Localize, Understand, Collaborate: Semantic-Aware Dragging via Intention Reasoner

Xing Cui[1], Peipei Li[1]*, Zekun Li[2], Xuannan Liu[1], Yueying Zou[1], Zhaofeng He[1]
[1]Beijing University of Posts and Telecommunications
[2]University of California, Santa Barbara
{cuixing, lipeipei, liuxuannan, zouyueying2001, zhaofenghe}@bupt.edu.cn
zekunli@cs.ucsb.edu
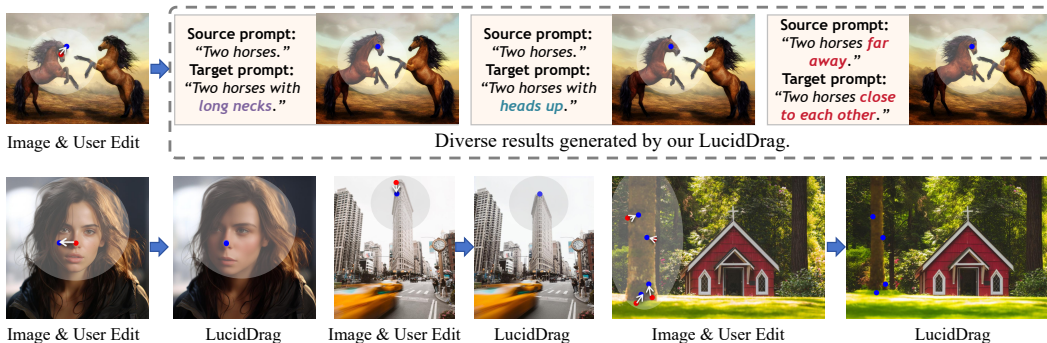
Figure 1: Given an input image, the user draws a mask specifying the editable region and clicks dragging points (handle points (red) and target points (blue)). Our LucidDrag considers the ill-posed nature of drag-based editing and can produce diverse results (the first row). Besides, it achieves outstanding performance in editing accuracy and image fidelity (the second row).

Abstract

Flexible and accurate drag-based editing is a challenging task that has recently garnered significant attention. Current methods typically model this problem as automatically learning "how to drag" through point dragging and often produce one deterministic estimation, which presents two key limitations: 1) Overlooking the inherently ill-posed nature of drag-based editing, where multiple results may correspond to a given input, as illustrated in Fig. 1; 2) Ignoring the constraint of image quality, which may lead to unexpected distortion. To alleviate this, we propose LucidDrag, which shifts the focus from "how to drag" to "what-then-how" paradigm. LucidDrag comprises an intention reasoner and a collaborative guidance sampling mechanism. The former infers several optimal editing strategies, identifying what content and what semantic direction to be edited. Based on the former, the latter addresses "how to drag" by collaboratively integrating existing editing guidance with the newly proposed semantic guidance and quality guidance. Specifically, semantic guidance is derived by establishing a semantic editing direction based on reasoned intentions, while quality guidance is achieved through classifier guidance using an image fidelity discriminator. Both qualitative and quantitative comparisons demonstrate the superiority of LucidDrag over previous methods. Code is available at: `https://github.com/cuixing100876/LucidDrag-NeurIPS2024`.

---

*Corresponding author

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

1 Introduction

The impressive success of diffusion models [27, 12, 51] has promoted the advancements in the field of image editing [26, 69, 17]. The conventional paradigms for editing conditions typically encompass text [13, 47, 25], instruction [4, 71, 68, 16], or image [64, 61, 21]. However, these conditions prove inadequate in effectively communicating specific image aspects, such as shape and location [15].

