , associated with the base model D , our objective is to fully leverage the weights of both LoRA sets and enable their effective fusion. To achieve this, we propose a method, called K-LoRA, to seamlessly combine the two LoRA weight sets, expressed as

$$\Delta W_x = K(\Delta W_c, \Delta W_s),$$

where K denotes our method, which can efficiently integrate the contributions of the content LoRA and style LoRA.

In what follows, we will explain the proposed approach in detail. Our approach is based on two findings. (i) In the diffusion steps, applying LoRA to only a subset of layers per step can achieve comparable effects comparing to applying LoRA to all layers; (ii) Using the subject LoRA in earlier diffusion steps tends to generate better subject information, while using the style LoRA in later steps is more effective for generating style and details without affecting the construction of the content.

3.2. K-LoRA

It has been pointed out in [26] that using a smaller set of key elements when finetuning with LoRA can achieve the same generative results as the original approach. However, the authors did not provide relevant experiments to explain this in the field of image generation. We first attempt to leverage this method by assigning zeros to the elements whose values are small following a similar approach to that of Magmax [19]. We found that the results obtained by modifying the elements of the matrix in this way are similar to the ones produced by [26, 30] because the model does not correctly interpret the concepts it has previously learned, resulting in a suboptimal quality of image generation.

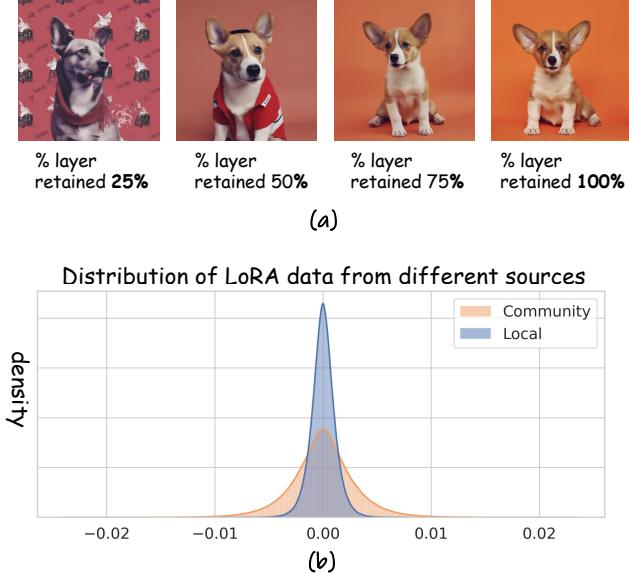


Figure 3. Experimental visualization results. (a) Generated images by randomly loading a portion of the LoRA attention layers according to a certain ratio. (b) Visualization of LoRA data distribution from different sources: one trained locally and the other one downloaded from a community repository.

Given the complexities and limitations involved in directly modifying attention elements, a question rises: Can we exploit the sparse characteristics of the LoRA matrix during the denoising process? The aim is to find an alternative method that can identify a good weight selection method and precise LoRA positioning for each step or layer without modifying the original LoRA weights. Based on Multi-LoRA Composition [41], we randomly apply the content LoRA attention layer to the diffusion steps, affecting the object using $x\%$ of the attention layers to observe the generated outcomes. As shown in Fig. 3(a), we found that when $x > 50$, the results are virtually indistinguishable from those of the original model. However, when $x < 25$, the ability of the model to maintain the original personalized concepts significantly diminished.

Inspired by recent studies [20, 29, 35], we further extend the aforementioned experiments in Fig. 2 and found that applying the style LoRA in earlier timesteps has a significant impact on the reconstruction of the original object, whereas applying it in later timesteps preserves the style information without affecting the original object. Additionally, we observe that for content LoRA, applying it in earlier timesteps yields significantly better results than applying it in later timesteps.

The above analysis motivates us to achieve the merging of generated objects and styles by adaptively selecting the LoRA module for each attention layer. According to finding (i), the selection strategy should preserve the overall object

and style information. Furthermore, according to finding (ii), the generation process should be achieved by arranging the object and style components appropriately. That is in the early diffusion steps, the model should focus more on object reconstruction while introducing style textures, and in later steps, it is better to refine the style with subtle object details. Therefore, we present K-LoRA, as shown in Fig. 4, which can adaptively select the appropriate LoRA layer for merging learned subject and style.

First, we take the absolute value of each element in the LoRA Layer to determine whether a particular value plays a significant role in the generation process,

$$\Delta W'_c = |\Delta W_c|, \quad (1)$$

$$\Delta W'_s = |\Delta W_s|, \quad (2)$$

where W_c and W_s denote the content and style LoRA weights, respectively. Because a small subset of dominant elements can achieve the original generation effect while the data distribution (see Fig. 3(b)) shows that smaller elements occupy a large proportion of the positions, which will influence the selection of the important elements, we use a smaller number of the largest elements to represent the importance of each layer.

Specifically, we select the top K elements with the highest values from $\Delta W'_c$ and $\Delta W'_s$, respectively. By accumulating the Top-K elements, we assess the importance of the two matrices at a given attention layer:

$$S_c = \sum_{i \in \text{Top-K}(\Delta W'_c)} \Delta W'_{c,i}, \quad (3)$$

$$S_s = \sum_{j \in \text{Top-K}(\Delta W'_s)} \Delta W'_{s,j}, \quad (4)$$

where Top-K returns the indices of the largest K values. For the selection of K , we note that the rank number in the LoRA training process reflects, to some extent, the amount of information contained within the matrix. Thus, our choice of K is aligned with the rank of each LoRA:

$$K = r_c \cdot r_s, \quad (5)$$

where r_c and r_s represent the ranks of the content and style LoRA layers, respectively. This formulation allows us to determine the appropriate weights within an attention layer by comparing the two sums

$$C(S_c, S_s) = \begin{cases} \Delta W_c, & \text{if } S_c \geq S_s \\ \Delta W_s, & \text{otherwise} \end{cases} \quad (6)$$

To more effectively leverage finding (ii) and allow both object and style to play their respective roles at different stages while ensuring a smooth transition from object-focused to style-focused representation, we introduce a

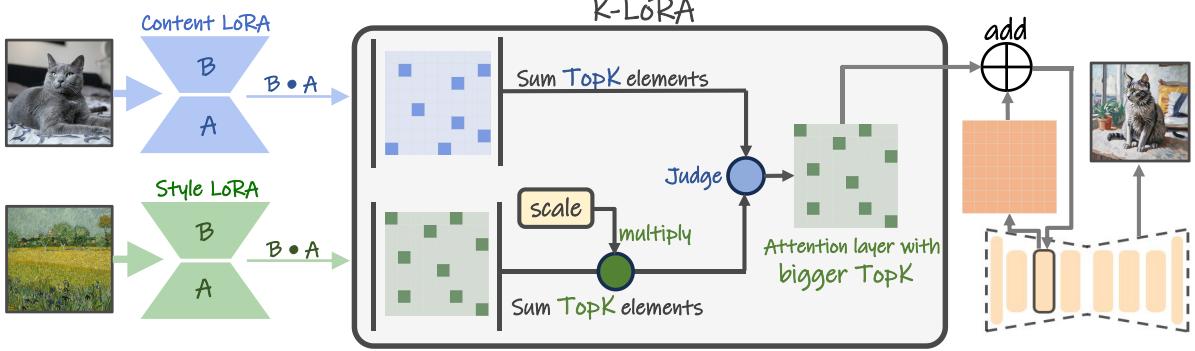


Figure 4. **Overview of the proposed K-LoRA.** We propose K-LoRA, which utilizes the Top-K function to select the important LoRA weights in each forward layer based on the sum of matrix elements. This enables us to preserve both stylistic details and object features.

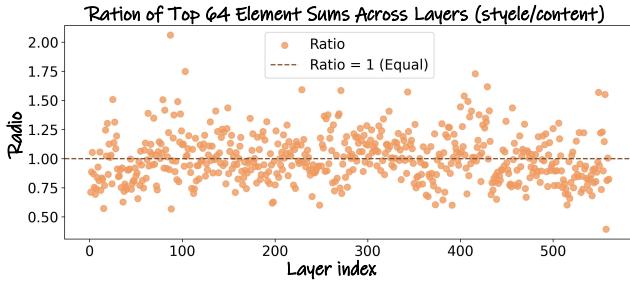


Figure 5. **Radio visualization.** This image reflects the ratio results after summing the Top-K elements, where the ratio differences at each corresponding position are quite significant.

scaling factor S . This factor S is directly applied to the Top-K selection process, enhancing object content in the early stages of generation and gradually emphasizing style in the later stages

$$S = \alpha \cdot \frac{t_{now}}{t_{all}} + \beta, \quad (7)$$

where t_{now} denotes the current step in the backward denoising process, t_{all} is the total step number, and α, β are hyperparameters.

