

# Tuning-Free Visual Customization via View Iterative Self-Attention Control

Xiaojie Li<sup>1</sup>, Chenghao Gu<sup>2</sup>, Shuzhao Xie<sup>1</sup>, Yunpeng Bai<sup>3</sup>, Weixiang Zhang<sup>1</sup>, Zhi Wang<sup>1\*</sup>

<sup>1</sup> Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen, China

<sup>2</sup> Jiluan Academy, Nanchang University, Nanchang, China

<sup>3</sup> Department of Computer Science, The University of Texas at Austin, US

{li-xj23, zhang-wx22}@mails.tsinghua.edu.cn, wangzhi@sz.tsinghua.edu.cn

## Abstract

Fine-Tuning Diffusion Models enable a wide range of personalized generation and editing applications on diverse visual modalities. While Low-Rank Adaptation (LoRA) accelerates the fine-tuning process, it still requires multiple reference images and time-consuming training, which constrains its scalability for large-scale and real-time applications. In this paper, we propose *View Iterative Self-Attention Control (VisCtrl)* to tackle this challenge. Specifically, *VisCtrl* is a training-free method that injects the appearance and structure of a user-specified subject into another subject in the target image, unlike previous approaches that require fine-tuning the model. Initially, we obtain the initial noise for both the reference and target images through DDIM inversion. Then, during the denoising phase, features from the reference image are injected into the target image via the self-attention mechanism. Notably, by iteratively performing this feature injection process, we ensure that the reference image features are gradually integrated into the target image. This approach results in consistent and harmonious editing with only one reference image in a few denoising steps. Moreover, benefiting from our plug-and-play architecture design and the proposed Feature Gradual Sampling strategy for multi-view editing, our method can be easily extended to edit in complex visual domains. Extensive experiments show the efficacy of *VisCtrl* across a spectrum of tasks, including personalized editing of images, videos, and 3D scenes.

## 1 Introduction

Imagine a world where visual creativity knows no bounds, liberated from the drudgery of manual editing and long waits. In this realm, you can swiftly manipulate diverse visual scenes: seamlessly integrating your beloved cat into any photograph, tailoring landscapes to your liking within VR/AR, or substituting individuals in videos with anyone you choose. This question lies at the heart of a challenging task—*rapidly personalized visual editing* which involves efficiently injecting user-specified visual features (e.g. appearance and structure) into the target visual representation.

The solutions for the personalized visual editing task fall into two paradigms: model-based and attention-based methods. Model-based methods [1, 2, 3] focus on collecting datasets to fine-tune the entire model, which requires substantial time and computational resources. To avoid the costly process, attention-based methods [4, 5, 6] have been proposed, with a special focus on manipulating the attention in the UNet of the diffusion model. Prompt-to-Prompt [4] can edit images by injecting cross-attention maps during the diffusion process through editing only the textual prompt. MasaCtrl [5] utilizes mutual self-attention to achieve non-rigid and consistent image editing by querying correlated

---

\*Corresponding author

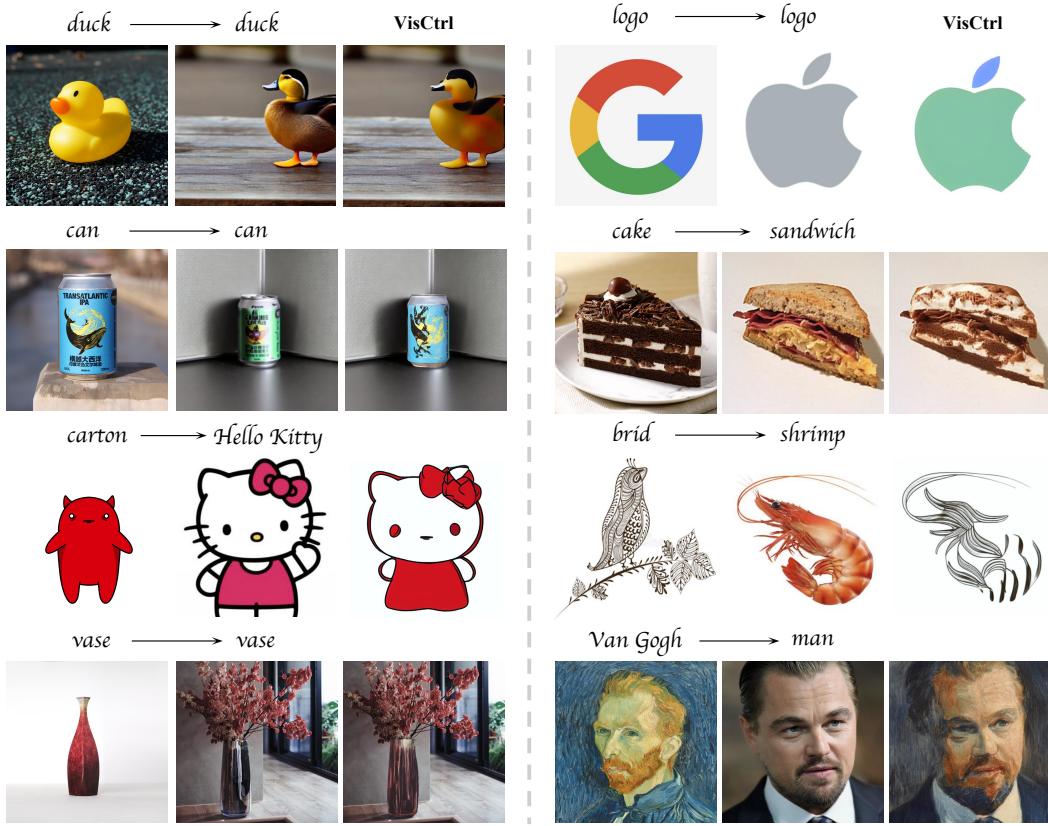


Figure 1: **VisCtrl** results span across various object and image domains, showcasing its broad applicability. From simple objects (cartoons, logos) to complex subjects (food, humans), the diversity in personalized image editing highlights the versatility and robustness of our framework across different usage scenarios.

local contents and textures from the source image. The editing methods for other visual modalities, such as video and 3D scenes, mostly build upon the aforementioned image editing techniques [7, 8]

However, previous methods still face several challenges in the efficiency of personalized visual editing: **1)** The prolonged DDIM inversion process causes the intermediate codes to diverge from the original trajectory, leading to unsatisfying image reconstruction [9]. **2)** The inherent ambiguity and inaccuracy of text often result in significant disparities between the user’s desired content and the generated output [5]. Furthermore, even minor adjustments to prompts in most text-to-image models can result in significantly different images [4]. **3)** These methods lack support for other visual representations, hindering their extension to video and 3D scene editing.

To tackle these challenges, we propose View Iterative Self-Attention Control (**VisCtrl**), a simple but effective framework that utilizes self-attention to inject personalized subject features into the target image. Specifically, we firstly obtain the initial noise for both the reference image and the target image through DDIM inversion [10]. Subsequently, during denoising reconstruction, we iteratively inject the features of user-specified subject into the target image using self-attention, while maintaining the overall structure of the target image using cross-attention. Additionally, we propose a Feature Gradually Sampling strategy for complex visual editing, which involves randomly sampling the latent feature from the reference images to achieve multi-view editing. Remarkably, We can generate outstanding results in Figure 1 with few denoising steps using only one reference image without retraining.

