Implicit Style-Content Separation using B-LoRA

Yarden Frenkel[1], Yael Vinker[1], Ariel Shamir[2], and Daniel Cohen-Or[1]

[1] Tel Aviv University
[2] Reichman University

Fig. 1: By implicitly decomposing a single image into its style and content representation, captured by B-LoRA, we can perform high quality style-content mixing and even swapping the style and content between two stylized images.

Abstract. Image stylization involves manipulating the visual appearance and texture (style) of an image while preserving its underlying objects, structures, and concepts (content). The separation of style and content is essential for manipulating the image's style independently from its content, ensuring a harmonious and visually pleasing result. Achieving this separation requires a deep understanding of both the visual and semantic characteristics of images, often necessitating the training of specialized models or employing heavy optimization.

In this paper, we introduce B-LoRA, a method that leverages LoRA (Low-Rank Adaptation) to *implicitly* separate the style and content components of a *single* image, facilitating various image stylization tasks.

By analyzing the architecture of SDXL combined with LoRA, we find that jointly learning the LoRA weights of two specific blocks (referred to as B-LoRAs) achieves style-content separation that cannot be achieved by training each B-LoRA independently. Consolidating the training into only two blocks and separating style and content allows for significantly improving style manipulation and overcoming overfitting issues often associated with model fine-tuning.

Once trained, the two B-LoRAs can be used as independent components to allow various image stylization tasks, including image style transfer, text-based image stylization, consistent style generation, and style-content mixing.

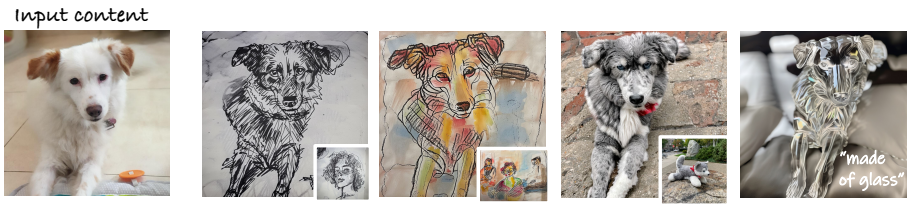

Fig. 2: Examples of image stylization generated with our approach. The content image is shown on the left. We show here three results of image style transfer based on a reference style, one (on the right) based on a guiding text prompt. Note that our method requires only a single image, and preserves the image's content and structure well while applying the desired style.

1 Introduction

Image stylization is a well-established task in computer vision, and has been actively researched for many years [16, 22]. This task involves changing the style of an image following some style reference, which can be text-based or image-based while preserving its content. Content refers to the semantic information and structure of the image, while style often refers to visual features and patterns such as colors and textures [46]. Image style manipulation is a highly challenging task, since style and content are strongly connected, leading to an inherent trade-off between style transformation and content preservation. On the other hand, many style manipulation tasks require a clear separation between style and content within an image.

In this paper, we present B-LoRA, a method for style-content separation of any given image. Our method distills the style and content from a single image to support various style manipulation applications.

In the realm of recent advancements in large language-vision models, existing approaches utilize the strong visual-semantic priors embedded within these models to facilitate style manipulation tasks. Common techniques involve fine-tuning a pre-trained text-to-image model to account for a new style or content [4, 19, 24, 42]. However, fine-tuned models often suffer from the inherent trade-off between style transformation and content preservation as they are prone to over-fitting. Unlike these methods, we unify the learning of style and content components by separating them per image (see Figure 1). This separation is performed by fitting a light-weight adapter (B-LoRA) that is less prone to over-fitting issues, and enables task flexibility, allowing for both text-based and reference style image conditions.

Our method utilizes LoRA (Low Rank Adaptation) [24], which has emerged as a popular approach due to its high-quality results and efficiency. LoRA incorporates optimizing external low-rank weight matrices for the attention layers of the base model, while the pretrained model weights remain "frozen". After training, these matrices define the adapted model that can be used for the desired task. LoRA is often utilized for image stylization by fine-tuning the base model

with respect to a set of images that can either represent the desired style or the desired content.

Specifically, we use LoRA with Stable Diffusion XL (SDXL) [38], a recently introduced text-to-image diffusion model renowned for its powerful style learning capabilities. Through detailed analysis of various layers within SDXL and their effect on the adaptation procedure, we made a surprising discovery: two specific transformer blocks can be used to separate the style and content of an input image, and to easily control them distinctly in generated images. For clarification, in this paper, we define a transformer block as a sequence of 10 consecutive attention layers.

Therefore, when provided with a *single* input image, we jointly optimize the LoRA weights corresponding to these two distinct transformer blocks with the objective of reconstructing the given image based on a provided text prompt. Since we only optimize the LoRA weights of these two transformer blocks, we refer to them as "B-LoRAs". The crucial aspect is that these B-LoRAs are trained on a single image only, yet they successfully disentangle its style and content, thereby circumventing the prevalent overfitting problem associated with common LoRA techniques that can struggle to change the style and/or content of an input image. Our technique benefits from the innate style-content disentanglement within the layers of the architecture. Another advantage of our method is that the B-LoRAs can be easily used as separate components, allowing various challenging style manipulation tasks without requiring any additional training or fine-tuning. In particular, we demonstrate style transfer, text-guided style manipulation and consistent style-conditioned image generation (see Figure 2).

We note that recent attempts have been made to combine trained LoRAs of style and content to a unified model [44]. This approach requires a new optimization process for each style-content combination. This is both time-consuming and raises challenges in achieving an effective trade-off between style transformation and content preservation. In contrast, our trained B-LoRAs can be easily replugged into a pre-trained model combined with other learned blocks from other reference images, without any further training.

We provide extensive evaluation of our method showing its advantages compared to alternative approaches that are often designed to achieve one of these tasks. Our method provides a practical and simple way for image stylization that can be broadly used with existing models.

2 Related Work

Style Transfer Image style transfer is a longstanding challenge in computer vision [12, 22], aimed at altering the style of an image based on a given reference. With the progress of deep learning research, Neural Style Transfer (NST) approaches rely on deep features extracted from pre-trained networks to merge content and style [16, 28, 29]. Subsequent GAN-based [17] techniques were proposed to transfer images across domains, using either paired [27] or unpaired [30, 35, 57] image sets, yet they require domain-specific datasets and training.