To avoid excessive weight disparities when using community LoRA models from different sources, which may make Top-K selection ineffective for attention allocation, we introduce a new factor γ to balance the two weights

$$S' = \gamma \cdot S. \quad (8)$$

Initially, we compute the sum of the absolute values of the elements within each layer l , and then accumulate these sums layer by layer to calculate γ

$$\gamma = \frac{\sum_l \sum_i \Delta W'_{cl,i}}{\sum_l \sum_j \Delta W'_{sl,j}}. \quad (9)$$

The introduction of γ addresses the significant numerical discrepancy between the elements in the two LoRA components, as shown in Fig. 3(b). This adjustment highlights the useful components within the LoRA layers. With γ , the proportional relationship between the content and style LoRA weights in each layer is shown in Fig. 5. It can be observed that, in each forward layer where LoRA is applied, there is a significant difference in the proportions of the dominant components' sums. This highlights the significance of the distinct LoRA weights within each layer, providing a solid basis for selection.

We then apply S' to the style LoRA and update S_s

$$S'_s = S_s \cdot S'. \quad (10)$$

By introducing S' , we can strengthen the influence of content during the earlier time steps, while amplifying the dominance of style in the later steps. This adjustment can effectively take advantage of finding (ii), optimizing the selection of both object and style to maximize their contributions in the image generation process. The final LoRA weights can be attained by computing $C(S_c, S'_s)$. To clarify, we present the pseudo code in Algorithm 1.

To better explain the weight selection process, we show the selection proportions in Fig. 6, where the object and style seamlessly interpenetrate and blend with each other. The first portion primarily focuses on the object, with a small amount of style incorporated, while the latter portion predominantly emphasizes style, retaining a subtle presence of the object which further substantiates our key findings.

4. Experiments

4.1. Experiment Setup

Datasets. Following the convention of ZipLoRA [26], for the LoRA obtained through local training, we choose a diverse set of content images from the DreamBooth [24] dataset, each containing 4-5 images of a given subject. For

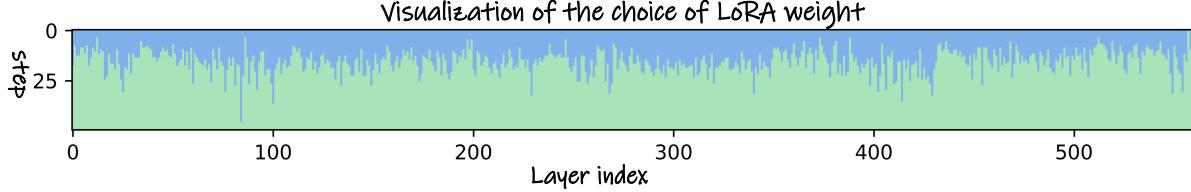


Figure 6. LoRA selection during the generation process. This figure illustrates the selection within each forward layer. The vertical axis represents the total 50 diffusion steps, while the horizontal axis indicates the number of LoRA layers. The color at each position denotes the selected layer. Blue bars correspond to objects, and green bars correspond to styles.

Algorithm 1 Pseudocode in a PyTorch-like style.

```

# timestep: current timestep
# content_lora_weight, style_lora_weight: input
# weights
# alpha, beta, gamma: scaling factors
# all_timesteps: total timesteps

# Set k based on rank
k = rank * rank

# Sum of TopK content values
abs_content_matrix = abs(content_lora_weight)
topk_content_values = topk(abs_content_matrix.fl(), k)
sum_topk_content = sum(topk_content_values)

# Sum of TopK style values
abs_style_matrix = abs(style_lora_weight)
topk_style_values = topk(abs_style_matrix.fl(), k)
sum_topk_style = sum(topk_style_values)

# Compute and apply scaling factor
scale = alpha + beta * timestep / all_timesteps
scale = scale * gamma
sum_topk_style *= scale

# Compare and return the result
if sum_topk_content >= sum_topk_style:
    return content_lora_weight
else:
    return style_lora_weight

```

fl: flatten;

style, we select the previous dataset provided by the authors of StyleDrop [28] and include several classic masterpieces along with some modern innovative styles. For each style, we only use a single image for training.

Experimental details. We perform our experiments using the SDXL v1.0 base model and FLUX model and test the performance of K-LoRA using locally trained LoRA and community-trained LoRA. For the community-trained LoRA, we use the widely available LoRA models from Hugging Face for testing. For the locally trained LoRA, we base on the method outlined in ZipLoRA [26] to obtain a set of style and content LoRAs. For the hyperparameters mentioned in Eqn. (7), we set $\alpha = 1.5$ and $\beta = 0.5$. This configuration was found to work effectively for nearly all cases, yielding consistently good generation results.

4.2. Results

Quantitative comparisons. We randomly select 18 combinations of objects and styles, each of which consists of

10 images to perform quantitative comparisons. We use CLIP [21] to measure the style similarity. We compute the subject similarity through CLIP score and DINO score [38]. We compare our method with popular approaches in the community as well as state-of-the-art methods, including direct arithmetic merging, joint training, ZipLoRA [26], and B-LoRA [8]. The results are shown in Tab. 1. It can be observed that our method significantly improves subject similarity metrics compared to previous approaches, while also achieving satisfactory style similarity.

Method	Style Sim \uparrow	CLIP Score \uparrow	DINO Score \uparrow
Direct	48.9%	66.6%	43.0%
Joint	68.2%	57.5%	17.4%
B-LoRA [8]	58.0%	63.8%	30.6%
ZipLoRA [26]	60.4%	64.4%	35.7%
K-LoRA (ours)	58.7%	69.4%	46.9%

Table 1. Quantitative comparisons. Comparison of alignment results across different methods.

Qualitative comparisons. In order to ensure a fair evaluation, all experiments at this stage are conducted using SD, the result is shown in Fig. 7, the method for merging LoRAs [25] struggles to preserve the original shape, color, and stylistic features of the object when the fusion ratio is set directly to 1:2 without extensive parameter adjustments or seed selection. B-LoRA [8] mainly captures the color and appearance of objects in the original image, often leading to overfitting of the color, which makes it difficult to distinguish the original objects in the generated image. In ZipLoRA [26] and joint training methods, while certain stylistic textures are incorporated, the model tends to focus on the background elements of the style rather than capturing the style itself, resulting in a lower success rate. In contrast, our method addresses these limitations by producing higher-quality output images with stable performance across a wide range of seed variations. Additionally, our approach eliminates the need for extra training or parameter fine-tuning.

We present a randomly selected set of 22 results to users for comparative evaluation. Each set includes outputs from

