A Diffusion-Based Framework for Occluded Object Movement

Zheng-Peng Duan[1, 2], Jiawei Zhang[2], Siyu Liu[1], Zheng Lin[5*], Chun-Le Guo[1, 3], Dongqing Zou[2, 4], Jimmy Ren[2], Chongyi Li[1, 3*]

[1]VCIP, CS, Nankai University
[2]SenseTime Research
[3]NKIARI, Shenzhen Futian
[4]PBVR
[5]BNRist, Department of Computer Science and Technology, Tsinghua University
{adamduan0211, zhjw1988}@gmail.com, liusiyu29@mail.nankai.edu.cn, frazer.linzheng@gmail.com, guochunle@nankai.edu.cn, dqzou.lhi@gmail.com, rensijie@sensetime.com, lichongyi@nankai.edu.cn

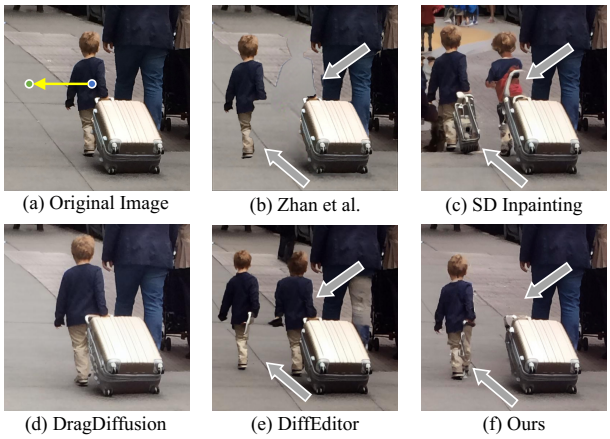
Abstract

Seamlessly moving objects within a scene is a common requirement for image editing, but it is still a challenge for existing editing methods. Especially for real-world images, the occlusion situation further increases the difficulty. The main difficulty is that the occluded portion needs to be completed before movement can proceed. To leverage the real-world knowledge embedded in the pre-trained diffusion models, we propose a **Diff**usion-based framework specifically designed for **O**ccluded **O**bject **M**ovement, named **DiffOOM**. The proposed DiffOOM consists of two parallel branches that perform object de-occlusion and movement simultaneously. The de-occlusion branch utilizes a background color-fill strategy and a continuously updated object mask to focus the diffusion process on completing the obscured portion of the target object. Concurrently, the movement branch employs latent optimization to place the completed object in the target location and adopts local text-conditioned guidance to integrate the object into new surroundings appropriately. Extensive evaluations demonstrate the superior performance of our method, which is further validated by a comprehensive user study.

Project — https://adam-duan.github.io/projects/diffoom/

Introduction

Seamlessly moving objects (Avrahami et al. 2024) within a scene is a common requirement for image editing (Nguyen et al. 2024b; Sajnani et al. 2024; Epstein et al. 2023; Brooks, Holynski, and Efros 2023). To move occluded objects, it involves three sub-tasks: completing the obscured object, moving the object to the target position, and inpainting the original region of the moved objects. To solve object de-occlusion, previous work (Zhan et al. 2020) employs two separate networks: the first predicts the complete mask of the object, and the second fills in the recovered mask with reasonable content. However, adopting discriminative networks significantly restricts the ability to generate new content, as illustrated in Figure 1(b). Recent advances in large-scale diffusion models, known for their powerful generative capability, present a new opportunity to generate the occluded portion (Liu et al. 2024; Zhan et al. 2024; Ozguroglu et al. 2024; Xu, Zhang, and Shi 2024). One intuitive solution is to utilize the SD Inpainting model (Rombach et al. 2022) to complete the missing regions, which is shown in Figure 1(c). With no constraints on the generated contents, the inpainting model may generate undesired elements rather than reconstructing the occluded portion of the target object. Recently, diffusion-based drag-style (Pan et al. 2023) editing methods, such as DragDiffusion (Shi et al. 2024) and DiffEditor (Mou et al. 2024), are proposed to drag objects to target positions with pre-trained diffusion models effectively. As illustrated in Figure 1(d), DragDiffusion focuses primarily on content dragging and therefore struggles to move the entire object. Although DiffEditor successfully moves the little boy to the target location in Figure 1(e), the occluded parts remain incomplete as de-occlusion is not considered.



(a) Original Image (b) Zhan et al. (c) SD Inpainting
(d) DragDiffusion (e) DiffEditor (f) Ours

Figure 1: Comparison with other methods for occluded object movement. Given a real-world image, our method can seamlessly move the occluded object to a user-specified position while completing the occluded portion.

Although existing diffusion-based editing methods cannot be directly employed for this task, the comprehensive real-world knowledge embedded in large-scale diffusion models may be useful for this task. To this end, we propose a diffusion-based framework specifically designed for the movement of occluded objects, called DiffOOM. Our method features two parallel Stable Diffusion-based

*Corresponding authors.

branches to handle object de-occlusion and movement.

For de-occlusion, our motivations mainly come from two aspects. 1) Diffusion models contain rich prior knowledge about the shape of various objects, which is crucial for identifying areas that require filling. 2) Diffusion models possess strong generative abilities to complete the occluded portion with reasonable content. Based on the motivations, the proposed de-occlusion branch utilizes cross-attention as well as self-attention maps to estimate the complete mask of the object, which is utilized to guide the object occlusion region generation during the diffusion process. To minimize the influence of irrelevant elements in the image, the input of the de-occlusion branch uses a color-fill strategy, where the background region of the target object is initialized as a uniform color. To make the visible region of the object unchanged, a latent hold strategy is adopted by replacing the diffusion-updated latent with the one from the inversion process in the visible region during the diffusion steps. Besides, LoRA (Hu et al. 2021) is adopted to ensure that the new content aligns with the characteristics of the target object.

