General Articulated Objects Manipulation in Real Images via Part-Aware Diffusion Process

Zhou Fang **Yong-Lu Li*** **Lixin Yang** **Cewu Lu***
Shanghai Jiao Tong University
{joefang, yonglu_li, siriusyang, lucewu}@sjtu.edu.cn

Abstract

Articulated object manipulation in real images is a fundamental step in computer and robotic vision tasks. Recently, several image editing methods based on diffusion models have been proposed to manipulate articulated objects according to text prompts. However, these methods often generate weird artifacts or even fail in real images. To this end, we introduce the Part-Aware Diffusion Model to approach the manipulation of articulated objects in real images. First, we develop Abstract 3D Models to represent and manipulate articulated objects efficiently. Then we propose dynamic feature maps to transfer the appearance of objects from input images to edited ones, meanwhile generating the novel-appearing parts reasonably. Extensive experiments are provided to illustrate the advanced manipulation capabilities of our method concerning state-of-the-art editing works. Additionally, we verify our method on 3D articulated object understanding for embodied robot scenarios and the promising results prove that our method supports this task strongly. The project page is at https://mvig-rhos.com/pa_diffusion.

1 Introduction

Image editing is a long-standing popular computer vision task. Specifically, manipulating articulated objects has garnered significant attention owing to its application in various fields, such as image augmentation for downstream tasks [43], building goal conditions to train reinforcement learning models for robotic manipulation [41, 55], creating videos with extra supervision information [33], detecting human-object interactions [21, 22, 25], reasoning object affordance [24, 23], *etc*. Thanks to the large-scale training data and immense computing power, diffusion-based [40] generative models have achieved surprising results in the field of image and video generation.

Inspired by these successes, several recent works have adopted diffusion models as the backbone and implemented text-guided object manipulation [19, 15, 7, 51]. We can properly divide these studies into a couple of groups. The first one is to directly edit 2D images by transferring the feature/attention maps from original images to edited ones such as [12, 31, 7]. However, weird artifacts are prone to appear when the objects are rotated and deformed, or novel views appear. Consequently, these methods are restricted to structure-preserving image editing. Another group relies on reconstructing 3D object models. As the most related work to ours, [51] reconstructed 3D object models for manipulation and projected them back to images later. Nevertheless, this approach depends on the quality of reconstructed 3D models heavily. And the reconstruction model has to be fine-tuned when dealing with new categories. Moreover, manipulation has to be done manually which is laborious and impractical to support editing large quantities of images.

To address these problems, we propose the Part-Aware Diffusion Model (PA-Diffusion model) for articulated object manipulation in real images, as illustrated in Fig. 1. Firstly, we introduce the

*Corresponding authors.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

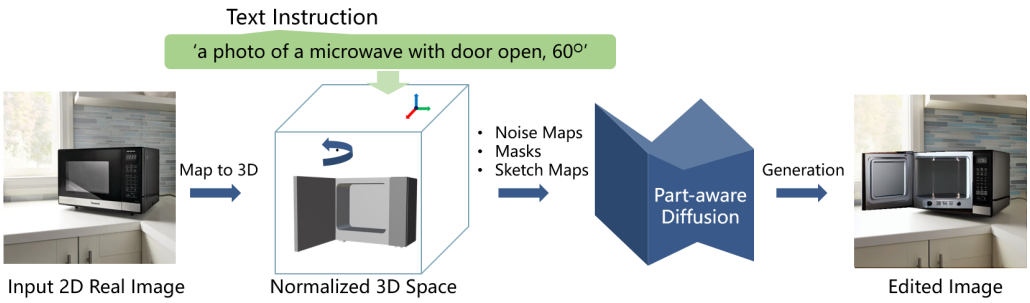

Figure 1: We propose the **Part-Aware Diffusion Model**: Abstract 3D model of the articulated object is constructed referring to the input 2D real image. Various manipulations can be done based on the text instruction or human interaction in the 3D space, the generation model then creates the edited image according to the manipulation.

concept of Abstract 3D Model and build a Primitive Prototype Library to represent articulated objects in the 3D space, so that our method can not only cover many common objects but also handle novel categories without extra training data or fine-tuning processes. Besides, various manipulation can be done efficiently. Second, we proposed dynamic feature maps to assist generation models in accurately transferring object appearances to accurate locations in edited images. As a result, weird artifacts are eliminated, and meanwhile, the novel-appearing parts are generated more reasonably. Finally, owing to the simple manipulation and editing process, the procedure is automatic and the model can strongly support other tasks by editing a large volume of images.

Our main contributions are summarized as follows:

(1) We introduce the concept of the Abstract 3D Model which accurately and robustly represents various articulated object categories with primitive prototypes. Meanwhile, novel categories can also be incorporated quickly. In addition, the articulated objects can be efficiently manipulated with text instructions or human interactions in the 3D space.

(2) We propose dynamic feature maps that let the diffusion model comprehend the object structure. Consequently, the diffusion model can generate novel views or novel-appearing parts of objects reasonably and preserve the appearance of the seen parts simultaneously.

(3) We present comprehensive experiments to highlight the advantages of our PA-Diffusion model including comparing with state-of-the-art editing methods both qualitatively and quantitatively, choosing a 3D articulated object understanding experiment to demonstrate how our method supports the tasks in embodied robot scenarios.

2 Related Work

2.1 Diffusion Model for Image Generation

In recent years, diffusion models [39, 38, 10] have achieved great success in the fields of image/video generation [6, 16], segmentation [4, 50], and many downstream computer vision tasks. To make the generation results controllable, [34] first proposed to extract and incorporate text features into the denoising process. Following this concept, [40, 11, 42, 14, 8] improved the performance of text-guided diffusion models with more effective text embedding methods.

However, as an implicit instruction, text guidance is still not strong enough to finish fine-grained image control such as determining the image layouts, objects' shape and texture, and so on. To make up this gap, [46] provided structural guidance by enhancing the similarity between the features of other conditions and the text guidance. [13, 5] proposed to modify the cross-attention maps and then guide the denoising process. To handle more complex scenarios and achieve more precise control, [52] and [32] proposed adding an extra module to the diffusion model. Then extra condition information can be imported to guide the denoising process.

2.2 Diffusion Based Image Editing

Considering the remarkable capability of understanding images, several recent works have also reported editing real and synthetic images with using diffusion models as the backbone. These methods can generally be summarized into two groups: Inversion-Based and Feature-Sharing Based.

The first group is primarily based on adding extra control to the inverted noise maps of images, then re-generating the image such as [19]. However, because the deterministic DDIM sampling process cannot be reversed perfectly, these methods struggle to preserve the appearance of original objects and backgrounds precisely. The second group attempts to maintain the appearance of objects by transferring the feature/attention/activation maps between guidance and generation branches or by adding extra loss items during the denoising process, as seen in [7, 31, 12]. Recent approaches like DragGAN [35] and DragDiffusion [31] propose to utilize a point-to-point dragging scheme, which can achieve refined content dragging. Nonetheless, these approaches often perform poorly on articulated object manipulation in real images, resulting in weird and blurry artifacts in edited images.

2D-3D-2D is another promising way of image editing, the recent work [51] introduced reconstructing 3D models from 2D images and projecting them back after manipulation. However, this approach highly relies on the quality of 3D reconstructed models, and reconstructing 3D models from a single 2D image is still a challenging task.

In contrast to the aforementioned approaches, our method demonstrates advantages when manipulating articulated objects in real images - high fidelity edited images, easy and various manipulations, covering multiple categories, and incorporating novel categories quickly.

3 Method

3.1 Overview

In this session, we go through the proposed PA-Diffusion model in detail. The overall architecture is demonstrated in Fig. 2. Initially, we reconstruct abstract 3D models for articulated objects with the Primitive Prototype Library. Then various manipulations can be done according to text instructions or human interactions. Next, leveraging DDIM Inversion [44, 30], initial inverted noise maps are created and manipulated following the previous actions. During the generation stage, we introduce dynamic feature maps, including manipulated inverted noise maps and compositional activation maps and images. These ensure that the appearance of seen parts of objects can be preserved accurately and that novel-appearing parts are generated reasonably. Besides, Texture and Style Consistency Score Loss are introduced to alleviate the blurry and style mismatch problems.

3.2 Preliminary

Diffusion models aim to convert random Gaussian noise into high-resolution images through a sequential denoising and sampling process [12]. Given the conditioning y, we start from the initial Gaussian noise map z_t, and then iteratively estimate the reduced noise $\hat{\epsilon_t}$ at each time step t:

$$\begin{aligned} \hat{\epsilon_t} &= \epsilon_\theta(z_t; t, y), \\ z_{t-1} &= update(z_t, \hat{\epsilon_t}, t, t-1, \epsilon_{t-1}), \end{aligned} \tag{1}$$

The update function could be DDPM [17], DDIM [44], or other sampling methods. Nevertheless, conventional sampling from conditional diffusion models often fails to produce high-quality images that align well with the condition y. To enhance the effect of the desired condition, extra class loss guidance is added to the reduced noise during the sampling process such as Classifier or Classifier-free guidance [45, 18].