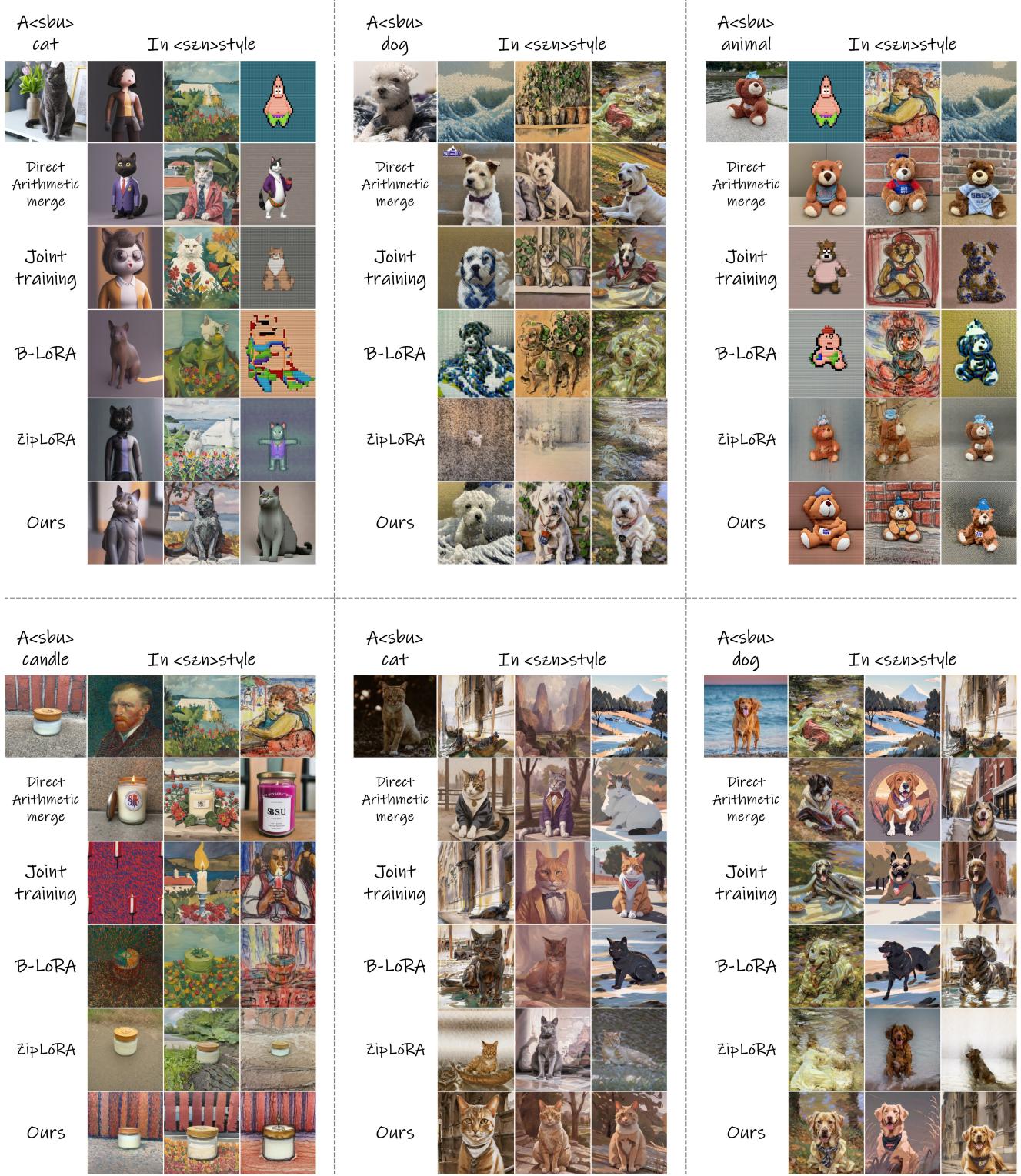


Figure 7. **Qualitative comparisons.** We present images generated by K-LoRA and the compared methods. K-LoRA generally achieves a seamless integration of objects and styles, effectively preserving fidelity and preventing distortion.

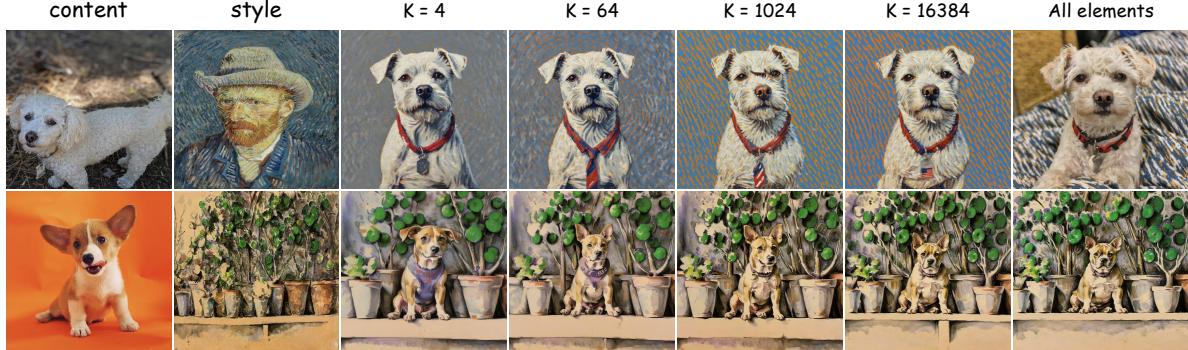


Figure 8. **Selection of K .** Evaluations of the impact on different K in K-LoRA.

ZipLoRA, B-LoRA, and our method, along with reference images for both the training subject and the style. Users were asked to identify which method best preserves both style and subject. The results, shown in Tab. 2, indicate that our method is the most preferred. In addition, we consulted with GPT-4o for a similar assessment. Our method shows a significant advantage in GPT-4o evaluations, further reflecting the superiority of our method.

Method	User Preference	GPT-4o Feedback
ZipLoRA [26]	29.2%	5.6%
B-LoRA [8]	18.1%	11.1%
Ours	52.7%	83.3%

Table 2. **User study results and GPT-4o feedback.**

4.3. Ablation Analysis

Top-K selection. We conduct two experiments to validate the effectiveness of the Top-K selection method: fixed selection and random selection. Finding (ii) suggests a straightforward approach: If the scale factor is greater than one, the content LoRA is selected; Otherwise, the style LoRA is chosen. This approach, which we refer to as “Fixed Selection” serves as a useful baseline to test the ablation of the Top-K selection method. It can also be seen as an extension and refinement of Multi-LoRA composition [41], which has shown promising results in certain scenarios. However, under specific style LoRA conditions, this method may result in object blurring or alterations in the content’s appearance, as shown in Fig. 9.

To ensure that our module performs consistently within the specified forward layer arrangement rather than relying on arbitrary configurations, we conduct a controlled experiment termed “Random Selection” using a random seed. In this setup, the model uses a random number with a 1/3 probability of selecting content attention and a 2/3 probability of selecting style attention. As shown in Fig. 9, under these random selection conditions, the generated images often re-

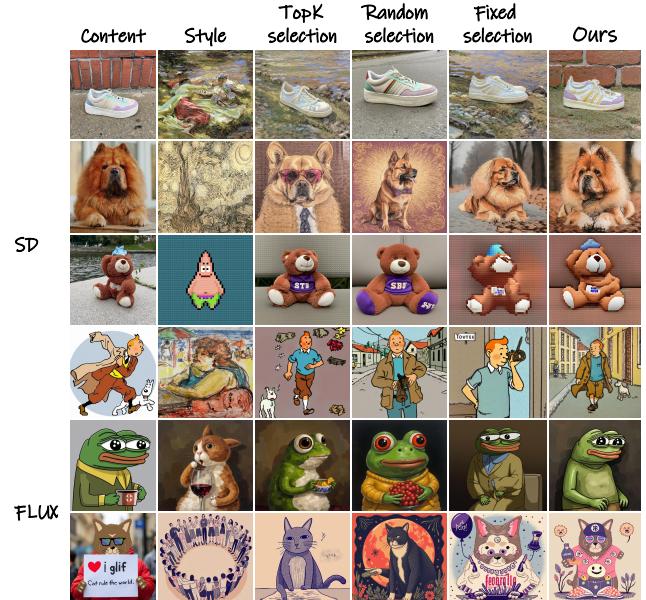


Figure 9. **Ablation of Top-K selection and scaling factor.** We compare different methods using five sets of images. The four rows above represent the results of SD, while the rows below present the results of FLUX, which include both locally trained LoRA and community trained LoRA.

tain only a single object feature or style feature, or fail to maintain either altogether. This outcome further validates our finding (ii), highlighting the distinct roles played by object and style components at earlier and later diffusion time steps, respectively.

Furthermore, we evaluate the impact of different choices of K on the generated images, as illustrated in Fig. 8. Within the Top-K approach, we systematically vary the values of K . Our observations indicate that when K is relatively small, neither the style nor the characteristics of the object are sufficiently prominent. This issue gradually improves as K increases. However, if K becomes excessively large, the style may not be preserved, and the shape of the object can undergo significant distortions.

Scaling factor. To evaluate the effectiveness of the scaling factor, we remove it and focus solely on the original Top-K approach. In the first experiment, as shown in Fig. 9, our analysis reveals that while the exclusive use of Top-K can produce satisfactory results under certain conditions, expanding the experimental scope uncovers instances of object distortion and style loss. To further assess the significance of gamma within the scaling factor, we test the performance of two LoRA models with distinct sources, characterized by substantial differences in their element sums. As illustrated in the bottom row of Fig. 9, it is evident that Top-K selection fails to capture the style accurately, while the fusion of object and style in fixed selection is noticeably weaker compared to our approach. We also experiment with an alternative scale. The detailed procedure is provided in the supplementary material (Sec. D).

In conclusion, the removal of these two modules leads to a decrease in generative performance, underscoring their critical contributions to the overall effectiveness of the model.

5. Conclusions

In this paper, we introduce K-LoRA, which can seamlessly merge independently trained style and subject LoRA models. K-LoRA enables precise object fine-tuning while preserving the intricate details of the original style. Our approach effectively leverages the contributions of both object and style LoRAs at each diffusion step through Top-K selection and scaling factors, maximizing the use of the original weights and allowing for accurate style fusion without the need for retraining or manual hyperparameter tuning.