With the object mask and the completed object from the de-occlusion branch, the movement branch aims to place the target object at the target location harmoniously. Specifically, latent optimization minimizes the distance between the latents of the completed object and the target region, guiding the diffusion process to generate the de-occluded object in the target region. To ensure relocated objects blend seamlessly into their new surroundings, local text-conditioned guidance is applied to the target region. Another issue is to avoid filling inadequate contents into the original location of the target object like the result of DiffEditor shown in Figure 1(e). To solve this issue, we fill the original region with noise and utilize a similar mask-guided strategy to direct the diffusion process, ensuring that it inpaints the region with information from the surrounding background.

Our **contributions** can be summarized as follows:

- We utilize the rich real-world knowledge embedded in pre-trained diffusion models to identify the occlusion portion of the object as well as generate the content.
- We introduce a dual-branch framework where the diffusion-based de-occlusion and movement branches process concurrently.
- Extensive experiments and a user study demonstrate the effectiveness of our method in de-occluding diverse objects and achieving satisfactory editing results.

Methodology

In real-world scenarios with occluded objects, our goal is to enable users to relocate these objects to specified target positions while completing the occluded portions. The necessary inputs for this process include the source image, denoted as $\mathbf{I}_s$, and a mask, denoted as $\mathbf{M}_v$, which highlights the visible portion of the object. This mask can either be provided by the user or generated through automated segmentation methods. Additionally, the user specifies the target position by indicating the target point $\mathbf{g}$. In the following subsections, we first introduce the preliminaries on diffusion models, and then outline our overall framework in detail.

Preliminaries

Diffusion Models Our method is built upon Stable Diffusion V1.5 (Rombach et al. 2022), which improves both the training and sampling efficiency of DDPM (Ho, Jain, and Abbeel 2020) by applying the diffusion processes in the latent space rather than pixel space. With pre-trained encoder $\mathcal{E}$ and decoder $\mathcal{D}$, Stable Diffusion can efficiently obtain the latent space representation $\mathbf{Z}$ of $\mathbf{X}$ by $\mathbf{Z} = \mathcal{E}(\mathbf{X})$, and transform the latent space samples to the pixel space through $\mathcal{D}$. To control the synthesis process through the text condition $\mathbf{c}$, Stable Diffusion adopts the conditional denoising model $\boldsymbol{\epsilon}_\theta(\mathbf{Z}_t, t, \mathbf{c})$, where $\mathbf{Z}_t$ is the noisy latent at timestep t.

Attention Mechanism The underlying backbone of the denoising model $\boldsymbol{\epsilon}_\theta$ is a time-conditional U-Net, which consists of a series of basic blocks. Each basic block is equipped with a residual block, a self-attention module, and a cross-attention module sequentially (Dosovitskiy et al. 2020; Vaswani 2017). There is also a text encoder τ_θ to project text prompt $\mathbf{c}$ of length N to an intermediate representation $\tau_\theta(\mathbf{c})$. At timestep t, the residual block first takes the features from $(l-1)$-th basic block as input, and generates the intermediate features $\mathbf{F}_{l,t}$. Then, the self-attention module mines the relationship between the features and themselves, while the cross-attention module captures the connection between visual and textual information (Hertz et al. 2022; Tumanyan et al. 2023; Chefer et al. 2023). Specifically, the cross-attention map $\mathbf{A}^C_{l,t}$ and self-attention map $\mathbf{A}^S_{l,t}$ at l-th layer and t-th timestep can be obtained by

$$\mathbf{A}^C_{l,t} = \text{softmax}(\frac{\mathbf{Q_F}\mathbf{K_c}^T}{\sqrt{d}}), \mathbf{A}^S_{l,t} = \text{softmax}(\frac{\mathbf{Q_F}\mathbf{K_F}^T}{\sqrt{d}}), \quad (1)$$

where d is the dimension of features. $\mathbf{Q_F}$ and $\mathbf{K_F}$ are different projections of the flattened representation of $\mathbf{F}_{l,t}$, while $\mathbf{K_c}$ is the projection of the text embedding $\tau_\theta(\mathbf{c})$.

Refined Cross-attention Map As detailed in Equ. (1), the cross-attention map $\mathbf{A}^C_{l,t}$ illustrates the activation degree of each pixel for each text token, while the self-attention map $\mathbf{A}^S_{l,t}$ captures the correlations between each pixel and others. The cross-attention map related to the token representing the target object provides a rough indication of the object's location and shape, and it can be further refined by utilizing the self-attention map to propagate the activated pixels to highly similar positions (Nguyen et al. 2024a). Concretely, we start by extracting the cross-attention map corresponding to the target object, denoted as $\tilde{\mathbf{A}}^C_{l,t}$. Next, we average both the cross-attention and self-attention maps at a resolution of 32×32 across all layers, which can be formulated as

$$\tilde{\mathbf{A}}^C_t = \frac{1}{L}\sum_{l=0}^{L} \tilde{\mathbf{A}}^C_{l,t}, \mathbf{A}^S_t = \frac{1}{L}\sum_{l=0}^{L} \mathbf{A}^S_{l,t}. \quad (2)$$

We then refine the cross-attention map via:

$$\mathbf{R}^C_t = (\mathbf{A}^S_t)^\lambda \tilde{\mathbf{A}}^C_t, \quad (3)$$

where λ is used to modify the influence of the self-attention map on the cross-attention map. In common practice, we extract the refined cross-attention map corresponding to the target object, which we denoted as $\mathbf{R}^C_t$.