Classifier guidance is introduced to generate conditional samples from an unconditional model by combining the unconditional score ϵ_t with a classifier $p(y|z_t)$, where $p(y|z_t)$ is the probability distribution of condition y based on the noise at time step t:

$$\hat{\epsilon_t} = \epsilon_\theta(z_t; t, y) + \beta \bigtriangledown_{z_t} p(y|z_t), \tag{2}$$

Classifier-free guidance eliminates the need for a separate classifier by incorporating the class information directly into the generative model as follows:

$$\hat{\epsilon_t} = (1+\alpha)\epsilon_\theta(z_t; t, y) - \alpha\epsilon_\theta(z_t; t), \tag{3}$$

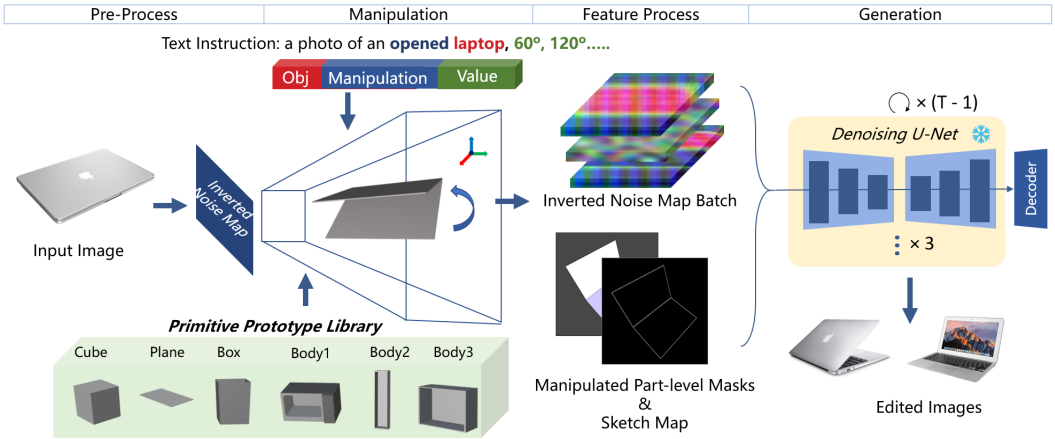

Figure 2: The overall image editing process. (1) In the Pre-Process stage, articulated objects in 2D images are part-level segmented and reconstructed to abstract 3D models. Meanwhile, inverted noise maps of input images are created with DDIM Inversion. (2) In the Manipulation stage, various manipulations can be implemented based on text instructions or human interaction in the 3D space . (3) After manipulation, part-level masks and sketches are rendered and exported. The inverted noise maps are transformed according to these masks. (4) Finally, with the transformed inverted noise maps, sketch maps, and part-level masks, the generation model creates the edited images.

Following these concepts, custom energy functions can also be utilized to guide the denoising process, instead of the probability function. In [12] [29] [54], various energy functions g are incorporated alongside classifier-free guidance to obtain high-fidelity samples as follows:

$$\hat{\epsilon_t} = (1+\alpha)\epsilon_\theta(z_t;t,y) - \alpha\epsilon_\theta(z_t;t) + \beta \bigtriangledown_{z_t} g(z_t;t,y), \tag{4}$$

Our proposed PA-Diffusion model is built on the diffusion model with classifier-free guidance. Extra energy functions are employed during the image editing process.

3.3 Various Manipulations in the 3D Space

As a promising workaround to the methods of dealing with images directly, the 2D-3D-2D pipeline has successfully handled many articulated objects with precise 3D models. Unfortunately, creating 3D models for various categories from a single image remains challenging, particularly for novel categories or instances. In this work, we introduce the concept of Abstract 3D Model that reconstructs accurate 3D models, supports efficient object manipulation, and incorporates novel objects easily.

Abstract 3D Model. Unlike previous methods, there is no need for precise 3D models of our method, the conditional information we have to provide to the diffusion model is coarse sketch maps and part-level masks. Therefore, we introduce the use of an abstract 3D model to represent the articulated object. As an abstract 3D model, the object is represented by combining several basic prototypes. As depicted at the bottom of Fig. 2, the laptop can be represented by two planes, storage furnitures and microwaves can be represented by a plane and a box. Primitive Prototype Library, which includes basic 3D prototypes such as planes, cubes, and boxes, can support common articulated object categories involving both rotation and translation joint types.

Camera Alignment. Next, we compute the camera pose in the 3D space and align the 2D real image view with the 3D space camera view. The pose computation problem is to calculate the intrinsic and extrinsic matrices for the camera that minimize the reprojection error from 3D-2D point correspondences [3]. Thus, in this work, we first employ Large-scale Segmentation Models to obtain the initial part-level segmentation masks of articulated objects M^{Init} and then detect the extreme corner points A, B, C, D (pts_1) with simple corner detection functions. These 2D extreme points are aligned with their 3D counterparts $A^{'}, B^{'}, C^{'}, D^{'}$ (pts_2, pre-defined in Primitive Prototype Library) as shown in Fig. 3. Finally, based on Perspective n-Points 2D-3D method [49], the camera matrices can be extracted, and 2D-3D views are aligned.

Manipulation. By representing objects with primitive prototypes, multiple types of manipulations can be implemented in the 3D space efficiently with the assistance of 3D computer graphics software.

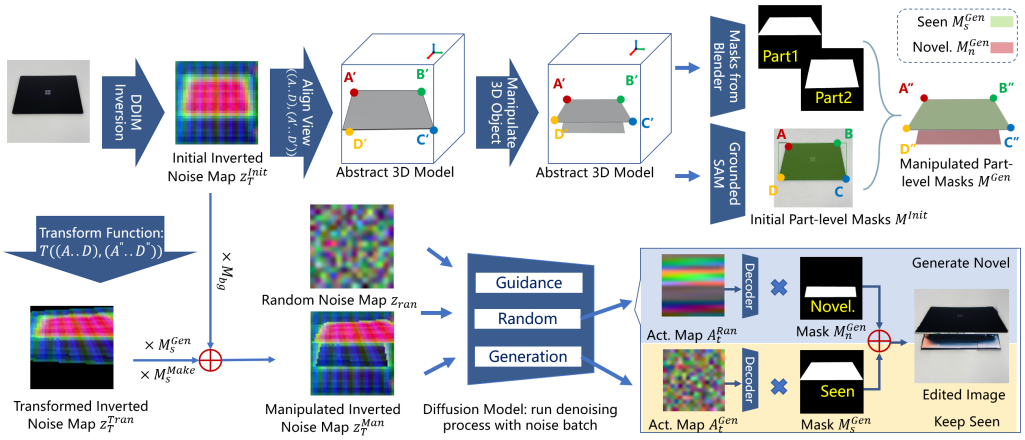

Figure 3: Algorithm pipeline of the PA-Diffusion model. Symbols and procedures in the figure are the same as those in the content.

As shown in Fig. 2, manipulating objects through text instructions is a concise approach. For example, the instruction *opening the laptop* 120° is converted to a script, then the manipulation will be done by running the script in 3D software. Our PA-Diffusion model also supports human interaction, which could be even more efficient. Additional manipulation guidance is provided in the Appendix. Contrary to the tedious manipulation experience of previous SOTA works, our proposed PA-Diffusion model offers a more flexible approach to editing articulated objects.

Structure Disentangle. Some object parts could still be seen after manipulation, and some unseen parts would appear. As shown in the top right part in Fig. 3, the laptop shell can be seen in the input image. After opening the laptop, the shell can still be seen, however, the keyboard is newly revealed. Therefore, to distinguish them and implement part-aware diffusion, we disentangle articulated objects into **seen parts** and **novel-appearing parts**. The appearance of seen parts should be consistent between input and edited images, and the style of novel-appearing parts should be consistent with the objects' overall appearance. In this work, we regard all the initial part-level masks M^{Init} as seen. Then after manipulation, we obtain manipulated seen parts mask M_s^{Gen} and manipulated novel-appearing parts mask M_n^{Gen}, both of them can be exported from the 3D software automatically. We use M^{Gen} to present the union of M_s^{Gen} and M_n^{Gen}.