Our method is validated through extensive experiments and shows promise for extension to other visual personalized tasks. Our contributions are summarized as follows: **1)** We propose a training-free framework for image editing with only one reference image, emphasizing speed and efficiency. **2)** We

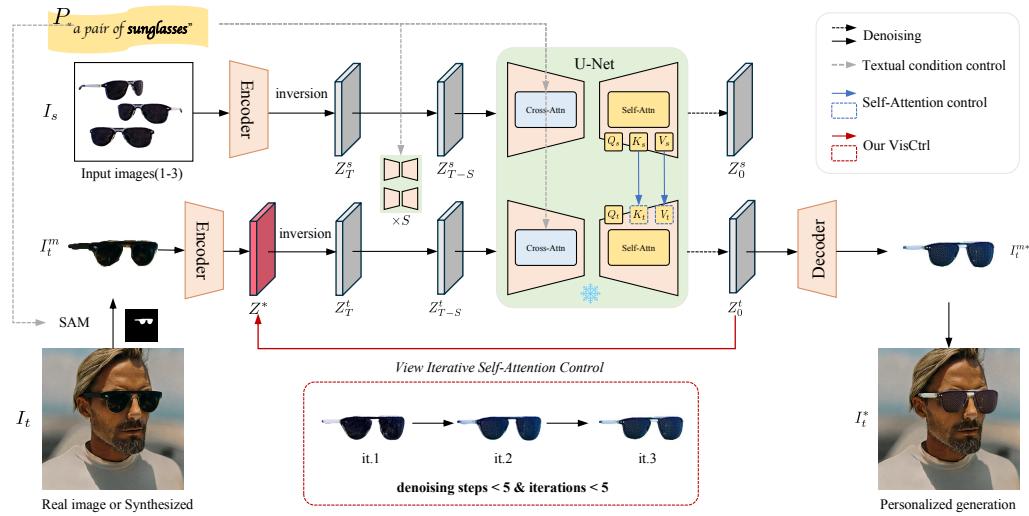


Figure 2: Pipeline of the proposed *VisCtrl*. Given one or several reference images of a new concept, we first encoder them to the latent space, followed by adding noise and denoising via DDIM [10]. The upper part of the process entails generating the reference image, while the bottom part involves generating the target image. Specifically, during the denoising process, we replace the  $K_t, V_t$  of the target image self-attention layer with  $K_s, V_s$  from the reference image self-attention layer. Additionally, we update  $Z^*$  with  $Z_0^t$  iteratively throughout this process. Finally, we decoder  $Z_0^t$  to obtain the target image. Please refer to Section 3.2 for further details.

propose an iterative self-attention control that utilizes the reference image and corresponding textual conditions to govern the editing process. 3) We propose a Feature Gradually Sampling strategy which effectively extends our framework to other visual domains, such as video and 3D scenes.

## 2 Related work

### 2.1 Text-guided Visual Generation and editing

Early image generation methods conditioned on text description mainly based on GANs [11, 12, 13, 14, 15], due to their powerful capability of high fidelity image synthesis. Recent advancements in Text-to-Image (T2I) generation have witnessed the scaling up of text-to-image diffusion models [16, 17, 18] through the utilization of billions of image-text pairs [19] and efficient architectures [20, 21, 22, 23, 24]. These models demonstrate remarkable proficiency in synthesizing high-quality, realistic, and diverse images guided by textual input. Additionally, they have extended their utility to various applications, including image-to-image translation [25, 26, 4, 27, 1, 9, 28], controllable generation [29], and personalization [30, 31]. Recent research has explored various extensions and applications of text-to-image (T2I) models. For instance, Tune-A-Video [32] utilizes T2I diffusion models to achieve high-quality video generation. Additionally, leveraging 3D representations such as NeRF [33] or 3D Gaussian splatting [34], T2I models have been employed for 3D object generation [35, 36, 37, 38, 39] and editing [7, 40]. Text-guided image editing has evolved from early GAN-based approaches [41, 42, 43, 44], which were limited to specific object domains, to more versatile diffusion-based methods [29, 18, 45]. However, existing diffusion model methods [18, 46, 25, 4, 9] often require manual masks for local editing, and struggle with layout preservation.

### 2.2 Subject-driven image editing

Exemplar-guided image editing covers a broad range of applications, and most of the works [47, 48] can be categorized as exemplar-based image translation tasks, conditioning on various information, such as stylized images [49, 50, 51], layouts [52, 53, 54], skeletons [53], sketches/edges [55]. With

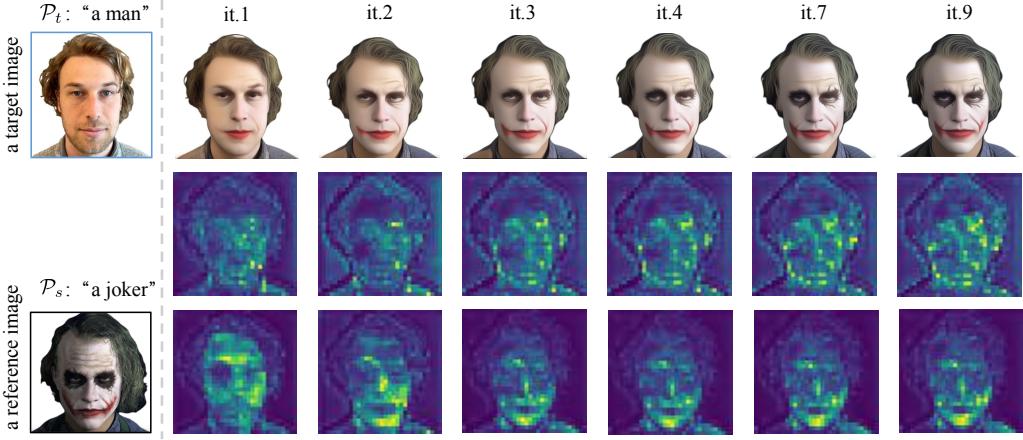


Figure 3: **Cross-Attention maps under different iterations.** On the left, using the *VisCtrl* method, the appearance of a reference image with text condition  $\mathcal{P}_s$  is inserted into a target image with text condition  $\mathcal{P}_t$ . On the right are the changes in the target image during the iterations, as well as the changes in the cross-attention computed between its intermediate latent and  $\mathcal{P}_s$  and  $\mathcal{P}_t$  respectively. Please refer to Section 4.1 for more details.

the convenience of stylized images, image style transfer [56, 57, 58] receives extensive attention, replying to methods to build a dense correspondence between input and reference images, but it cannot deal with local editing and shape editing. To achieve local editing with non-rigid transformation, conditions like bounding boxes and masks are introduced, but require drawing efforts from users, which sometimes are hard to obtain [3, 2, 59]. A recent work [60] learns the visual concept of the subject from reference images and then swaps it into the target image using pre-trained diffusion models. However, it requires multiple reference images to learn the corresponding visual concepts that need to fine-tune diffusion model and a significant number of DDIM inversion and denoising steps, which are time-consuming. Our method leverages attention mechanisms to enable personalized editing without the need for additional training while preserving the identity of the original image.

### 3 Method

In this section, we first provide a short preliminary Section 3.1 and then describe our method Section 3.2. An illustration of our method is shown in Figure 2 and Algorithm 1.

