FastDrag: Manipulate Anything in One Step

Xuanjia Zhao[1], **Jian Guan**[1,*], **Congyi Fan**[1], **Dongli Xu**[4],
Youtian Lin[2], **Haiwei Pan**[1], **Pengming Feng**[3]

[1]College of Computer Science and Technology, Harbin Engineering University
[2]School of Intelligence Science and Technology, Nanjing University
[3]State Key Laboratory of Space-Ground Integrated Information Technology
[4]Independent Researcher

Abstract

Drag-based image editing using generative models provides precise control over image contents, enabling users to manipulate anything in an image with a few clicks. However, prevailing methods typically adopt n-step iterations for latent semantic optimization to achieve drag-based image editing, which is time-consuming and limits practical applications. In this paper, we introduce a novel one-step drag-based image editing method, *i.e.*, FastDrag, to accelerate the editing process. Central to our approach is a latent warpage function (LWF), which simulates the behavior of a stretched material to adjust the location of individual pixels within the latent space. This innovation achieves one-step latent semantic optimization and hence significantly promotes editing speeds. Meanwhile, null regions emerging after applying LWF are addressed by our proposed bilateral nearest neighbor interpolation (BNNI) strategy. This strategy interpolates these regions using similar features from neighboring areas, thus enhancing semantic integrity. Additionally, a consistency-preserving strategy is introduced to maintain the consistency between the edited and original images by adopting semantic information from the original image, saved as key and value pairs in self-attention module during diffusion inversion, to guide the diffusion sampling. Our FastDrag is validated on the DragBench dataset, demonstrating substantial improvements in processing time over existing methods, while achieving enhanced editing performance. Project page: `https://fastdrag-site.github.io/`.

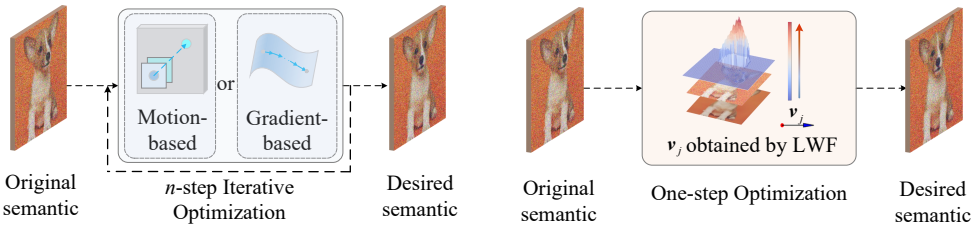
1 Introduction

The drag editing paradigm [28, 15, 20] leverages the unique properties of generative models to implement a point-interaction mode of image editing, referred to as drag-based image editing. Compared with text-based image editing methods [18, 10, 3, 30], drag-based editing enables more precise spatial control over specific regions of the image while maintaining semantic logic coherence, drawing considerable attention from researchers.

However, existing methods typically involve n-step iterative semantic optimization in latent space to obtain optimized latent with desired semantic based on the user-provided drag instructions. They focus on optimizing a small region of the image at each step, requiring n small-scale and short-distance adjustments to achieve overall latent optimization, leading to a significant amount of time. These optimization approaches can be categorized primarily into motion-based [23, 28, 15, 16, 32, 4] and gradient-based [20, 19] n-step iterative optimizations, as shown in Fig. 1(a). n-step iterations in motion-based methods are necessary to avoid abrupt changes in the latent space, preventing image

*Corresponding author: j.guan@hrbeu.edu.cn

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



(a) Existing methods with n-step iterative optimization. (b) FastDrag with one-step warpage optimization.

Figure 1: (a) Existing methods usually require multiple iterations to transform an image from its original semantic to desired semantic; (b) Our method utilizes latent warpage function (LWF) to calculate the warpage vectors (*i.e.*, $\boldsymbol{v}_j$) to move each individual pixel on feature map and achieve semantic optimization in one step.

distortions and ensuring a stable optimization process. This is exemplified in DragDiffusion [28] and GoodDrag [32], which require 70 to 80 iterations of point tracking and motion supervision for optimization. In addition, gradient-based methods align sampling results with the drag instructions through gradient guidance [6]. In this way, they also require multiple steps due to the optimizer [13, 25] needing multiple iterations for non-convex optimization. For instance, DragonDiffusion [20] requires around 50 gradient steps to accomplish the latent optimization. Therefore, existing drag-based image editing methods often suffer from significant time consumption due to n-step iterations required for latent semantic optimization, thus limiting the practical applications.

To this end, we present a novel one-step drag-based image editing method based on diffusion, *i.e.*, FastDrag, which significantly accelerates editing speeds while maintaining the quality and precision of drag operations. Specifically, a novel one-step warpage optimization strategy is proposed to accelerate editing speeds, which can achieve the latent semantic optimization in a single step with an elaborately designed latent warpage function (LWF), instead of using motion or gradient-based n-step optimizations, as illustrated in Fig. 1(b). By simulating strain patterns in stretched materials, we treat drag instructions on the noisy latent as external forces stretching a material, and introduce a stretch factor in LWF, which enables the LWF to generate warpage vectors to adjust the position of individual pixels on the noisy latent with a simple latent relocation operation, thus achieving one-step optimization for drag-based editing. Meanwhile, a bilateral nearest neighbor interpolation (BNNI) strategy is proposed to enhance the semantic integrity of the edited content, by interpolating null values using similar features from their neighboring areas to address semantic losses caused by null regions emerging after latent relocation operation, thus enhancing the quality of the drag editing.

Additionally, a consistency-preserving strategy is introduced to maintain the consistency of the edited image, which adopts the original image information saved in diffusion inversion (*i.e.*, key and value pairs of self-attention in the U-Net structure of diffusion model) to guide the diffusion sampling for desired image reconstruction, thus achieving precise editing effect. To further reduce time consumption for inversion and sampling, the latent consistency model (LCM) [17] is employed in the U-Net architecture of our diffusion-based FastDrag. Therefore, our FastDrag can significantly accelerate editing speeds while ensuring the quality of drag effects.

Experiments on DragBench demonstrate that the proposed FastDrag is the fastest drag-based editing method, which is nearly 700% faster than the fastest existing method (*i.e.*, DiffEditor [19]), and 2800% faster than the typical baseline method (*i.e.*, DragDiffusion [28]), with comparable editing performance. We also conduct rigorous ablation studies to validate the strategies used in FastDrag.

Contributions: 1) We propose a novel drag-based image editing approach based on diffusion *i.e.*, FastDrag, where a LWF strategy is proposed to achieve one-step semantic optimization, tremendously enhancing the editing efficiency. 2) We propose a novel interpolation method (*i.e.*, BNNI), which effectively addresses the issue of null regions, thereby enhancing the semantic integrity of the edited content. 3) We introduce a consistency-preserving strategy to maintain the image consistency during editing process.

2 Related Work

2.1 Text-based Image Editing

Text-based image editing has seen significant advancements, allowing users to manipulate images through natural language instructions. DiffusionCLIP [12] adopts contrastive language-image pre-

training (CLIP) [24] for diffusion process fine-tuning to enhance the diffusion model, enabling high-quality zero-shot image editing. The study in [7] manipulates the cross-attention maps within the diffusion process and achieves text-based image editing. Imagic [11] further enhances these methods by optimizing text embeddings and using text-driven fine-tuning of the diffusion model, enabling complex semantic editing of images. InstructPix2Pix [1] leverages a pre-trained large language model combined with a text-to-image model to generate training data for a conditional diffusion model, allowing it to edit images directly based on textual instructions during forward propagation. Moreover, Null-text Inversion [18] enhances text-based image editing by optimizing the default null-text embeddings to achieve desired image editing. Although text-based image editing methods enable the manipulation of image content using natural language description, they often lack the precision and explicit control provided by drag-based image editing.