References

- [1] Yuval Alaluf, Elad Richardson, Gal Metzer, and Daniel Cohen-Or. A neural space-time representation for text-to-image personalization. *ACM Transactions on Graphics (TOG)*, 42(6):1–10, 2023. 3
- [2] Omri Avrahami, Kfir Aberman, Ohad Fried, Daniel Cohen-Or, and Dani Lischinski. Break-a-scene: Extracting multiple concepts from a single image. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–12, 2023. 3
- [3] black-forest labs. Flux.1. <https://github.com/black-forest-labs/flux>, 2024. 1
- [4] Dongdong Chen, Lu Yuan, Jing Liao, Nenghai Yu, and Gang Hua. Stylebank: An explicit representation for neural image style transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1897–1906, 2017. 1
- [5] Jiwoo Chung, Sangeek Hyun, and Jae-Pil Heo. Style Injection in Diffusion: A Training-free Approach for Adapting Large-scale Diffusion Models for Style Transfer. *arXiv e-prints*, art. arXiv:2312.09008, 2023. 11, 17
- [6] Yingying Deng, Fan Tang, Weiming Dong, Wen Sun, Feiyue Huang, and Changsheng Xu. Arbitrary style transfer via multi-adaptation network. In *Proceedings of the 28th ACM international conference on multimedia*, pages 2719–2727, 2020. 1
- [7] Jiahua Dong, Wenqi Liang, Hongliu Li, Duzhen Zhang, Meng Cao, Henghui Ding, Salman Khan, and Fahad Shahbaz Khan. How to continually adapt text-to-image diffusion models for flexible customization? *arXiv preprint arXiv:2410.17594*, 2024. 3
- [8] Yarden Frenkel, Yael Vinker, Ariel Shamir, and Daniel Cohen-Or. Implicit style-content separation using b-lora. *arXiv preprint arXiv:2403.14572*, 2024. 1, 2, 3, 6, 8
- [9] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016. 1
- [10] Yuchao Gu, Xintao Wang, Jay Zhangjie Wu, Yujun Shi, Yunpeng Chen, Zihan Fan, Wuyou Xiao, Rui Zhao, Shuning Chang, Weijia Wu, et al. Mix-of-show: Decentralized low-rank adaptation for multi-concept customization of diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024. 3
- [11] Soufiane Hayou, Nikhil Ghosh, and Bin Yu. Lora+: Efficient low rank adaptation of large models. *arXiv preprint arXiv:2402.12354*, 2024. 3
- [12] 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, 3
- [13] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pages 1501–1510, 2017. 1
- [14] Jiaxiu Jiang, Yabo Zhang, Kailai Feng, Xiaohe Wu, and Wangmeng Zuo. Mc²: Multi-concept guidance for customized multi-concept generation. *arXiv preprint arXiv:2404.05268*, 2024. 3
- [15] Dawid J Kopickzo, Tijmen Blankevoort, and Yuki M Asano. Vera: Vector-based random matrix adaptation. *arXiv preprint arXiv:2310.11454*, 2023. 3
- [16] 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. 3
- [17] Zhen Li, Mingdeng Cao, Xintao Wang, Zhongang Qi, Ming-Ming Cheng, and Ying Shan. Photomaker: Customizing realistic human photos via stacked id embedding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1
- [18] Zhiheng Liu, Ruili Feng, Kai Zhu, Yifei Zhang, Kecheng Zheng, Yu Liu, Deli Zhao, Jingren Zhou, and Yang Cao. Cones: Concept neurons in diffusion models for customized generation. *arXiv preprint arXiv:2303.05125*, 2023. 3
- [19] Daniel Marczak, Bartłomiej Twardowski, Tomasz Trzcinski, and Sebastian Cygert. Magmax: Leveraging model merging for seamless continual learning. *arXiv preprint arXiv:2407.06322*, 2024. 3

- [20] Or Patashnik, Daniel Garabi, Idan Azuri, Hadar Averbuch-Elor, and Daniel Cohen-Or. Localizing object-level shape variations with text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23051–23061, 2023. 2, 4
- [21] 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. 6
- [22] Pengjie Ren, Chengshun Shi, Shiguang Wu, Mengqi Zhang, Zhaochun Ren, Maarten Rijke, Zhumin Chen, and Jiahuan Pei. Melora: Mini-ensemble low-rank adapters for parameter-efficient fine-tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3052–3064, 2024. 3
- [23] 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. 3
- [24] 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*, 2023. 1, 3, 5, 11
- [25] Simo Ryu. Merging loras. <https://github.com/cloneofsimo/lora>, 2023. 2, 3, 6, 11
- [26] Viraj Shah, Nataniel Ruiz, Forrester Cole, Erika Lu, Svetlana Lazebnik, Yuanzhen Li, and Varun Jampani. Ziplora: Any subject in any style by effectively merging loras. In *European Conference on Computer Vision*, pages 422–438. Springer, 2024. 1, 2, 3, 5, 6, 8
- [27] Jing Shi, Wei Xiong, Zhe Lin, and Hyun Joon Jung. Instantbooth: Personalized text-to-image generation without test-time finetuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8543–8552, 2024. 3
- [28] Kihyuk Sohn, Nataniel Ruiz, Kimin Lee, Daniel Castro Chin, Irina Blok, Huiwen Chang, Jarred Barber, Lu Jiang, Glenn Entis, Yuanzhen Li, et al. Styledrop: Text-to-image generation in any style. *arXiv preprint arXiv:2306.00983*, 2023. 1, 6, 11
- [29] Andrey Voynov, Qinghao Chu, Daniel Cohen-Or, and Kfir Aberman. p+: Extended textual conditioning in text-to-image generation. *arXiv preprint arXiv:2303.09522*, 2023. 2, 3, 4
- [30] Taiqiang Wu, Jiahao Wang, Zhe Zhao, and Ngai Wong. Mixture-of-subspaces in low-rank adaptation. *arXiv preprint arXiv:2406.11909*, 2024. 3
- [31] Guangxuan Xiao, Tianwei Yin, William T Freeman, Frédo Durand, and Song Han. Fastcomposer: Tuning-free multi-subject image generation with localized attention. *International Journal of Computer Vision*, pages 1–20, 2024. 3
- [32] Shaoan Xie, Zhifei Zhang, Zhe Lin, Tobias Hinz, and Kun Zhang. Smartbrush: Text and shape guided object inpainting with diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22428–22437, 2023. 3
- [33] Yu xin Zhang, Weiming Dong, Fan Tang, Nisha Huang, Haibin Huang, Chongyang Ma, Tong-Yee Lee, Oliver Deussen, and Changsheng Xu. Prospect: Prompt spectrum for attribute-aware personalization of diffusion models. *ACM Transactions on Graphics (TOG)*, 42:1 – 14, 2023. 11
- [34] Peng Xing, Haofan Wang, Yanpeng Sun, Qixun Wang, Xu Bai, Hao Ai, Renyuan Huang, and Zechao Li. Csgo: Content-style composition in text-to-image generation. *arXiv preprint arXiv:2408.16766*, 2024. 1
- [35] Youcan Xu, Zhen Wang, Jun Xiao, Wei Liu, and Long Chen. Freetuner: Any subject in any style with training-free diffusion. *arXiv preprint arXiv:2405.14201*, 2024. 2, 4
- [36] Yang Yang, Wen Wang, Liang Peng, Chaotian Song, Yao Chen, Hengjia Li, Xiaolong Yang, Qinglin Lu, Deng Cai, Boxi Wu, et al. Lora-composer: Leveraging low-rank adaptation for multi-concept customization in training-free diffusion models. *arXiv preprint arXiv:2403.11627*, 2024. 3
- [37] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ipadapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023. 1
- [38] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022. 6
- [39] Longteng Zhang, Lin Zhang, Shaohuai Shi, Xiaowen Chu, and Bo Li. Lora-fa: Memory-efficient low-rank adaptation for large language models fine-tuning. *arXiv preprint arXiv:2308.03303*, 2023. 3
- [40] Yuxin Zhang, Nisha Huang, Fan Tang, Haibin Huang, Chongyang Ma, Weiming Dong, and Changsheng Xu. Inversion-based style transfer with diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10146–10156, 2023. 3
- [41] Ming Zhong, Yelong Shen, Shuhang Wang, Yadong Lu, Yizhu Jiao, Siru Ouyang, Donghan Yu, Jiawei Han, and Weizhu Chen. Multi-lora composition for image generation. *arXiv preprint arXiv:2402.16843*, 2024. 2, 3, 4, 8, 11, 18
- [42] Donghao Zhou, Jiancheng Huang, Jinbin Bai, Jiaze Wang, Hao Chen, Guangyong Chen, Xiaowei Hu, and Pheng-Ann Heng. Magictailor: Component-controllable personalization in text-to-image diffusion models. *arXiv preprint arXiv:2410.13370*, 2024. 1
- [43] Hongyun Zhou, Xiangyu Lu, Wang Xu, Conghui Zhu, and Tiejun Zhao. Lora-drop: Efficient lora parameter pruning based on output evaluation. *arXiv preprint arXiv:2402.07721*, 2024. 3
- [44] Bojia Zi, Xianbiao Qi, Lingzhi Wang, Jianan Wang, Kam-Fai Wong, and Lei Zhang. Delta-lora: Fine-tuning high-rank parameters with the delta of low-rank matrices. *arXiv preprint arXiv:2309.02411*, 2023. 3

Supplementary Material

The supplementary material is structured as follows:

1. We first evaluated our results on extensive datasets and community LoRAs on different models to validate the effectiveness of our approach in section A.
2. We compared our method with the other methods in section B.
3. We assessed the influence of complex prompts on the model’s performance in section C.
4. We experimented with a new scale and tested its comparative effects in section D.
5. We utilized Community LoRA in combination with local LoRA to conduct integrated performance evaluations and examined random seeds on model performance through comprehensive testing in section E.
6. We tested the choice of different parameters in scale factors in section F.