Recent advancements in language-vision models and diffusion models have revolutionized the field of image stylization. Leveraging the vast knowledge encoded in pre-trained language-vision models, modern approaches explore zero-shot image stylization and editing [6, 10, 11, 13, 32, 34, 36, 52], where images are manipulated without additional fine-tuning or data adaptation by intervening in the generation process. Prompt-to-Prompt [20] proposes an approach to edit generated images by manipulating their cross-attention maps. In Plug-and-Play [47] the appearance of a content image is manipulated with respect to a given text prompt by adjusting spatial features from the guidance image via the self-attention mechanism. Cross Image Attention (CIA) [2] presents a method to modify the image appearance based on a reference image through alterations in cross-attention mechanisms. While these approaches effectively transform the appearance of the content image, they may encounter challenges in transferring appearance between subjects with differing semantics.

StyleAligned [21] utilizes attention features sharing combined with the AdaIN mechanism [25] to achieve style alignment between a sequence of generated images. However, the method is not explicitly designed to control the content of the generated image, potentially resulting in style image structure leakage. Similarly, the lack of style-content separation is also evident in encoder-based methods, such as IP-Adapter [53]. InstantStyle [51] is a concurrent work to ours, aiming to improve IP-Adapter for image stylization tasks by injecting the CLIP embedding of the style image into specific blocks within SDXL. In our work, we decompose the style and the content and learn a separate representation for each.

Text-to-Image Personalization In another line of work [3, 4, 14, 19, 42, 50], optimization techniques are proposed to extend pre-trained Text-to-Image models to support the generation of novel visual concepts, including both style and content, based on a small set of input images with the same concept. This allows utilizing the rich semantic-visual prior of pre-trained models for customized tasks such as producing images of a desired style. Existing methods employ either token optimization techniques [1, 14, 49, 50, 55, 56], fine-tuning the model's weights [42], or a combination of both [3–5, 7]. Token optimization requires longer training times and often results in sub-optimal reconstruction. While model fine-tuning provides better reconstruction, it consumes substantial memory and tends to overfit. To address the memory inefficiency, and to facilitate more efficient model fine-tuning, Parameter Efficient Fine-Tuning (PEFT) approaches have been proposed [23, 24, 31]. StyleDrop [45] utilizes Muse [9] as a base model, and adjusts its styles to align with a reference image. StyleDrop trains a lightweight adapter layer at the end of each attention block within the transformer model. However, similar to StyleAligned [21], their approach is designed for style adaptation, but for content preservation, another optimization is required. Among existing PEFT methods, Low-Rank Adaptation (LoRA) [24] is a popular fine-tuning technique, widely used by researchers and practitioners for its versatility and high-quality results.

LoRA for Image Stylization LoRA is often used for image stylization by fine-tuning a model to produce images of a desired style. Commonly, a LoRA is trained on a set of images, and then it is combined with control methods such as stylistic Concept-Sliders [15] or ControlNet [40,54], along with a text prompt to condition the generated image content. While LoRA-based approaches have demonstrated significant abilities in capturing style and content, two separate LoRA models are required for this task, and there is no trivial way to combine them. A common naïve approach is to combine two LoRAs by directly interpolating their weights [43], relying on a manual search for the desired coefficients. Alternative approaches [18, 37] propose an optimization-based strategy to find the optimal coefficients for such a combination. However, they focus on combining two objects and not on image stylization tasks.

Recently, Shah et al. introduced ZipLoRA [44], proposing to merge two individual LoRAs trained for style and content into a new 'zipped' LoRA by learning mixing coefficients for their columns. This work is closely related to ours, as we also mix LoRA weights trained on different images to facilitate image stylization. However, ZipLoRA requires an additional optimization stage for each new combination of content and style, thereby restricting the flexibility of reusing trained LoRA weights, which is LoRA's primary advantage. In contrast, our approach allows for the direct reuse of learned styles and contents without additional training, enhancing efficiency and versatility. Moreover, we demonstrate that our implicit approach is more robust to challenging styles and contents.

3 Preliminaries

SDXL Architecture In our work, we utilize the recently introduced publicly available text-to-image Stable Diffusion XL (SDXL) [38], which is an upgraded version of the known Stable Diffusion [41]. Both models are types of latent diffusion models (LDM), where the diffusion process is applied in the latent space of a pre-trained image autoencoder. The SDXL architecture leverages a three times larger UNet backbone compared to Stable Diffusion. The UNet consists of a total number of 70 attention layers. Each layer consists of a cross and self-attention. These attention layers are often referred to as attention blocks. In this paper, for clarity, we refer to them as *layers* so they are not confused with the larger transformer *blocks* we optimize. These attention layers are divided into 11 transformer blocks where the first two and last three blocks are comprised of four and six attention layers, respectively. The six inner blocks consist of 10 attention layers each, as illustrated in Figure 3).

Text condition generation is also extended in SDXL in the following way: given a text prompt y, it is encoded twice, with both OpenCLIP ViT-bigG [26] and CLIP ViT-L [39]. The resulting embeddings are then concatenated to define the conditioning encoding c. Then this text embedding is fed into the cross-attention layers of the network, following the attention mechanism [48].

Specifically, in each layer, the deep spatial features x are projected to a query matrix $Q = l_Q(x)$, and the textual embedding is projected to a key matrix

$K = l_K(c)$ and a value matrix $V = l_V(c)$ via learned linear projections l_Q, l_K, l_V. The attention maps are then defined by:

$$A_t = Softmax(\frac{QK^T}{\sqrt{d}})V, \tag{1}$$

where d is the latent projection dimension of the keys and queries.

LoRA Low-Rank Adaptation [24] is a method for efficiently fine-tuning large pre-trained models for specific tasks or domains. LoRA has emerged as a very popular approach for fine-tuning pre-trained text-to-image diffusion models [43] due to its high-quality results and efficiency.