2.2 Drag-based Image Editing

Drag-based image editing achieves precise spatial control over specific regions of the image based on user-provided drag instructions. Existing drag-based image editing methods generally rely on n-step latent semantic optimization in latent space to achieve image editing. These methods fall into two main categories: motion-based [23, 28, 32, 4, 16, 15, 9] and gradient-based [20, 19] optimizations. For example, DragGAN [23] employs generative adversarial network (GAN) for drag-based image editing with iterative point tracking and motion supervision steps. However, the image quality of the methods using GAN for image generation is worse than diffusion models [5]. Therefore, a series of diffusion-based methods have been proposed for drag-based image editing. For instance, DragDiffusion [28] employs iterative point tracking and motion supervision for latent semantic optimization to achieve drag-based editing. Building on this foundation, GoodDrag [32], StableDrag [4], DragNoise [16], and FreeDrag [15] have made significant improvements to the motion-based methods. Without coincidence, by utilizing feature correspondences, DragonDiffusion [20] and its improved version DiffEditor [19] formulate an energy function that conforms to the desired editing results, thereby transforming the image editing task into a gradient-based process that enables drag-based editing. However, these methods inherently require n-step iterations for latent optimization, which significantly increases the time consumption. Although SDEDrag [22] does not require n-step iterative optimization, it is still time-consuming due to the stochastic differential equation (SDE) process for diffusion. In addition, while EasyDrag [9] offers user-friendship editing, its requirement for over 24GB of memory (*i.e.*, a 3090 GPU) limits its broad applicability. To this end, based on latent diffusion model (LDM) [26], we propose a novel one-step optimization method that substantially accelerates the image editing speeds.

3 Proposed Method

FastDrag is based on LDM [26] to achieve drag-based image editing across four phases. The overall framework is given in Fig. 2, and the detailed description of strategies in FastDrag are presented as follows: (1) Initially, FastDrag is based on a traditional image editing framework including diffusion inversion and sampling processes, which will be elaborated in Sec. 3.1. (2) The core phase in Sec. 3.2 is a one-step warpage optimization, employing LWF and a latent relocation operation to simulate the behavior of stretched material, allowing for fast semantic optimization. (3) BNNI is then applied in Sec. 3.3 to enhance the semantic integrity of the edited content, by interpolating the null regions emerging after the one-step warpage optimization. (4) The consistency-preserving strategy is introduced in Sec. 3.4 to maintain the desired image consistency with original image, by utilizing the key and value of self-attention in inversion to guide the sampling.

3.1 Diffusion-based Image Editing

Similar to most existing drag editing methods [28, 32, 20], FastDrag is also built upon diffusion model (*i.e.*, LDM), including diffusion inversion and diffusion sampling.

Diffusion Inversion [29] is about mapping a given image to its corresponding noisy latent representation in the model's latent space. We perform semantic optimization on the noisy latent $\boldsymbol{z}_t \in \mathbb{R}^{w \times h \times c}$, due to it still captures the main semantic features of the image but is perturbed by noise, making it suitable as a starting point for controlled modifications and sampling [28]. Here, w, h, c represent the width, height and channel of $\boldsymbol{z}_t$, respectively. This process for a latent variable at diffusion step t can

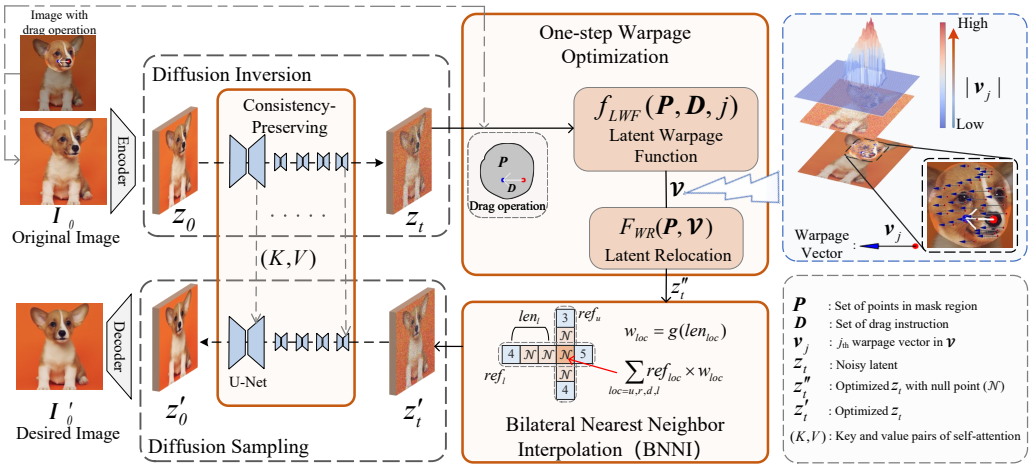


Figure 2: Overall framework of FastDrag with four phases: diffusion inversion, diffusion sampling, one-step warpage optimization and BNNI. Diffusion inversion yields a noisy latent $\boldsymbol{z}_t$ and diffusion sampling reconstructs the image from the optimized noisy latent $\boldsymbol{z}'_t$. One-step warpage optimization is used for noisy latent optimization, where LWF is proposed to generate warpage vectors to adjust the location of individual pixels on the noisy latent with a simple latent relocation operation. BNNI is used to enhance the semantic integrity of noisy latent. A consistency-preserving strategy is introduced to maintain the consistency between original image and edited image.

be expressed as:

$$\boldsymbol{z}_t = \frac{\sqrt{\alpha_t}}{\sqrt{\alpha_{t-1}}}(\boldsymbol{z}_{t-1} - \sqrt{1-\alpha_{t-1}} \cdot \boldsymbol{\epsilon}_t) + \sqrt{1-\alpha_t} \cdot \boldsymbol{\epsilon}_t, \tag{1}$$

where $z_0 = \mathcal{E}(\boldsymbol{I}_0)$ denotes the initial latent of the original image $\boldsymbol{I}_0$ from the encoder [14] $\mathcal{E}(\cdot)$. α_t is the noise variance at diffusion step t, and $\boldsymbol{\epsilon}_t$ is the noise predicted by U-Net. Subsequently, we perform a one-step warpage optimization on $\boldsymbol{z}_t$ in Sec. 3.2.

Diffusion Sampling reconstructs the image from the optimized noisy latent $\boldsymbol{z}'_t$ by progressively denoising it to the desired latent $\boldsymbol{z}'_0$. This sampling process can be formulated as:

$$\boldsymbol{z}'_{t-1} = \sqrt{\alpha_{t-1}} \cdot \left(\frac{\boldsymbol{z}'_t - \sqrt{1-\alpha_t} \cdot \boldsymbol{\epsilon}_t}{\sqrt{\alpha_t}} \right) + \sqrt{1-\alpha_{t-1}-\sigma^2} \cdot \boldsymbol{\epsilon}_t + \sigma^2 \cdot \boldsymbol{\epsilon}, \tag{2}$$

where $\boldsymbol{\epsilon}$ is the Gaussian noise and σ denotes the noise level. By iterating the process from t to 1, $\boldsymbol{z}'_0$ is reconstructed, and the desired image can be obtained by $\boldsymbol{I}'_0 = \mathcal{D}(\boldsymbol{z}'_0)$, with $\mathcal{D}(\cdot)$ being the decoder [14].