3.4 Dynamic Feature Maps

To maintain the object's appearance including color and texture, previous editing methods introduced a **guidance branch** to invert and re-generate the input image, and a **generation branch** to create the edited image, the attention/feature/activation maps are transferred from the guidance to the generation branch directly [7]. However, when changing the object location or shape during manipulation, directly sharing these maps would transfer the feature from the input image to undesired locations in the edited image. Furthermore, these methods cannot reasonably generate novel views or novel-appearing parts.

To overcome these problems, we propose dynamic feature maps including manipulated inverted noise maps and compositional activation maps and images. To keep appearance accurate, manipulated inverted noise maps transfer the feature of seen parts in input images to the manipulated seen parts in edited images. Simultaneously, to make novel-appearing parts reasonably, compositional activation maps and images let the diffusion model create these parts from random noise. The following content describes how to manipulate the noise maps, how to construct compositional maps, and how they work. For clarity, the process is presented in Fig. 3.

Manipulated inverted noise map. As shown at the top of Fig. 3, we firstly reverse the input image to the initial inverted noise map z_T^{Init} with DDIM inversion. After 3D manipulation, we calculate the transform function T based on the initial M^{Init} and manipulated seen part-level masks M_s^{Gen} and then compute the transformed inverted noise map z_T^{Tran} with T. The manipulated inverted noise map is created as the following equation:

$$z_T^{Man} = z_T^{Tran} \times M_s^{Gen} + z_T^{Tran} \times M_s^{Make} + z_T^{Init} \times M_{bg}, \tag{5}$$

where M_s^{Make} is the makeup mask generated by $XOR(M^{Init},\ M^{Init} \cap M_s^{Gen})$. M_{bg} is the background mask created by $1 - M_s^{Gen}$. In addition, we also create a random noise map z_T^{Ran} to generate novel-appearing parts. The three noise maps z_T^{Init}, z_T^{Ran}, and z_T^{Man} will be sent to the denoising UNet as a batch in the next step.

Compositional activation map. As shown at the bottom of Fig.3, the diffusion model runs a sequential denoising process with the three noise maps as a batch and generates three output activation maps and images: According to previously defined masks, activation maps A_t^{Gui} and A_t^{Gen} from the guidance and generation branches are merged. Then novel appearing parts are pasted from the random branch as follows:

$$\begin{aligned} A_t^{Gen} &= A_t^{Gen} \times M_s^{Gen} + A_t^{Gui} \times \left(1 - M_s^{Gen}\right) \\ I^{Gen} &= I^{Ran} \times M_n^{Gen} + I^{Gen} \times \left(1 - M_n^{Gen}\right) \end{aligned} \tag{6}$$

Owing to the above steps, we transfer the feature of seen parts in input images to the accurate location in edited images. At the same time, all the other contents and the background in input images can be preserved, as the highlighted yellow part in Fig. 3. Besides, as the highlighted blue part in Fig. 3, an extra image is synthesized with random noise map z_T^{Ran}, the novel-appearing parts are cropped and pasted to edited images from the extra image, which makes these parts more reasonable and consistent with the original inputs.

3.5 Score Function

Texture Consistency Score Loss. However, simply manipulating the inverted noise map will lead to a serious blurry problem. This is due to the denoising process includes several convolution steps. As the initial inverted noise map z_T^{Init} is not rotation invariant, manipulating z_T^{Init} will disturb the original distribution and make the denoising process fail. To alleviate this limitation, we construct Texture Consistency Score Loss (TCSL) [31] as an extra supervision that lets the specific region in the generation branch match with the one in the guidance branch,

$$Loss_t = \frac{\varphi_{fg}}{cos(A_t^{Gui}[M^{Init}], A_t^{Gen}[M_s^{Gen}])} + \frac{\varphi_{bg}}{cos(A_t^{Gui}[1 - M^{Init}], A_t^{Gen}[1 - M_s^{Gen}])}, \tag{7}$$

where φ_{fg} and φ_{bg} are hyper-parameters. We add this loss item as an extra loss in classifier guidance in each denoising iteration step to calibrate the appearance of objects.

Style Consistency Score Loss. For novel-appearing parts, the diffusion model is prone to randomly select a style to generate them with text guidance or sketch maps. As a result, the texture and style could be different from the objects in input images. Therefore, we introduce Style Consistency Score Loss (SCSL) to calibrate the style of seen parts and the novel-appearing parts.

Different from Texture Consistency Score Loss, there is no need to match every pixel in input images and edited images. Thus we calculate L1 loss between the activation maps of the guidance and the generation branch [12]. The loss function is as follows:

$$Loss_s = \left| sum(A_t^{Gui}[M^{Init}]) - sum(A_t^{Gen}[M_n^{Gen}]) \right|_1, \tag{8}$$

This loss item is also added as an extra classifier guidance. The final reduced noise in each denoising iteration is as follows, where γ_1 are γ_2 are the hyper-parameter weights of TCSL and SCSL,

$$\hat{\epsilon_t} = (1 + \alpha)\epsilon_\theta(z_t; t, y) - \alpha\epsilon_\theta(z_t; t) + \gamma_1 \bigtriangledown_{z_t} Loss_t + \gamma_2 \bigtriangledown_{z_t} Loss_s. \tag{9}$$

4 Experiment

In this section, we provide two kinds of experiments to prove the advantages of our proposed PA-Diffusion model. First, various image editing tasks are conducted to showcase the model's image editing capabilities. To highlight the superiority of our model compared with state-of-the-art methods, we collect a testbench and evaluate all the methods both qualitatively and quantitatively. Second, we create a synthetic training set to support the challenging 3D articulated object understanding task in the robotic scenarios.

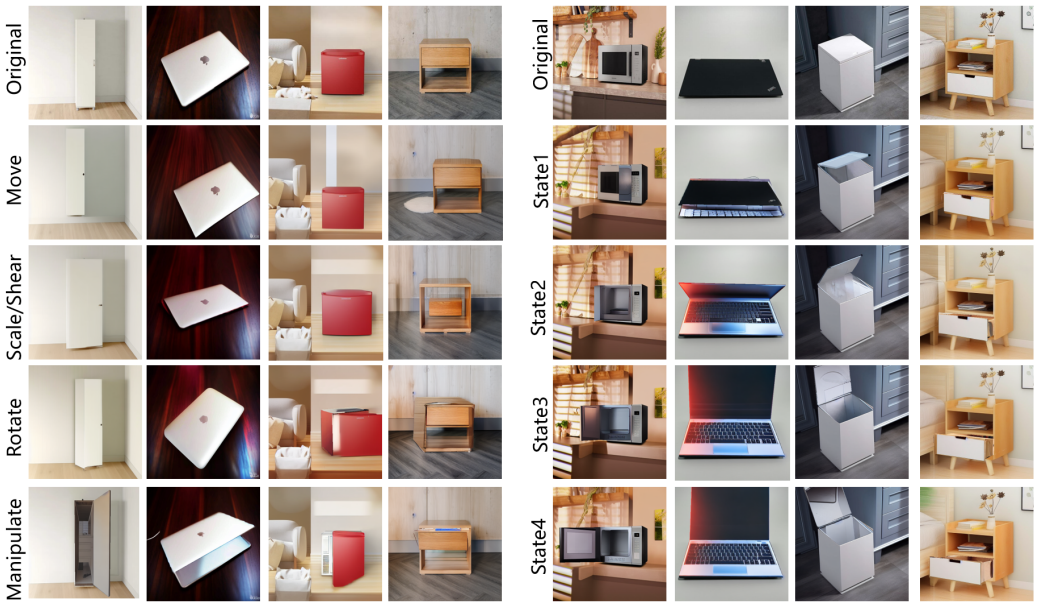

Figure 4: Results of basic manipulations: move, scale/shear. rotate, and manipulate. Blank regions caused by the manipulation are in-painted automatically. The novel views and novel-appearing parts match with the style of the seen parts (left). Articulated objects are opened from the initial close state with 4 steps. The appearance of the increasing novel-appearing parts keeps being consistent throughout the whole process (right).

4.1 Implementation

In this work, we select Grounded Segment Anything [20, 28] to obtain the initial part-level object segmentation masks. T2I Adapter [32] is chosen as the conditional generation model, and the condition we used is the sketch map. The fundamental diffusion model is Stable Diffusion V1-5. All experiments run on a single NVIDIA A100 GPU. Notably, **NO** models need to be trained or fine-tuned in the image editing process.

Primitive Prototype Library is built within Blender [9]. 3D planes, cubes, boxes, and other 3D primitive shapes are created and combined to represent different objects. In this work, 5 primitive shapes are created to represent 6 categories of articulated objects. The ease of creating prototypes allows for the quick incorporation of novel categories or instances. Various manipulations can be implemented in Blender efficiently.