#### 3.1 Preliminary

**Latent Diffusion Models.** Latent Diffusion Model (LDM) [61] is composed of two main components: an autoencoder and a latent diffusion model. The encoder  $\mathcal{E}$  from the autoencoder component of the LDMs maps an image  $\mathcal{I}$  into a latent code  $z_0 = \mathcal{E}(\mathcal{I})$  and the decoder reverses the latent code back to the original image as  $\mathcal{D}(\mathcal{E}(\mathcal{I})) \approx \mathcal{I}$ . Let  $\mathcal{C} = \tau_\theta(\mathcal{P})$  be the conditioning mechanism that maps a textual condition  $\mathcal{P}$  into a conditional vector for LDMs, the LDM model is updated by the loss:

$$L_{LDM} := \mathbb{E}_{z_0 \sim \mathcal{E}(\mathcal{I}), \mathcal{P}, \epsilon \sim \mathcal{N}(0,1), t \sim U(1,T)} \left[ \|\epsilon - \epsilon_\theta(z_t, t, \mathcal{C})\|_2^2 \right] \quad (1)$$

The denoiser  $\epsilon_\theta$  is typically a conditional U-Net [62] which predicts the added gaussian noise  $\epsilon$  at timestep  $t$ . Text-to-image diffusion models [16, 17, 24, 18] are trained by Equation 1 with  $\epsilon_\theta$  that estimates the noise conditioned on the text prompt  $\mathcal{P}$ .

**DDIM inversion.** Inversion involves finding an initial noise  $z_T$  that reconstructs the input latent code  $z_0$  conditioned on  $\mathcal{P}$ . As our goal is to precisely reconstruct a given image with a reference image, we utilize deterministic DDIM sampling [10]:

$$z_{t+1} = \sqrt{\bar{\alpha}_{t+1}} f_\theta(z_t, t, \mathcal{C}) + \sqrt{1 - \bar{\alpha}_{t+1}} \epsilon_\theta(z_t, t, \mathcal{C}) \quad (2)$$

where  $\bar{\alpha}_{t+1}$  is noise scaling factor defined in DDIM [10] and  $f_\theta(z_t, t, \mathcal{C})$  predicts the final denoised latent code  $z_0$  as  $f_\theta(z_t, t, \mathcal{C}) = [z_t - \sqrt{1 - \bar{\alpha}_t} \epsilon_\theta(z_t, t, \mathcal{C})] / \sqrt{\bar{\alpha}_t}$ .

**Attention Mechanism in LDM.** The U-Net in the Diffusion model, consists of a series of basic blocks, and each basic block contains a residual block [63], a self-attention module, and a cross-attention [64] module. The attention mechanism can be formulated as follows:

$$\text{Attention}(Q_t, K, V) = \text{softmax}\left(\frac{Q_t K^T}{\sqrt{d}}\right)V, \quad (3)$$

where  $d$  represents the latent dimension, and  $Q$  denotes the query features projected from spatial features, while  $K$  and  $V$  signify the key and value features projected from the spatial features in self-attention layers or the textual embedding in cross-attention layers. The attention map is  $A_t = \text{softmax}(Q_t \cdot K^T / \sqrt{d})$  which is the first component of Equation 3.

### 3.2 VisCtrl: View iterative Self-Attention Control

In this section, we introduce *View iterative self-attention Control* (*VisCtrl*) for Tuning-Free personalized visual editing. The overall architecture of the proposed pipeline to perform synthesis and editing is shown in Figure 2, and the algorithm is summarized in Algorithm 1. Our goal is to inject the features of the personalized subject in reference images  $\{I_s\}_1^N$  (typically 1-3) into another subject  $I_t^m$  in a given target image  $I_t$ . Firstly, we use SAM [65] to segment the target subject  $I_t^m$  based on the target text prompt  $\mathcal{P}_t$ . Then, we obtain the initial noise  $Z_T^s$  for the reference images and the initial noise  $Z_T^t$  for the target subject through DDIM inversion [10], which are used for the reconstruction of images. Next, through the U-Net, we obtain the features  $K$  and  $V$  of the images. Finally, during the target image reconstruction process conditioned on the noise  $Z_T^t$  and the target text prompt  $\mathcal{P}_t$ , the target subject features  $(K_t, V_t)$  are replaced with the reference image features  $(K_s, V_s)$  obtained during the reference image reconstruction process. Hence, we can seamlessly integrate the generated subject back into the target image in a harmonious manner.

As shown in Figure 2, the architecture includes the reference image branch (top) and the target image branch (bottom), both branches perform inversion and denoising, but the denoising process will be different. Specifically, the reference image branch provides personalized subject features through self-attention. Then, in the target image branch, we assemble the inputs for the self-attention by 1) keeping the current Query features  $Q_t$  unchanged, and 2) obtaining the Key and Value features  $K_s$  and  $V_s$  from the self-attention layer in the reference image branch. 3) Continuously perform the denoising process described above to obtain  $Z_0^t$ . 4) Finally, utilize  $Z_0^t$  as a replacement for  $Z^*$ , followed by an inversion process and iterate through steps (1), (2), and (3) for  $N$  iterations to gradually inject the feature of reference images into the target image. We initialize  $Z^*$  as  $\mathcal{E}(I_t)$ . During the iterative process,  $Z^*$  is updated according to the following formulation:

$$Z_{(n+1)}^* = Z_{0(n)}^t, \quad 1 < n < N \quad (4)$$

where  $n$  denotes the iteration number,  $N$  is the total number of iterations. It is noteworthy that significant improvement can be achieved within 5 iterations. Each iteration involves an inversion process and denoising process, both of which do not exceed 5 steps. Remarkably, We can control the level of the appearance and structure of reference images into target image with proper starting denoising step  $S$  and layer  $L$  for editing, please refer to Figure 9. Thus the Edit function in Algorithm 1 can be formulated as follows:

---

**Algorithm 1:** View Iterative Self-Attention Control

---

```

1 Input: The reference images  $\{I_s\}_1^N$  and
   corresponding prompt  $\mathcal{P}_s$ , a target image  $I_t$ 
   and corresponding prompt  $\mathcal{P}_t$ .
2 Output: Edited latent map  $z_0^t$ .
3  $\{z_T^s\}_1^N \leftarrow \text{DDIMInversion}(\mathcal{E}(\{I_s\}_1^N), \mathcal{P}_s);$ 
4  $z^* = \mathcal{E}(I_t);$ 
5  $z_T^t \leftarrow \text{DDIMInversion}(z^*, \mathcal{P}_t);$ 
6 for  $n = N, N-1, \dots, 1$  do
7    $z_T^t \leftarrow$ 
      $\alpha * \text{DDIMInversion}(z^*, \mathcal{P}_t) + (1-\alpha) * z_T^t;$ 
8    $z_T^s = \text{DataSampler}(\{z_T^s\}_1^N);$ 
9   for  $t = T, T-1, \dots, 1$  do
10    |  $\epsilon_s, \{Q_s, K_s, V_s\} \leftarrow \epsilon_\theta(z_T^s, P_s, t);$ 
11    |  $z_{t-1}^s \leftarrow \text{DDIMSampler}(z_T^s, \epsilon_s);$ 
12    |  $\{Q_t, K_t, V_t\} \leftarrow \epsilon_\theta(z_t, P, t);$ 
13    |  $\{Q_t^*, K_t^*, V_t^*\} \leftarrow$ 
      |    $\text{Edit}(\{Q_t, K_t, V_t\}, \{Q_s, K_s, V_s\});$ 
14    |  $\epsilon_t = \epsilon_\theta(z_t, P, t; \{Q_t^*, K_t^*, V_t^*\});$ 
15    |  $z_{t-1}^t \leftarrow \text{DDIMSampler}(z_t^t, \epsilon_t);$ 
16   end
17    $z^* = z_T^t;$ 
18 end
19 Return  $z_0^t$ 

```

---

Table 1: **Comparison to prior exemplar-guided image editing methods.** We compare our method with several prior Exemplar-guided Image Editing approaches across three distinct tasks. The initial two editing tasks (dog → dog, teddy bear → teddy bear) are assessed using CLIP-I, BG LPIPS, and SSIM. Definitions and details of these metrics can be found in the Appendix C.2. Specifically, we contrast the generated images with both the reference image and the source image, resulting in two CLIP-I scores. In the CLIP-I column, the left value denotes the score between the reference image and the generated image, while the right represents the score between the source image and the generated image. For the remaining task (man → van gogh), only CLIP-I and SSIM metrics are utilized, as background reconstruction is deemed irrelevant.