3.2 One-step Warpage Optimization

Building upon the phases in Sec.3.1, we propose a one-step warpage optimization for fast drag-based image editing. The core idea involves simulating strain patterns in stretched materials, where drag instructions on the noisy latent are interpreted as external forces stretching the material. This enables us to adjust the position of individual pixels on the noisy latent, optimizing the semantic of noisy latent in one step, thus achieving extremely fast drag-based editing speeds. To this end, we design the LWF in Sec. 3.2.1 to obtain warpage vector, which is utilized by a straightforward latent relocation operation in Sec. 3.2.2 to adjust the position of individual pixels on the noisy latent.

3.2.1 Warpage Vector Calculation using LWF

In drag-based image editing, each drag instruction $\boldsymbol{d}_i$ in a set of k drag instructions $\boldsymbol{D} = \{\boldsymbol{d}_i \mid i = 1, \ldots, k; k \in \mathbb{Z}\}$ can simultaneously influence a feature point p_j on the mask region $\boldsymbol{P} = \{p_j \mid j = 1, \ldots, m; m \in \mathbb{Z}\}$ provided by the user. As shown in Fig. 3, the mask region is represented by the brighter areas in the image, indicating the specific image area to be edited. To get a uniquely determined vector, *i.e.*, warpage vector $\boldsymbol{v_j}$ to adjust the position of feature point p_j (will be discussed in Sec.3.2.2), we propose a latent warpage function $f_{LWF}(\cdot)$ to aggregate multiple component warpage vectors caused by different drag instructions, *i.e.*, $\boldsymbol{v}_j^{i*}$, with balanced weights to avoid

deviating from the desired drag effect. The function is given as follows:

$$\boldsymbol{v}_j = f_{LWF}(\boldsymbol{P}, \boldsymbol{D}, j) = \sum_i^k w_j^i \cdot \boldsymbol{v}_j^{i*}, \tag{3}$$

where w_j^i is the normalization weight for component warpage vector $\boldsymbol{v}_j^{i*}$. Here, drag instruction $\boldsymbol{d}_i$ is considered as a vector form handle point s_i to target point e_i. During dragging, we aim for the semantic changes around the handle point s_i to be determined by the corresponding drag instruction $\boldsymbol{d}_i$, rather than other drag instructions far from the s_i. Therefore, w_j^i is calculated as follows:

$$w_j^i = \frac{1/|p_j s_i|}{\sum_i^k (1/|p_j s_i|)}, \tag{4}$$

where s_i is considered as the "point of force" of $\boldsymbol{d}_i$, and the weight w_j^i is inversely proportional to the Euclidean distance from s_i to p_j.

It is worth noting that under an external force, the magnitude of component forces at each position within the material is inversely proportional to the distance from the force point, while the movement direction at each position typically aligns with the direction of the applied force [21]. Similarly, the component warpage vector $\boldsymbol{v}_j^{i*}$ on each p_j aligns with the direction of drag instruction $\boldsymbol{d}_i$, and magnitudes of $\boldsymbol{v}_j^{i*}$ are inversely proportional to the distance from s_i. Hence, $\boldsymbol{v}_j^{i*}$ can be simplified as:

$$\boldsymbol{v}_j^{i*} = \lambda_j^i \cdot \boldsymbol{d}_i, \tag{5}$$

where λ_j^i is the stretch factor that denotes the proportion between $\boldsymbol{v}_j^{i*}$ and $\boldsymbol{d}_i$.

To appropriately obtain the stretch factor λ_j^i and facilitate the calculation, we delve into the geometric representation of the component warpage vector $\boldsymbol{v}_j^{i*}$. As shown in Fig. 3, $\boldsymbol{v}_j^{i*}$ can be depicted as the guidance vector from point p_j to point p_j^{i*}, where p_j^{i*} is the expected new position of p_j under the drag effect of $\boldsymbol{d}_i$. Recognizing that the content near to mask edge should remain unaltered, we define a reference circle O where every $\boldsymbol{v}_j^{i*}$ will gradually reduce to 0 as p_j approaches the circle. Consequently, since $\boldsymbol{v}_j^{i*}$ and $\boldsymbol{d}_i$ are parallel, magnitudes of $\boldsymbol{v}_j^{i*}$ are inversely proportional to the distance from s_i and $\boldsymbol{v}_j^{i*}$ is reduced to 0 on circle O, the extended lines from $s_i p_j$ and $e_i p_j^{i*}$ will intersect at q_j^i on circle O. Hence, based on the Eq. (5) and the geometric principle in Fig. 3, we calculate λ_j^i as follows:

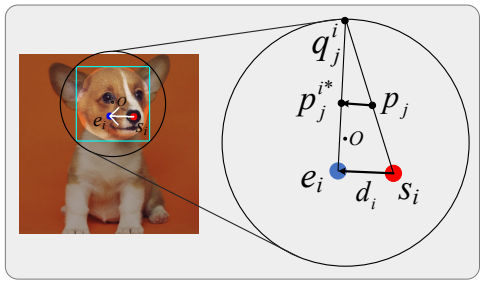


Figure 3: Geometric representation of $\boldsymbol{v}_j^{i*}$. Circle O is the circumscribed circle of the circumscribed rectangle enclosing the mask's shape. p_j is the feature point requiring relocation, and p_j^{i*} is its new position following the drag instruction $\boldsymbol{d}_i$

$$\lambda_j^i = \frac{|\boldsymbol{v}_j^{i*}|}{|\boldsymbol{d}_i|} = \frac{|\overrightarrow{p_j p_j^{i*}}|}{|\overrightarrow{s_i e_i}|} = \frac{|p_j q_j^i|}{|s_i q_j^i|}. \tag{6}$$

Finally, we obtain the warpage vector $\boldsymbol{v}_j$ using only $\boldsymbol{d}_i$ and two factors as follows:

$$f_{LWF}(\boldsymbol{P}, \boldsymbol{D}, j) = \sum_i^k w_j^i \cdot \lambda_j^i \cdot \boldsymbol{d}_i \tag{7}$$

Note that, for the special application of drag-based editing, such as object moving as shown in Fig. 8, drag editing is degenerated to a mask region shifting operation, requiring the spatial semantics of the mask region to remain unchanged. In that case, we only process a single drag instruction, and all component drag effects will be set equal to the warpage vector, *i.e.*, $\boldsymbol{v}_j = \boldsymbol{d}_1$ and $\boldsymbol{D} = \{\boldsymbol{d}_1\}$.

3.2.2 Latent Relocation with Warpage Vector

Consequently, we utilize the warpage vector v_j to adjust the position of feature point p_j via a latent relocation operation F_{WR}, achieving the semantic optimization of noisy latent for drag-based editing.