4.2 Results

Fig. 4 demonstrates the editing results of some basic manipulations and a sequential manipulation process. Move, scale/shear, and other basic manipulations are finished with text instructions. Opening is implemented with human interaction. As shown in the left part, our PA-Diffusion model naturally manipulates articulated objects with rotation or translation joint types. The edited objects blend seamlessly with other contents and backgrounds in the original images. When we move or rotate the objects, the blank regions in the background are in-painted semantically according to the surroundings. Moreover, in-painting and editing are completed in a single denoising process by the PA-Diffusion model, no extra in-paint model or process is required. Last but not least, novel-appearing parts of objects are generated reasonably, and the style matches with the object. For example, the storage furniture is empty after opening, and the keyboard of the laptop is designed logically.

The right part of Fig. 4 presents a complete operation process of opening articulated objects, from the initial closed state to the open state with 4 steps progressively. The appearance of objects' seen parts is transferred from the original input to different states accurately. The point that needs to be mentioned is that along with the operation progress, more novel-appearing parts of objects appear, our proposed PA-Diffusion model keeps the style and texture of these novel-appearing parts

Figure 5: Manipulate non-rigid objects, non-uniform shapes, and objects with weird or multiple joint types.

being consistent throughout the process. This capability potentially allows our method to generate a complete manipulation video from a single input image, maintaining consistent object appearance and style even as novel views or parts increase.

Generally, articulated objects' parts are rigid and connected with one of the typical joint types - rotation and translation. However, we are surprised to notice that our PA-Diffusion model can also handle non-rigid objects with non-uniform shapes, weird joints, and manipulations. As illustrated in Fig. 5, we first select toys as examples of non-rigid with non-uniform shapes, the tail of the shark is moved up together with other close parts as deformable objects. Then, we broke the cup in a real image. Third, the kitchen pot and storage furniture are opened to illustrate the case of weird and multiple joint types within one object. No matter whether the shape of object parts has changed after manipulation or joint types are unconventional, the PA-Diffusion model can edit all the objects successfully. Meanwhile, the background is preserved or inpainted well.

4.3 Ablation Study

As mentioned in Section 3, TCSL is added to the denoising loss function to release the serious blurry problem. As shown in the left of Fig. 6, we move the storage furniture, the microwave, and the drawer to different directions. It can be seen that without TCSL, the objects are prone to be blurry. More seriously, the edited image could be destroyed, as in the example of the drawer. On the other hand, with TCSL, the texture of objects can be transferred to the desired location, meanwhile, the blank region caused by the movement is in-painted well. SCSL is another loss to keep the style consistent between seen parts and novel-appearing parts. Its effectiveness is shown in the left bottom of Fig. 6 (with SCSL). We notice that when opening the storage furniture, the style of the novel-appeared inner door and body part is more likely to be consistent with the outer body part with SCSL, which makes the edited image natural. More quantitative ablation studies about the two score losses are provided in the Appendix.

4.4 Comparison

To further evaluate our PA-Diffusion model, we compare it with four state-of-the-art image editing approaches that are based on diffusion models: Imagic, DragDiffusion, MasaCtrl (with T2I Adapter), and Image Sculpting. In this experiment, we require these models to manipulate different categories of articulated objects, including both rotation and translation joint types. The results are shown in the right part of Fig. 6. It is hard for Imagic to finish the tasks as articulated objects cannot be opened at all or the wrong part is manipulated. This is because the text instruction is too weak and the fundamental generation model cannot understand the structure of objects. Similarly, DragDiffusion cannot finish the tasks even though human interaction is applied.

MasaCtrl performs better than Imagic and DragDiffusion. The manipulation can be finished, while the edited images are prone to be either unrealistic or unreasonable. Take the laptop as an example (second column in the right part of Fig. 6), the object has been opened, while the region highlighted with the red bounding box in the edited image remains unchanged, which does not make sense. This issue is prevalent across other categories as well. The reason is that MasaCtrl simply shares the whole feature/attention maps between the input and edited image, features of seen parts cannot be transferred to the desired new location when objects move or the shape changes. Finally, Image Sculpting works well on the storage furniture. However it is likely to fail to reconstruct precise 3D models of the laptop, trashcan, and drawer, consequently, the edited images are undesirable.

In contrast, our PA-Diffusion model consistently produces high-fidelity and reasonable edits. The appearance of seen parts is kept accurate no matter whether objects move or the shapes of parts have changed. The novel parts are reasonable and semantically consistent with the original.

Figure 6: TCSL and SCSL are employed to release the blurry and style mismatch problem. The model is required to move and open the object (left). Comparison of Imagic, DragDiffusion, MasaCtrl (with T2I adapter), Image Sculpting, and our PA-Diffusion model. The target state is *'a photo of an opened object'* (right).

4.5 Quantitative Evaluation

To quantitatively evaluate our method, we built an articulated object manipulation testbench. The testbench comprises 6 object categories including storage furniture, laptop, trashcan, microwave, drawer, and refrigerator, which covers both rotation and translation joint types. In total, 660 real images are collected from the website. Considering articulated objects are typically rigid with uniform shapes, this testbench can represent the characteristics of common articulated object categories.

The comparison methods we select are Imagic and MasaCtrl (with T2I Adapter). DragDiffusion is excluded here, as it simply reconstructs the original image and cannot complete the manipulation tasks. Due to the long manual processing time and frequent failures in generating 3D models, Image Sculpting is also excluded here. To assess the realism of the edited images, the evaluation metric used is the Frechet Inception Distance (FID) score. The quantitative evaluation results are summarized in Tab. 1.

Since Imagic relies solely on text instructions, the edited images often do not align well with the original inputs, resulting in poor scores. Due to previously discussed reasons, the edited images of MasaCtrl are confusing and lack coherence. Sequentially, the FID score is not satisfying. In comparison, the PA-Diffusion model outperforms other methods with an obvious improvement.

4.6 Articulated Object Understanding

In this session, we demonstrate how our proposed method supports the task of 3D articulated object understanding. As one of the fundamental steps to understanding 3D articulated objects, estimating the axes and surface normal is still challenging because of the lack of data. To release the data limitation, we create a synthetic dataset with the PA-Diffusion model. The dataset includes 660 sequential samples, each sample includes the sequence of opening objects from the close state with 4 steps. [37] introduced a 3-step training process to develop the object understanding model including

Category	Imagic ↓	MasaCtrl ↓	**Ours** ↓
Storage.	5.93	2.23	0.81
Laptop	9.48	1.17	1.94
Microwave	1.34	3.15	0.87
Trashcan	5.62	1.59	0.82
Refrigerator	7.75	1.76	1.38
Drawer	30.7	0.98	0.60
Avg.	10.1	1.81	**1.07**

Table 1: FID Score of edited images with Imagic, MasaCtrl (with T2I adapter) and ours.

Category	bbox ↑	bbox+axis ↑	normal $< 30°$ ↑
Storage.	67.5/65.0	65.0/65.0	59.9/61.5
Laptop	87.5/90.0	64.2/75.2	22.8/35.7
Microwave	80.0/85.0	80.0/85.0	72.4/70.7
Trashcan	95.0/92.5	74.4/84.9	40.5/49.2
Refrigerator	70.0/80.0	70.0/80.0	63.2/80.0
Drawer	70.0/82.5	70.0/78.8	67.5/82.5
Avg.	78.3/**82.5**	70.6/**78.2**	54.1/**63.2**

Table 2: Prediction accuracy of the model developed with half (left) and full (right) training set separately.

Dataset	AUROC ↑	bbox ↑	bbox+axis(rot) ↑	bbox+axis(rot) normal ↑	bbox ↑	bbox+axis(tran) ↑	bbox+axis(tran) normal ↑
InternetVideo	74.0	62.1	28.5	16.5	32.0	26.2	14.3
Mixed	**74.7**	**66.1**	**29.2**	**18.3**	**38.1**	**30.0**	**19.3**

Table 3: Mix the edited images with the training set of the InternetVideo dataset, then evaluate the fine-tuned model on the testing set of the InternetVideo dataset.

Figure 7: Detection results on the testing samples, including rotation and translation joint types.

BBox detection, axis prediction, and plane normal estimation. Following this schedule, we evaluate the feasibility of edited images by two kinds of experiments.

First, the generated sequential samples are divided into training/testing sets (612/48). Specifically, we follow the 3-step to develop the model with half and full training samples separately, and then evaluate the model with three matrices, BBox IoU, Axes EA-score, and surface normal error smaller than 30° [37]. Fig. 7 demonstrates the prediction results, the model can understand the structures of articulated objects after training with edited images, including moving plane, joint types, axis and surface normal. Quantitative evaluation results in Tab. 2 indicate that the prediction accuracy improves with more training samples, illustrating that the edited images are comparable to the real ones.