Method	dog → dog			teddy bear → teddy bear			man → van gogh	
	CLIP-I (↑)	BG LPIPS (↓)	SSIM(↑)	CLIP-I (↑)	BG LPIPS (↓)	SSIM(↑)	CLIP-I (↑)	SSIM(↑)
AnyDoor [2]	<b>79.9%</b> / 75.3%	0.379	0.580	70.2% / 80.1%	0.378	0.546	59.6% / 48.8%	0.289
Paint-by-Example [3]	75.6% / 75.5%	0.287	0.674	<b>76.4%</b> / 75.1%	0.388	0.601	64.5% / 41.1%	0.522
Photoswap [60]	69.8% / <b>80.8%</b>	0.225	0.768	62.6% / 78.2%	0.228	0.640	47.7% / 51.2%	0.635
<i>VisCtrl</i> (ours)	76.7% / 71.9%	<b>0.211</b>	<b>0.822</b>	72.8% / <b>85.7%</b>	<b>0.205</b>	<b>0.838</b>	<b>72.1%</b> / <b>69.1%</b>	<b>0.746</b>

$$\text{Edit} := \begin{cases} \{Q_t, K_s, V_s\}, & \text{if } t > S \text{ and } l > L, \\ \{Q_t, K_t, V_t\}, & \text{otherwise,} \end{cases} \quad (5)$$

where  $S$  and  $L$  are the time step and layer index to start *VisCtrl*, respectively.

### 3.3 Feature Gradually Sampling strategy for multi-view editing

When applying the *VisCtrl* method to complex visual domains where the target content is distributed across multiple views, such as video editing and 3D editing, we encounter two key challenges: **1) Limited usability of single reference Image:** In complex scenarios with multiple perspectives, relying on single reference image often leads to blurring due to significant changes between different views. This occurs because retrieving insufficient useful information from a single reference image can cause the target image to lose its original structure during the iterative process. Once the structure is compromised, it becomes difficult to restore, as the missing structure is no longer present in the query of the target image, please refer to Figure 8. **2) Consistent injection from multiple reference images:** When incorporating multiple reference images, it's crucial to ensure that the injection of information from these images is consistent. Drastic variations can lead to jitter in video and artifacts in 3D scenes.

Therefore, we propose the Feature Gradual Sampling strategy (FGS) for multi-view editing, which involves randomly sampling the data from the reference images to allow the target image to perceive as much useful information as possible. Additionally, to mitigate forgetting, we will let  $z$  with weighted updates during the iterative process.  $Z_{T(n+1)}^t$  is updated according to the following formulation:

$$Z_{T(n+1)}^t = \alpha * \mathcal{F}(Z_{(n)}^*, \mathcal{P}_t) + (1 - \alpha) * Z_{T(n+1)}^t, \quad 1 < n < N \quad (6)$$

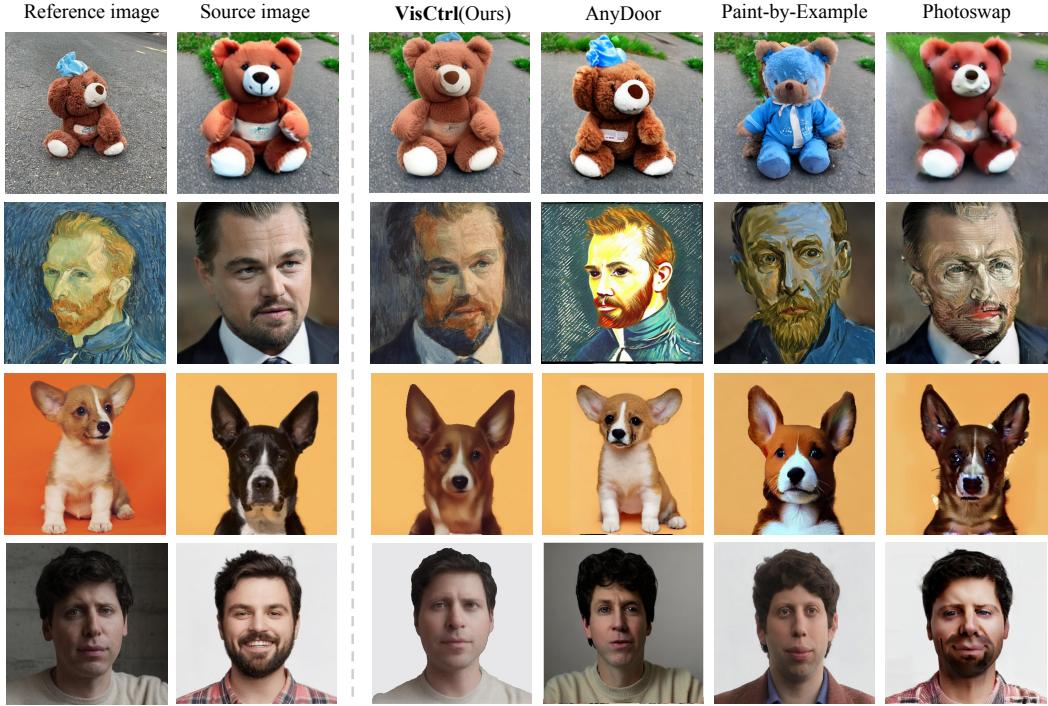
where  $n$  denotes the iteration number,  $\mathcal{F}$  represents the process of target branch DDIM inversion, obtaining the initial noise using  $Z_{(n)}^*$  under the condition of  $\mathcal{P}_t$ . The parameter  $\alpha$  denotes the sampling coefficient, which controls the degree of feature injection. A smaller  $\alpha$  results in more gradual feature changes.

## 4 Experiments

Our *VisCtrl* can be used to edit images, videos, and 3D scenes. We validate the effectiveness of FGS and demonstrate that *VisCtrl* can control the degree of subject personalization, including its shape and appearance, please refer to Appendix B. We showcase the capabilities of our method in various experiments, please refer to Appendix A.

### 4.1 Personalized Subject Editing in images

Figure 1 showcases the effectiveness of *VisCtrl* for personalized subject editing in images. Our approach excels at preserving crucial aspects such as spatial layout, geometry, and the pose of the



**Figure 4: Results of different methods on personalized image editing.** Our proposed *VisCtrl* method yields compelling results across various object and image domains, showcasing its broad applicability. From left to right: the reference image and the source image with their respective prompts, editing results with the proposed *VisCtrl* method, and Other Exemplar-guided Image Editing results with existing methods AnyDoor [2], Paint by Example [3], and Photoswap [60]. Please refer to Section 4.2 for more details.

original subject while seamlessly introducing a reference subject into the source image. Our method can not only achieve personalized injection of similar subject (e.g. duck to personalized duck, vase to personalized vase) but also enable editing between different subject (e.g. injecting cake features into a sandwich, incorporating Van Gogh’s art style into a portrait).