Let us denote the weights of a pre-trained text-to-image diffusion model with W_0, and the learned residuals after fine-tuning the model for a specific task with ΔW. The key idea in LoRA is that $\Delta W \in \mathbb{R}^{m \times n}$ can be decomposed into two low intrinsic rank matrices $B \in \mathbb{R}^{m \times r}$ and $A \in \mathbb{R}^{r \times n}$, such that $\Delta W = BA$, and the rank $r << \min(m, n)$. During training, the original model weights W_0 remain frozen, and only A and B are updated. Thus, by the end of the training, we can obtain the tuned model weights by using $W = W_0 + \Delta W$.

LoRA is commonly used in text-to-image diffusion models only in the cross and self-attention layers. As discussed, the attention mechanism in each layer relies on four projection matrices: l_Q, l_K, l_V, and l_{out}. The LoRA weights ΔW_Q, ΔW_K, ΔW_V, and ΔW_{out} are optimized for each of these pre-trained matrices. We denote by ΔW the LoRA weights of all four matrices.

4 Method

Our objective is to decouple the style and content aspects of an input image I into separate components, enabling both text-based and image-based stylization applications. Our approach harnesses the capabilities of a pre-trained SDXL text-to-image generation model [38], known for its robustness in capturing stylistic features [44]. We conduct an analysis of the SDXL architecture to gain insight into the contributions of individual layers to either the style or the content of the generated image. Guided by our observations, we employ LoRA [24] to train update matrices of only two specific transformer blocks within the SDXL model. These matrices capture the representation of the content and the style of the input image and they suffice to facilitate a number of image stylization tasks.

4.1 SDXL Architecture Analysis

Similar to previous works [1, 50] we examine the effect of different layers within the base text-to-image model on the generated image. We adopt a similar approach to the one proposed in Voynov et al. [50]. The key idea is to inject a different text prompt into the cross-attention layers of one of the transformer blocks within SDXL. Then examine the similarity between the different prompts and the resulting image. If we only change the input prompt corresponding to

the i'th block, and the i'th block dominates a certain quality of the generated image, this will be apparent in the resulting image. Specifically, we examine six intermediate transformer blocks $\{W_0^1, ..W_0^6\}$ of SDXL, each containing 10 attention layers (see Figure 3). These layers have been selected based on previous works [1,50], which demonstrate that they are most likely to affect the important visual properties of the generated images.

We define two random sets of text prompts $P_{content}$ and P_{style} describing different objects with different colors. The prompts in $P_{content}$ are defined by placing random objects in the template text "A photo of a <object>". For P_{style} we use the template "A photo of a <color> <object>". The random objects and colors are generated with ChatGPT. Note that color is used as a proxy for style since we use CLIP [39] to evaluate results (as will be described next), and we found CLIP to be a better indicator for changes in color than changes in style. We sample a pair of prompts $(p, \hat{p}) \in P_{content}$ and $(p, \hat{p}) \in P_{style}$ such that $p \neq \hat{p}$.

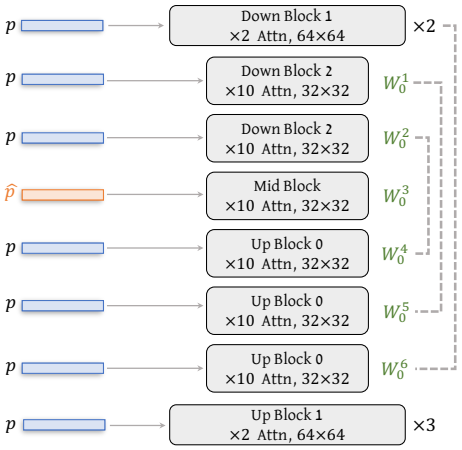

Fig. 3: Illustration of SDXL architecture and our text-based analysis. To examine the effect of the i'th transformer block on the generated image, we inject a different text prompt $\hat{p}$ to it, while p is injected into all other blocks.

For each pair $(\hat{p}, p)$, we generate an image $I_{\hat{p}\to i, p\to j\neq i}$ by injecting the embedding of $\hat{p}$ to W_0^i while injecting the embedding of p to all other layers $W_0^j, j \neq i$ (illustrated in Figure 3). This is performed for each of the six transformer blocks we target, resulting in six images per pair.

Next, to measure the effect of injecting $\hat{p}$ into the i'th block on the generated image, we estimate the following similarity score:

$$\mathcal{C}(I_{\hat{p}\to i, p\to j}, \hat{p}) = sim(CLIP_I(I_{\hat{p}\to i, p\to j}), CLIP_T(\hat{p})), \tag{2}$$

where $CLIP_I(I_{\hat{p}\to i, p\to j})$ and $CLIP_T(\hat{p})$ are the CLIP image embedding of the generated image, and the CLIP text embedding of the prompt, respectively. $sim(x, y) = \frac{x \cdot y}{||x|| \cdot ||y||}$ indicates the cosine similarity between the clip embeddings.

In total, we examined 400 pairs of content and style prompts and averaged the scores of each layer. The three topmost layers that show similarity to one type of prompt are W_0^2 and W_0^4 which dominate the content of the generated image, and W_0^5 which dominates its color. We visually demonstrate these conclusions in Figure 4. On the left, we show the effect of blocks 2 and 4 on the generated content. Note that $I_{\hat{p}\to 2, p\to j}$ and $I_{\hat{p}\to 4, p\to j}$ demonstrate that when "A photo of

a tiger" is injected to only one block (2 or 4), while "A photo of a bunny" is injected to the rest of the blocks, the generated images depict a tiger, while in all other options, the generated image will depict a bunny. Similarly, on the right we show the effect of block 5 on the generated image's color.

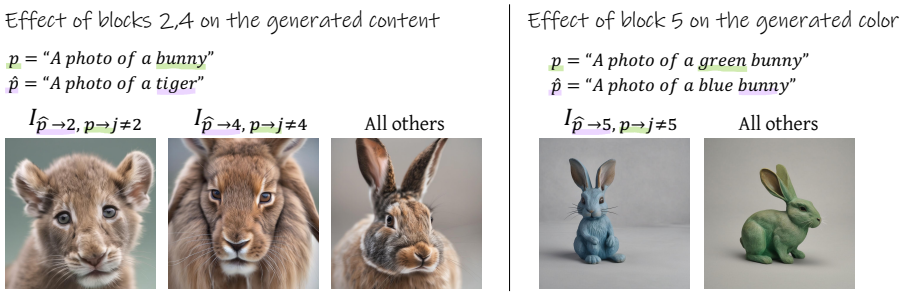

Fig. 4: Prompt injection effect on the generated image. On the left, we demonstrate how blocks 2 and 4 affect the content in the generated image (turning into a tiger), whereas the rightmost image shows that injecting $\hat{p}$ to a block $i \neq 2, 4$ has no effect on the generated image. On the right we show how the fifth block controls the generated image's color.