Second, to further evaluate the edited images, we merge them with the original training set of Internet Video dataset [37] and fine-tune the pre-trained model. The fine-tuned model is evaluated on the testing set (6,231 real images) of the InternetVideo Dataset. *Baseline* refers to the model trained on the InternetVideo dataset only. Here, to facilitate the comparison, surface normal accuracy is multiplied with the accuracy of BBox and axis, other evaluation metrics are the same as above. The evaluation results are summarized in Tab. 3. Compared with the baseline, the overall performance has been improved by enlarging the training set with edited images. The above two experiments illustrate how our PA-Diffusion model can benefit robotic vision tasks.

5 Limitations

Even though our method can handle common articulated objects, there are still some limitations. First, as edited images are generated from inverted noise maps, the quality of the original input images significantly affects the editing outcomes. Blurry or low-resolution inputs will degrade the edited images. Second, when the object undergoes substantial deformation or movement, this editing method is likely to fail. Besides, manipulating deformable objects and fluids remains challenging with this approach. Further explanation is provided in the Appendix.

6 Conclusion

This work introduces the PA-Diffusion model, a novel articulated object manipulation method that covers common object categories and supports various manipulations. Both the qualitative and quantitative experiments have proven the feasibility and effectiveness of our method. Besides, the 3D articulated object understanding experiment illustrates that the PA-Diffusion model has positive impacts on helping build robots that interact with the real world smartly.

Acknowledgments and Disclosure of Funding

This work is supported by the National Natural Science Foundation of China under Grants 62306175, the National Key R&D Program of China (No.2021ZD0110704), CCF-Tencent Rhino-Bird Open Research Fund, the National Key Research, Development Project of China (No.2022ZD0160102), the National Key Research and Development Project of China (No.2021ZD0110704), Shanghai Artificial Intelligence Laboratory, XPLORER PRIZE grants, STCSM 2023 Pujiang X Program Project (No.23511103104), and STCSM 2024 Qimingxing-Yangfan Fund (No.24YF2722000).

References

[1] Opencv find contour. https://docs.opencv.org/3.4/d4/d73/tutorial_py_contours_begin.html. Accessed: 2010-09-30.

[2] Opencv find corner. https://pyimagesearch.com/2016/04/11/finding-extreme-points-in-contours-with-opencv/. Accessed: 2010-09-30.

[3] Opencv solve pnp. https://docs.opencv.org/4.x/d5/d1f/calib3d_solvePnP.html. Accessed: 2010-09-30.

[4] Tomer Amit, Tal Shaharbany, Eliya Nachmani, and Lior Wolf. Segdiff: Image segmentation with diffusion probabilistic models. *arXiv preprint arXiv:2112.00390*, 2021.

[5] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, et al. ediffi: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022.

[6] Omer Bar-Tal, Dolev Ofri-Amar, Rafail Fridman, Yoni Kasten, and Tali Dekel. Text2live: Text-driven layered image and video editing. In *European conference on computer vision*, pages 707–723. Springer, 2022.

[7] Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. *arXiv preprint arXiv:2304.08465*, 2023.

[8] Minghao Chen, Iro Laina, and Andrea Vedaldi. Training-free layout control with cross-attention guidance. *arXiv preprint arXiv:2304.03373*, 2023.

[9] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.

[10] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[11] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. Cogview: Mastering text-to-image generation via transformers. *Advances in Neural Information Processing Systems*, 34:19822–19835, 2021.

[12] Dave Epstein, Allan Jabri, Ben Poole, Alexei A Efros, and Aleksander Holynski. Diffusion self-guidance for controllable image generation. *arXiv preprint arXiv:2306.00986*, 2023.

[13] Weixi Feng, Xuehai He, Tsu-Jui Fu, Varun Jampani, Arjun Akula, Pradyumna Narayana, Sugato Basu, Xin Eric Wang, and William Yang Wang. Training-free structured diffusion guidance for compositional text-to-image synthesis. *arXiv preprint arXiv:2212.05032*, 2022.

[14] Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-a-scene: Scene-based text-to-image generation with human priors. In *European Conference on Computer Vision*, pages 89–106. Springer, 2022.

[15] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.

[16] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

[17] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[18] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[19] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6007–6017, 2023.

[20] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023.

[21] Yong-Lu Li, Xinpeng Liu, Xiaoqian Wu, Yizhuo Li, Zuoyu Qiu, Liang Xu, Yue Xu, Hao-Shu Fang, and Cewu Lu. Hake: A knowledge engine foundation for human activity understanding. *TPAMI*, 2022.

[22] Yong-Lu Li, Liang Xu, Xinpeng Liu, Xijie Huang, Yue Xu, Shiyi Wang, Hao-Shu Fang, Ze Ma, Mingyang Chen, and Cewu Lu. Pastanet: Toward human activity knowledge engine. In *CVPR*, 2020.

[23] Yong-Lu Li, Yue Xu, Xiaohan Mao, and Cewu Lu. Symmetry and group in attribute-object compositions. In *CVPR*, 2020.

[24] Yong-Lu Li, Yue Xu, Xinyu Xu, Xiaohan Mao, Yuan Yao, Siqi Liu, and Cewu Lu. Beyond object recognition: A new benchmark towards object concept learning. In *ICCV*, 2023.

[25] Yong-Lu Li, Siyuan Zhou, Xijie Huang, Liang Xu, Ze Ma, Hao-Shu Fang, Yanfeng Wang, and Cewu Lu. Transferable interactiveness knowledge for human-object interaction detection. In *CVPR*, 2019.

[26] Chen Liu, Kihwan Kim, Jinwei Gu, Yasutaka Furukawa, and Jan Kautz. Planercnn: 3d plane detection and reconstruction from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4450–4459, 2019.

[27] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9298–9309, 2023.

[28] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.

[29] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021.

[30] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6038–6047, 2023.

[31] Chong Mou, Xintao Wang, Jiechong Song, Ying Shan, and Jian Zhang. Dragondiffusion: Enabling drag-style manipulation on diffusion models. *arXiv preprint arXiv:2307.02421*, 2023.

[32] Chong Mou, Xintao Wang, Liangbin Xie, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. *arXiv preprint arXiv:2302.08453*, 2023.

[33] Haomiao Ni, Changhao Shi, Kai Li, Sharon X Huang, and Martin Renqiang Min. Conditional image-to-video generation with latent flow diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18444–18455, 2023.

[34] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

[35] Xingang Pan, Ayush Tewari, Thomas Leimkühler, Lingjie Liu, Abhimitra Meka, and Christian Theobalt. Drag your gan: Interactive point-based manipulation on the generative image manifold. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–11, 2023.

[36] Shengyi Qian and David F Fouhey. Understanding 3d object interaction from a single image. *arXiv preprint arXiv:2305.09664*, 2023.

[37] Shengyi Qian, Linyi Jin, Chris Rockwell, Siyi Chen, and David F Fouhey. Understanding 3d object articulation in internet videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1599–1609, 2022.

[38] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

[39] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

[40] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[41] Javier Romero, Dimitris Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics*, 36(6), 2017.

[42] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.

[43] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

[44] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[45] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[46] Andrey Voynov, Kfir Aberman, and Daniel Cohen-Or. Sketch-guided text-to-image diffusion models. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–11, 2023.

[47] Wikipedia contributors. 3d projection, 2004. [Online; accessed 22-July-2004].

[48] Wikipedia contributors. Affine transformation, 2004. [Online; accessed 22-July-2004].

[49] Wikipedia contributors. Perspective-n-point, 2004. [Online; accessed 22-July-2004].

[50] Julia Wolleb, Robin Sandkühler, Florentin Bieder, Philippe Valmaggia, and Philippe C Cattin. Diffusion models for implicit image segmentation ensembles. In *International Conference on Medical Imaging with Deep Learning*, pages 1336–1348. PMLR, 2022.

[51] Jiraphon Yenphraphai, Xichen Pan, Sainan Liu, Daniele Panozzo, and Saining Xie. Image sculpting: Precise object editing with 3d geometry control. *arXiv preprint arXiv:2401.01702*, 2024.

[52] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023.

[53] Kai Zhao, Qi Han, Chang-Bin Zhang, Jun Xu, and Ming-Ming Cheng. Deep hough transform for semantic line detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):4793–4806, 2021.

[54] Min Zhao, Fan Bao, Chongxuan Li, and Jun Zhu. Egsde: Unpaired image-to-image translation via energy-guided stochastic differential equations. *Advances in Neural Information Processing Systems*, 35:3609–3623, 2022.

[55] Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3d-vla: 3d vision-language-action generative world model. *arXiv preprint arXiv:2403.09631*, 2024.