To demonstrate the effectiveness of our feature injecting method, we examined the changes in the generated images and the corresponding cross-attention maps with different prompts under different iterations. As shown in Figure 3, it can be seen that with only 4 iterations, the quality of the generated images can rival that of 9 iterations. As the iterations progress, the features from the reference image ‘joker’ gradually become richer (e.g. the black eye circles in the second iteration, the wrinkles on the forehead in the third iteration). We compute the cross-attention map related to  $\mathcal{P}_t$  and the latent of the target branch (Figure 2 below) by using Equation 3, where the features about “joker” continue to manifest, as shown in the middle row. Similarly, we compute the cross-attention map related to  $\mathcal{P}_s$  and the latent of the reference branch (Figure 2 above), where the features about “man” gradually diminish, as shown in the bottom row. In Figure 10, we also observe the changes in self-attention during different iterations of the generation process.

## 4.2 Comparison with Baseline Methods

We compared our method with several baselines for personalized image editing. Please refer to Appendix C for more details.

In Figure 4, we present a comparative analysis between our approach and the baselines. AnyDoor generated images exhibit favorable features related to subject from reference images, albeit with structural degradation of the source image. Paint-by-Example produces high-quality results but fails to inject subject-related features and adequately preserve the layout structure of the source image. Although Photoswap retains both subject features and the layout structure of the source image, it

suffers from inferior generation quality. Our method far surpass those baselines, effectively balancing the preservation of the source image’s layout structure and background while incorporating more features from the reference image.

In Table 1, we conduct a comparative analysis between our method and the baselines, revealing a consistent trend. AnyDoor exhibits the highest BG LPIPS score, indicating significant variations in the background of source images. Paint-by-Example generally achieves lower CLIP-I score, suggesting substantial disparities between the generated image and both source and reference image. Our method achieves The first and second highest CLIP-I score, striking a balance between incorporating the appearance features from the reference image and preserving the structural characteristics of the source image. This is evidenced by the lowest BG LPIPS score and the highest SSIM score.



Figure 5: **Results of different methods on personalized video editing.** We edit the foreground subject and background of various videos using different methods. Compare to baseline, Our method not only generates content that is more similar to the reference image but also maintains the continuity of the edited regions across different frames.

### 4.3 Personalized Subject editing in complex visual domains

Thanks to the following characteristics of our method *VisCtrl*, our approach can be easily adapted to other complex visual personalized editing tasks: **1)** The plug-and-play architecture allows direct usage on any method that utilizes Stable-diffusion. **2)** The distinguishing attribute of our method, Training-Free, is its capability to complete single-image editing within just a few denoising steps without fine-tune. **3)** The Feature Gradually Sampling strategy for Multi-view editing (Section 3.3) enables consistent editing across multiple views. We conducted a spectrum of experiments in complex visual scenarios, validating the scalability of our method.

**Video editing.** We adopt Pix2video [8] as our baseline, which utilizes a 2D diffusion model driven by text to achieve image editing. In the task of video editing, we use a single image as a reference subject, and insert its feature into corresponding subject in each frame of the video. As illustrated in Figure 5, our approach edits the content in the video to be most similar to the reference subject, while effectively controlling the influence on other content outside the editing region. Moreover, as shown in Table 2a, our method achieves the best scores in both CLIP Directional Similarity and LPIPS, indicating that our approach not only preserves the layout of the target image but also effectively achieves personalized editing of the video scene.

**3D scene editing.** Our method extends upon AnyDoor [2] by introducing the *VisCtrl* module (see more details in Appendix A.3), enabling to inject the features of the reference images into the target subject in the 3D scene. Moreover, leveraging the FGS enhances the performance of 2D image editing methods in 3D scene editing. As observed in Figure 7, Instruct-NeRF2NeRF (IN2N) generated

Table 2: **Comparison to prior complex visual editing methods.** We individually assess the quantitative metrics of *VisCtrl* in both video editing and 3D scenes, comparing them against other baseline methods.

(a) **Video editing.** Quantitative comparison of video editing. Our method, *VisCtrl*, is compared with Pix2video across two video scenarios: background editing (e.g. sky) and foreground subject manipulation (e.g. car). *VisCtrl* outperforms on par with existing method across almost metrics.

Method	sky → sky			car → car		
	CLIP Directional Similarity( $\uparrow$ )	CLIP-I( $\uparrow$ )	LPIPS( $\downarrow$ )	CLIP Directional Similarity( $\uparrow$ )	CLIP-I( $\uparrow$ )	LPIPS( $\downarrow$ )
Pix2video [8]	0.136	<b>84.3%</b> 82.2%	0.509	0.087	76.2% <b>77.9%</b>	0.392 <b>0.044</b>
<i>VisCtrl</i> (ours)	<b>0.226</b>	<b>82.2%</b>	<b>0.195</b>	<b>0.090</b>	<b>77.9%</b>	<b>0.044</b>

(b) **3D scene editing.** Quantitative comparison of on 3D scene editing. *VisCtrl* can achieve plug and play. After using *VisCtrl*, the capabilities of Anydoor have been significantly improved on 3D scenes editing, which be marked red in the table.

Method	CLIP Directional Similarity( $\uparrow$ )	CLIP-I( $\uparrow$ )	LPIPS( $\downarrow$ )
IN2N [1]	<b>0.210</b>	79.0%	<b>0.401</b>
AnyDoor [2]	0.180	76.3%	0.529
AnyDoor+ <i>VisCtrl</i>	0.189( <b>+5%</b> )	<b>79.9% (+4.7%)</b>	0.452( <b>+14.5%</b> )

sunglasses exhibit missing structures and even affect irrelevant backgrounds (as shown in the red circles in the figure). The sunglasses generated by AnyDoor differ significantly in appearance and shape (as shown in the blue circles in the figure) from the reference image. The noise in the sunglasses generated by AnyDoor is due to the inconsistent editing between different views. These inconsistent edits make it difficult for the 3DGS [34] to converge. Our method alleviates this issue by ensuring more consistent editing (as shown in the green circles in the figure). *VisCtrl* improves the subject similarity and structural continuity. Quantitative indications in Table 2b also clearly demonstrate the significant improvement in effectiveness brought about by the incorporation of the *VisCtrl* module.

## 5 Conclusion

In this paper, we propose View Iterative Self-Attention Control (*VisCtrl*), a simple but effective framework designed for personalized visual editing. *VisCtrl* is capable of injecting features between images using the self-attention mechanism without fine-tuning the model. Furthermore, we propose a Feature Gradually Sampling strategy to adapt *VisCtrl* to complex visual applications such as video editing and 3D scene editing. We demonstrate the effectiveness of our method in exemplar-guided visual editing, including images, videos, and real 3D scenes, outperforming previous methods both quantitatively and qualitatively.

**Limitations.** Since we use pre-trained diffusion models, there are instances where the results are imperfect due to the inherent limitations of these models. Additionally, our method relies on masks to specify the objects or regions to be edited, and incorrect masks can lead to disharmonious image editing results. Please refer to Appendix E for further details.

**Broader impacts.** Our research introduces a comprehensive visual editing framework that encompasses various modalities, including 2D images, videos, and 3D scenes. While it is important to acknowledge that our framework might be potentially misused to create fake content, this concern is inherent to visual editing techniques as a whole. Furthermore, our method relies on generative priors derived from diffusion models, which may inadvertently contain biases due to the auto-filtering process applied to the vast training dataset. However, *VisCtrl* has been meticulously designed to mitigate bias within the diffusion model. Please refer to Appendix D for further details.