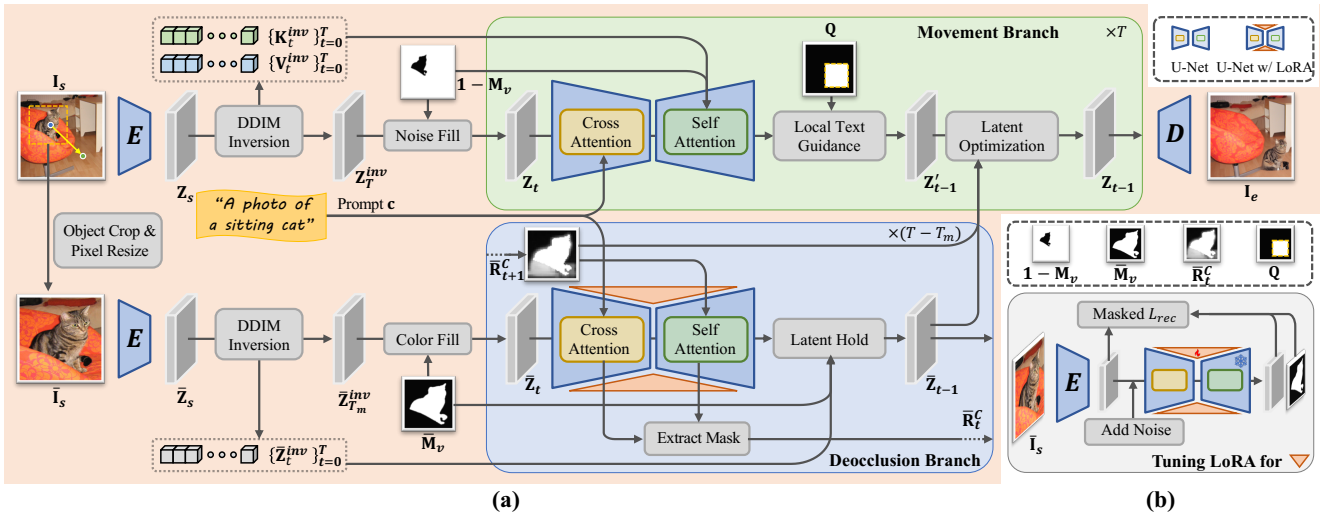


Figure 2: Overview of proposed framework (a) and LoRA tuning process (b). (a) We decouple the task of occluded object movement into de-occlusion and movement, handled by parallel branches. Both branches are built upon Stable Diffusion V1.5 and operate simultaneously. The de-occlusion branch leverages the prior knowledge within the diffusion models to complete the occluded portion, while the movement branch mainly places the completed object at the target position. (b) To ensure the content generated by the de-occlusion branch aligns with the characteristics of the target object, we equip this branch with LoRA, which is fine-tuned using a masked diffusion loss that applies exclusively to the visible portions of the object.

Framework Overview

We propose a diffusion-based framework specifically designed for occluded object movement. Our method effectively decouples this task into two sub-tasks: de-occlusion and movement, which are handled by parallel branches, as depicted in Figure 2. In the following two subsections, we will detail the key designs of the two branches.

Deocclusion Branch

Input Preparation To eliminate the influence of irrelevant elements in the image, the de-occlusion branch takes the image patch that exclusively contains the target object as input. Concretely, with the visible mask $\mathbf{M}_v$ denoting the visible portion of an object, we compute a square bounding box that tightly encloses the object. We denote the center point of this square as $\mathbf{b}$ and the side length as $\hat{r}$. To ensure that the square box covers the complete object, we adjust the side length r using a relax ratio η by $r = \eta \cdot \hat{r}$. Utilizing the center point $\mathbf{b}$ and the side length r, we can crop the source image $\mathbf{I}_s$ and the visible mask $\mathbf{M}_v$ into square patches, which we define as $\textbf{Crop}(\cdot, \mathbf{b}, r)$. Since Stable Diffusion V1.5 is trained on the resolution of 512, we further resize these square patches, which we denote as $\textbf{Resize}(\cdot, s)$, where s represents the desired side length. Thus, the input image $\bar{\mathbf{I}}_s$ and the input mask $\bar{\mathbf{M}}_v$ of the de-occlusion branch can be obtained via

$$\{\bar{\mathbf{I}}_s, \bar{\mathbf{M}}_v\} = \textbf{Resize}(\textbf{Crop}(\{\mathbf{I}_s, \mathbf{M}_v\}, \mathbf{b}, r), \{512, 64\}). \quad (4)$$

Clean latent can be obtained using the pre-trained encoder, represented as $\bar{\mathbf{Z}}_s = \mathcal{E}(\bar{\mathbf{I}}_s)$. To establish starting points for the diffusion process, DDIM inversion (Song, Meng, and Ermon 2020) is employed, which maintains the consistency of the edited result. Inspired by DragonDiffusion, we store the intermediate noisy latents $\{\bar{\mathbf{Z}}_t^{inv}\}_{t=0}^{T}$ to provide precise guidance for the denoising process.

Key Designs The de-occlusion branch aims to leverage the rich real-world knowledge embedded in pre-trained foundation models to complete the occluded object. Our solution integrates two key motivations: 1) Diffusion models contain rich prior knowledge about the shape of various objects (Zhan et al. 2024); and 2) Diffusion models have strong generative abilities to generate the occluded content.