Appendix

In this appendix session, we first go through the pipeline of the PA-Diffusion model, and then provide more experiment results and detailed explanations, the arrangement is as follows:

Sec. A: Additional algorithm pipeline of the PA-Diffusion model.

Sec. B: Additional articulated object manipulation results with the PA-Diffusion model.

Sec. C: Additional ablation study and analysis.

Sec. D: Additional explanation of 3D articulated object understanding experiment.

Sec. E: Limitations and future research.

Sec. F: Societal impacts and potential risks.

A Additional algorithm pipeline of the PA-Diffusion model

To facilitate the understanding of our proposed PA-Diffusion model, we present the entire algorithm pipeline in Algorithm 1. $eq.1, eq.2,\ and\ eq.5$ refer to the equations in the main paper.

Algorithm 1 PA-Diffusion Model

Require: Manipulate the articulated objects in RGB images
Input: RGB image x, Primitive Prototype Library
Output: Edited RGB image x_{edit}
Pre-Process:
Generate initial inverted noise map z_T^{Init} with $DDIMInvertion$
Generate initial part-level masks M^{Init} with $GroundedSAM$
Create 3D Abstract model and calibrate 2D image - 3D camera view
Manipulation:
Manipulate articulated objects with text instructions or human interaction in the 3D space
Export manipulated part-level masks M_s^{Gen} of seen part and M_n^{Gen} of novel-appearing part
Feature Process:
Calculated manipulated inverted noise map z_T^{Man} as $eq.5$.
Generation:
Send initial z_T^{Init}, random z_T^{Ran}, and manipulated z_T^{Man} inverted noise map to diffusion model
for $t = T, ..., 1$ **do**
  Construct compositional activation map A_t^{Gen} as $eq.6$
  Add extra loss items TCSL $Loss_t$ and SCSL $Loss_s$ as $eq.7$ **and** $eq.8$
end
Output: Edited image $x_{edit} = Decoder(z_0)$

Dynamic Feature Maps: To make it clear, we simplify the $64 \times 64 \times 4$ inverted noise maps and compositional images to pure color block maps, as shown in Fig. 8. Simple actions such as moving, scaling, and shearing are finished with text instructions, and affine transform [48] is selected. However, for complex actions, perspective transform [47] is required. As the following equations show, we calculate the transform function $Transform$ based on pts_1 and pts_3, where pts_1 and pts_3 are corner points of the input image masks A, B, C, D (pts_1) and corner points of manipulated masks $A^{"}, B^{"}, C^{"}, D^{"}$ (pts_3) as introduced in the main paper. The corner points are automatically detected with a simple corner detection function. And then we can get the transformed inverted noise map z_T^{Tran} by transforming the initial inverted noise map z_T^{Init} as following equations. Finally, the manipulated inverted noise map is calculated by adding z_T^{Tran} and z_T^{Init}.

$$\begin{aligned} T &= Transform(pts_1, pts_3), \\ z_T^{Tran} &= T(z_T^{Init}), \end{aligned} \tag{10}$$

On the other hand, the edited images are generated by adding I^{Ran} generated with the random noise map and I^{Gen} generated with the manipulated inverted noise map, as shown in the right part of Fig. 8.

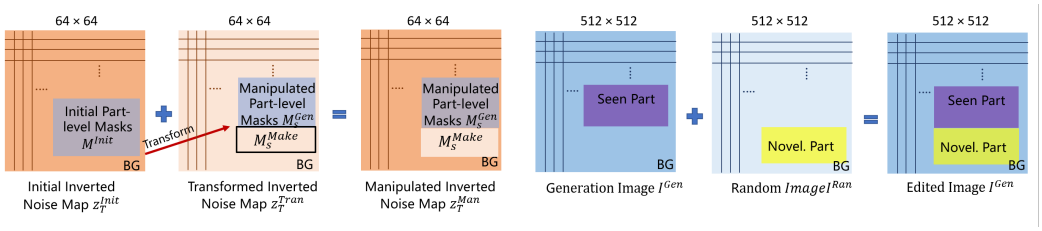

Figure 8: Manipulating noise maps and constructing compositional images. The manipulation is implemented with affine or perspective projection. The final edited image is constructed by merging two activation maps and images.

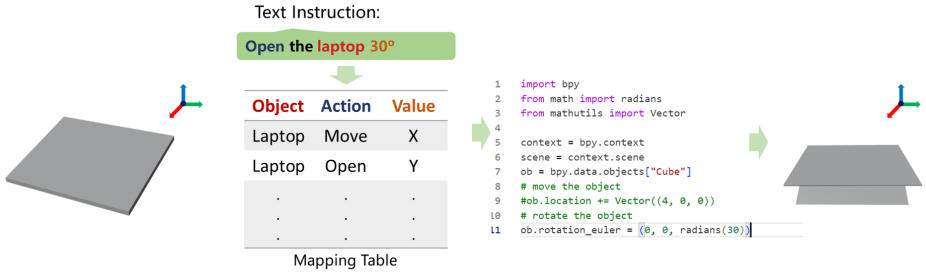

Figure 9: Example of manipulating 3D objects in Blender with text instructions.

Articulated Object Manipulation in 3D space: As noted in the main paper, different manipulations can be done with text instructions or human interaction. Manipulating the articulated objects in the 3D space is straightforward in this work.

For the text-based method, considering the objects can be manipulated with Python scripts in Blender, we first construct a table mapping the text instructions to actions. Then these actions can be implemented by running Python scripts. As shown in Fig. 9, we require the laptop to open 30 degree, this text instruction is converted to Python script. Finally, running this script can finish the open action. For the second type, users can directly manipulate any parts of articulated objects in Blender, which is more flexible and convenient.

B Additional Manipulation Results

In Fig. 10 and Fig. 11, more articulated object manipulation results synthesized by our proposed PA-Diffusion model are demonstrated. Novel categories including door, toilet, and book are experimented here. For various categories, joint types, and backgrounds, our proposed method can manipulate the objects and preserve other contents in the input images simultaneously.

C Additional Ablation Study

To analyze and explain our proposed PA-Diffusion model in detail, we provide more ablation studies in this session.

Additional Loss Items: In the main paper, we qualitatively demonstrate the effect of TCSL and SCSL in the experiment part. Here, we evaluate them quantitatively. Following the main paper, the evaluation metric is the FID score. The results are summarized in Tab. 4. It is obvious that without TCSL, the performance degrades significantly since the feature cannot be transferred to the edited images correctly leading to inconsistent appearance compared to the input images. On the other hand, SCSL only aims to adjust the style of novel-appearing parts, which does not determine the score. When including the two losses, the edited images are close to the input real images in the aspects of color, texture, and style.

Primitive Prototype Library: In the main paper, we create 5 primitive prototypes to represent 6 different kinds of articulated objects in the testbench. Since our method does not require precise 3D CAD models of objects, Primitive Prototype Library can cover a wide range of articulated objects with a small number of primitive prototypes. As shown in Fig. 10 and Fig. 11, books can be

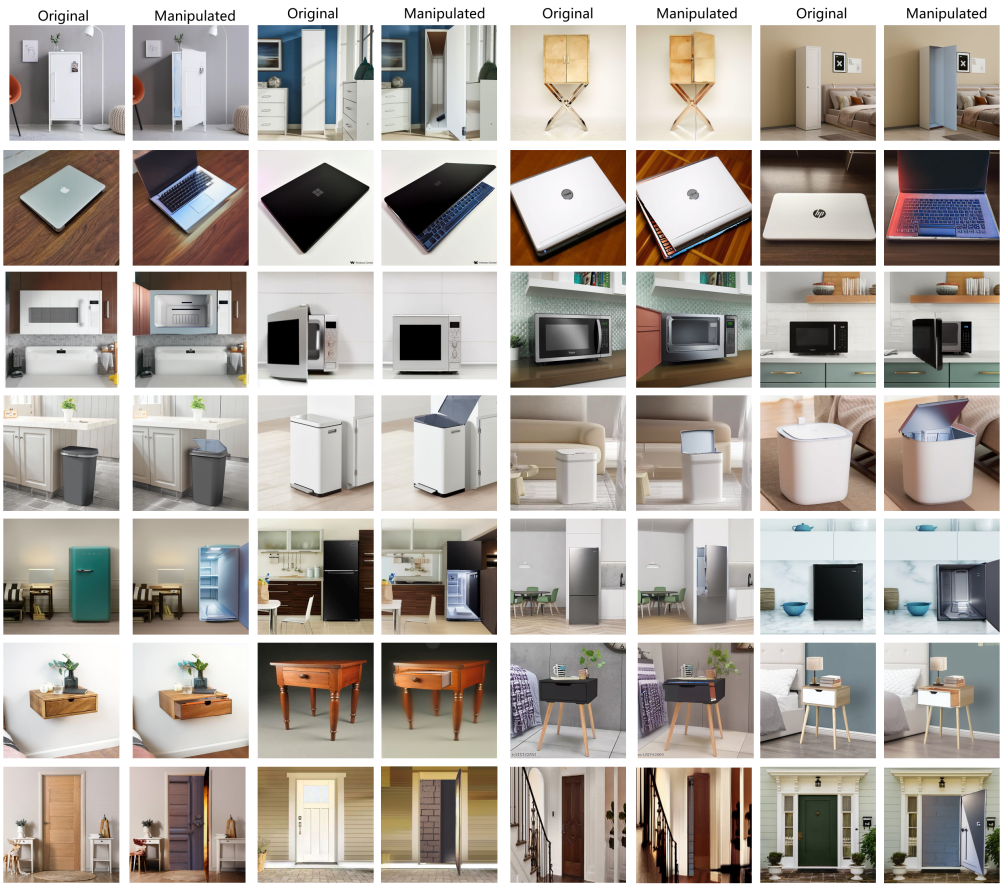

Figure 10: Additional demonstration of the images edited with our proposed PA-Diffusion model, including storage furniture, laptop, microwave, trashcan, door, drawer, and refrigerator.