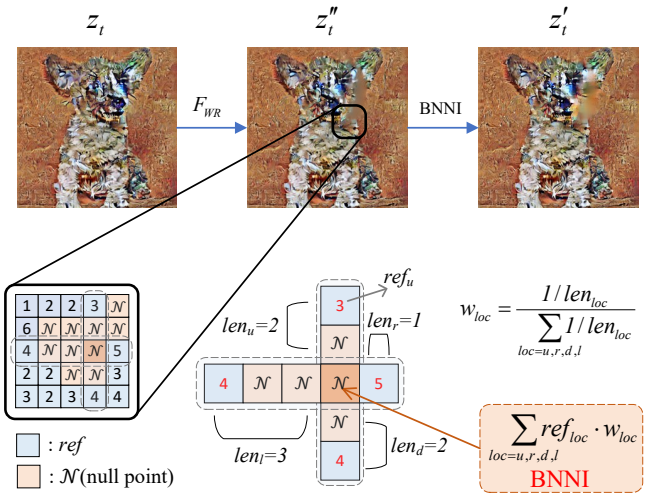


Figure 4: Illustration of bilateral nearest neighbor interpolation.

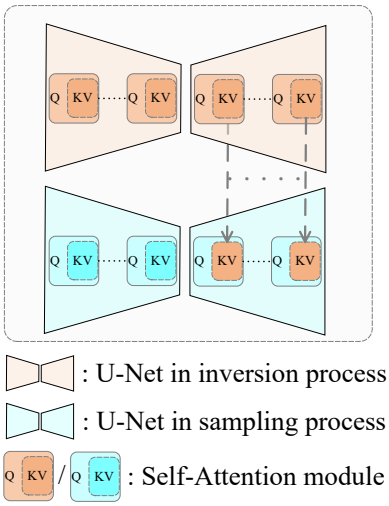


Figure 5: Illustration of consistency-preserving strategy.

Establishing a Cartesian coordinate system on the latent space, let (x_{p_j}, y_{p_j}) denote the position of point $p_j \in \boldsymbol{P}$ within this coordinate system. The new location of the point p_j^* after applying the vector $\boldsymbol{v}_j = (v_j^x, v_j^y)$ can be written as:

$$(x_{p_j}^*, y_{p_j}^*) = (x_{p_j}, y_{p_j}) + (v_j^x, v_j^y) \tag{8}$$

Then the new coordinates set $\boldsymbol{C}$ of all feature points in $\boldsymbol{P}$ can be written as:

$$\boldsymbol{C} = F_{WR}(\boldsymbol{P}, \boldsymbol{\mathcal{V}}) = \{(x_{p_j}^*, y_{p_j}^*) | (x_{p_j}^*, y_{p_j}^*) = (x_{p_j}, y_{p_j}) + (v_j^x, v_j^y); j = 1, \cdots, m\}, \tag{9}$$

where $\boldsymbol{\mathcal{V}} = \{\boldsymbol{v}_j | j = 1, \cdots, m, m \in \mathbb{Z}\}$. If (x_{p_j}, y_{p_j}) has already been a new position for a feature point, it no longer serves as a new position for any other points. Consequently, by assigning corresponding values to these new positions, the optimized noisy latent $\boldsymbol{z}_t''$ can be obtained as shown in the following equation:

$$\boldsymbol{z}_{t\,(x_{p_j}+v_j^x, y_{p_j}+v_j^y)}'' = \boldsymbol{z}_{t(x_{p_j}, y_{p_j})} \tag{10}$$

In essence, the latent relocation operation optimizes semantics efficiently by utilizing the LWF-generated warpage vector, eliminating the need for iterative optimization.

However, as certain positions in the noisy latent may not be occupied by other feature points, $\boldsymbol{z}_t''$ obtained from one-step warpage optimization may contain regions with null values as shown in Fig. 4, leading to semantic losses that can adversely impact the drag result. We address this issue in Sec. 3.3.

3.3 Bilateral Nearest Neighbor Interpolation

To enhance the semantic integrity, BNNI interpolates points in null region using similar features from their neighboring areas in horizontal and vertical directions, thus ensuring the semantic integrity and enhancing the quality of drag editing. Let $\mathcal{N}$ be a point with coordinate $(x_\mathcal{N}, y_\mathcal{N})$ in null regions, we identify the nearest points of $\mathcal{N}$ containing value in four directions: up, right, down, and left, as illustrated in Fig. 4, which are used as reference points for interpolation. Then, the interpolated value for null point $\mathcal{N}$ can be calculated as:

$$\boldsymbol{z}'_{t(x_\mathcal{N}, y_\mathcal{N})} = \sum_{loc=u,r,d,l} w_{loc} \cdot ref_{loc} \tag{11}$$

where ref_{loc} denotes the value of reference point, and loc indicates the direction, with u, r, d and l representing up, right, down and left, respectively. w_{loc} is the interpolation weight for each reference point, which is calculated based on its distance to $\mathcal{N}$, as follows:

$$w_{loc} = \frac{1/len_{loc}}{\sum_{loc=u,r,d,l} 1/len_{loc}} \tag{12}$$

where len_{loc} represents the distance between the reference point and $\mathcal{N}$. Such that we can obtain the optimized noisy latent $\boldsymbol{z}'_t$ with complete semantic information by using BNNI to exploit similar semantic information from surrounding areas, further enhancing the quality of the drag editing.

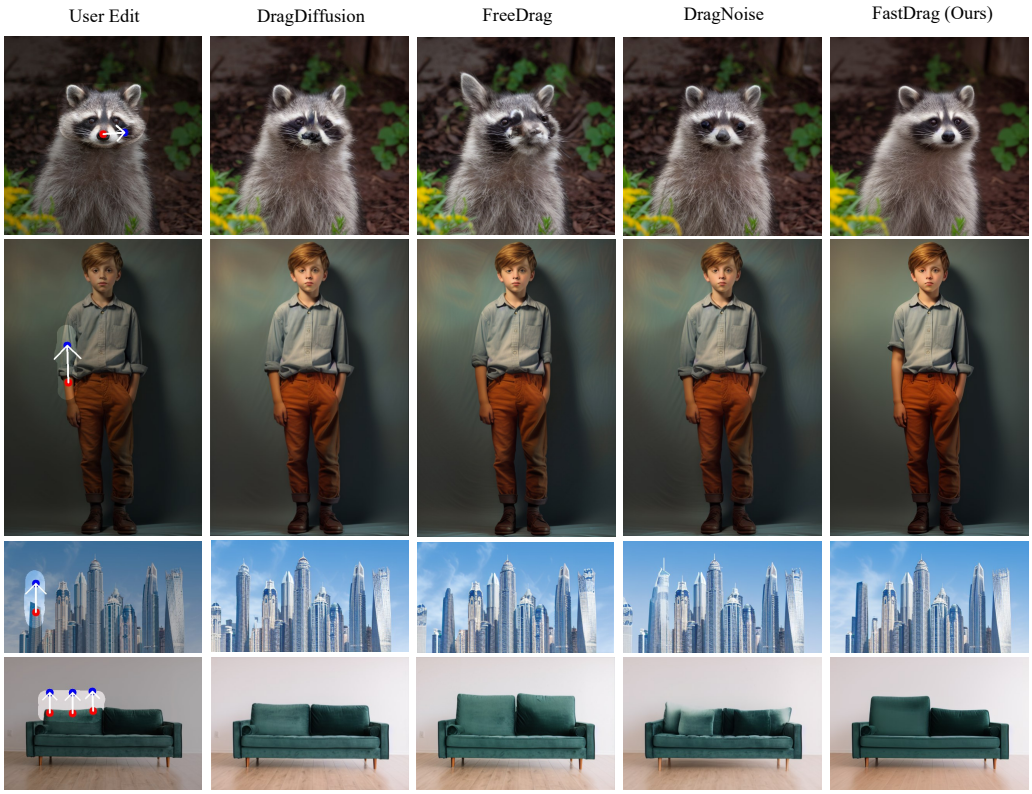


Figure 6: Illustration of qualitative comparison with the state-of-the-art methods.