4.2 LoRA-Based Separation with B-LoRA

While the observations above apply to a *generated* image, our goal is to examine if the layers we locate could be useful in capturing the content and style of a *given* input image I. To fine-tune the model to generate variations of our given image we utilize the LoRA [24] approach.

Let us denote the frozen weights of our base pre-trained SDXL model with W_0 and the learned residual matrices for each block with ΔW^i. We follow the default settings of DreamBooth LoRA [43] to finetune the model to reconstruct the given input image I.

However, instead of optimizing the LoRA weights of all eleven blocks (as usually done), we conduct two experiments, where in the first experiment we optimize the pair $\{\Delta W^2, \Delta W^5\}$, and in the second experiment we optimize $\{\Delta W^4, \Delta W^5\}$ (as we found W_0^2 and W_0^4 to dominate the content, and W_0^5 to dominate the color). In addition, we use a general prompt "A [v]" during training to prevent the model from being explicitly guided to capture either the image's style or content. This process and example results are depicted in Figure 5. As can be seen, we find that the best combination to optimize in terms of 1. Achieving a full reconstruction of the input concept, and 2. Capturing the input image's content, are $\Delta W^4, \Delta W^5$. Note that using the deeper layers of the UNet ΔW^4, rather than ΔW^2 during the LoRA training process, aligns with the goal of preserving finer details in the output image, as demonstrated in [47]. We

provide ablation and analysis of the effect of other layers and specific parts within them, as well as the effect of using different text prompts in the supplementary material.

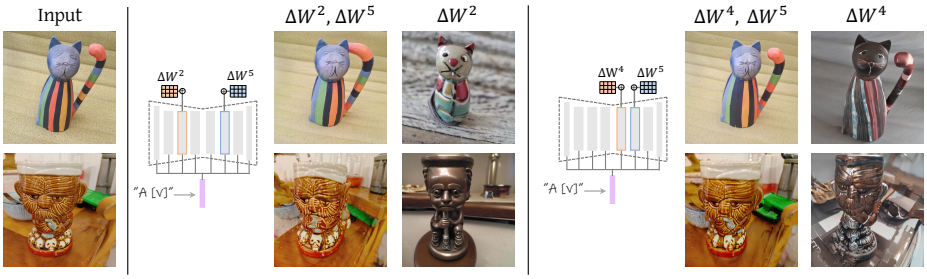

Fig. 5: Comparison of training B-LoRAs for the input images shown on the left for W_0^2, W_0^5 (middle) and W_0^4, W_0^5 (right). For each pair of trained LoRA weights, we show the results of applying both together (to reconstruct the input image) and applying the content layer separately (i.e. using only ΔW^2 and ΔW^4). The results demonstrate that ΔW^4 better captures the fine details of the input object.

We call such a training scheme *B-LoRA*, as it only trains two transformer Blocks instead of the full weights. Hence, apart from the style-content separation abilities such a method also reduces storage requirements by 70%.

4.3 B-LoRA for Image Stylization

Combining the insights from the above analyses, we now describe the B-LoRA training approach. Given an input image I, we only fine-tune the LoRA weights $\Delta W^4, \Delta W^5$ with the objective of reconstructing the image, w.r.t a general text prompt "A [v]". Besides increasing efficiency, we find that by training only these two layers, we can achieve an **implicit** style-content decomposition, where ΔW^4 captures the content and ΔW^5 captures the style.

Once we find these update matrices, we can easily use them by updating the corresponding block weights of the pre-trained SDXL model for style manipulation applications as described next and demonstrated in Figure 6.

Image stylization based on image style reference Given two input images I_c, I_s depicting the desired content and style respectively, we use the process described above to learn their corresponding B-LoRA weights: $\Delta W_c^4, \Delta W_c^5$ for I_c and $\Delta W_s^4, \Delta W_s^5$ for I_s. We then directly use ΔW_c^4 and ΔW_s^5 to update the transformer blocks W_0^4 and W_0^5 of the pre-trained network. For the inference process, we use the prompt "A [c] in [s] style", as illustrated at the top of Figure 6.

Text-based image stylization By omitting ΔW_c^5 (capturing the style of I_c) and only using ΔW_c^4 to update the weights of the pre-trained model, we get a personalized model that is adapted to only the content of I_c. To manipulate the style of

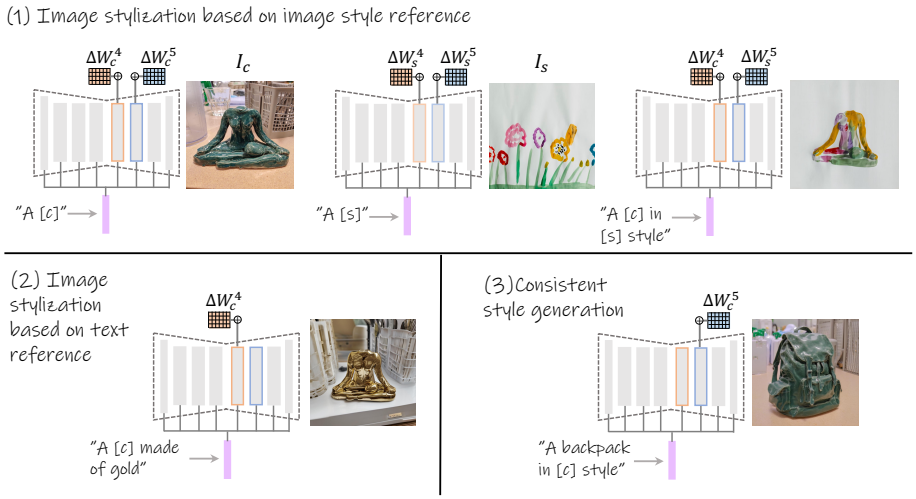

Fig. 6: B-LoRA for Image Stylization. (1) To stylize a given content image I_c w.r.t an given style image reference I_s, we train our B-LoRAs for both images and then combine ΔW_c^4 and ΔW_s^5 to a single adapted model. (2) For text-based stylization we simply plug only the trained ΔW_c^4 to adapt the model and then use the desired text prompt during inference. (3) The learned style weights ΔW_c^5 can be also used as is to adjust the backbone model to produce images with the style of I_c.