Category	None ↓	only SCSL ↓	only TCSL ↓	All Losses ↓
Storage.	1.48	1.54	1.00	0.81
Laptop	2.30	1.92	1.88	1.94
Microwave	1.52	1.71	0.71	0.87
Trashcan	1.39	1.65	0.70	0.82
Refrigerator	1.99	1.81	1.79	1.38
Drawer	1.63	1.25	0.51	0.60
Avg.	1.72	1.65	1.10	**1.07**

Table 4: FID score of edited images with different additional losses: no SCSL and TCSL, with SCSL only, with TCSL only, and with all losses. The performance improves 37.8% with the assistance of the two losses.

represented as laptops, doors can be represented with simple planes, and toilets can be represented with a plane and boxes. No extra prototypes are required when creating abstract 3D models for novel categories. Besides, the edited images are still high-fidelity and high-quality. Furthermore, the Primitive Prototype Library is easy to expand, more primitive prototypes can be created rapidly when dealing with novel articulated objects.

2D-3D Models analysis: Besides the advantage of convenience, we also compare the quality of reconstructed 3D models with state-of-the-art 2D-3D methods and abstract 3D models.

Following the method introduced in Image Sculpting [51], we use ClipDrop to remove the background of input images, and then reconstruct 3D models with Zero123 [27]. The tested images are the same as those in the main paper. The reconstructed 3D object mesh models are shown in Fig. 12 including the front view, side view, and the edited images after manipulation. We notice that the result of

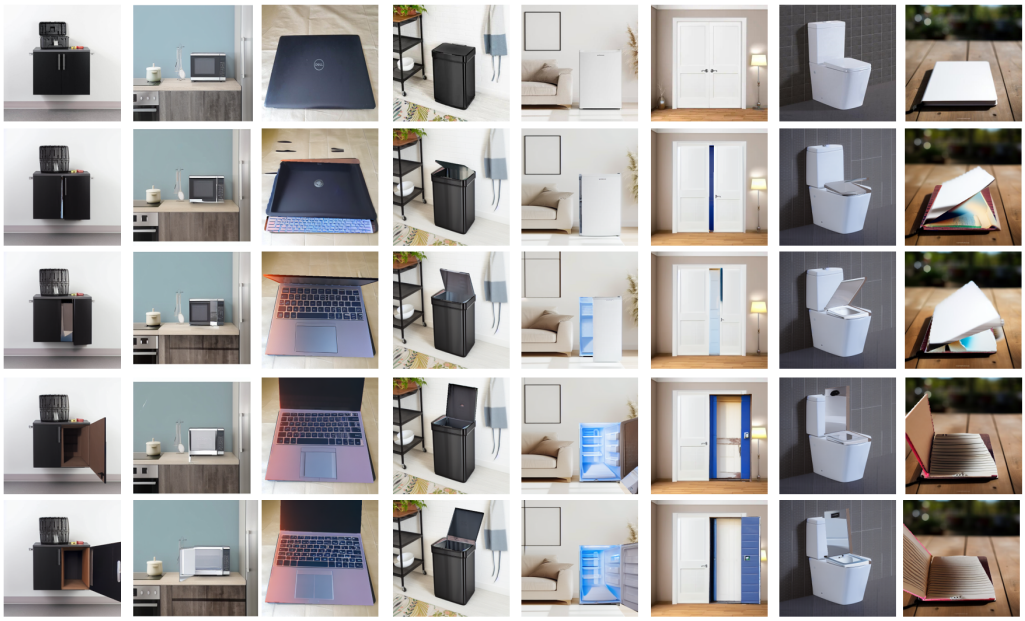

Figure 11: Additional demonstration of editing images with manipulation process based on our proposed PA-Diffusion model. The last three columns are novel articulated object categories.

storage furniture is acceptable, the shape closely matches the original, and the texture is stored in UV maps correctly. However, for other categories, the reconstructed models are poor which is the main reason for low-quality edited images.

In the main paper, we mentioned that manipulating the reconstructed 3D models created by 2D-3D methods is tedious and inaccurate. As shown in Fig. 12, the reconstructed models are not part-level (only one mesh object), tremendous human effort is required to cut the mesh into parts before manipulation. For example, when trying to split the door from the body of storage furniture, we need to cut the furniture into several parts first (top, bottom, four sides of the body), and then merge others except the door. This process, even with advanced 3D graphics software, is complex and labor-intensive. As a result, summarizing reconstruction, manipulation, and generation time, Image Sculpting [51] requires over 10 mins to manipulate one image, making it unsuitable for large-scale image editing tasks.

In summary, using abstract 3D models to present articulated objects offers several advantages: (1) State-of-the-art 2D-3D methods are still not robust enough to create precise 3D models such as laptops and trashcans. In comparison, it is easy to achieve abstract 3D models with primitive prototypes, as shown in Fig. 12. (2) Manipulating primitive prototypes in the 3D space is easier and more accurate than manipulating a single 3D object mesh. (3) Novel categories and instances can be handled with our method efficiently. (4) Our method is brief and thus can support various downstream tasks.

D 3D Articulated Objects Understanding

Overview. In the main paper, we introduce the 3D articulated object understanding experiment and demonstrate how our proposed PA-Diffusion model supports other research fields. Here, we discuss the experiment setup and implementation in more detail.

Annotation Generation. In previous works [37, 36, 26], human labeling is required for annotating the bounding boxes, rotation/translation axes, and surface normal, which is labor-intensive and inaccurate. On the contrary, by representing objects with abstract 3D models, our method can achieve these annotations automatically.

As shown in Fig. 13, $part1$ and $part2$ are the masks of articulated objects' parts that are exported from Grounded SAM or Blender software. The corner points can be calculated with [1, 2] automatically. Then, the bounding box of each object part can be calculated with these corner points. The rotation and

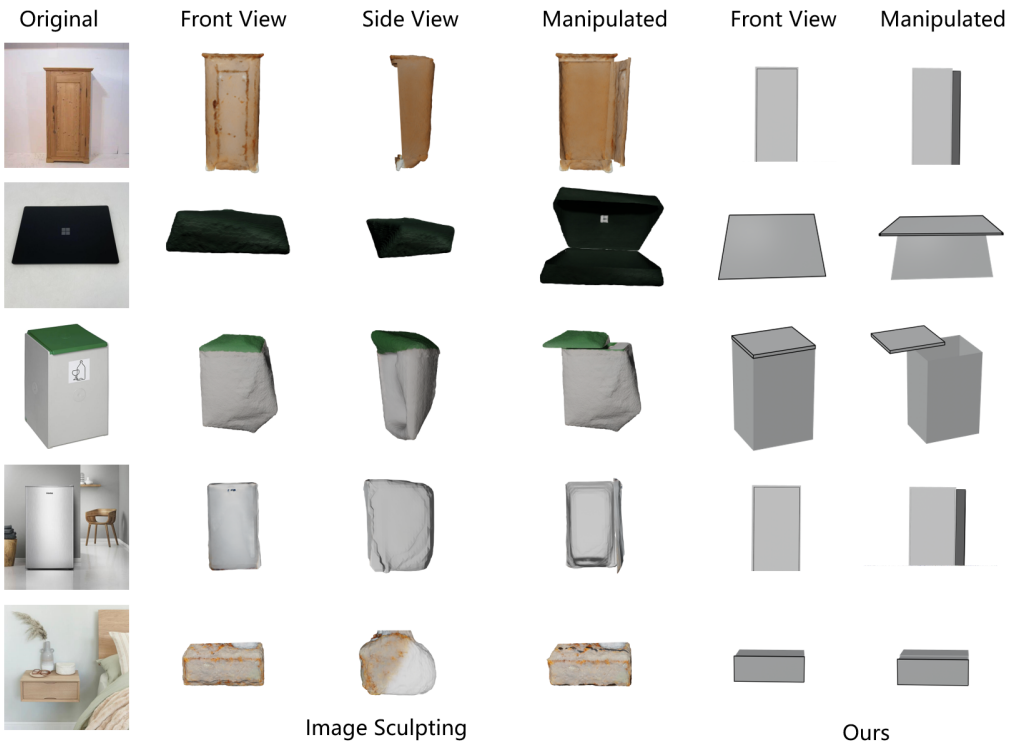

Figure 12: Reconstructed 3D object models with Image Sculpting, and our abstract 3D models created with Primitive Prototype Library.