To initially validate our motivations, we utilize the noisy latent $\bar{\mathbf{Z}}_T$ as the starting point of the diffusion process, which is derived by filling the regions of $\bar{\mathbf{Z}}_T^{inv}$ outside the visible portion with noise. To preserve the visible portion during the diffusion process, we introduce a **Latent Hold** strategy by replacing the visible region $\bar{\mathbf{M}}_v$ of the intermediate sampling latent $\bar{\mathbf{Z}}_t'$ with the corresponding region from $\bar{\mathbf{Z}}_t^{inv}$ in the same time step. This process can be formulated as

$$\bar{\mathbf{Z}}_t = \begin{cases} \boldsymbol{\epsilon}_T \otimes (1 - \bar{\mathbf{M}}_v) + \bar{\mathbf{Z}}_T^{inv} \otimes \bar{\mathbf{M}}_v & \text{if } t = T \\ \bar{\mathbf{Z}}_t' \otimes (1 - \bar{\mathbf{M}}_v) + \bar{\mathbf{Z}}_t^{inv} \otimes \bar{\mathbf{M}}_v & \text{otherwise} \end{cases}, \quad (5)$$

where $\boldsymbol{\epsilon}_T$ is the T-th step noise map, and $\otimes$ denotes the Hadamard product. As shown in Figure 3(c), the stable diffusion process successfully generates a complete realistic donut. However, the generated donut is much larger than the original occluded donut, resulting in over-generation issues.

The potential reason behind over-generation is that, in the early stages of the diffusion, noise level is so high that the model fails to accurately capture the visible portion. Consequently, the model generates content freely according to the input prompt but completely ignores the visible portion. To address this issue, we adopt two strategies: 1) We skip the early stages and start the diffusion process from the T_m-th step. 2) We fill the regions of $\bar{\mathbf{Z}}_{T_m}^{inv}$ outside the visible portion with uniform color. The process can be formulated as

$$\bar{\mathbf{Z}}_t = \begin{cases} \mathbf{J}_{T_m} \otimes (1 - \bar{\mathbf{M}}_v) + \bar{\mathbf{Z}}_t^{inv} \otimes \bar{\mathbf{M}}_v & \text{if } t = T_m \\ \bar{\mathbf{Z}}_t' \otimes (1 - \bar{\mathbf{M}}_v) + \bar{\mathbf{Z}}_t^{inv} \otimes \bar{\mathbf{M}}_v & \text{otherwise} \end{cases}, \quad (6)$$

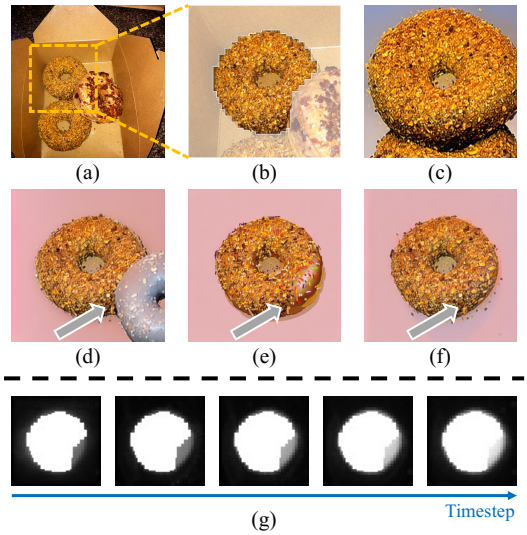


Figure 3: (a)-(b) showcase process of obtaining $\bar{\mathbf{I}}_s$ as Equ. (4). (b) marks $1-\bar{\mathbf{M}}_v$ with white mask. (c) - (f) are results from variants of De-occlusion Branch. (c) is generated by filling $1-\bar{\mathbf{M}}_v$ with noise as Equ. (5). (d) introduces color-fill strategy as Equ. (6). (e) is generated under the guidance of progressively updating masks. (f) is the full Deocclusion Branch. (g) showcases the progressively updating masks based on the refined cross-attention map $\bar{\mathbf{R}}_t^C$.

where $\mathbf{J}_{T_m}$ is a randomly colored image added with the T_m-th step noise. The **Color Fill** strategy not only decreases the difficulties in capturing the visible portion, but also encourages the model to focus on the target object by minimizing distractions from background generation.

However, relying solely on these strategies does not guarantee that the model will avoid regenerating undesired elements in the obscured areas, as shown in Figure 3(d). This necessitates the acquisition of a complete mask for the object to guide the diffusion process. To explicitly exploit the shape priors of the diffusion model, we extract the refined cross-attention map $\bar{\mathbf{R}}_t^C$ corresponding to the target object in each diffusion step. The progress of $\bar{\mathbf{R}}_t^C$ throughout the diffusion process is displayed in Figure 3(g). It shows that the refined cross-attention map can somehow represent the complete shape of the object and it becomes more and more accurate during the diffusion process. To utilize this shape prior to guide the diffusion process, we store the map $\bar{\mathbf{R}}_{t+1}^C$ from the previous timestep, and send it into the self-attention module, which restricts the attention module to query information exclusively from the target object.

Another challenge is the inconsistency in style between the generated and visible portions of the object, as depicted in Figure 3(e). Inspired by prior research (Shi et al. 2024; Gu et al. 2024; Avrahami et al. 2023), we conduct a style-preserving fine-tuning on the diffusion U-Net. The finetune process is implemented with Low-Rank Adaptation (LoRA), and the supervision is applied exclusively to the visible portion. Equipped with LoRA, the diffusion model is specifically tuned to ensure the generative style aligns with the visible parts of the object.