I_c with text-based guidance, we simply inject the desired text into the adapted layers during inference (see Figure 6 bottom-left). Note that because the style and content are separated and encoded in different blocks, our approach allows challenging style manipulations.

Consistent style generation Lastly, in a similar manner, one can adapt the model for a specific style provided in I_s by excluding ΔW_s^4 and using only ΔW_s^5. This results in a model adapted to the desired style, and one can use text-based conditions to generate any content with the desired style (see Figure 6 bottom-right).

4.4 Implementation details

We train the B-LoRA weights on SDXL v1.0 [38] while keeping both the model weights and text encoders frozen during the fine-tuning process. All LoRA training was performed on a single image. We utilize the Adam optimizer with a learning rate of $5e-5$. For data augmentations, we only use center cropping during training. We set the LoRA weights rank to $r = 64$ and use the prompt "A [v]" for 1000 optimization steps, requiring approximately 10 minutes per image on a single A100 GPU. Note that while other methods typically train LoRA for 400 steps to mitigate overfitting concerns, this was not an issue in our case.

5 Results

To produce the various results of our approach we optimized our B-LoRAs ($\Delta W^4, \Delta W^5$) once for each image and then plugged either one of them or both of them (depending on the application) at inference time to receive image stylization without any further optimization or fine-tuning.

We present some qualitative results of the three applications discussed in Section 4.3 in Figure 7. In the first two rows of Figure 7, our method manages to transfer the style of the image references (top row) while preserving the content of the input image on the left. Notable, this can be done for challenging content inputs such as stylized images (first row) and images of whole scenes (second row). Our method is robust to many types of different styles and manages to preserve the essence of the content reference even in very abstract styles such as the one depicted in the third style column. In the third row, we show examples of text-based image stylization. As can be seen with our implicit style-content separation, the content of the input object is preserved well while the style is governed by the desired text prompt. In the last row, we demonstrate how our method can be used for consistent style generation where only the B-LoRA weights of the style are used. Observe that the object's style is well preserved across all text-based generated images. Please refer to the supplementary material for many more examples.

Comparisons We next compare our method with alternative approaches, both qualitatively and quantitatively. Note that since we rely on SDXL as our backbone model, for a fair comparison we applied alternative approaches on SDXL as well. As a naïve baseline we employ DB-LoRA [43] (fine-tuned for style) with a ControlNet [54] for content conditioning. We additionally compare to three recent approaches for image stylization that rely on the prior of large pre-trained text-to-image models, namely, ZipLoRA [44], StyleDrop [45], and StyleAligned [21]. StyleAligned is applied using the author's official implementation. With the lack of official implementations for StyleDrop and ZipLoRA, we implemented StyleDrop on SDXL (as described in [21]), and utilized a non-official implementation of ZipLoRA [33].

Note that for content preservation, all three alternative methods require *multiple* content images, while our method can be applied to a *single* image. Thus, for a fair comparison, we collected a total set of 23 objects from existing personalization works [14, 31, 42, 49], where a small set of images is provided for each object. We collected 20 style image references from [21, 45], along with 5 additional style images of our own. From these sets, we randomly sampled 50 pairs of style and content images to compose our final evaluation set.

In terms of runtime, StyleAligned is zero-shot only for consistent style generation, while for content preservation it relies on LoRA to adapt the model to the desired concepts. Similarly, StyleDrop and ZipLoRA require LoRA training for content and style. Thus, our runtime is comparable to theirs. However, ZipLoRA entails an extra training phase to merge the two LoRAs, making it more time-consuming than our approach.