3.4 Consistency-Preserving Strategy

Following [20, 2, 28], we introduce a consistency-preserving strategy to maintain the consistency between the edited image and the original image by adopting the semantic information of the original image (*i.e.*, key and value pairs) saved in self-attention module during diffusion inversion to guide the diffusion sampling, as illustrated in Fig. 5. Specifically, during the diffusion sampling, the calculation of self-attention $\text{Attention}_{\text{Sa}}$ within the upsampling process of the U-Net is as follows:

$$\text{Attention}_{\text{Sa}}(\boldsymbol{Q}_{\text{Sa}}, \boldsymbol{K}_{\text{In}}, \boldsymbol{V}_{\text{In}}) = \text{softmax}(\frac{\boldsymbol{Q}_{\text{Sa}} \cdot \boldsymbol{K}_{\text{In}}}{\sqrt{d}}) \cdot \boldsymbol{V}_{\text{In}} \tag{13}$$

where query $\boldsymbol{Q}_{\text{Sa}}$ is still used from diffusion sampling but key $\boldsymbol{K}_{\text{In}}$ and value $\boldsymbol{V}_{\text{In}}$ are correspondingly from diffusion inversion. Thus, the consistency-preserving strategy maintains the overall content consistency between the desired image and original image, ensuring the effect of drag-based editing.

4 Experiments

4.1 Qualitative Evaluation

We conduct experiments to demonstrate the drag effects of our FastDrag method, comparing it against state-of-the-art techniques such as DragDiffusion [28], FreeDrag [15], and DragNoise [16]. The qualitative comparison results are presented in Fig. 6. Notably, FastDrag maintains effective drag performance and high image quality even in images with complex textures, where n-step iterative methods typically falter. For instance, as shown in the first row of Fig. 6, FastDrag successfully rotates the face of an animal while preserving intricate fur textures and ensuring strong structural integrity. In contrast, methods like DragDiffusion and DragNoise fail to rotate the animal's face, and FreeDrag disrupts the facial structure.

In the stretching task, FastDrag outperforms all other methods, as shown in the second row of Fig. 6, where the goal is to move a sleeve to a higher position. The results show that other methods lack robustness to slight deviations in user dragging, where the drag point is slightly off the sleeve.

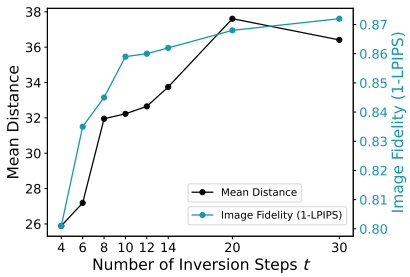


Figure 7: Ablation study on number of inversion steps in terms of quantitative metrics.

Table 1: Quantitative comparison with state-of-art methods on DragBench. Here, lower MD indicates more precise drag results, while higher 1-LPIPS reflects greater similarity between the generated and original images. The time metric represents the average time required per point based on RTX 3090. Preparation denotes LoRA training. † means FastDrag without LCM-equipped U-Net.

Approach	Venue	MD ↓	1 − LPIPS ↑	Time	
				Preparation	Editing(s)
DragDiffusion [28]	CVPR2024	33.70	0.89	1 min (LoRA)	21.54
DragNoise [16]	CVPR2024	33.41	0.63	1 min (LoRA)	20.41
FreeDrag [15]	CVPR2024	35.00	0.70	1 min (LoRA)	52.63
GoodDrag [32]	arXiv2024	22.96	0.86	1 min (LoRA)	45.83
DiffEditor [19]	CVPR2024	28.46	0.89	✗	21.68
FastDrag†(Ours)		33.22	0.87	✗	5.66
FastDrag (Ours)		32.23	0.86	✗	**3.12**

Despite this, FastDrag accurately moves the sleeve to the desired height, understanding the underlying semantic intent of dragging the sleeve.

Additionally, we perform multi-point dragging experiments, illustrated in the fourth row of Fig. 6. Both DragDiffusion and DragNoise fail to stretch the back of the sofa, while FreeDrag incorrectly stretches unintended parts of the sofa. Through the LWF introduced in Sec. 3.2.1, FastDrag can manipulate all dragged points to their target locations while preserving the content in unmasked regions. More results of FastDrag are illustrated in supplementary Sec. E.

4.2 Quantitative Comparison

To better demonstrate the superiority of FastDrag, we conduct quantitative comparison using DragBench dataset [28], which consists of 205 different types of images with 349 pairs of handle and target points. Here, mean distance (MD) [23] and image fidelity (IF) [11] are employed as performance metrics, where MD evaluates the precision of drag editing, and IF measures the consistency between the generated and original images by averaging the learned perceptual image patch similarity (LPIPS) [31]. Specifically, 1-LPIPS is employed as the IF metric in our experiment to facilitate comparison. In addition, we compare the average time required per point to demonstrate the time efficiency of our proposed FastDrag. The results are given in Table 1.

Apart from [28],[16],[15], two other state-of-the-art methods, *i.e.*, GoodDrag [32] and DiffEditor [19], are also adopted for comparison, with DiffEditor being the current fastest drag-based editing method. Due to well-designed one-step warpage optimization and consistency-preserving strategy, our FastDrag does not require LoRA training preparation, resulting in significantly reduced time consumption (*i.e.*, 3.12 seconds), which is nearly 700% faster than DiffEditor (*i.e.*, 21.68 seconds), and 2800% faster than the typical baseline DragDiffusion (*i.e.*, 1 min and 21.54 seconds). Moreover, even using standard U-Net without LCM, our method is still much faster than DiffEditor and far outperforms all other state-of-the-art methods. It is particularly noteworthy that, even with an A100 GPU, DiffEditor still requires 13.88 seconds according to [19], whereas FastDrag only requires 3.12 seconds on an RTX 3090.

In addition, our FastDrag also achieves competitive quantitative evaluation metrics (*i.e.*, IF and MD) comparable to the state-of-the-art methods, and even better drag editing quality, as illustrated in Fig. 6. These results demonstrate the effectiveness and superiority of our method.

4.3 Ablation Study

Inversion Step: To determine the number of inversion steps in diffusion inversion with LCM-equipped U-Net, we conduct an ablation experiment with number of inversion steps set as t = 4, 6, 8, 10, 12, 14, 20, and 30, where IF and MD are used to evaluate the balance between the consistency with original image and the desired drag effects. The results are given in Fig. 8 and Fig. 7, where we can see that when $t < 6$, the generated images lack sufficient detail to accurately reconstruct the original images. Conversely, when $t > 6$, it can successfully recover complex details such as intricate fur textures and dense stone while maintaining high image quality. However, when $t > 14$, some image details lost, which negatively impacts the effectiveness of the drag effect. By comprehensive

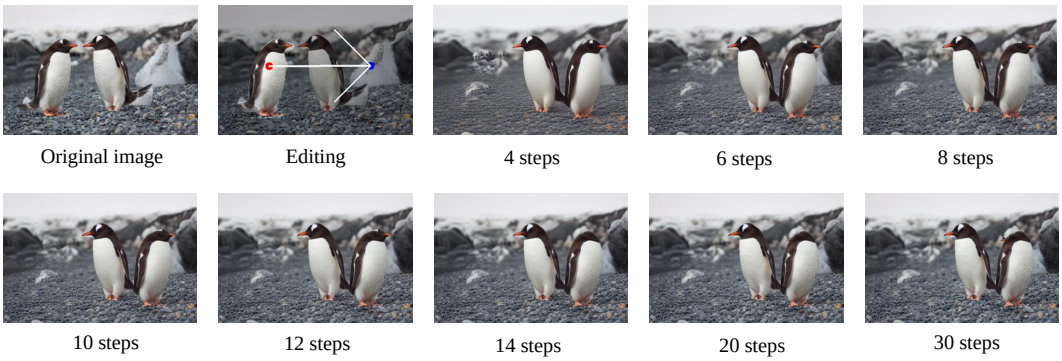


Figure 8: Ablation study on number of inversion steps in terms of drag effect.