## References

- [1] T. Brooks, A. Holynski, and A. A. Efros. Instructpix2pix: Learning to follow image editing instructions. *arXiv preprint arXiv:2211.09800*, 2022.
- [2] X. Chen, L. Huang, Y. Liu, Y. Shen, D. Zhao, and H. Zhao. Anydoor: Zero-shot object-level image customization. *arXiv preprint arXiv:2307.09481*, 2023.
- [3] B. Yang, S. Gu, B. Zhang, T. Zhang, X. Chen, X. Sun, D. Chen, and F. Wen. Paint by example: Exemplar-based image editing with diffusion models. *arXiv*, 2022.
- [4] A. Hertz, R. Mokady, J. Tenenbaum, K. Aberman, Y. Pritch, and D. Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.

- [5] M. Cao, X. Wang, Z. Qi, Y. Shan, X. Qie, and Y. Zheng. Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. *arXiv*, 2023.
- [6] G. Parmar, K. K. Singh, R. Zhang, Y. Li, J. Lu, and J.-Y. Zhu. Zero-shot image-to-image translation. *arXiv*, 2023.
- [7] A. Haque, M. Tancik, A. Efros, A. Holynski, and A. Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. 2023.
- [8] D. Ceylan, C.-H. P. Huang, and N. J. Mitra. Pix2video: Video editing using image diffusion. *arXiv preprint arXiv:2303.12688*, 2023.
- [9] R. Mokady, A. Hertz, K. Aberman, Y. Pritch, and D. Cohen-Or. Null-text Inversion for Editing Real Images using Guided Diffusion Models. In *arXiv*, 2022.
- [10] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. URL <https://openreview.net/forum?id=St1giarCHLP>.
- [11] A. Brock, J. Donahue, and K. Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv*, 2018.
- [12] K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng, and F.-Y. Wang. Generative adversarial networks: introduction and outlook. *IEEE/CAA Journal of Automatica Sinica*, 4(4):588–598, 2017.
- [13] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.
- [14] H. Ye, X. Yang, M. Takac, R. Sunderraman, and S. Ji. Improving text-to-image synthesis using contrastive learning. *arXiv preprint arXiv:2107.02423*, 2021.
- [15] M. Tao, H. Tang, F. Wu, X.-Y. Jing, B.-K. Bao, and C. Xu. Df-gan: A simple and effective baseline for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16515–16525, 2022.
- [16] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Goncalves Lopes, B. Karagol Ayan, T. Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.
- [17] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.
- [18] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.
- [19] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402*, 2022.
- [20] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.
- [21] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.
- [22] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2020.
- [23] W. Peebles and S. Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022.

- [24] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022.
- [25] C. Meng, Y. Song, J. Song, J. Wu, J.-Y. Zhu, and S. Ermon. Sdedit: Image synthesis and editing with stochastic differential equations. *arXiv*, 2021.
- [26] O. Bar-Tal, D. Ofri-Amar, R. Fridman, Y. Kasten, and T. Dekel. Text2live: Text-driven layered image and video editing. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XV*, pages 707–723. Springer, 2022.
- [27] B. Kawar, S. Zada, O. Lang, O. Tov, H. Chang, T. Dekel, I. Mosseri, and M. Irani. Imagic: Text-based real image editing with diffusion models. *arXiv*, 2022.
- [28] A. Voynov, K. Aberman, and D. Cohen-Or. Sketch-guided text-to-image diffusion models. *arXiv preprint arXiv:2211.13752*, 2022.
- [29] L. Zhang and M. Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv*, 2023.
- [30] R. Gal, Y. Alaluf, Y. Atzman, O. Patashnik, A. H. Bermano, G. Chechik, and D. Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv*, 2022.
- [31] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. *arXiv preprint arXiv:2208.12242*, 2022.
- [32] J. Z. Wu, Y. Ge, X. Wang, S. W. Lei, Y. Gu, Y. Shi, W. Hsu, Y. Shan, X. Qie, and M. Z. Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7623–7633, 2023.
- [33] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.
- [34] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023. URL <https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/>.
- [35] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.
- [36] J. Tang. Stable-dreamfusion: Text-to-3d with stable-diffusion, 2022. <https://github.com/ashawkey/stable-dreamfusion>.
- [37] Z.-X. Zou, Z. Yu, Y.-C. Guo, Y. Li, D. Liang, Y.-P. Cao, and S.-H. Zhang. Triplane meets gaussian splatting: Fast and generalizable single-view 3d reconstruction with transformers. *arXiv preprint arXiv:2312.09147*, 2023.
- [38] J. Tang, J. Ren, H. Zhou, Z. Liu, and G. Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*, 2023.
- [39] R. Liu, R. Wu, B. Van Hoorick, P. Tokmakov, S. Zakharov, and C. Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. *arXiv preprint arXiv:2303.11328*, 2023.
- [40] J. Fang, J. Wang, X. Zhang, L. Xie, and Q. Tian. Gaussianeditor: Editing 3d gaussians delicately with text instructions. In *CVPR*, 2024.
- [41] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. In *International conference on machine learning*, pages 1060–1069. PMLR, 2016.