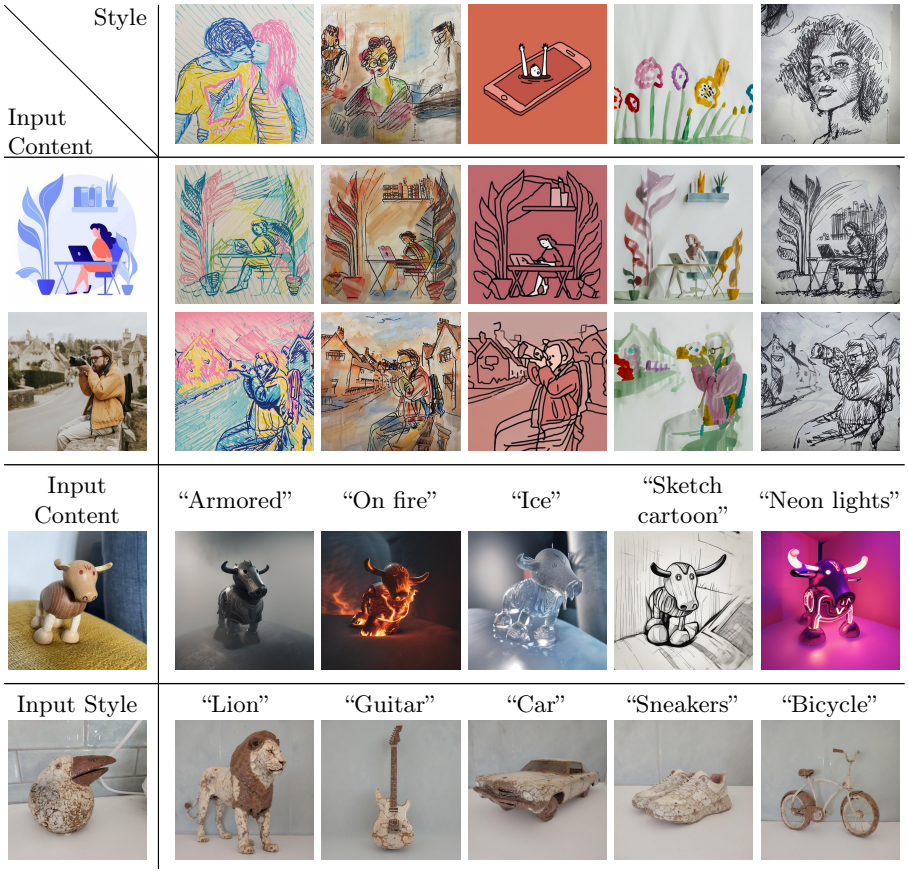

Fig. 7: Results produced by our method for three image stylization tasks. Rows 1-3: image style transfer. Our method can operate on scene images and extract content from a stylized image. Fourth row: text-based image stylization applied to the content image reference on the left. Note how the pose and identity are preserved well. Last row: consistent style generation, where that style is extracted from the image on the left and used to generate new objects. In this row, we use $\alpha = 1.1$ to enhance the style effect.

Qualitative Evaluation We show representative comparison results in Figure 8, where on the left we show the style and content reference images. On the first four rows, we show the results of alternative approaches when applied with multiple content images, whereas our method uses a single image. As can be seen, our method effectively preserves the subject from the content image while transferring the desired style. In contrast, other methods either overfit the content subject, thereby failing to alter its style (e.g., cat and sloth in ZipLoRA and StyleDrop), or they suffer from style image "leakage". For instance, in the cat example of StyleAligned (first row), the model generates two cats, matching

Fig. 8: Comparison with alternative approaches. The input style and content references are shown on the left, where multiple content images were used for alternative methods. In the last row, we applied other approaches to a single content image. ZipLoRA tends to overfit the content, and thus struggles with depicting the desired style. StyleDrop also struggles to preserve the content when trained on multiple images. In the case of a single content image (last row), both methods preserve the content but lose the style. StyleAligned preserves the style well; however, it tends to include semantic content originating in the *style* image, such as creating a couple in row 1.Additional comparisons to InstantStyle [51] are provided in the supplementary material.

the number of people in the style reference image. We also include an example of alternative methods applied to a single content image, where StyleDrop and ZipLoRA exhibit increased overfitting.

Quantitative Evaluation We measure content and style preservation by computing the cosine similarity between the embeddings of the input content and style references and the output image, utilizing the DINO ViT-B/8 embeddings [8]. The average scores are presented in Table 1. Our method achieves the highest style alignment score, indicating its superior ability to adapt styles effectively. However, we observe lower object similarity scores, possibly due to content overfitting issues observed in alternative approaches. To further support this observation, we conducted the same experiment using a single content image as a reference (scores shown in the "single" row). The results indicate a decrease

in style consistency scores across all methods, accompanied by an increase in content preservation scores, suggesting overfitting.

User Study We conducted a user study to further validate the findings presented above. Using 30 random images from our evaluation set, we compared our results with the three alternative approaches. The participants were presented with the reference style and content images along with two combined results, one produced by our method and the other by an alternative method (with the results presented in random order). Participants were asked to choose the result that "better transfers the style from the style image while preserving the content of the content image". We collected responses from 34 participants for the survey, which contained a total of 1020 answers. The results demonstrate a strong preference for our method, with 94% of participants favoring our method over StyleAligned, 91% over ZipLoRA, and 88% over StyleDrop.

Table 1: Quantitative comparison. We measure the average cosine similarity between the DINO features of the output image and the reference style and content. Our method performs best at adapting to the style without overfitting the content image.

	Input	StyleDrop	StyleAligned	ZipLoRA	DB-LoRA	Ours
Style Transfer	Multiple	0.826 ± 0.07	0.855 ± 0.05	0.796 ± 0.07	0.863 ± 0.06	$\mathbf{0.881} \pm 0.05$
	Single	0.790 ± 0.06	0.829 ± 0.05	0.782 ± 0.05		
Content	Multiple	0.817 ± 0.06	0.779 ± 0.05	$\mathbf{0.841} \pm 0.05$	0.769 ± 0.05	0.790 ± 0.05
	Single	0.874 ± 0.08	0.792 ± 0.06	$\mathbf{0.933} \pm 0.05$		

6 Conclusions and Future work

We have presented a simple yet effective method to disentangle the style and content of a single input image. The style and content components are encoded separately with two B-LoRAs, providing high flexibility for independent use in various image stylization tasks. In contrast to existing methods that focus on style extraction, we employ a compound style-content learning approach that enables a better separation of style and content, enhancing stylization fidelity.

As for future research, one possible avenue is to further explore separation techniques within LoRA fine-tuning, to achieve more concrete separation into sub-components such as structure, shape, color, texture, etc. This could provide users with more control over the desired output. Another direction for future work is to leverage the robustness of our approach and extend it to combine LoRA weights from multiple distinct objects or combine a few styles.

Acknowledgements

We would like to thank Amir Hertz and Yuval Alaluf for their insightful feedback. Additionally, some of the artistic paintings presented in this paper were created by the artist Judith Kondor Mochary. We thank the artist's family for granting us the privilege to use Judith's drawings. This work was supported by the Israel Science Foundation under Grant No. 2492/20, 3441/21 and 1390/19, and Joint NSFC-ISF Research Grant Research Grant no. 3077/23.