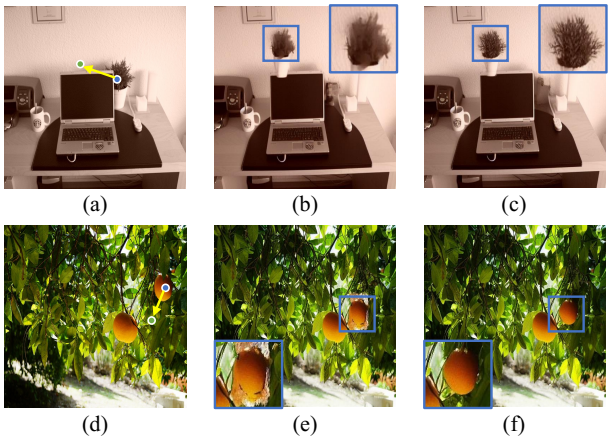


Figure 4: (a) and (d) are source images, and the others are results from variants of Movement Branch. The starting and ending points of the yellow arrows represent the original and target positions of the moved object. (b) is result with direct resizing as Equ. (8). (c) introduces the latent resizing operation as Equ. (9), alleviating the severe degradation. (e) and (f) are results w/o and w/ local text guidance, which helps the object integrate into surroundings more appropriately.

Movement Branch

Input Preperation The input of the movement branch is the source image $\mathbf{I}_s$ and the visible mask $\mathbf{M}_v$. After implementing the DDIM inversion on the source image $\mathbf{I}_s$, we can obtain the output noisy latent $\mathbf{Z}_T^{inv}$, which sets an appropriate starting point for the movement branch to preserve the consistency between the source and edited images. Besides, the intermediate keys $\{\mathbf{K}_t^{inv}\}_{t=0}^T$ and values $\{\mathbf{V}_t^{inv}\}_{t=0}^T$ are stored to provide guidance for subsequent diffusion process.

Key Designs The movement branch aims to place the fully de-occluded object at the target location accurately, preserve the background information, and inpaint the original region of the moved object. The input of the movement branch is initialized as $\mathbf{Z}_T^{inv}$ while the region $\mathbf{M}_v$ left at the original position of the object is filled with noise **(Noise Fill)**, which can be denoted as

$$\mathbf{Z}_T = \boldsymbol{\epsilon} \otimes \mathbf{M}_v + \mathbf{Z}_T^{inv} \otimes (1 - \mathbf{M}_v), \tag{7}$$

where $\mathbf{Z}_T$ is the noisy latent at the T-th step in the movement branch. During the forward propagation of the self-attention modules in the denoising process, we replace the keys $\mathbf{K}_t$ and values $\mathbf{V}_t$ generated from $\mathbf{Z}_t$ with the stored $\mathbf{K}_t^{inv}$ and $\mathbf{V}_t^{inv}$. Under the guidance of $\mathbf{M}_v$, the queries $\mathbf{Q}_t$ generated from $\mathbf{Z}_t$ are directed to retrieve the background contents from $\mathbf{K}_t^{inv}$ and $\mathbf{V}_t^{inv}$. The above operations can help the background contents consistent with the input without generating undesired elements and inpaint the original region of the moved object appropriately.

To place the de-occluded object at the target position, we introduce **Latent Optimization**, which strives to minimize the distance between the de-occluded object and the target region. Through latent optimization, the diffusion process is guided to generate the de-occluded object in the target region. Specifically, during each denoising step, after passing the noisy latent $\mathbf{Z}_{t+1}$ through the U-Net, we can obtain

$\mathbf{Z}'_t$. Then, we utilize the L2 distance between the complete object and the target region as the optimization objective, which is formulated as

$$\mathcal{L}_{mv}(\mathbf{Z}'_t) = \|\textbf{Crop}(\mathbf{Z}'_t, \mathbf{g}, r') - \textbf{Resize}(\bar{\mathbf{Z}}_t \otimes \bar{\mathbf{R}}_t^C, r')\|_2, \quad (8)$$

where $\mathbf{g}$ is the user-specified target position. r' is the crop size in the latent space, which equals $\lceil r/8 \rceil$ because the latent space is downsampled from the pixel space by a factor of 8. Then we can obtain the latent $\mathbf{Z}_t$ by optimizing $\mathbf{Z}'_t$ through gradient descent, denoted as $\mathbf{Z}_t = \mathbf{Z}'_t - \gamma \frac{\partial \mathcal{L}_{mv}(\mathbf{Z}'_t)}{\partial \mathbf{Z}'_t}$. However, direct resizing with bilinear interpolation is not suitable for latent space. As depicted in Figure 4 (b), when the latent is resized in the latent space, a significant degradation can be observed (Hwang, Park, and Jo 2024). We have found a straightforward solution to this issue: 1) Decode the latent to the pixel space. 2) Perform the resizing at the pixel level using bilinear interpolation. 3) Encode the resized pixel data back into the latent space. Thus, the latent resizing operation can be defined as

$$\textbf{L-Resize}(\mathbf{Z}, r') = \mathcal{E}(\textbf{Resize}(\mathcal{D}(\mathbf{Z}), r)). \quad (9)$$

Then, we replace the resizing operation in Equ. (8) with Equ. (9). The result, shown in Figure 4(c), demonstrates that this approach alleviates the degradation issues.

As depicted in Figure 4(e), relying solely on latent optimization is insufficient to guarantee harmonious integration. Therefore, we leverage the priors from the diffusion model by applying classifier-free guidance at the target position. We generate a mask $\mathbf{Q}$ marking the square region centered by $\mathbf{g}$ with the side length of r. After passing the noisy latent $\mathbf{Z}_t$ into the denoising U-Net, we apply text guidance (Ho and Salimans 2022) (**Local Text Guidance** in Figure 2) exclusively in the target region, which is denoted as

$$\epsilon_{pred,t} = \boldsymbol{\epsilon}_\theta(\mathbf{Z}_t, t) + \omega \mathbf{Q} \otimes (\boldsymbol{\epsilon}_\theta(\mathbf{Z}_t, t, \mathbf{c}) - \boldsymbol{\epsilon}_\theta(\mathbf{Z}_t, t)), \quad (10)$$

where ω represents the guidance scale. Guided by the text prompt, diffusion model adjusts the moved object in the target region to better align with the distribution of natural images. This adaptation ensures that the object integrates seamlessly into new surroundings, as observed in Figure 4(f).