To address this, recent studies [14, 50, 53] define a new task called drag-based editing, which incorporates dragging points as conditions. These studies specifically regard drag-based editing as the problem of "how to drag" and tackle it by designing an editing guidance loss, enabling the model to implicitly learn the appropriate solutions. Despite their considerable success [50, 45, 46], these methods have two limitations: ***Firstly, they neglect the inherent ambiguity of semantic intention.*** Drag-based editing is an ill-posed problem. As depicted in Fig. 1, the drag points starting from the horse's head and ending at its upper right can indicate various semantic intentions, such as "make the neck longer," "raise the head," or "bring the two horses closer." However, existing methods mainly follow point dragging principles [53, 45], focusing on positional movement by constraining feature correlation between the source and target points. This current position optimization strategy inherently overlooks semantic diversity, making it challenging to generate images with precise semantic perception. ***Secondly, they overlook the preservation of the overall image quality.*** Current methods prioritize editing accuracy while neglects the overall image quality. Some methods [45, 46] utilize score-based classifier guidance for image editing, which can cause mismatches between the distribution of the edited image and the input image, compromising image fidelity.

In this study, we divide the drag-based editing task into two steps. We introduce a preliminary step, "what to drag", to determine the specific content and semantic direction for editing before addressing "how to drag". That is, we shift the focus from "how to drag" to a paradigm of "what-then-how". As shown in the first row of images in Fig. 1, before editing, we need to determine what we are going to edit, such as the horse's head, and the semantic strategy for editing it towards the upper right. For example, we could make the horse lift its head, elongate its neck, or shorten the distance between the two horses. With the "what to drag" information established, we can then proceed to address "how to drag". To achieve this, we construct an intention reasoner that integrates a Large Language-Vision Model (LVLM) and a Large Language Model (LLM) to deduce possible intentions. As illustrated in Fig. 1, given an input image and drag points that begin at the horse's face and end in its upper-right, the intention reasoner acts as an AI agent to infer potential intentions, subsequently providing corresponding source and target prompts. Once the intention is determined, we then address "how to drag" by injecting the reasoned semantic information into the model by developing collaborative guidance sampling, which integrates editing guidance with the proposed semantic guidance and quality guidance. Specifically, the semantic guidance is derived from the source and target prompts determined by the intention reasoner. The asymmetric prompts establish a clear semantic editing direction towards the target intention. Additionally, a discriminator is employed as the score function to provide quality guidance. The quality gradient is generated based on image fidelity and incorporated into the model via the classifier guidance mechanism. For the editing guidance, we follow previous work [45] to maximize the feature correspondence between the source and target positions.

LucidDrag is an intuitive framework for "what-then-how" drag-based editing, demonstrating outstanding performance in terms of semantic perception ability, diversity, and editing quality. LucidDrag enjoys several attractive attributes: ***Firstly, clear, diverse, and reasonable semantic intentions.*** Our innovative method employs LVLM and LLM to explicitly deduce the intention by localizing the drag points and reasoning several probable intentions. By explicitly incorporating semantics, we enhance semantic perception and offer diverse editing modes, enriching the variety of outcomes. ***Secondly, enhanced overall generation quality.*** By introducing collaborative guidance sampling, we significantly promote the generation quality of drag-based editing. Specifically, we achieve diverse and accurate image editing by introducing an additional semantic editing direction through semantic guidance. Additionally, we maintain better image quality by explicitly constraining the image distribution using quality guidance. In summary, our main contributions are:

- We propose a new "what-then-how" paradigm for drag-based editing and introduce Lucid-Drag. To address the "what" problem, we present an intention reasoner, which employs LVLM and LLM to determine what content and semantic direction should be edited.

2

- We then use the inferred semantic results of "what" to guide "how", enhancing editing accuracy and overall image quality. Additionally, a quality discriminator is also employed to provide quality gradients via score-based classifier guidance. This quality guidance, combined with editing and semantic guidance from "what," improves the precision and fidelity of the results.

- We present quantitative and qualitative results demonstrating the applicability and superiority of our method, in terms of editing accuracy, fidelity, and diversity.