Figure 9: Ablation study on bilateral nearest neighbor interpolation.

evaluation of both the drag effect and the similarity to the original images, we select 10 as the number of inversion steps for our method with LCM to balance the drag effect.

BNNI: To demonstrate the effectiveness of BNNI, we compare it with several interpolation methods on null point $\mathcal{N}$, including maintaining the original value of this position, interpolation by zero-value, and interpolation by random noise, denoted as "original value", "0 interpolation", and "random interpolation", respectively. The results are given in Fig. 9, where we can see that, by effectively utilizing surrounding feature values to interpolate null points, BNNI can address semantic losses, and enhance the quality of the drag editing.

Consistency-Preserving: We also conduct an experiment to validate the effectiveness of the consistency-preserving strategy in maintaining image consistency. The results are illustrated in Fig. 10, where "w/ CP" and "w/o CP" denote our FastDrag with and without using consistency-preserving strategy, respectively. It is obviously that our method with consistency-preserving strategy can effectively preserve image consistency, resulting in better drag editing effect.

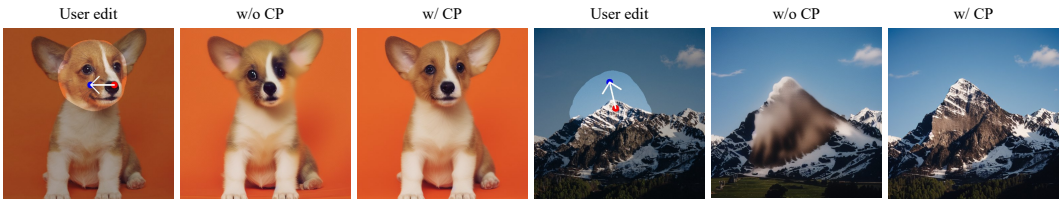
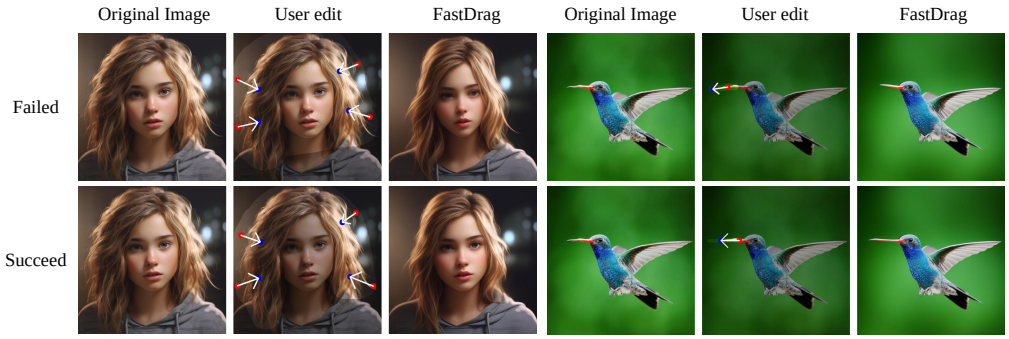


Figure 10: Ablation study on consistency-preserving strategy.

Figure 11: Illustration of failure cases for limitation analysis under extremely long-distance drag editing. Our FastDrag method may lose some detailed information in these cases but still achieves better editing performance compared to state-of-the-art (SOTA) methods.



(a) Desired editing effect: Thin the hair while keeping the face size

(b) Desired editing effect: Lengthen the beak

Figure 12: Illustration of the limitation analysis with failed and successful drag editing for highly relying on precise drag instruction. (a) It is best to exclude the face from the mask region. (b) The handle point should ideally be placed where the “beak" feature is more prominent.

5 Limitations

Despite FastDrag’s impressive editing speed compared to SOTA methods, it shares some common limitations. 1) **Overly Smooth and Finer Details Loss:** Similar to other diffusion-based methods [3, 28], FastDrag occasionally loses fine textures from the original images, as shown in Fig. 6, row 4. Despite this, FastDrag outperforms other methods in speed and overall performance. 2) **Extremely Long-distance Drag Editing:** In such case, object details may be lost due to the lower-dimensional latent space, in which significant changes in detail (i.e., long-drag editing) can disrupt the semantics, making it harder to preserve all details. Nevertheless, FastDrag handles long-distance editing better than other SOTA methods, as illustrated in Fig. 11, where our method successfully achieves long-distance drag editing that others fail to achieve. 3) **Highly Relying on Precise Drag Instruction:** Achieving optimal results depends heavily on clear drag instructions. As with other SOTA methods, precise input, such as excluding irrelevant areas from the mask (e.g., the face in Fig. 12, row 2) or correctly placing the handle point (e.g., beak in Fig. 12), is essential for better performance.

6 Conclusion

This paper has presented a novel drag-based image editing method, *i.e.*, FastDrag, which achieved faster image editing speeds than other existing methods. By proposing one-step warpage optimization and BNNI strategy, our approach achieves high-quality image editing according to the drag instructions in a very short period of time. Additionally, through the consistency-preserving strategy, it ensures the consistency of the generated image with the original image. Moving forward, we plan to continue refining and expanding our approach to further enhance its capabilities and applications.

7 Acknowledgments

This work was partly supported by Beijing Nova Program (20230484261).

References

[1] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. InstructPix2Pix: Learning to follow image editing instructions. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023.

[2] Ming 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. *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2023.

[3] Hansam Cho, Jonghyun Lee, Seoung Bum Kim, Tae-Hyun Oh, and Yonghyun Jeong. Noise map guidance: Inversion with spatial context for real image editing. In *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2024.

[4] Yutao Cui, Xiaotong Zhao, Guozhen Zhang, Shengming Cao, Kai Ma, and Limin Wang. StableDrag: Stable dragging for point-based image editing. *arXiv preprint arXiv:2403.04437*, 2024.

[5] Prafulla Dhariwal and Alex Nichol. Diffusion models beat GANs on image synthesis. In *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2021.

[6] Dave Epstein, Allan Jabri, Ben Poole, Alexei A Efros, and Aleksander Holynski. Diffusion self-guidance for controllable image generation. In *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2023.

[7] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. In *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2023.

[8] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2021.

[9] Xingzhong Hou, Boxiao Liu, Yi Zhang, Jihao Liu, Yu Liu, and Haihang You. Easydrag: Efficient point-based manipulation on diffusion models. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 8404–8413, 2024.