Experiment

Evaluation Dataset

To compare the performance of our method with other existing arts, we develop a specialized evaluation dataset derived from COCOA (Zhu et al. 2017) training and validation sets. Given that our work primarily concentrates on object-level movement with occlusion, we filter the dataset to include images that feature occluded objects of considerable size. The final evaluation dataset comprises 120 images with a total of 150 sample objects. For each sample, the visible mask $\mathbf{M}_v$ is provided by the COCO dataset (Lin et al. 2014). As input text prompts $\mathbf{c}$ are needed for diffusion-based methods, we designed a prompt template `A photo of <category name>`, where the category name is also provided by COCO dataset. For each sample, we randomly set 8 different target positions, which results in 1200 testing cases in total.

Comparison on De-occlusion

Evaluation Metrics For quantitative evaluation of the realism of the de-occluded objects, we adopt the KID score (Bińkowski et al. 2018), comparing the de-occluded objects with the ground-truth complete objects in the COCOA dataset. Note that all object images are set against a white background to ensure a fair comparison. Following DreamBooth (Ruiz et al. 2023), we also use the CLIP-T metric to evaluate the prompt fidelity, which is measured as the average cosine similarity between prompt and image CLIP (Radford et al. 2021) embeddings.

Comparison with Other Methods To validate the effectiveness of our de-occlusion branch, we conduct a comparison with the object de-occlusion method, PCNet (Zhan et al. 2020). Table 1 reports the quantitative comparison results, demonstrating that our method outperforms PCNet in terms of image realism and prompt fidelity. Furthermore, the qualitative comparisons illustrated in Figure 5 highlight that our method can produce more satisfactory results than PCNet. Although PCNet can achieve relatively good results for simple cases, it struggles with the completion of large-scale occlusions and complex subjects. Leveraging the extensive real-world knowledge embedded in pre-trained diffusion models, our method excels at de-occluding complex objects and generating high-quality content.

Comparison on Occluded Object Movement

Evaluation Metrics For the evaluation of the occluded object movement, we mainly focus on the original position (OP) and the target position (TP) of the moved object. We adopt three evaluation metrics: DINO-OP, DINO-TP, and CLIP-TP. DINO-OP measures whether the target object leaves no residual at the original position. To this end, we crop the box area around the original position for both the source image $\mathbf{I}_s$ and the edited result $\mathbf{I}_e$, and use DINOv2 (Oquab et al. 2023) to measure the similarity between the box areas. A higher DINO-OP score indicates that the target object remains at the original position, which is undesirable. To measure whether the target object is indeed moved to the target position, we crop the box area around the original position in $\mathbf{I}_s$ and around the target position in $\mathbf{I}_e$. Similarly, we utilize DINOv2 to measure the cosine similarity between the crops. Higher DINO-TP scores represent that the target object is successfully moved to the target position. Additionally, we also compare the similarity of the image CLIP embeddings of these two patches, which reflects the object's fidelity and harmony with its surroundings.

Comparison with Other Methods In this section, we compare our method against feasible editing methods for occluded object movement. We divide these methods into four categories. 1) Paint-By-Example (PBE) (Yang et al. 2023) and AnyDoor (Chen et al. 2024) are designed to add an object to an image. To adapt them for object movement, we apply the image inpainting at the original position of the object, and then paste the object at the target position. 2) SD Inpainting (Rombach et al. 2022) is the state-of-the-art inpainting method, which we can also convert into occluded

Figure 5: Qualitative comparison on de-occlusion. PCNet struggles with the completion of large-scale occlusion and complex objects, while our method can generate high-quality content consistent with the target object.

Method	KID ↓	CLIP-T ↑
PCNet	0.0186	29.57
Ours	**0.0142**	**30.49**

Table 1: Quantitative comparison on de-occlusion. Our method outperforms PCNet for both image realism and prompt fidelity according to KID and CLIP-T, respectively.

object movement. With the visible mask $\mathbf{M}_v$, we can extract the object and paste it to the target position. By masking the original position and the areas around the target position, the model is forced to fill these regions with reasonable content. 3) DragDiffusion (Shi et al. 2024) is a point-dragging editing method, which we adapt for object movement by selecting multiple points on the target object. 4) DragonDiffusion (Mou et al. 2023) and DiffEditor (Mou et al. 2024) can be directly applied since it can tackle object movement.

The quantitative comparison is reported in Table 2. PBE and AnyDoor receive low scores in DINO-TP due to their inability to preserve the original pose and appearance of the edited object. SD Inpainting achieves high performance on DINO-TP because it effectively relocates the object to the target position by directly pasting it. However, this leads to a lack of harmony between the object and its new surroundings, reflected by the low CLIP-TP score. Additionally, it may generate undesired objects at the original position, leading to a poor DINO-OP score. DragDiffusion performs poorly across all three metrics as it primarily focuses on content dragging rather than object movement. DragonDiffusion and DiffEditor encounter significant issues with residual artifacts, which negatively impacts their DINO-OP scores. Our method outperforms the baselines, which is also supported by the qualitative comparison in Figure 6.