2 Related Work

2.1 Diffusion Models

Diffusion model [54, 20, 56] aims to estimate the noise ϵ_t added to the image $z_t = \alpha_t x + \sigma_t \epsilon_t$, where α_t and σ_t are non-learned parameters. The training loss is to minimize the distance between the added noise and the estimated noise:

$$L(\theta) = \mathbb{E}_{t \sim \mathcal{U}(1,T), \epsilon_t \sim \mathcal{N}(0,I)} ||\epsilon_t - \epsilon_\theta(z_t; t, y)||_2^2, \tag{1}$$

where t refers to the time step, ϵ_t is the ground-truth noise, y is an additional condition. Diffusion models can be regarded as score-based models [56]. In this context, ϵ_θ serves as an approximation of the score function for the noisy marginal distributions: $\epsilon_\theta(z_t) \approx \nabla_{z_t} \log p(z_t)$.

we can sample images given conditioning y by starting from a random noise $z_T \sim \mathcal{N}(0, I)$, and then alternating between estimating the noise component and sampling z_{t-1}. The noise is estimated as:

$$\hat{\epsilon}_t = \epsilon_\theta(z_t; t, y). \tag{2}$$

The sampling could be based on DDPM [20] or DDIM [55]. In this paper, we utilize DDIM, which denoising z_t to a previous step z_{t-1} with a deterministic process:

$$z_{t-1} = \sqrt{\frac{\alpha_{t-1}}{\alpha_t}} z_t + \left(\sqrt{\frac{1}{\alpha_{t-1}} - 1} - \sqrt{\frac{1}{\alpha_t} - 1} \right) \cdot \hat{\epsilon}_t. \tag{3}$$

2.2 Classifier Guidance

Classifier guidance moves the sampling process towards images that are more likely according to the classifier [12]. As a powerful conditional sampling strategy, it has been used in various tasks, including generating diverse results [8], refining generative process [31, 65] and image editing [45, 46]. Specifically, it combines the unconditional score function for $p(z_t)$ with a classifier $p(y|z_t)$ to produce samples from $p(z_t|y) \propto p(y|z_t)p(z_t)$ [12, 56]. Classifier Guidance requires access to a labeled dataset and the training of a noise-dependent classifier $p(y|z_t)$ which can be differentiated concerning the noisy image z_t. During the sampling process, classifier guidance can be incorporated as follows:

$$\nabla_{z_t} \log q(z_t|y) \propto \nabla_{z_t} \log q(z_t) + \nabla_{z_t} \log q(y|z_t), \tag{4}$$

The first term is the original diffusion denoiser, and the second term refers to the conditional gradient produced by an energy function $g(z_t; t, y) = q(y|z_t)$. Thereby, we apply classifier guidance by modifying $\hat{\epsilon}_t$:

$$\hat{\epsilon}_t = \epsilon_\theta(z_t; t, y) + \eta \nabla_{z_t} \log g(z_t; t, y), \tag{5}$$

where η is an additional parameter parameter that modulates the strength of the guidance.

2.3 Visual Programming

As LLMs [49, 7, 58] and LVLMs [74, 11, 67] demonstrated remarkable emergency abilities [1, 62, 40, 41], researches [18, 37, 59] explore to leverage them for planning and reasoning in multi-modal image generation. For example, ChatEdit [9] utilizes pre-trained language models to track the user intents. Some approaches [18, 37, 59] augment the original prompt through paraphrasing. Recent, visual programmer methods [22, 24] translate complex input prompts into programmatic operations and data flows. Despite the effectiveness of these strategies in augmenting the input text instructions, they overlook the capacity to reason within visual-modal instructions, such as the dragging points in drag editing tasks. In contrast, our LucidDrag addresses this gap by integrating LVLM and LLM.