[10] Xuan Ju, Ailing Zeng, Yuxuan Bian, Shaoteng Liu, and Qiang Xu. Direct inversion: Boosting diffusion-based editing with 3 lines of code. In *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2024.

[11] 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 *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023.

[12] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. DiffusionCLIP: Text-guided diffusion models for robust image manipulation. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022.

[13] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[14] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013.

[15] Pengyang Ling, Lin Chen, Pan Zhang, Huaian Chen, Yi Jin, and Jinjin Zheng. FreeDrag: Feature dragging for reliable point-based image editing. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024.

[16] Haofeng Liu, Chenshu Xu, Yifei Yang, Lihua Zeng, and Shengfeng He. Drag your noise: Interactive point-based editing via diffusion semantic propagation. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024.

[17] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023.

[18] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023.

[19] Chong Mou, Xintao Wang, Jiechong Song, Ying Shan, and Jian Zhang. DiffEditor: Boosting accuracy and flexibility on diffusion-based image editing. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024.

[20] Chong Mou, Xintao Wang, Jiechong Song, Ying Shan, and Jian Zhang. DragonDiffusion: Enabling drag-style manipulation on diffusion models. In *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2024.

[21] D. Naylor. Theoretical elasticity, by A. E. Green and W. Zerna (Second edition). Clarendon Press, Oxford, 1968. xv + 457 pages. *Can. Math. Bull*, 12:537–538, 1969.

[22] Shen Nie, Hanzhong Allan Guo, Cheng Lu, Yuhao Zhou, Chenyu Zheng, and Chongxuan Li. The blessing of randomness: SDE beats ODE in general diffusion-based image editing. In *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2024.

[23] 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 *Proc. ACM Conf. SIGGRAPH*, 2023.

[24] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proc. Int. Conf. Machine Learning (ICML)*, 2021.

[25] Herbert E. Robbins. A stochastic approximation method. *Ann. Math. Stat.*, 22:400–407, 1951.

[26] Robin Rombach, A. Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021.

[27] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022.

[28] Yujun Shi, Chuhui Xue, Jiachun Pan, Wenqing Zhang, Vincent YF Tan, and Song Bai. DragDiffusion: Harnessing diffusion models for interactive point-based image editing. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024.

[29] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021.

[30] Sihan Xu, Yidong Huang, Jiayi Pan, Ziqiao Ma, and Joyce Chai. Inversion-free image editing with natural language. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024.

[31] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018.

[32] Zewei Zhang, Huan Liu, Jun Chen, and Xiangyu Xu. GoodDrag: Towards good practices for drag editing with diffusion models. *arXiv preprint arXiv:2404.07206*, 2024.

A Supplementary Experiments

We conduct supplementary quantitative experiments on BNNI and consistency-preserving strategies to further validate their effectiveness. The quantitative metrics used are consistent with those described in Sec. 4.1.

BNNI: Following the setup in Sec.4.3, we compare it with several interpolation methods on null point $\mathcal{N}$, including maintaining the original value of this position, interpolation by zero-value, and interpolation by random noise, denoted as "origin", "0-inter", and "random-inter", respectively. As illustrated in Fig. 13, FastDrag with BNNI achieves the best MD levels compared to other interpolation methods, while its IF is second only to "origin". However, "origin" can lead to negative drag effects, as shown in Fig. 9. Therefore, by effectively utilizing surrounding feature values to interpolate null points, BNNI can address semantic losses and enhance the quality of drag editing.

Consistency-Preserving: We also conduct experiments to assess the impact of initiating the consistency-preserving strategy at different sampling steps. The results, as shown in Fig. 14, indicate that as the starting step increases (*i.e.*, the frequency of key and value replacements decreases), the IF decreases, leading to poorer image consistency. Meanwhile, the MD initially decreases and then increases as the starting step increases. It is evident that consistency-preserving strategy can effectively maintain the consistency between the generated images and the original images.

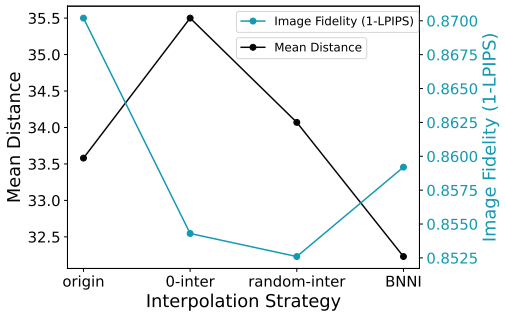


Figure 13: Ablation study on BNNI in terms of quantitative metrics.

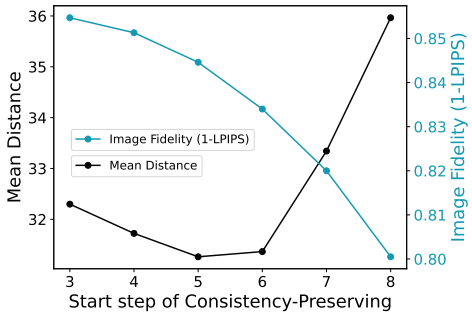


Figure 14: Ablation study on consistency-preserving in terms of quantitative metrics.

B Single-step Diffusion

When integrating DragDiffusion with a single-step diffusion model, the editing time is still much longer than that of FastDrag. For DragDiffusion and FastDrag under diffusion steps of 1, 20, and 50, we calculate the time required for inversion, sampling, and latent optimization respectively. The results provided in Fig. 15 show that even with a single diffusion step (i.e., diffusion step set as 1), DragDiffusion still requires significantly more time (20.7 seconds) compared to FastDrag (2.88 seconds).

In addition, as observed in Fig. 15, DragDiffusion spends significantly more time on latent optimization compared to diffusion inversion and sampling. Therefore, reducing the time spent on latent optimization is crucial for minimizing overall editing time, which is precisely what FastDrag accomplishes.

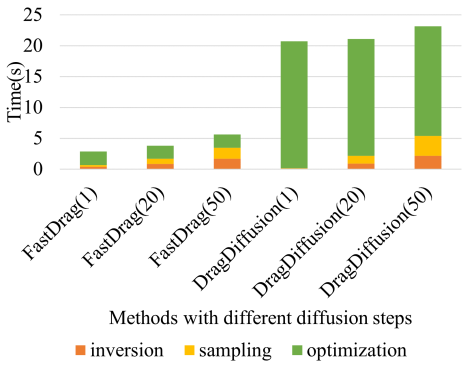


Figure 15: Overall editing time comparison with different diffusion steps between FastDrag and DragDiffusion. All experiments are conducted on RTX 3090 with diffusion step set as 1, 20, and 50 respectively. Optimization means latent optimization.

C Statistical Rigor

To further validate the superiority of our work and to achieve statistical rigor, we conducted an additional experiment by repeating our experiment 10 times under the same experimental settings. We observed that the variances of the performance metrics obtained from 10 realizations of our FastDrag are MD (0.000404), 1-LPIPS (9.44E-11), and Time (0.018), all of which fall within a reasonable range. These statistical results further demonstrate the effectiveness and stability of our method for drag editing.

D Implementation Details

We utilize a pretrained LDM (*i.e.*, Stable Diffusion 1.5 [27]) as our diffusion model, where the U-Net structure is adapted with LCM-distilled weights from Stable Diffusion 1.5. It is worth emphasizing that the U-Net structure used in our model is widely used in image generation methods [28, 15, 16, 32, 4]. Unless otherwise specified, the default setting for inversion and sampling step is 10. Following DragDiffusion [28], classifier-free guidance (CFG) [8] is not applied in diffusion model, and we optimize the diffusion latent at the 7th step. All other configurations follow that used in DragDiffusion. Our experiments are conducted on an RTX 3090 GPU with 24G memory.