User Study To further evaluate the visual quality, we invite 60 volunteers for a user study. We select four methods (AnyDoor, SD Inpainting, DragDiffusion, and DiffEditor) to make the comparison, and each method generated 40 edited results. For each comparison, volunteers are required to choose whether our result is better than one of the methods. As shown in Table 3, our method is preferred over these methods with a higher winning percentage.

Method	DINO-OP ↓	DINO-TP ↑	CLIP-TP ↑
PBE	0.575	0.690	0.950
AnyDoor	0.651	0.728	0.952
SD Inpainting	0.745	**0.742**	0.949
DragDiff	0.647	0.706	0.933
DragonDiff	0.673	0.730	0.957
DiffEditor	0.678	0.731	0.958
Ours	**0.561**	**0.742**	**0.960**

Table 2: Quantitative comparison on occluded object movement. Our method outperforms the compared methods.

Ours vs.	OP	TP	Realism
AnyDoor	86.8%	84.3%	85.0%
SD-Inpainting	78.0%	79.3%	75.8%
DragDiffusion	81.0%	80.5%	80.9%
DiffEditor	80.8%	79.0%	78.8%

Table 3: User Study. Users are asked to choose a better result (Ours vs. the baseline) in terms of: 1) No residual artifacts at the original position (OP). 2) Placement of the completed object at the target position (TP). 3) Maintenance of image realism (Realism). The numbers indicate the winning rate of our method over the compared method.

Ablation Study

We conduct ablation study for the following components and the results are reported in Table 4. 1) **Color Fill Strategy (CF)**: Removing CF harms the DINO-TP score, as it may cause the problem of over-generation, which reduces the similarity to the original object. 2) **Attention Guidance (AG)**: For the de-occlusion branch, without the progressive updating mask restricting the attention module to query information exclusively from the target object, it may generate undesired elements, resulting in a lower DINO-TP score. For the movement branch, removing the mask indi-

Figure 6: Qualitative comparison on occluded object movement. The target objects are marked by yellow masks. The starting and ending points of the orange arrows represent the original and target positions of the moved object.

Method	DINO-OP ↓	DINO-TP ↑	CLIP-TP ↑
Ours	**0.561**	**0.742**	**0.960**
w/o CF	0.565	0.717	0.957
w/o AG	0.612	0.731	0.959
w/o LoRA	0.564	0.733	0.958
w/o LR	0.561	0.695	0.958
w/o LTG	0.563	0.732	0.955

Table 4: Ablation Study. We conduct ablation study on the following components: 1) w/o Color Fill Strategy (CF), 2) w/o Attention Guidance (AG), 3) w/o LoRA, 4) w/o Latent Resizing(LR), and 5) w/o Local Text Guidance (LTG).

cating the background region, the diffusion model may regenerate the original object at the original position, resulting in a high DINO-OP score. 3) **LoRA**: Removing the LoRA causes the generated portion inconsistent with the original portion, which reduces the DINO-TP score. 4) **Latent Resizing(LR)**: When removing the LR and adopting the pixel resizing directly, the target object encounters severe degradation, significantly lowering the DINO-TP score. 5) **Local Text Guidance (LTG)**: LTG aims to integrate the target object seamlessly into new surroundings. Therefore, removing LTG harms the harmony of the target region, which is reflected in the lower score of CLIP-TP.

Integration with Other Methods

Our dual-branch framework allows for the decomposition of the object from its background, enabling focused edits directly on the object. The de-occlusion branch is compatible with most existing editing methods, enhancing their ability to perform precise edits in complex scenes. Figure 7 provides two examples where our de-occlusion branch is integrated with two typical editing methods, text-conditioned method (MasaCtrl (Cao et al. 2023)) and drag-style method

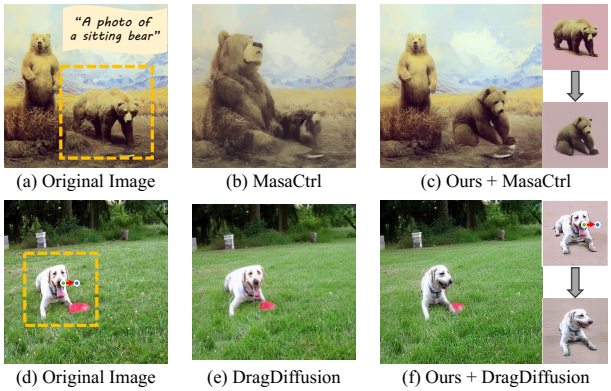


Figure 7: Examples of integrating our framework with MasaCtrl and DragDiffusion. Our framework enables two editing methods to perform precise editing in complex scenes.

(DragDiffusion). The main advantages of our framework in enhancing these methods are: 1) In scenarios with multiple similar objects, our framework allows for the editing of a specific object without affecting others. 2) By isolating the object from a complex background, our method permits more precise control over the object.

Conclusion

In this paper, we propose a diffusion-based framework for occluded object movement. We demonstrate that extensive prior knowledge within diffusion models is helpful for this task. In our dual-branch network, the de-occlusion branch completes the occluded portion of the target object, while the movement branch places the restored object at the target position. Moreover, our framework can be integrated with existing editing methods, enabling them to perform precise editing in complex scenes. We hope that our framework will serve as a valuable tool for image editing tasks in the future.

Acknowledgments

This work is funded by the National Natural Science Foundation of China (62306153), the Fundamental Research Funds for the Central Universities (Nankai University, 070-63243143), the China Postdoctoral Science Foundation (GZB20240357, 2024M761682), and Shui Mu Tsinghua Scholar (2024SM079). The computational devices of this work is supported by the Supercomputing Center of Nankai University (NKSC).

References

Avrahami, O.; Aberman, K.; Fried, O.; Cohen-Or, D.; and Lischinski, D. 2023. Break-a-scene: Extracting multiple concepts from a single image. In *SIGGRAPH*.