3
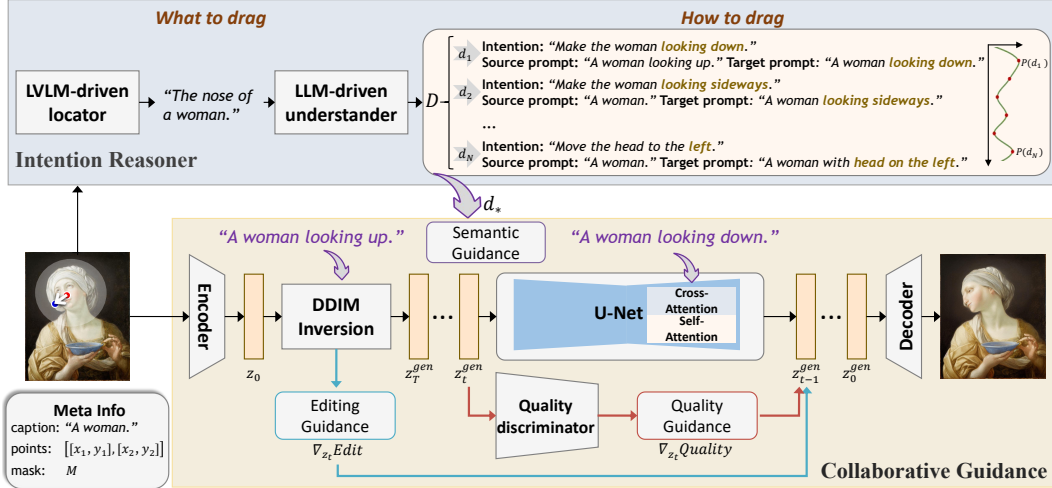
Figure 2: Overview of LucidDrag. LucidDrag comprises two main components: an intention reasoner and a collaborative guidance sampling mechanism. *Intention Reasoner* leverages an LVLM and an LLM to reason N possible semantic intentions. *Collaborative Guidance Sampling* facilitates semantic-aware editing by collaborating editing guidance with semantic guidance and quality guidance.

2.4 Image Editing

Image editing aims to manipulate an image according to specific conditions. Prior methodologies [33, 36, 34] can only manipulate specific attributes. The prevailing approaches have primarily relied on text conditions. For example, [9, 2] manipulate images in the GAN [29] latent space by learning an edit direction. Motivated by the success of the diffusion model [27, 60], state-of-the-art methods extend their exploration into diffusion-based image editing by exploring the initial noise [10, 73, 43], attention maps [6, 26, 63, 15], or prompts [35, 44, 13, 4, 57]. Recently, DragGAN [50] explores a novel editing scheme that drags any points of the image to reach target points with the help of StyleGAN [29] latent space. FreeDrag [38] improves point tracking by introducing adaptive feature updating and backtracking. Readout Guidance [42] solves the challenging task by leveraging the video dataset. The following works [45, 46] utilize the feature correspondence to direct the editing process. Nevertheless, they only use the drag points as a control, which is insufficient due to the potential diversity in semantic intentions. In contrast, our approach introduces an intention reasoner to achieve semantic-aware editing. The intention reasoner can reduce cognitive load, handle vague requests, and discover potential needs. Additionally, it generates precise descriptions automatically, ensuring accurate and consistent manipulations.

3 Method

In this section, we introduce LucidDrag, a unified framework for Drag Manipulation via **L**ocaling, **U**nderstanding, **C**ollaborate Gu**id**ing. Within our framework, image drag editing is decomposed into two stages. Firstly, the ***Intention Reasoner*** translates the user-drag points into potential semantic intentions and generates corresponding source prompt and target prompt, thus solving the problem of "what to edit". Then, the ***Collaborative Guidance Sampling*** is designed to facilitate image editing. Concretely, prompts generated from the intention reasoner are utilized in the DDIM Inversion and Diffusion U-Net, providing semantic guidance for the generation of semantically controllable results. Besides, LucidDrag employs a discriminator to provide quality guidance for improved image fidelity. The semantic guidance and the quality guidance collaborate with the original editing guidance, offering a novel perspective of "how to edit". We elaborate on more details of our techniques below.

3.1 Intention Reasoner

As depicted in Fig. 2, the first stage of our LucidDrag is the intention reasoner, which bridges the gap between the input point condition and semantic intention. This intention reasoner consists of two key

4

components, *i.e.*, an LVLM locator that identifies the interested position, and an LLM understander that interprets input conditions into semantic intentions.