For the special application of drag-based editing, *i.e.*, object moving, as shown in Fig. 8, significant null region may be left at the original position of the object due to long-distance relocation, posing challenges for BNNI. To ensure semantic integrity in the image and facilitate user interaction, we adopted two straightforward strategies. For first strategy, specifically, we introduce a parameter r, set as 2, centered at the unique target point e_1, defining a rectangular area with dimensions $2r$. We then extract the noised-latent representation within this rectangular area and fill it into the mask, effectively restoring the semantics at the original position of the object. For second strategy, we just maintain the original semantic of original position to avoid null region. Under second strategy, object moving will produce the effect of object replication, as shown in the third row of Fig. 16.

E More Results

We apply FastDrag to drag-based image editing in various scenarios, including face rotation, object movement, object stretching, object shrinking and so on. The experimental results, as shown in Fig. 16, demonstrate that FastDrag achieves excellent drag-based effects across multiple scenarios.

F Societal Impacts

FastDrag has the potential to bring about several positive societal impacts. Firstly, they offer intuitive and efficient image editing tools, catering to artists, designers, and creators, thereby fostering creativity and innovation. Secondly, their user-friendly nature simplifies the image editing process, increasing accessibility and participation among a wider audience. In addition, our FastDrag improves efficiency and saves the user's time and effort, thus increasing productivity. Moreover, the innovative and flexible nature of these methods opens up possibilities for various applications, spanning art creation, design, education, and training.

However, along with these positive aspects, FastDrag can also have certain negative societal implications. They may be exploited by unethical individuals or organizations to propagate misinformation and fake imagery, potentially contributing to the spread of false news and undermining societal trust. Furthermore, widespread use of image editing tools may encroach upon individual privacy rights, particularly when unauthorized information or imagery is manipulated. Moreover, inappropriate or irresponsible image editing practices could lead to social injustices and imbalances, such as distorting facts or misleading the public, thereby influencing public opinion and policies negatively.

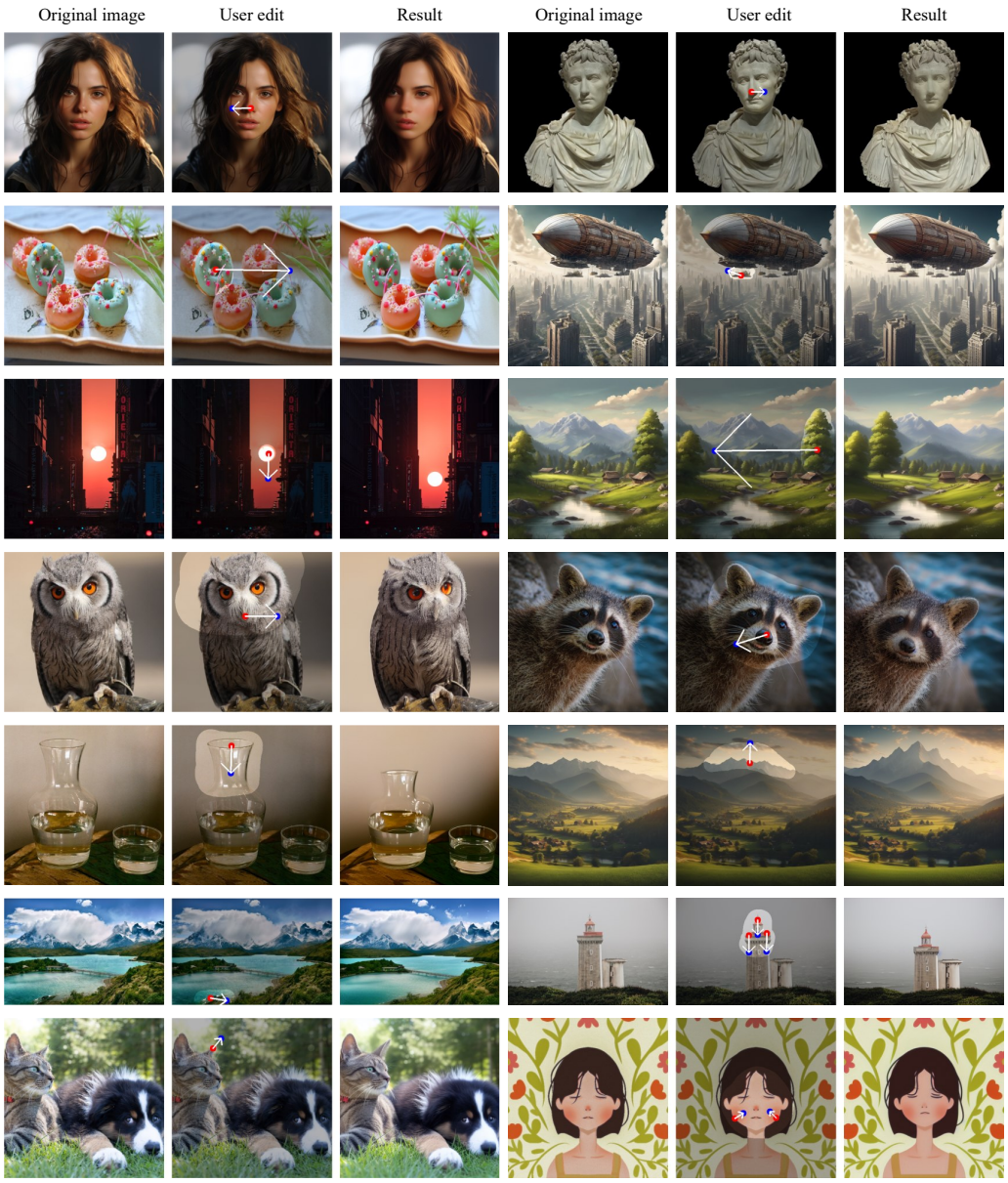


Figure 16: More visualized results of FastDrag.

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: We clearly outline the contributions of our approach in both the abstract and the introduction in Sec.1 of our paper.

   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: We discuss the limitations of the our method in Section 5

   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 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: Our paper does not involve theoretical results.

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 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: Our experimental results are reproducible. We provide a detailed description of our method's workflow in Sec.3 and implementation details in Sec.D.

   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: We include the implementation code for our method in the supplementary materials submitted.

Guidelines:

- The answer NA means that 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: The main experimental details of our method are discussed in Appendix D.

   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 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: Our experiments, conducted on the DragBench dataset, yield average metrics that are statistically significant and provide a reliable basis for comparing our method with others. And we discuss experimental statistical significance in Appendix C.

   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 than 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 compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have provided detailed information about the computer resources used for all our experiments in Appendix D

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers 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 compute 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: We have thoroughly reviewed the NeurIPS Code of Ethics and can confirm that our research conducted in the paper adheres to 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: The potential positive societal impacts and negative societal impacts of the our work is discussed in Appendix F

    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 responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our paper does not introduce any new datasets or pretrained models.

    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 a 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: The models and baselines utilized in our method are detailed in Appendix D of 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: Our paper does not introduce any new assets.

    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: During our research, we do not employ crowdsourcing or conduct studies involving human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor 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: In our research process, human subjects were not required to assist with the study.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor 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.