References

1. Agarwal, A., Karanam, S., Shukla, T., Srinivasan, B.V.: An image is worth multiple words: Multi-attribute inversion for constrained text-to-image synthesis. ArXiv **abs/2311.11919** (2023), https://api.semanticscholar.org/CorpusID:265295502
2. Alaluf, Y., Garibi, D., Patashnik, O., Averbuch-Elor, H., Cohen-Or, D.: Cross-image attention for zero-shot appearance transfer. ArXiv **abs/2311.03335** (2023), https://api.semanticscholar.org/CorpusID:265043677
3. Alaluf, Y., Richardson, E., Metzer, G., Cohen-Or, D.: A neural space-time representation for text-to-image personalization. ACM Transactions on Graphics (TOG) **42**(6), 1–10 (2023)
4. Arar, M., Voynov, A., Hertz, A., Avrahami, O., Fruchter, S., Pritch, Y., Cohen-Or, D., Shamir, A.: Palp: Prompt aligned personalization of text-to-image models. ArXiv **abs/2401.06105** (2024), https://api.semanticscholar.org/CorpusID:266933184
5. Avrahami, O., Aberman, K., Fried, O., Cohen-Or, D., Lischinski, D.: Break-a-scene: Extracting multiple concepts from a single image. In: SIGGRAPH Asia 2023 Conference Papers. SA '23, Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3610548.3618154, https://doi.org/10.1145/3610548.3618154
6. Avrahami, O., Fried, O., Lischinski, D.: Blended latent diffusion. ACM Transactions on Graphics (TOG) **42**, 1 – 11 (2022), https://api.semanticscholar.org/CorpusID:249394540
7. Avrahami, O., Hertz, A., Vinker, Y., Arar, M., Fruchter, S., Fried, O., Cohen-Or, D., Lischinski, D.: The chosen one: Consistent characters in text-to-image diffusion models. arXiv preprint arXiv:2311.10093 (2023)
8. Caron, M., Touvron, H., Misra, I., J'egou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. 2021 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 9630–9640 (2021), https://api.semanticscholar.org/CorpusID:233444273
9. Chang, H., Zhang, H., Barber, J., Maschinot, A., Lezama, J., Jiang, L., Yang, M., Murphy, K.P., Freeman, W.T., Rubinstein, M., Li, Y., Krishnan, D.: Muse: Text-to-image generation via masked generative transformers. ArXiv **abs/2301.00704** (2023), https://api.semanticscholar.org/CorpusID:255372955
10. Chen, M., Laina, I., Vedaldi, A.: Training-free layout control with cross-attention guidance. ArXiv **abs/2304.03373** (2023), https://api.semanticscholar.org/CorpusID:258041377

11. Couairon, G., Verbeek, J., Schwenk, H., Cord, M.: Diffedit: Diffusion-based semantic image editing with mask guidance. ArXiv **abs/2210.11427** (2022), https://api.semanticscholar.org/CorpusID:253018768
12. Efros, A.A., Freeman, W.T.: Image quilting for texture synthesis and transfer. Proceedings of the 28th annual conference on Computer graphics and interactive techniques (2001), https://api.semanticscholar.org/CorpusID:9334387
13. Epstein, D., Jabri, A., Poole, B., Efros, A.A., Holynski, A.: Diffusion self-guidance for controllable image generation. ArXiv **abs/2306.00986** (2023), https://api.semanticscholar.org/CorpusID:258999106
14. Gal, R., Alaluf, Y., Atzmon, Y., Patashnik, O., Bermano, A.H., Chechik, G., Cohen-Or, D.: An image is worth one word: Personalizing text-to-image generation using textual inversion. arXiv preprint arXiv:2208.01618 (2022)
15. Gandikota, R., Materzynska, J., Zhou, T., Torralba, A., Bau, D.: Concept sliders: Lora adaptors for precise control in diffusion models. ArXiv **abs/2311.12092** (2023), https://api.semanticscholar.org/CorpusID:265308675
16. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 2414–2423 (2016), https://api.semanticscholar.org/CorpusID:206593710
17. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A.C., Bengio, Y.: Generative adversarial networks. Communications of the ACM **63**, 139 – 144 (2014), https://api.semanticscholar.org/CorpusID:1033682
18. Gu, Y., Wang, X., Wu, J.Z., Shi, Y., Chen, Y., Fan, Z., Xiao, W., Zhao, R., Chang, S., Wu, W., Ge, Y., Shan, Y., Shou, M.Z.: Mix-of-show: Decentralized low-rank adaptation for multi-concept customization of diffusion models. ArXiv **abs/2305.18292** (2023), https://api.semanticscholar.org/CorpusID:258960192
19. Han, L., Li, Y., Zhang, H., Milanfar, P., Metaxas, D.N., Yang, F.: Svdiff: Compact parameter space for diffusion fine-tuning. 2023 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 7289–7300 (2023), https://api.semanticscholar.org/CorpusID:257631648
20. Hertz, A., Mokady, R., Tenenbaum, J.M., Aberman, K., Pritch, Y., Cohen-Or, D.: Prompt-to-prompt image editing with cross attention control. ArXiv **abs/2208.01626** (2022), https://api.semanticscholar.org/CorpusID:251252882
21. Hertz, A., Voynov, A., Fruchter, S., Cohen-Or, D.: Style aligned image generation via shared attention. arXiv preprint arXiv:2312.02133 (2023)
22. Hertzmann, A., Jacobs, C.E., Oliver, N., Curless, B., Salesin, D.H.: Image analogies p. 327–340 (2001). https://doi.org/10.1145/383259.383295, https://doi.org/10.1145/383259.383295
23. Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., de Laroussilhe, Q., Gesmundo, A., Attariyan, M., Gelly, S.: Parameter-efficient transfer learning for nlp. ArXiv **abs/1902.00751** (2019), https://api.semanticscholar.org/CorpusID:59599816
24. Hu, J.E., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Chen, W.: Lora: Low-rank adaptation of large language models. ArXiv **abs/2106.09685** (2021), https://api.semanticscholar.org/CorpusID:235458009
25. Huang, X., Belongie, S.J.: Arbitrary style transfer in real-time with adaptive instance normalization. 2017 IEEE International Conference on Computer Vision