LVLM-driven loactor   Given an input image, it may contain various objects situated at different positions. To accurately identify the objects of interest, we employ an off-the-shelf pre-trained large vision-language model (LVLM) Osprey [67]. Osprey is trained with fine-grained mask regions, enabling it to comprehend images at the pixel level. With the input image x, we instruct the Osprey model with drag points P to generate a descriptive representation of the objects of interest O, *i.e.*, $O = LVLM(x, P)$. As shown in Fig. 2, O="*The nose of a woman*", which subsequently serves as input to prompt the LLM to understand the condition and reason potential intentions.

LLM-driven reasoner   As shown in Fig. 2, each point condition may encompass various semantic intentions. For instance, it may represent non-rigid manipulation, such as "looking down", while it may also represent rigid manipulation, such as "moving to the left". Our LLM-driven reasoner is designed to discern potential semantic intentions to facilitate semantic-aware drag-based editing. We leverage the capabilities of the large language model, GPT 3.5 [49], acting as an AI agent to reason the possible intentions. We take the generated description of the object of interest O, the original image caption C, and drag points P as input. Then, we prompt the LLM with in-context examples to generate N possible intentions, *i.e.* $D = LLM(O, C, P)$, where each output sample can have different intentions or levels of complexity. Specifically, $D = \{(d_j, P(d_j))\}_{j=1}^{N}$, where $d_j = \{i_j, p_j^s, p_j^t\}$ is the generated text output. i_j, p_j^s, p_j^t represents the predicted intention, the predicted description of the source image (source prompt), and the predicted description of the target image (target prompt), respectively. $P(d_j)$ is the corresponding confidence probabilities of the j-*th* generate text output. Finally, we can select n outputs $\{d_{s*}\}_{s=1}^{n}$ by sampling based on the confidence probabilities. The confidence probability reflects the quality of the output. A higher confidence probability indicates that the intention is more reasonable, leading to better editing results.

$$\{d_{s*}\}_{s=1}^{n} = \underset{d_i \in D}{argmax} \left(P(d_i), n \right). \tag{6}$$

3.2   Collaborative Guidance Sampling

As shown in Fig. 2, the objective of collaborative guidance sampling is to modify the intended content while ensuring the preservation of irrelevant components. The input image is inverted to noise z_T^{gen} through DDIM Inversion [55, 44]. During the inversion process, the intermediate noise z_t^{gud}, along with the corresponding key k_t^{gud}, and value v_t^{gud} of the self-attention layer, are recorded in the memory bank, which serves in guiding subsequent generation process. Subsequently, we generate the edited images employing collaborative guidance sampling which incorporates three fundamental components: semantic guidance, quality guidance, and editing guidance. Each of these components contributes to the overall editing process distinctively, thereby ensuring a balanced and comprehensive approach to image editing.

Semantic guidance   As textual conditions can convey semantic information, we leverage the source and target prompts generated by the intention reasoner to facilitate semantic-aware dragging. Specifically, during the inversion process, we employ the source prompt to transform the input image into its corresponding noise by iterating DDIM inversion, *i.e.*, $z_{t+1}^{gud} = DDIM\_inversion(z_t^{gud}, p_*^s)$. During the sampling process, we utilize the target prompt to generate the target image, *i.e.*, $z_{t-1}^{gen} = DDIM(z_t^{gen}, p_*^t)$. As there exists a divergence between the source and target prompts, the asymmetric textual condition introduces a distinct editing direction that is oriented toward the target image, thereby offering semantic guidance. This differentiation facilitates a semantically guided editing process, enhancing the semantic coherence of the edited image.