A. Visual Results

We employ datasets from StyleDrop [28] and DreamBooth [24] with Stable Diffusion (SD), as depicted in Fig. 13 and Fig. 14, we also evaluated our method on FLUX using LoRAs from Hugging Face, as shown in Fig. 11 and Fig. 12. By systematically combining these object and style LoRAs, we obtained a sequence of images that demonstrates the effectiveness of our approach in seamlessly integrating both object and style, yielding consistent and high-quality visual outputs.

B. Additional Comparisons

We have added a comparison with StyleID [5], as shown in Fig. 15. It can be observed that StyleID [5] effectively achieves style transfer while preserving texture quality. However, the generated objects might be slightly blurred or the style generated may not be distinct. Additionally, compared to our method, their approach is based on the fixed layout of original image, which may not generalize well to backgrounds and actions.

C. Prompt Control

We conduct experiments to evaluate whether our method can modify the object’s actions, the surrounding environment, or introduce new elements through prompt adjustments. As illustrated in Fig. 18 and Fig. 19, after modifying the prompts, our method effectively retains the original object’s features and stylistic attributes, while also integrating new elements or scene details seamlessly.

D. New Scale

In the main text of our paper, we employ the scale given by Eqn. (7) as follows:

$$S = \alpha \cdot \frac{t_{now}}{t_{all}} + \beta. \quad (11)$$

Inspired by [33], we also introduce an alternative scale factor:

$$S^* = \left(\alpha' \cdot \frac{t_{now}}{t_{all}} + \beta' \right) \% \alpha. \quad (12)$$

In this equation, we set $\alpha' = 1.5$ and $\beta' = 1.3$, which means that the style information is enhanced to some extent at the beginning of the generation process, allowing the model to capture certain block information from the style LoRA. Fig. 10 below illustrates the primary differences between the two scales.

For S^* results, since the style information is enhanced during the early diffusion steps, the generated images capture the background and color block information from the style LoRA. However, this approach results in a weakened learning effect for the texture and brushstrokes information in the style LoRA. This represents a trade-off, and users can select different scale factors based on their preferences.



Figure 10. **Results of different scaling factors.** Corresponding generation results of K-LoRA with differernt scaling factor and for each object-style pair, two seeds are randomly selected.

E. Robustness Analysis

We evaluate LoRA models from various sources, where the object LoRA is sourced from the community, while the style LoRA is trained locally. We also compare Direct-Merge [25], Multi-LoRA composition [41], and our proposed Fixed Selection approach. As shown in Fig. 16, our method demonstrates superior performance in learning both object and style characteristics, surpassing other approaches. Furthermore, we test the robustness of our approach by selecting random seeds to assess stability. The results, presented in Fig. 17, indicate that our method consistently achieves stable fusion across a broad range of seed selections, ensuring reliable integration.

F. Additional Ablations

In the main text, we employ a scale with two hyperparameters, α and β . Specifically, we set α to 0.5 and β to 1.5, enabling objects and styles to exert varying levels of influence at different positions. To validate the suitability of the selected parameters, we compute the CLIP similarity scores between 18 randomly chosen sets of generated images and their corresponding original object/style references. The results shown in the table below represent the summation of CLIP similarity scores.

$\beta \setminus \alpha$	1.0	1.5	2.0
0.25	125.3%	126.7%	127.0%
0.50	126.5%	128.1%	126.2%
0.75	124.5%	125.8%	125.3%

We can see that the optimal setting for α and β is 1.5 and 0.5, respectively. This weight configuration satisfies almost all content-style pairs according to our experiments, and users do not need to make further adjustments.



Figure 11. **Additional Generated Results using FLUX.** The images in each position correspond to the object above and the style on the left, showing the results generated by applying the different LoRAs with our method.

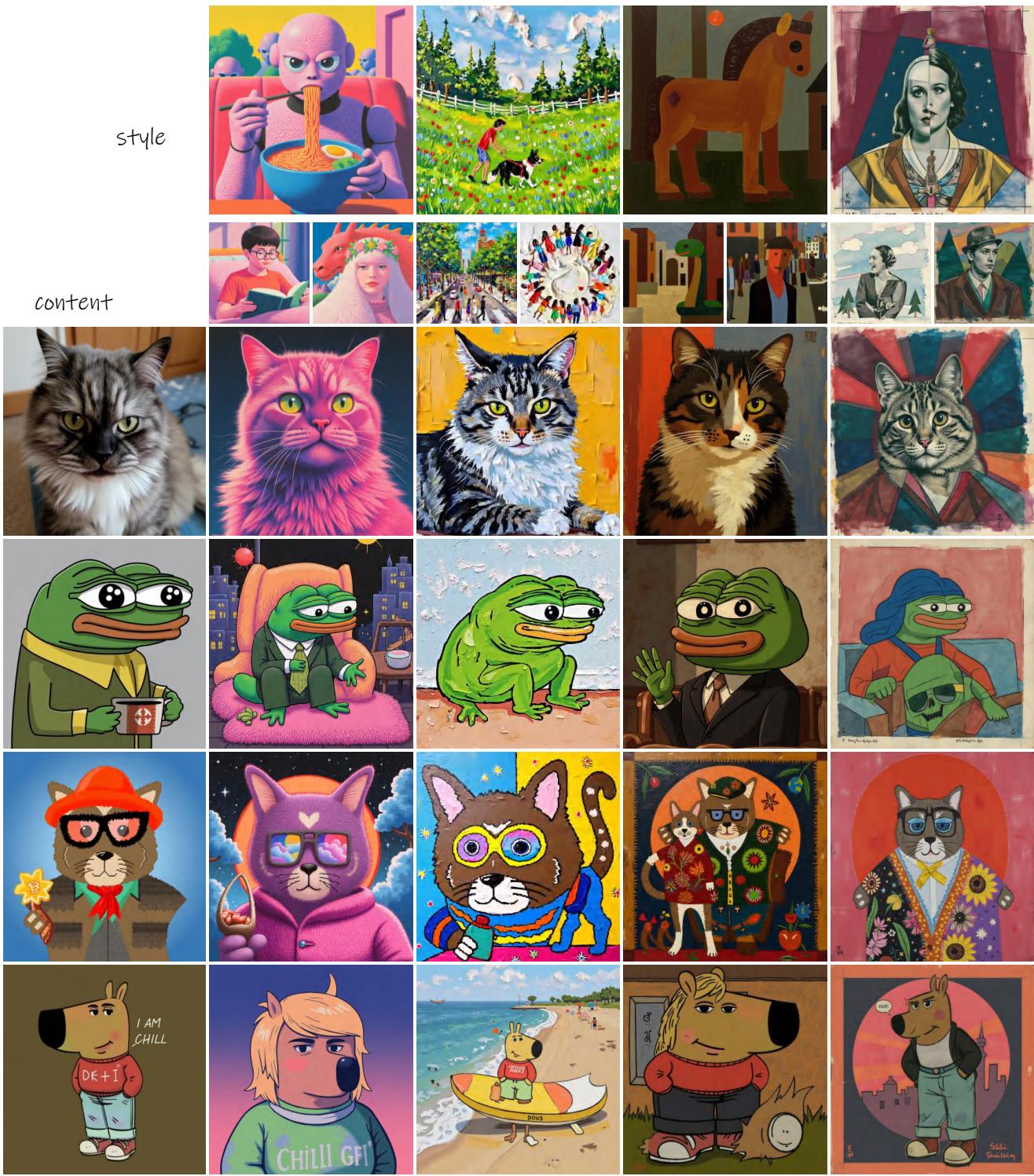


Figure 12. **Additional Generated Results using FLUX.** The images in each position correspond to the object above and the style on the left, showing the results generated by applying the different LoRAs with our method.