translation axis annotations are represented as $[x_0, y_0, x_1, y_1]$, where x_0, y_0, x_1, y_1 are the coordinates of corner points in part-level masks, different object category uses different corner points. For the 3D surface normal annotations, as shown in Fig. 14, we first export the object's part world transform matrix M_{obj} and camera world transform matrix M_{camera}. Then the two matrices are normalized and calibrated to create the aligned transform matrix $M_{aligned}$. Consequently, we simply select the normal of one plane to represent the orientation of the object part, and the surface normal V_{surf} is equal to the multiplication of the aligned matrix and local plane vector V_{plane}:

$$\begin{aligned} M_{aligned} &= Align(Norm(M_{camera}), Norm(M_{obj})), \\ V_{surf} &= M_{aligned} \times V_{plane}. \end{aligned} \tag{11}$$

Evaluation Matrix. For quantitative evaluation, we follow [37] to calculate the average precision of the bounding box, axis, and surface normal. The bounding box is the traditional horizontal type, the threshold of IoU is set as 0.5. The predicted axes are measured with EA score as [53]. To demonstrate the results clearly, we calculate the surface normal error and measure the accuracy that the error is smaller than the threshold $30°$.

E Limitations and Future Research

The limitations of our proposed PA-Diffusion models have been discussed in the main paper. Due to the inaccuracy of DDIM inversion, the inverted noise map might be poor if the input image quality is low. Unfortunately, the poor noise map will lead to mismatch error accumulation and propagation during the iterative denoising process. As in the left of Fig. 15, when the original input image is of low resolution (it is normalized to 512×512 before manipulation), the PA-Diffusion model cannot re-generate the original image with the inverted noise map. Simple actions like moving and scaling also cannot be completed.

Manipulating the initial inverted noise maps is critical to preserve the appearance of seen parts. However, as discussed in the main paper, this step disturbs the original data distribution. The problem

will be too serious to be fixed when the object shape deformation is large. As shown in the right of Fig. 15, when reshaping the laptop to a slim non-uniform diamond shape, the object's appearance cannot be preserved.

Considering this situation, one promising solution is to add stronger and more precise supervision loss in each denoising step. This is beyond the scope of this work, we plan to implement this later.

In the future, more categories of articulated objects will be covered and the edited image dataset will be expanded to millions-scale for supporting various computer vision and robotic manipulation tasks. Next, we will extend this method to handle deformable objects and fluids.

F Societal impacts and potential risks

The articulated object manipulation method presented in this work has profound positive societal implications. This method can serve as a fundamental tool to benefit other computer vision or robot vision tasks. Consequently, the artificial intelligent algorithm can understand and interact with the real world better. Humans will have stronger AI assistants including smart offices, intelligent home or medical robots, and so on.

All the models and data used in this work are collected from the Website. No personal information is used. The code and data created in this work have a low risk of misuse.

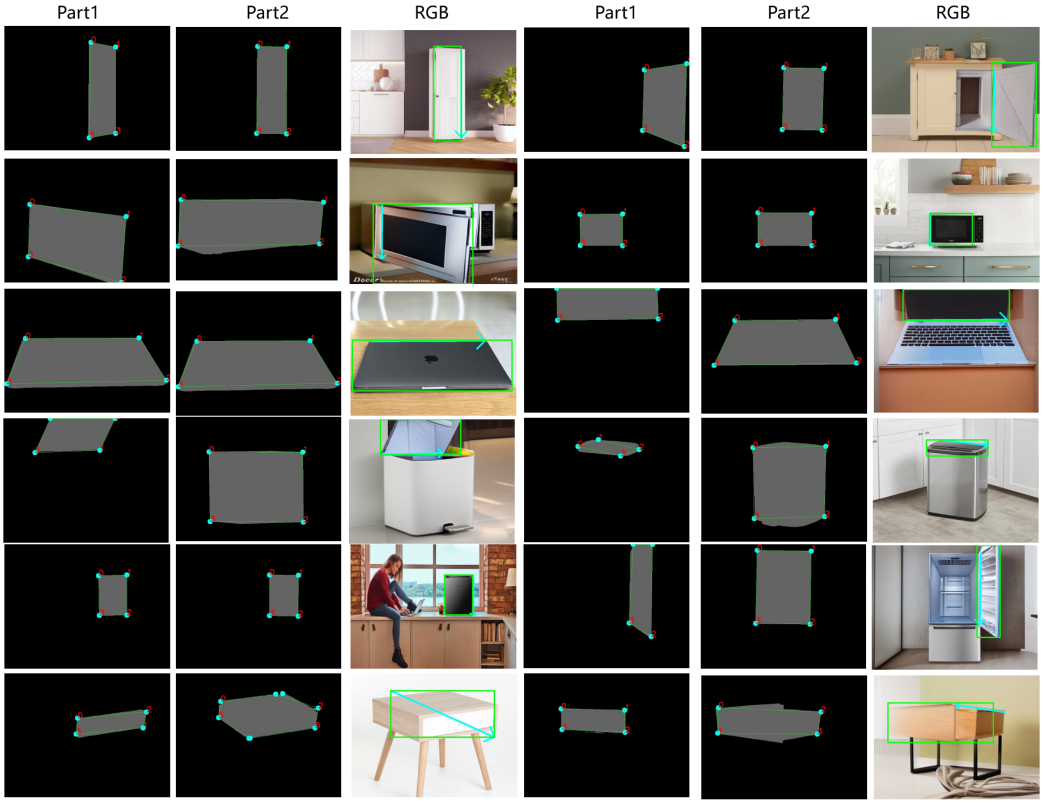

Figure 13: Demonstration of extracting bounding box annotations and rotation/translation axis annotations from part-level masks.

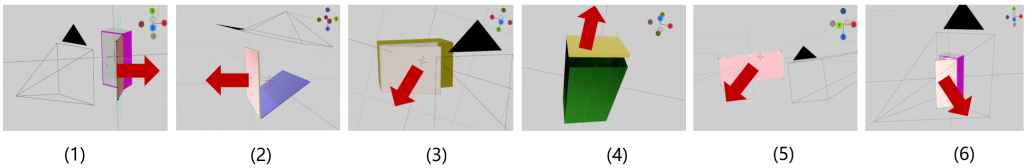

Figure 14: The camera poses and orientations of 6 planes in Blender, (1)-(6) refers to abstract 3D models of cabinet, laptop, microwave, trashcan, drawer, and refrigerator separately. Red arrows are the surface normal directions.

Figure 15: Limitation of the PA-Diffusion model: dealing with low-quality images or large deformation.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The contribution: PA-Diffusion model, a novel articulated object manipulation method in real images has been claimed in the abstract.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitation is discussed in both the main paper and the Appendix.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when the image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The paper introduces a novel method, there is no theoretical result.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in the appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The completed algorithm pipeline is provided in the Appendix. Each step is also introduced in both the main paper and the Appendix. The context explanation and figures are aligned to explain the proposed method. Besides, to make the algorithm easy to understand, we provide a simpler explanation in the Appendix as well. All the experimental results can be reproduced.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code will be published.

   Guidelines:

   - The answer NA means that the paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All the training and test details are described in the session of the experiment in the main paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in the appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The paper reports the experiment results on certain datasets.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar and then state that they have a 96% CI if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of computing workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The required computing resources are claimed in the session of the experiment in the main paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute worker CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more computing than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g. if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: It is discussed in the Appendix.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for the responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: It is described in the Appendix. The code and data of this work have a quite low risk of misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make the best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited, and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All the models and data used in this work are published, and all the citations have been included in the paper.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: No new assets are introduced in the paper.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: There is no crowdsourcing experiments and research in this work.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing or research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This work has no crowdsourcing experiments and research.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing or research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.