Avrahami, O.; Gal, R.; Chechik, G.; Fried, O.; Lischinski, D.; Vahdat, A.; and Nie, W. 2024. DiffUHaul: A Training-Free Method for Object Dragging in Images. *arXiv preprint arXiv:2406.01594*.

Bińkowski, M.; Sutherland, D. J.; Arbel, M.; and Gretton, A. 2018. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*.

Brooks, T.; Holynski, A.; and Efros, A. A. 2023. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*.

Cao, M.; Wang, X.; Qi, Z.; Shan, Y.; Qie, X.; and Zheng, Y. 2023. Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. In *ICCV*.

Chefer, H.; Alaluf, Y.; Vinker, Y.; Wolf, L.; and Cohen-Or, D. 2023. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. *ACM TOG*.

Chen, X.; Huang, L.; Liu, Y.; Shen, Y.; Zhao, D.; and Zhao, H. 2024. Anydoor: Zero-shot object-level image customization. In *CVPR*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Epstein, D.; Jabri, A.; Poole, B.; Efros, A.; and Holynski, A. 2023. Diffusion self-guidance for controllable image generation. In *NeurIPS*.

Gu, Y.; Wang, X.; Wu, J. Z.; Shi, Y.; Chen, Y.; Fan, Z.; Xiao, W.; Zhao, R.; Chang, S.; Wu, W.; et al. 2024. Mix-of-show: Decentralized low-rank adaptation for multi-concept customization of diffusion models. In *NeurIPS*.

Hertz, A.; Mokady, R.; Tenenbaum, J.; Aberman, K.; Pritch, Y.; and Cohen-Or, D. 2022. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. In *NeurIPS*.

Ho, J.; and Salimans, T. 2022. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Hwang, J.; Park, Y.-H.; and Jo, J. 2024. Upsample guidance: Scale up diffusion models without training. *arXiv preprint arXiv:2404.01709*.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *ECCV*.

Liu, Z.; Liu, Q.; Chang, C.; Zhang, J.; Pakhomov, D.; Zheng, H.; Lin, Z.; Cohen-Or, D.; and Fu, C.-W. 2024. Object-level Scene Deocclusion. In *SIGGRAPH*.

Mou, C.; Wang, X.; Song, J.; Shan, Y.; and Zhang, J. 2023. Dragondiffusion: Enabling drag-style manipulation on diffusion models. *arXiv preprint arXiv:2307.02421*.

Mou, C.; Wang, X.; Song, J.; Shan, Y.; and Zhang, J. 2024. Diffeditor: Boosting accuracy and flexibility on diffusion-based image editing. In *CVPR*.

Nguyen, Q.; Vu, T.; Tran, A.; and Nguyen, K. 2024a. Dataset diffusion: Diffusion-based synthetic data generation for pixel-level semantic segmentation. In *NeurIPS*.

Nguyen, T.-T.; Nguyen, D.-A.; Tran, A.; and Pham, C. 2024b. FlexEdit: Flexible and Controllable Diffusion-based Object-centric Image Editing. *arXiv preprint arXiv:2403.18605*.

Oquab, M.; Darcet, T.; Moutakanni, T.; Vo, H.; Szafraniec, M.; Khalidov, V.; Fernandez, P.; Haziza, D.; Massa, F.; El-Nouby, A.; et al. 2023. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*.

Ozguroglu, E.; Liu, R.; Surís, D.; Chen, D.; Dave, A.; Tokmakov, P.; and Vondrick, C. 2024. pix2gestalt: Amodal segmentation by synthesizing wholes. In *CVPR*.

Pan, X.; Tewari, A.; Leimkühler, T.; Liu, L.; Meka, A.; and Theobalt, C. 2023. Drag your gan: Interactive point-based manipulation on the generative image manifold. In *SIGGRAPH*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *ICML*.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *CVPR*.

Ruiz, N.; Li, Y.; Jampani, V.; Pritch, Y.; Rubinstein, M.; and Aberman, K. 2023. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *CVPR*.

Sajnani, R.; Vanbaar, J.; Min, J.; Katyal, K.; and Sridhar, S. 2024. GeoDiffuser: Geometry-Based Image Editing with Diffusion Models. *arXiv preprint arXiv:2404.14403*.

Shi, Y.; Xue, C.; Liew, J. H.; Pan, J.; Yan, H.; Zhang, W.; Tan, V. Y.; and Bai, S. 2024. Dragdiffusion: Harnessing diffusion models for interactive point-based image editing. In *CVPR*.

Song, J.; Meng, C.; and Ermon, S. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Tumanyan, N.; Geyer, M.; Bagon, S.; and Dekel, T. 2023. Plug-and-play diffusion features for text-driven image-to-image translation. In *CVPR*.

Vaswani, A. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Xu, K.; Zhang, L.; and Shi, J. 2024. Amodal completion via progressive mixed context diffusion. In *CVPR*.

Yang, B.; Gu, S.; Zhang, B.; Zhang, T.; Chen, X.; Sun, X.; Chen, D.; and Wen, F. 2023. Paint by example: Exemplar-based image editing with diffusion models. In *CVPR*.

Zhan, G.; Zheng, C.; Xie, W.; and Zisserman, A. 2024. Amodal ground truth and completion in the wild. In *CVPR*.

Zhan, X.; Pan, X.; Dai, B.; Liu, Z.; Lin, D.; and Loy, C. C. 2020. Self-supervised scene de-occlusion. In *CVPR*.

Zhu, Y.; Tian, Y.; Metaxas, D.; and Dollár, P. 2017. Semantic amodal segmentation. In *CVPR*.