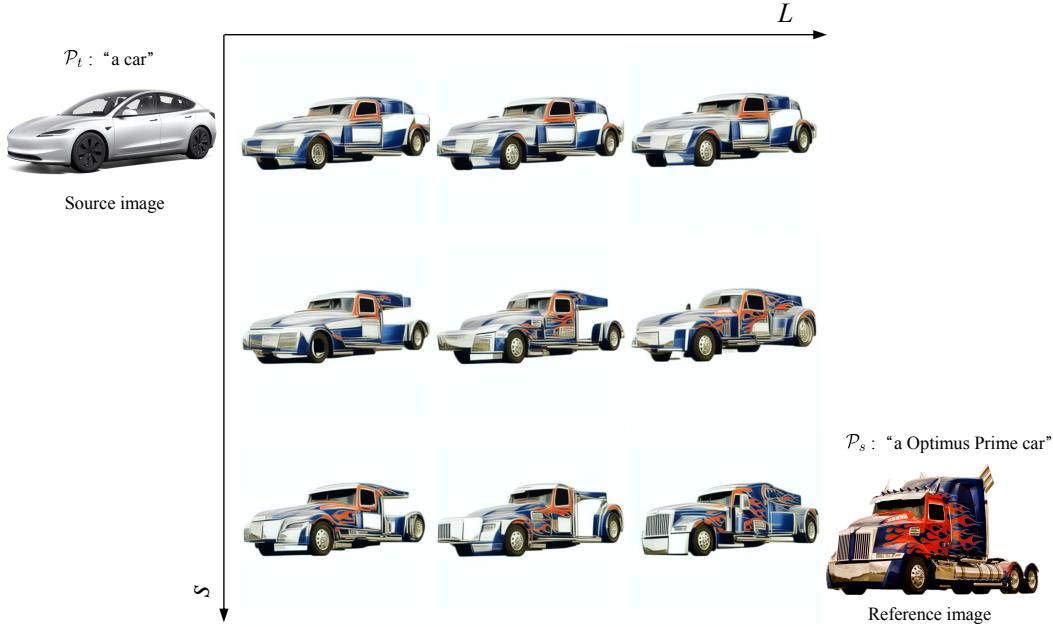
- [42] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1947–1962, 2018.
- [43] B. Li, X. Qi, T. Lukasiewicz, and P. Torr. Controllable text-to-image generation. *Advances in neural information processing systems*, 32, 2019.
- [44] R. Gal, O. Patashnik, H. Maron, A. H. Bermano, G. Chechik, and D. Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators. *ACM Transactions on Graphics (TOG)*, 41(4):1–13, 2022.
- [45] W. Feng, X. He, T.-J. Fu, V. Jampani, A. Akula, P. Narayana, S. Basu, X. E. Wang, and W. Y. Wang. Training-Free Structured Diffusion Guidance for Compositional Text-to-Image Synthesis. In *International Conference on Learning Representations*, 2023.
- [46] O. Avrahami, D. Lischinski, and O. Fried. Blended diffusion for text-driven editing of natural images. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022.
- [47] M. Wang, G.-Y. Yang, R. Li, R.-Z. Liang, S.-H. Zhang, P. M. Hall, and S.-M. Hu. Example-guided style-consistent image synthesis from semantic labeling. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.
- [48] X. Zhou, B. Zhang, T. Zhang, P. Zhang, J. Bao, D. Chen, Z. Zhang, and F. Wen. Cocosnet v2: Full-resolution correspondence learning for image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11465–11475, 2021.
- [49] S. Liu, T. Lin, D. He, F. Li, M. Wang, X. Li, Z. Sun, Q. Li, and E. Ding. Adaatttn: Revisit attention mechanism in arbitrary neural style transfer. In *IEEE International Conference on Computer Vision*, 2021.
- [50] Y. Deng, F. Tang, W. Dong, C. Ma, X. Pan, L. Wang, and C. Xu. Stytr2: Image style transfer with transformers. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022.
- [51] Y. Zhang, N. Huang, F. Tang, H. Huang, C. Ma, W. Dong, and C. Xu. Inversion-based creativity transfer with diffusion models. *arXiv*, 2022.
- [52] Z. Yang, J. Wang, Z. Gan, L. Li, K. Lin, C. Wu, N. Duan, Z. Liu, C. Liu, M. Zeng, et al. Reco: Region-controlled text-to-image generation. *arXiv*, 2022.
- [53] Y. Li, H. Liu, Q. Wu, F. Mu, J. Yang, J. Gao, C. Li, and Y. J. Lee. Gligen: Open-set grounded text-to-image generation. *arXiv*, 2023.
- [54] M. Jahn, R. Rombach, and B. Ommer. High-resolution complex scene synthesis with transformers. *arXiv*, 2021.
- [55] J. Seo, G. Lee, S. Cho, J. Lee, and S. Kim. Midms: Matching interleaved diffusion models for exemplar-based image translation. *arXiv*, 2022.
- [56] J. Liao, Y. Yao, L. Yuan, G. Hua, and S. B. Kang. Visual attribute transfer through deep image analogy. *ACM Transactions on Graphics*, 2017.
- [57] P. Zhang, B. Zhang, D. Chen, L. Yuan, and F. Wen. Cross-domain correspondence learning for exemplar-based image translation. In *CVPR*, pages 5143–5153, 2020.
- [58] N. Tumanyan, O. Bar-Tal, S. Bagon, and T. Dekel. Splicing vit features for semantic appearance transfer. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022.
- [59] T. Li, M. Ku, C. Wei, and W. Chen. Dreamedit: Subject-driven image editing, 2023.
- [60] J. Gu, Y. Wang, N. Zhao, T.-J. Fu, W. Xiong, Q. Liu, Z. Zhang, H. Zhang, J. Zhang, H. Jung, and X. E. Wang. Photoswap: Personalized subject swapping in images, 2023.
- [61] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022.

- [62] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.
- [63] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [64] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017.
- [65] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick. Segment anything. *arXiv:2304.02643*, 2023.
- [66] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.
- [67] C. Vachha and A. Haque. Instruct-gs2gs: Editing 3d gaussian splats with instructions, 2024. URL <https://instruct-gs2gs.github.io/>.
- [68] M. Tancik, E. Weber, E. Ng, R. Li, B. Yi, J. Kerr, T. Wang, A. Kristoffersen, J. Austin, K. Salahi, et al. Nerfstudio: A modular framework for neural radiance field development. *arXiv preprint arXiv:2302.04264*, 2023.
- [69] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. 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.
- [70] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.
- [71] R. Gal, O. Patashnik, H. Maron, G. Chechik, and D. Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators, 2021.
- [72] A. Sasha Luccioni, C. Akiki, M. Mitchell, and Y. Jernite. Stable bias: Analyzing societal representations in diffusion models. *arXiv e-prints*, pages arXiv–2303, 2023.
- [73] M. V. Perera and V. M. Patel. Analyzing bias in diffusion-based face generation models. *arXiv preprint arXiv:2305.06402*, 2023.

# Appendix

## A Implementation Details

We demonstrate our method in various experiments using Stable Diffusion v1.5 [61]. The segmentation model utilized in the experiment is the LangSAM segmentation algorithm, which is built upon SAM [65], and the GroundingDINO [66] detection model. All of our experiments were performed using a single NVIDIA V100 GPU.



**Figure 6: Results at different injecting layers and denoising steps.** The top left corner shows the source image and the corresponding text prompt  $\mathcal{P}_t$ . The bottom right corner displays the reference image and the corresponding text prompt  $\mathcal{P}_s$ . The middle section presents the generated results with different combinations of the time step  $S$  and the layer index  $L$ , with the values gradually decreasing in the direction indicated by the arrows.

### A.1 2D image personalized editing

AnyDoor [2] and Paint-by-Example [3] are model-based approaches that require extensive fine-tuning with large datasets. In our experiment, we utilized the default models and parameters as described in their respective papers. Given a source image and mask, the reference image is inserted into the corresponding mask region. Photoswap [60] and VisCtrl are attention-based methods that manipulate the attention in UNet to edit images. However, unlike Photoswap, which requires Dreambooth [31] to learn new concepts from reference images, VisCtrl does not need any additional training or learning. Since VisCtrl utilizes only one reference image in our experiments, for fairness, we also used a single image for learning new concepts in Photoswap. We set the Dreambooth training steps 1000 for each image in Photoswap, while keeping other parameters at their defaults.

For our method, We set both the noise addition and denoising steps to  $T = 5$ , with classifier-free guidance set to  $\omega = 6$ , and the number of iterations set to  $N = 5$ . Initially, we utilized DDIM Inversion [10] to transform both the reference and target images into initial noise, and then denoising and iteration until convergence. Setting the number of steps higher injects and generates more details, but should not be excessively large to avoid introducing significant biases from DDIM Inversion. In general, during the initial iteration, a higher number of steps can be set to capture more detail, while the denoising steps remain at 5 for subsequent iterations. Our algorithm is highly efficient,



Figure 7: **Results of different methods on personalized 3D scene editing.** The image on the leftmost is a rendering from the original 3DGS. The image on the rightmost is the reference image used to edit the 3D scene. The images in the middle are rendered from the same viewpoint as the original 3DGS after editing the 3D scene using different methods. We analyze the results of these methods in Section 4.3.

typically converging to satisfactory image results within three iterations. It’s important to note that in 2D image experiments, only one reference image was used.

## A.2 Video personalized editing

For video editing, we apply our method to edit videos frame by frame. Since few video editing methods support the input of reference images, we compare our model with other text-driven tuning-free video editing models, such as Pix2Video, which can represent common video editing methods. For our work, a reference image is provided to edit each frame of the original video, aiming to achieve the overall editing effect. For the Pix2Video model, we obtain the text description of the reference image and use it as the textual input to achieve the video’s editing effect. We set classifier-free guidance  $\omega = 3.5$ , and DDIM steps  $T = 50$  for Pix2Video. Since Pix2Video does not support the input of reference images, we do not overly discuss the similarity between the editing result and the reference image. Instead, we focus more on the temporal consistency of the edited video and the preservation of the background.