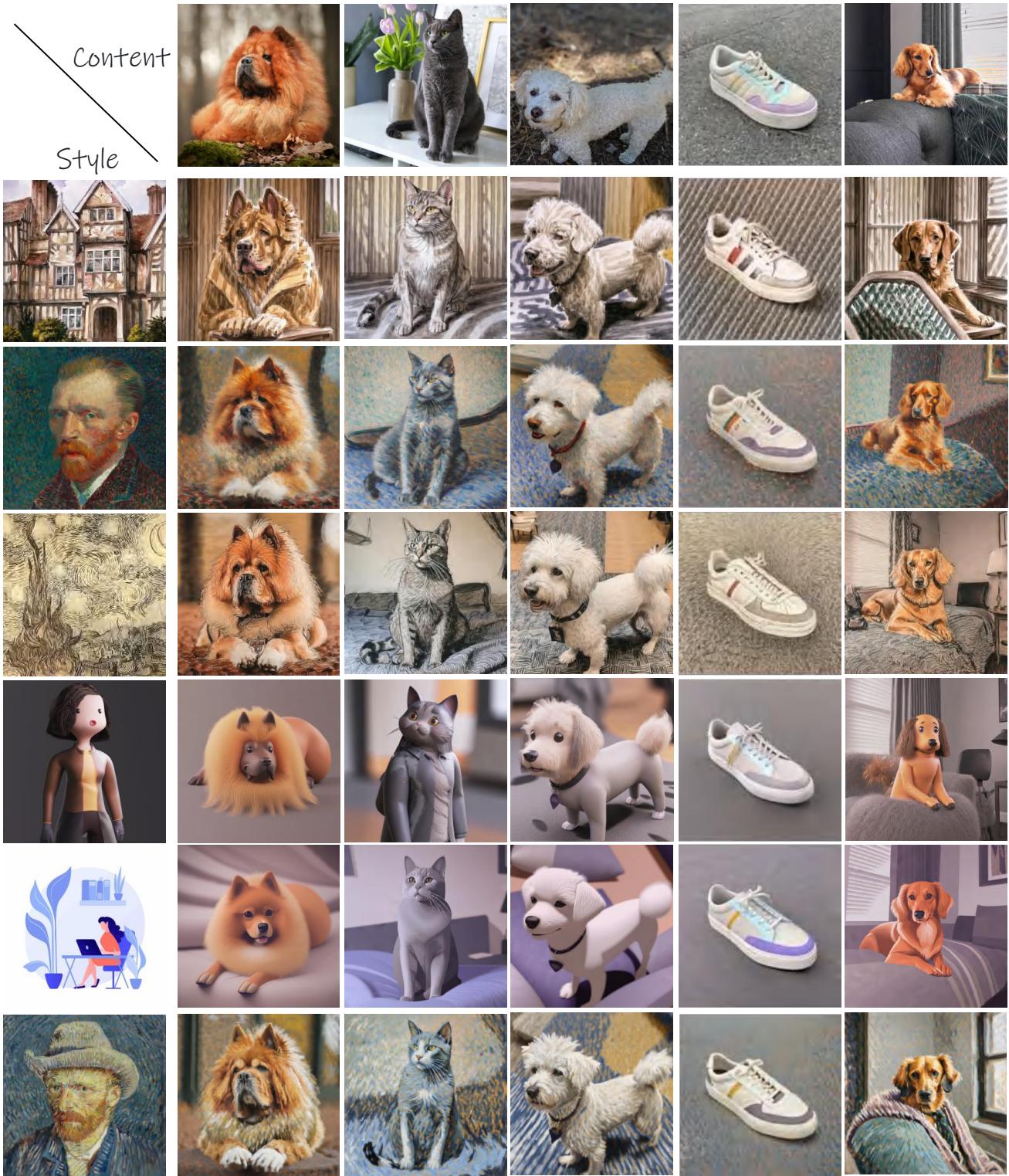


Figure 13. **Additional Generated Results using SD.** The images in each position correspond to the object above and the style on the left, showing the results generated by applying the different LoRAs with our method.

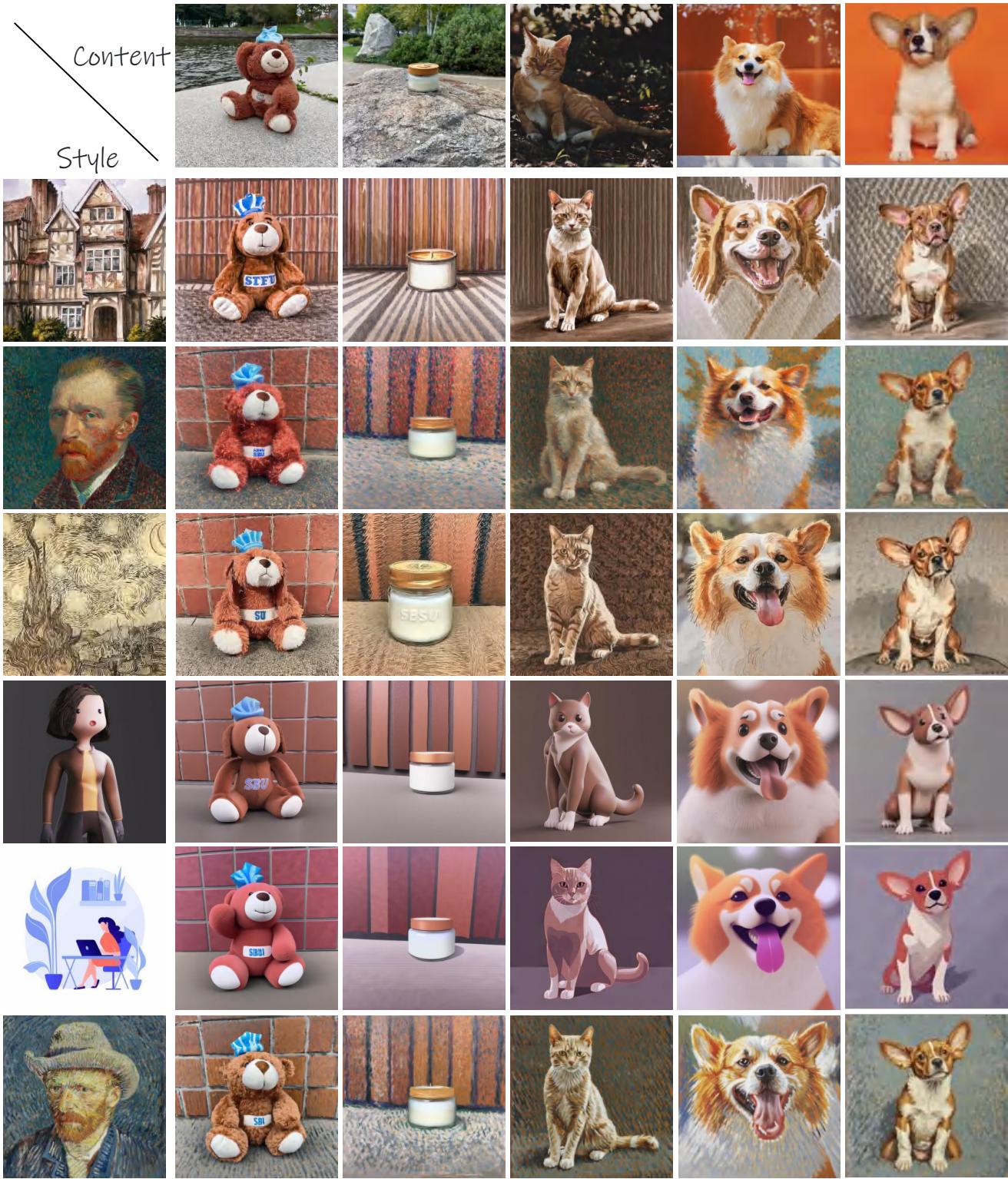


Figure 14. **Additional Generated Results using SD.** The images in each position correspond to the object above and the style on the left, showing the results generated by applying the different LoRAs with our method.