Quality guidance   As shown in Fig. 2, we design quality guidance to ensure the quality of the generated image. A discriminator is trained to distinguish between high-fidelity images and low-fidelity images at any step t. In particular, given a real image and its corresponding text description y, we utilize a stochastic process to simulate potential points of drag and their respective directions. Then, we generate images using DragonDiffusion [45]. The images with an aesthetic score [52]

5

below 5, are classified as low-fidelity. Their corresponding real images are considered high-fidelity. The selected low-fidelity images, along with the high-fidelity images, constitute the training dataset of the discriminator. Finally, the training dataset comprises a total of 10,000 high-fidelity images and 10,000 low-fidelity images.

As the intermediate representation of the diffusion U-Net captures semantic information of the input image [32], we utilize this hidden representation to evaluate image quality. The discriminator comprises the down block and middle block of the Diffusion U-Net to capture the semantic information, followed by a linear classifier layer. During training, we froze the down blocks which are initialized with Stable Diffusion v2.1-base [51]. The middle block and the prediction layer are fine-tuned to classify images as real or fake. The conditional discriminator $d(X_t|y;t)$ is trained by minimizing the canonical discrimination loss:

$$
\begin{aligned}
\mathcal{L} = \mathbb{E}_{y,t}\big[&\mathbb{E}_{z_t \sim p(z_t|y)}[-\log d(z_t|y;t)] \\
&+\mathbb{E}_{z_t \sim q(z_t|y)}[-\log(1 - d(z_t|y;t))]\big].
\end{aligned}
\tag{7}
$$

The energy function to constrain the image quality is defined as [19, 31]:

$$
g_{quality} = \frac{p(z_t|y)}{q(z_t|y)} \approx \frac{d^*(z_t|y;t)}{1 - d^*(z_t|y;t)}.
\tag{8}
$$

Editing guidance   Following DragonDiffusion [45], we extract intermediate features F_t^{gen} and F_t^{gud} from z_t^{gen} and z_t^{gud} via UNet denoiser $\epsilon_{\boldsymbol{\theta}}$ respectively. The energy function is built by calculating the correspondence between F_t^{gen} and F_t^{gud}. The editing guidance includes editing target contents and preserving unrelated regions. We denote the original content position, target content position, and the unrelated content position as binary masks, *i.e.*, m^{orig}, m^{tar}, and m^{share}, respectively. The energy function to drag the content is defined as:

$$
g_{drag} = \frac{1}{\alpha + \beta \cdot \mathcal{S}(F_t^{gen}, m^{tar}, F_t^{gud}, m^{orig})},
\tag{9}
$$

where $\mathcal{S}(F_t^{gen}, m^{tar}, F_t^{gud}, m^{orig})$ calculates the similarity between the two regions of F_t^{gen} and F_t^{gud}. Similarly, the energy function to preserve the unrelated region is defined as:

$$
g_{content} = \frac{1}{\alpha + \beta \cdot \mathcal{S}_{local}(F_t^{gen}, m^{share}, F_t^{gud}, m^{share})}.
\tag{10}
$$

The final editing energy function is defined as:

$$
g_{edit} = w_e \cdot g_{drag} + w_c \cdot g_{content},
\tag{11}
$$

where w_e and w_c are hyper-parameters to balance these guidance terms. Finally, the editing guidance collaborates with the quality guidance during sampling:

$$
g(z_t; t, y) = g_{edit} + g_{quality}.
\tag{12}
$$

Additionally, following [5], to ensure content consistency between the edited image and the input images, we replace the keys and values within the self-attention module of the UNet decoder with those retrieved from the memory bank.

4   Experiments

4.1   Implementation Details

To train the quality discriminator, we employ the Adam optimizer with a learning rate of 1e-4. We set the training epochs as 100 and the batch size as 128. For the denoising process, we adopt Stable Diffusion [51] as the base model. During sampling, the number of denoising steps is set to $T = 50$ with a classifier-free guidance of 5. The energy weights for $g_{quality}$, g_{drag} and $g_{content}$ are set to $1e-3$, $4e-4$ and $4e-4$, respectively. The training of the discriminator can be conducted on a NVIDIA V100 GPU and the inference can be conducted on a NVIDIA GeForce RTX 3090 GPU.