## A.3 3D Scenes personalized editing

For text-based 3D editing scenes, we use Instruct-NeRF2NeRF as one of our baselines [7]. We first pretrain 3DGS [34] using the *splatfacto* method [67] from NeRFStudio [68], training it for 30,000 steps in 10 minutes on an NVIDIA Tesla V100. Then, we use ‘give him a pair of sunglasses’ as the IN2N textual condition, iteratively editing the 3D scene and corresponding dataset. There are currently few personalized 3D scene editing methods. Therefore, we adopt 2D editing methods (e.g., AnyDoor [2]) as another baseline for 3D scene editing. We use these methods to edit the 3D scene dataset and then train a model to obtain the edited 3D scene. When editing each image with AnyDoor, we keep the model’s default parameters and turn on shape control.



Figure 8: **Ablation study.** The top row depicts the insertion of features from a single reference image into the source image, along with the changes in the generated image at each iteration step. The bottom row illustrates the utilization of Feature Gradually Sampling to insert features from multiple reference images into the source image, as well as the changes in the generated image during each iteration. See Appendix B for more details

## B Ablation study

**Ablation on the components of FGS.** Feature Gradually Sampling strategy (See Section 3.3) is designed to address the issue where, in a single image scenario, insufficient subject information in the reference image may lead to the loss of certain structural details in the source image. As shown in Figure 8 (top row), features highlighted within the red circle gradually weaken and eventually disappear during iterative layers (e.g. loss of the logo 'N'). Once these structures are lost, it becomes difficult to recover them in subsequent stages. FGS effectively mitigates this problem, as illustrated in Figure 8 (bottom row), by preserving the structural details of the source image while injecting features from multiple reference images.

**Controlling Subject Identity.** We can control at which step of denoising and which layer of the U-Net to start *VisCtrl* by setting  $S$  and  $L$ , respectively. Different settings of  $S$  and  $L$  parameters lead to different outcomes (See Figure 6). As  $S$  and  $L$  decrease, the number of iterations of *VisCtrl* increases. This means that as more features from the reference image are injected into the source image, the generated result not only becomes visually more similar to the reference image but also structurally more alike. Conversely, the opposite is true.

## C Evaluation details

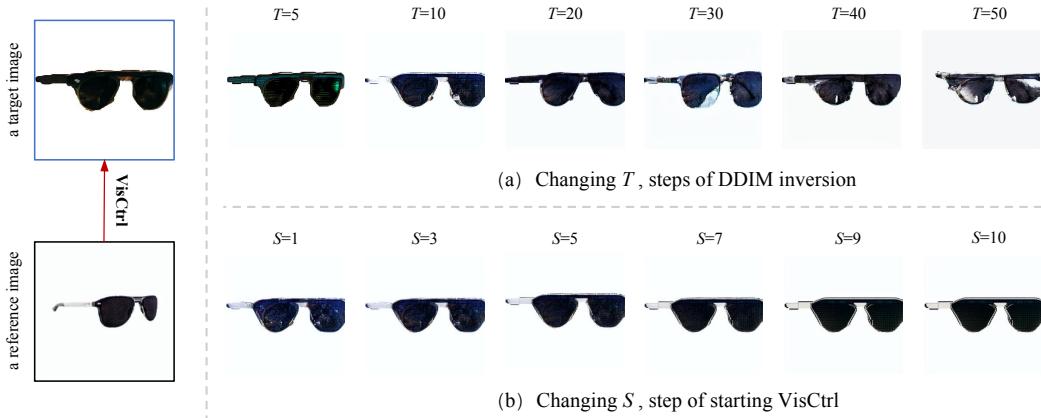


Figure 9: **Results at different denoising steps.** The top right row of the figure showcases the generated results with different denoising steps, while the bottom right row presents the generated results with different insertion steps when  $T = 10$ .



Figure 10: **Self-Attention maps under different iterations.** This representation reveals that the layout of the edited image is intrinsically embedded in the self-attention map from the initial iteration. At different stages of iteration, the attention map in the self-attention varies.

### C.1 Tasks

We compared *VisCtrl* with three other different methods, evaluating the editing results of four images (See Figure 4) and selecting three of these results for quantitative evaluations (See Table 1). Some input images are sourced from the DreamBooth dataset [69], while others are obtained from the internet.

### C.2 Metrics

For quantitative evaluations, we assess three criteria: (1) the adequacy of injected features from reference images, (2) the preservation of the source image’s structure in the edited image, and (3) the consistency of background regions between images. We measure the fidelity of subjects between reference and generated images using CLIP-I [69], which computes the average pairwise cosine similarity between CLIP [70] embeddings of generated and real images. Additionally, we calculate the background LPIPS error (BG LPIPS) to quantify the preservation of background regions post-editing. This involves computing the LPIPS distance between background regions in the source and edited images, with background regions identified using the SAM object detector [65]. A lower BG LPIPS score indicates better preservation of the original image background. Finally, we employ the Structural Similarity Index Measure (SSIM) to gauge the similarity between the source image and the generated image, ensuring that the generated results maintain the overall structure of the source image.

In our work on video editing and 3D scene manipulation tasks, we employ the CLIP-I and LPIPS metrics. Additionally, we utilize CLIP Directional Similarity [71], which quantifies the alignment between textual modifications and corresponding image alterations.

## D Ethics Exploration

Similar to many AI technologies, text-to-image diffusion models may exhibit biases reflective of those inherent in the training data [72, 73]. Trained on extensive text and image datasets, these models might inadvertently learn and perpetuate biases, including stereotypes and prejudices, present within the data. For instance, if the training data contains skewed representations or descriptions of specific demographic groups, the model may produce biased images in response to related prompts.

However, *VisCtrl* has been meticulously designed to mitigate bias within the text-to-image diffusion model generation process. It achieves this by first, not requiring retraining of the model and avoiding

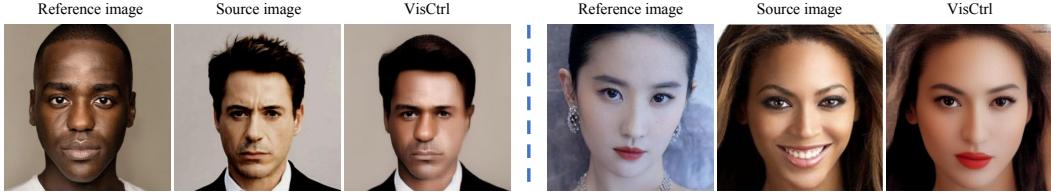


Figure 11: **Results on real human images across different races.** Evidently, the appearance features of the reference image can be seamlessly integrated into the source image, unaffected by skin color or gender.

parameter updates; second, directly performing feature matching and injection in the latent space, thereby preventing bias introduction.

In Figure 11, we present our evaluation of facial feature injection across various skin tones and genders. It is crucial to note that significant disparities between the source and reference images tend to homogenize the skin color in the results. Consequently, we advocate for using *VisCtrl* on subjects with similar racial backgrounds to achieve more satisfactory and authentic outcomes. Despite these potential disparities, the model ensures the preservation of most of the target subject’s specific facial features, thereby reinforcing the credibility and accuracy of the final image.

## E Failure Cases



Figure 12: **Failure cases.** Our algorithm relies on SAM [65] to obtain masks. Occasional inaccuracies in segmentation can result in errors in our generated results, as indicated by the red circles in the figure.

In this section, we highlight common failure cases. When intending to edit a specific subject within a source image, it is necessary to segment this subject using a segmentation algorithm. Subsequently, utilizing the reference image, *VisCtrl* operations are performed to generate the desired subject. The final generated result is obtained by overlaying this generated subject with the corresponding mask. Consequently, if the mask produced by the segmentation algorithm is of poor quality, it may result in missing portions in the resulting image, as illustrated by the mouth of the horse and the tail of the cat in Figure 12.