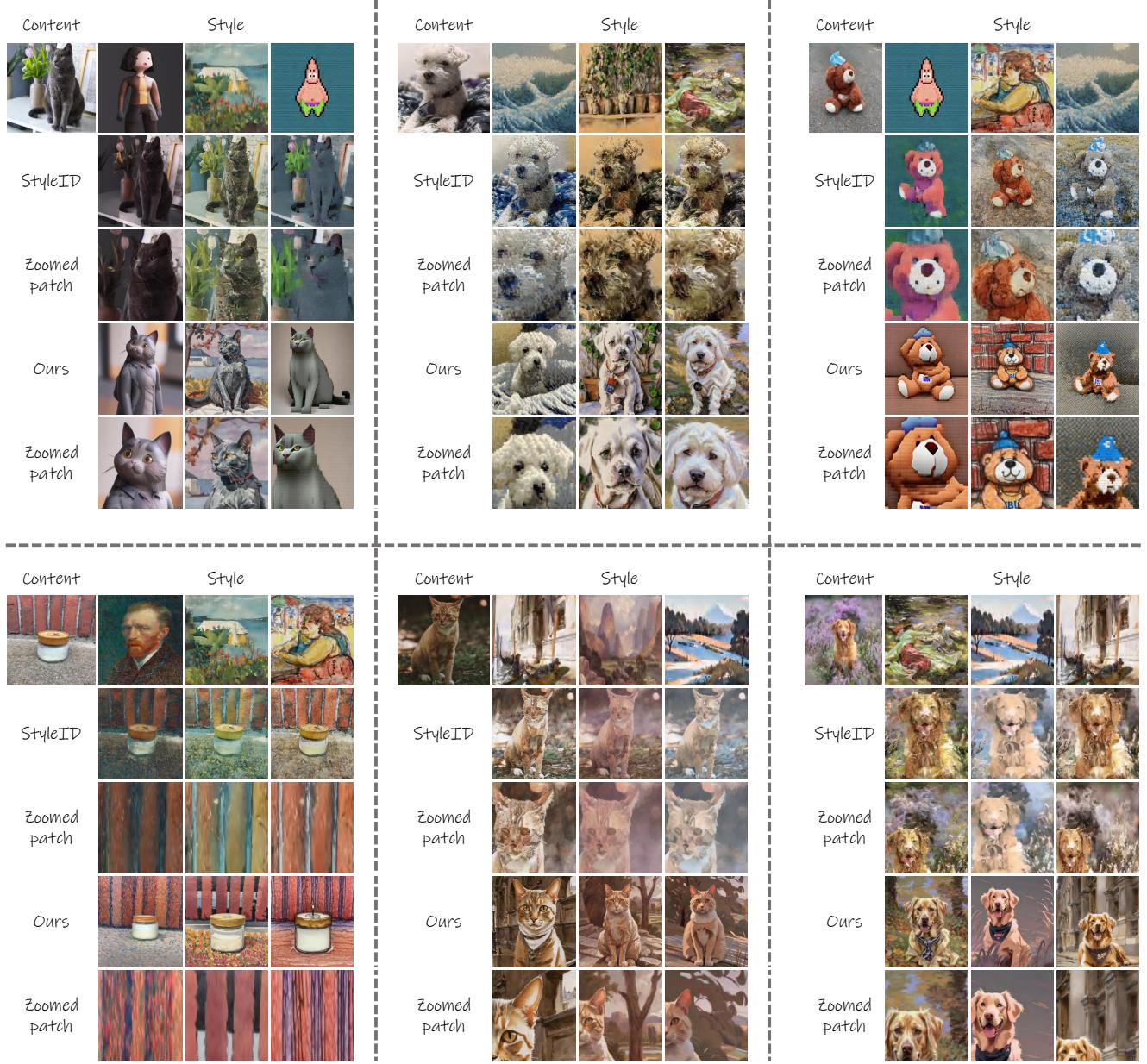


Figure 15. **Additional Comparisons.** We compare the StyleID [5] method and then capture zoomed patches in the output image to observe detailed texture information and stylistic features. Within each block, the second and third rows represent StyleID results along with its corresponding zoomed patch, while the subsequent two rows illustrate the result of our method and the associated zoomed patch.

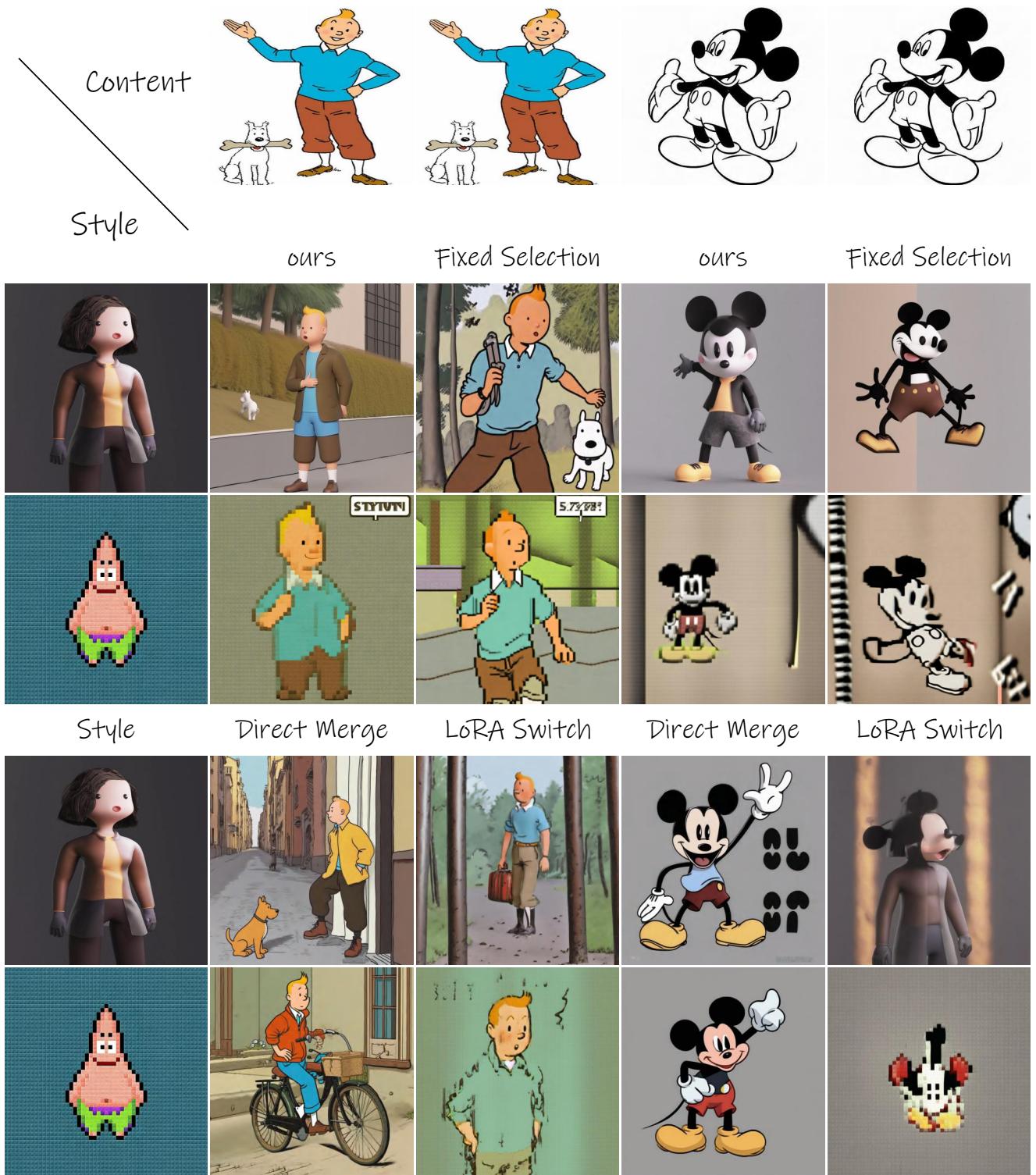


Figure 16. Robustness Validation. We utilize community LoRAs and locally trained LoRAs to compare the Fixed Selection proposed in the main text, direct arithmetic merging LoRA as a baseline comparison, Multi-LoRA Composition [41] methods, in order to validate generalizability and robustness.

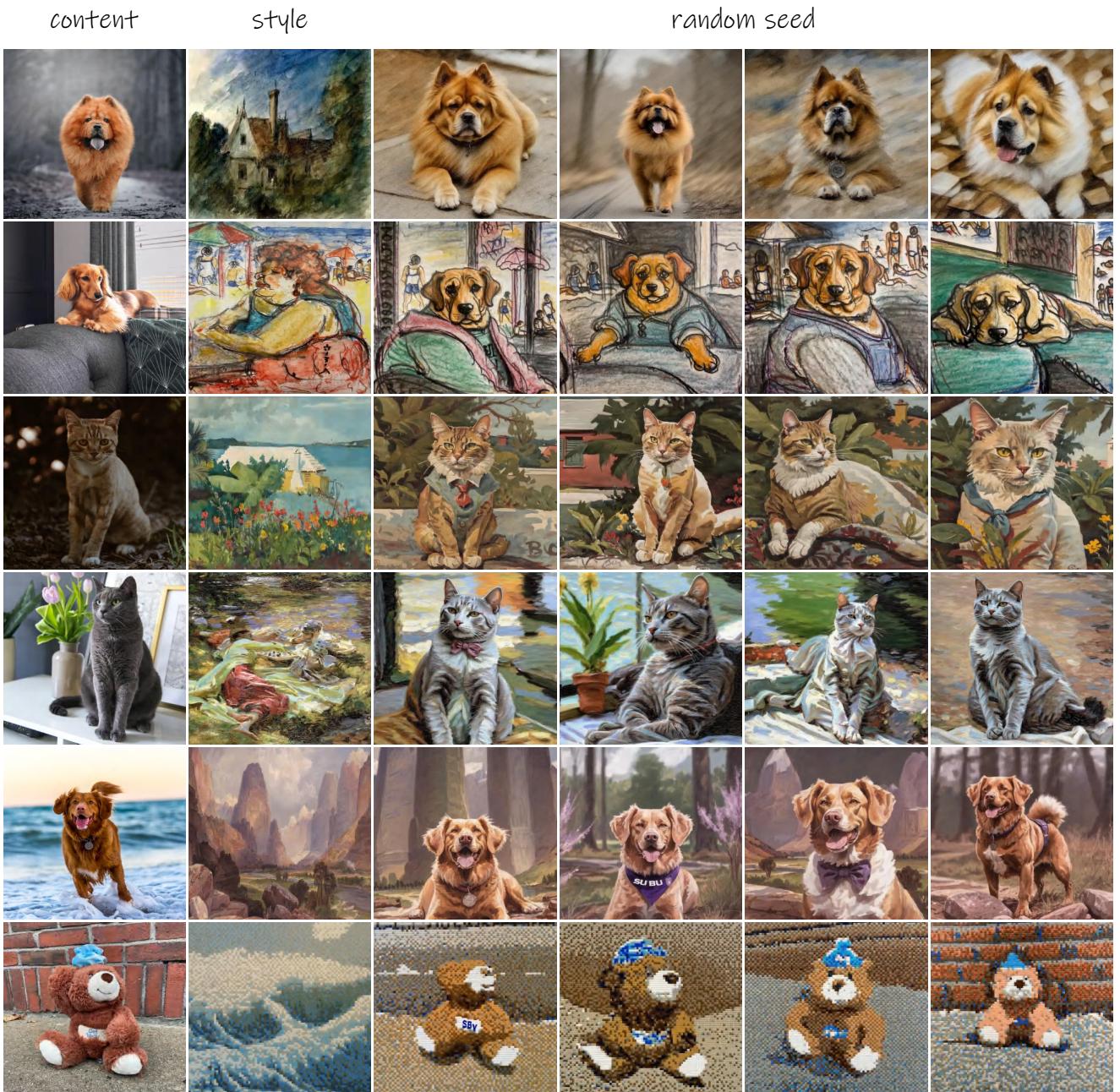


Figure 17. **Robustness Validation.** We randomly select seeds to further validate stability.