(ICCV) pp. 1510–1519 (2017), https://api.semanticscholar.org/CorpusID:6576859
26. Ilharco, G., Wortsman, M., Wightman, R., Gordon, C., Carlini, N., Taori, R., Dave, A., Shankar, V., Namkoong, H., Miller, J., Hajishirzi, H., Farhadi, A., Schmidt, L.: Openclip (Jul 2021). https://doi.org/10.5281/zenodo.5143773, https://doi.org/10.5281/zenodo.5143773, if you use this software, please cite it as below.
27. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 5967–5976 (2016), https://api.semanticscholar.org/CorpusID:6200260
28. Jing, Y., Yang, Y., Feng, Z., Ye, J., Song, M.: Neural style transfer: A review. IEEE Transactions on Visualization and Computer Graphics **26**, 3365–3385 (2017), https://api.semanticscholar.org/CorpusID:4875951
29. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. ArXiv **abs/1603.08155** (2016), https://api.semanticscholar.org/CorpusID:980236
30. Katzir, O., Lischinski, D., Cohen-Or, D.: Cross-domain cascaded deep translation. In: European Conference on Computer Vision (2020), https://api.semanticscholar.org/CorpusID:209315529
31. Kumari, N., Zhang, B., Zhang, R., Shechtman, E., Zhu, J.Y.: Multi-concept customization of text-to-image diffusion. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 1931–1941 (2022), https://api.semanticscholar.org/CorpusID:254408780
32. Li, S., van de Weijer, J., Hu, T., Khan, F.S., Hou, Q., Wang, Y., Yang, J.: Stylediffusion: Prompt-embedding inversion for text-based editing. ArXiv **abs/2303.15649** (2023), https://api.semanticscholar.org/CorpusID:257771440
33. mkshing: Ziplora-pytorch. https://github.com/mkshing/ziplora-pytorch
34. Mokady, R., Hertz, A., Aberman, K., Pritch, Y., Cohen-Or, D.: Null-text inversion for editing real images using guided diffusion models. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023. pp. 6038–6047. IEEE (2023). https://doi.org/10.1109/CVPR52729.2023.00585, https://doi.org/10.1109/CVPR52729.2023.00585
35. Park, T., Efros, A.A., Zhang, R., Zhu, J.Y.: Contrastive learning for unpaired image-to-image translation. In: European Conference on Computer Vision (2020), https://api.semanticscholar.org/CorpusID:220871180
36. Parmar, G., Singh, K.K., Zhang, R., Li, Y., Lu, J., Zhu, J.Y.: Zero-shot image-to-image translation. ACM SIGGRAPH 2023 Conference Proceedings (2023), https://api.semanticscholar.org/CorpusID:256616002
37. Po, R., Yang, G., Aberman, K., Wetzstein, G.: Orthogonal adaptation for modular customization of diffusion models. ArXiv **abs/2312.02432** (2023), https://api.semanticscholar.org/CorpusID:265659333
38. Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Muller, J., Penna, J., Rombach, R.: Sdxl: Improving latent diffusion models for high-resolution image synthesis. ArXiv **abs/2307.01952** (2023), https://api.semanticscholar.org/CorpusID:259341735
39. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning (2021), https://api.semanticscholar.org/CorpusID:231591445

40. Research, F.: Cog sdxl canny controlnet with lora support. https://replicate.com/batouresearch/sdxl-controlnet-lora
41. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 10674–10685 (2021), https://api.semanticscholar.org/CorpusID:245335280
42. Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., Aberman, K.: 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. pp. 22500–22510 (2023)
43. Ryu, S.: Low-rank adaptation for fast text-to-image diffusion fine-tuning. https://github.com/cloneofsimo/lora
44. Shah, V., Ruiz, N., Cole, F., Lu, E., Lazebnik, S., Li, Y., Jampani, V.: Ziplora: Any subject in any style by effectively merging loras. arXiv preprint arXiv:2311.13600 (2023)
45. Sohn, K., Ruiz, N., Lee, K., Chin, D.C., Blok, I., Chang, H., Barber, J., Jiang, L., Entis, G., Li, Y., Hao, Y., Essa, I., Rubinstein, M., Krishnan, D.: Styledrop: Text-to-image generation in any style (2023)
46. Tenenbaum, J., Freeman, W.: Separating style and content. In: Mozer, M., Jordan, M., Petsche, T. (eds.) Advances in Neural Information Processing Systems. vol. 9. MIT Press (1996), https://proceedings.neurips.cc/paper_files/paper/1996/file/70222949cc0db89ab32c9969754d4758-Paper.pdf
47. Tumanyan, N., Geyer, M., Bagon, S., Dekel, T.: Plug-and-play diffusion features for text-driven image-to-image translation. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 1921–1930 (2022), https://api.semanticscholar.org/CorpusID:253801961
48. Vaswani, A., Shazeer, N.M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Neural Information Processing Systems (2017), https://api.semanticscholar.org/CorpusID:13756489
49. Vinker, Y., Voynov, A., Cohen-Or, D., Shamir, A.: Concept decomposition for visual exploration and inspiration. ACM Trans. Graph. **42**(6) (dec 2023). https://doi.org/10.1145/3618315, https://doi.org/10.1145/3618315
50. Voynov, A., Chu, Q., Cohen-Or, D., Aberman, K.: $p+$: Extended textual conditioning in text-to-image generation. arXiv preprint arXiv:2303.09522 (2023)
51. Wang, H., Spinelli, M., Wang, Q., Bai, X., Qin, Z., Chen, A.: Instantstyle: Free lunch towards style-preserving in text-to-image generation. ArXiv **abs/2404.02733** (2024), https://api.semanticscholar.org/CorpusID:268876474
52. Yang, S., joo Hwang, H., Ye, J.C.: Zero-shot contrastive loss for text-guided diffusion image style transfer. 2023 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 22816–22825 (2023), https://api.semanticscholar.org/CorpusID:257532402
53. Ye, H., Zhang, J., Liu, S., Han, X., Yang, W.: Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. ArXiv **abs/2308.06721** (2023), https://api.semanticscholar.org/CorpusID:260886966
54. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. 2023 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 3813–3824 (2023), https://api.semanticscholar.org/CorpusID:256827727

55. xin Zhang, Y., Huang, N., Tang, F., Huang, H., Ma, C., Dong, W., Xu, C.: Inversion-based style transfer with diffusion models. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 10146–10156 (2022), https://api.semanticscholar.org/CorpusID:257427673
56. Zhang, Y., Dong, W., Tang, F., Huang, N., Huang, H., Ma, C., Lee, T.Y., Deussen, O., Xu, C.: Prospect: Prompt spectrum for attribute-aware personalization of diffusion models. ACM Transactions on Graphics (TOG) **42**(6), 1–14 (2023)
57. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. 2017 IEEE International Conference on Computer Vision (ICCV) pp. 2242–2251 (2017), https://api.semanticscholar.org/CorpusID:206770979