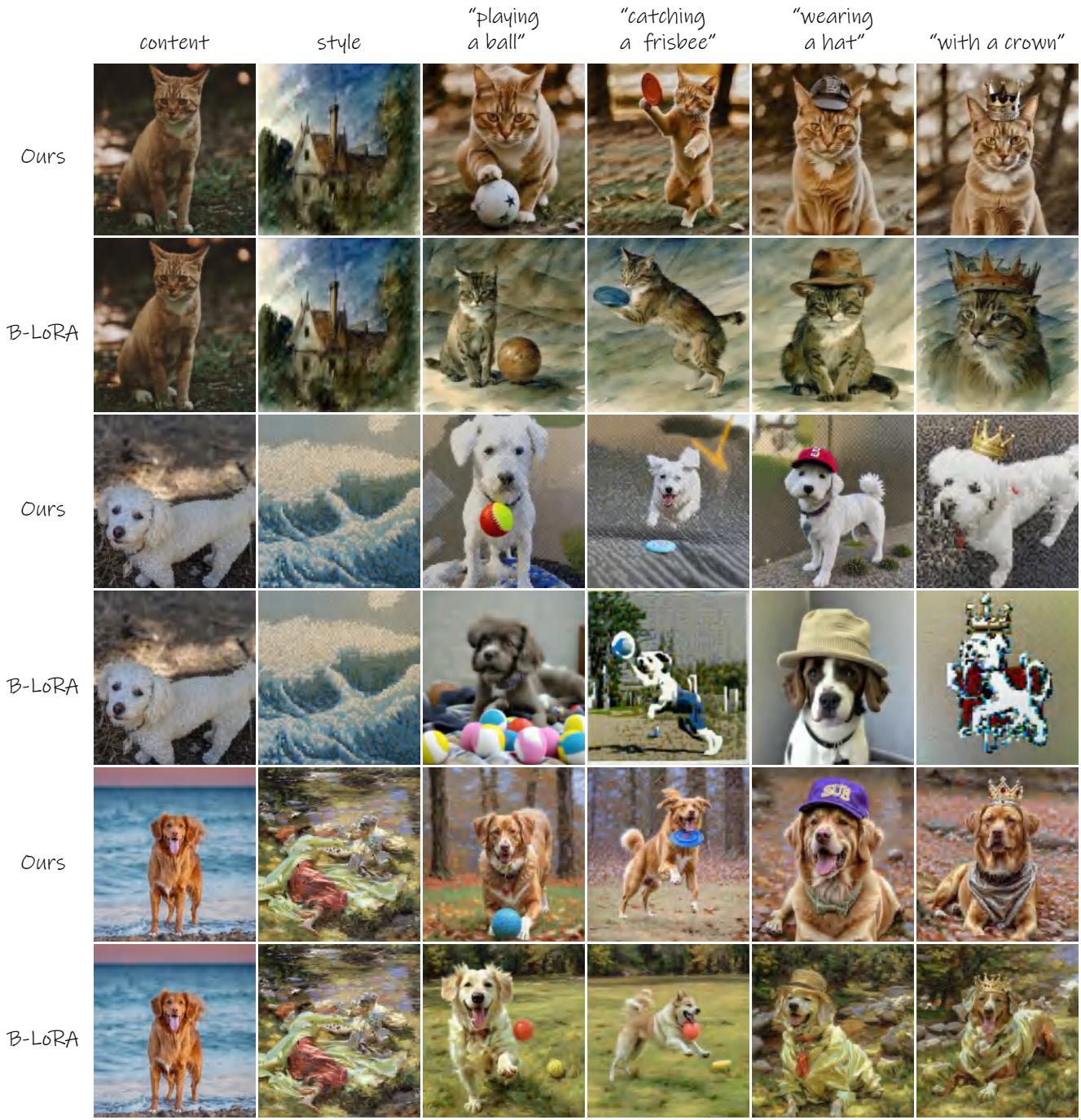


Figure 18. **Prompt Control.** We introduce prompts for new scenes, new actions, and new objects to validate our method’s ability to re-contextualize content and maintain stylistic consistency.

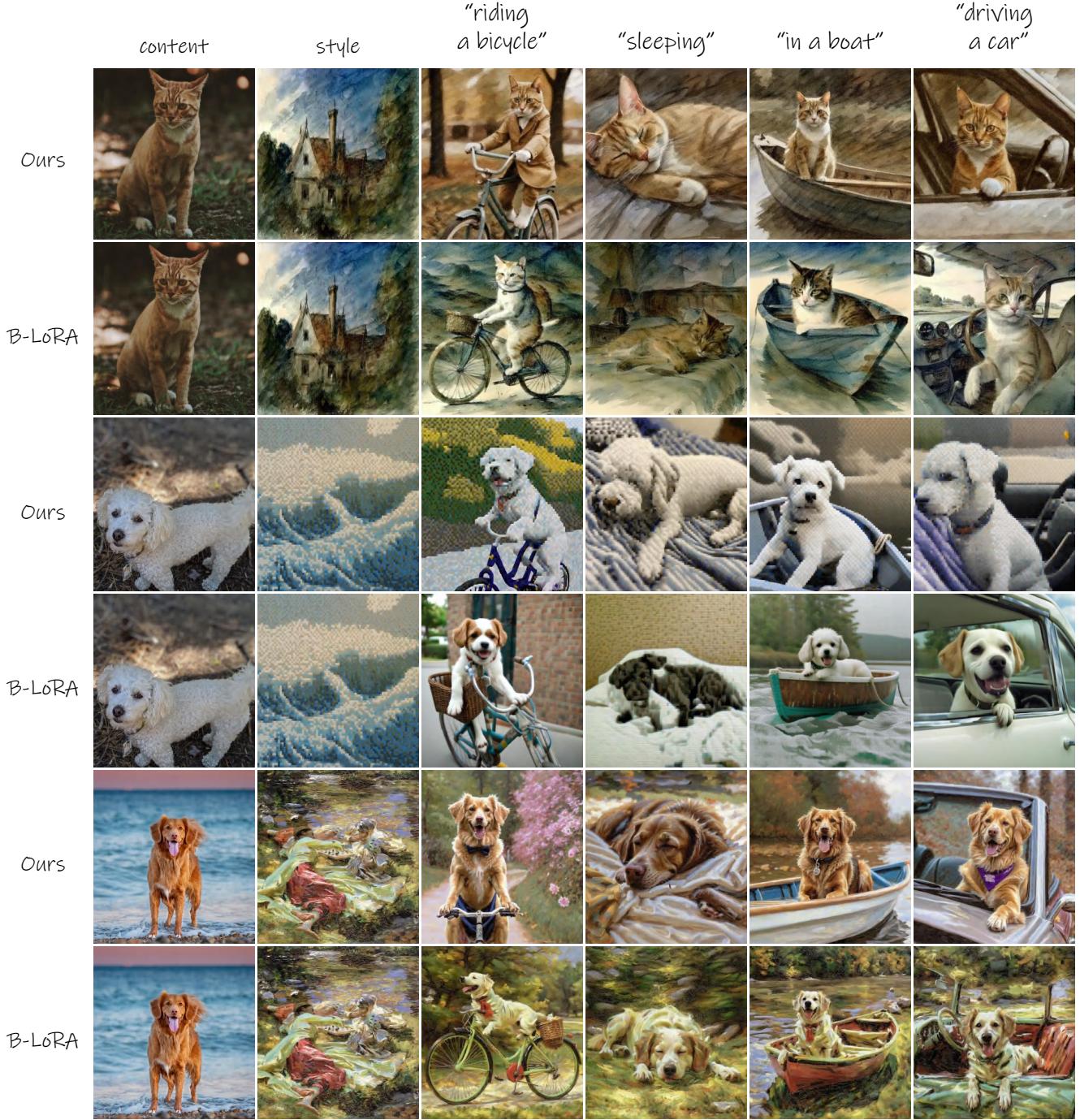


Figure 19. **Prompt Control.** We introduce prompts for new scenes, new actions, and new objects to validate our method’s ability to re-contextualize content and maintain stylistic consistency.