6
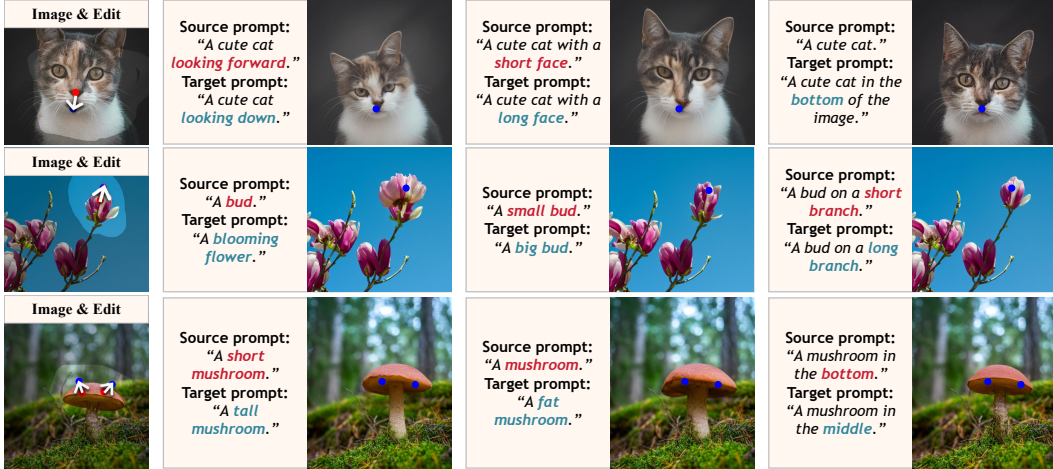
Figure 3: LucidDrag allows generating diverse results conforming to the intention.

| Image&Edit | DragDiffusion | FreeDrag | DragonDiffusion | DiffEditor | Ours |

Qualitative comparison between our LucidDrag and other methods in drag-based editing.

4.2 Comparisons

Semantic-aware dragging   Since our LucidDrag effectively discerns potential intentions, we first evaluate its semantic-awareness ability. Specifically, given an input image and corresponding dragging conditions, we sample several intentions and obtain corresponding results. The results are shown in Fig. 3. On the one hand, the intention reasoner deduces reasonable intentions that align with the input dragging points and generate semantic-aware images, demonstrating both an enhanced understanding of semantic intentions and increased diversity. On the other hand, our method can generate high-fidelity images aligned with the input prompts, improving the quality of the results.

Content dragging   We evaluate the proposed LucidDrag against existing drag editing models [38, 53, 45, 46]. We first conduct quantitative comparisons. Following DragDiffusion [53], we utilize the DragBench benchmark which is designed for the image-dragging task. In DragBench, each image is accompanied by a set of dragging instructions, including several pairs of handle and target points

7

Table 1: Comparisons of content dragging on DragBench.

	DragDiffusion	FreeDrag	DragonDiffusion	DiffEditor	Ours
Mean Distance (↓)	32.60	30.40	29.16	26.01 ± 0.81	$\mathbf{20.46 \pm 0.77}$
GScore (↑)	6.94	7.15	6.26	6.42	**7.37**

and a mask indicating the editable region. For editing precision, we use the Mean Distance [50] to evaluate the model's ability to move the contents to the target points. For image quality, existing Image Quality Assessment methods [30, 70] rely on handcrafted features or are trained on limited image samples, which do not always align well with human perception [72]. Thereby, we employ GScore [72] to provide a human-aligned assessment of image quality via Large Multimodal Models.

The quantitative results are presented in Tab. 1. For dragging precision, our LucidDrag consistently outperforms other methods by a significant margin across all categories in Mean Distance, indicating higher accuracy in dragging handle contents to target positions. In terms of image quality, our method achieves an average GScore of 7.37, surpassing DragDiffusion (6.94), FreeDrag (7.15), DragonDiffusion (6.26), and DiffEditor (6.42).

We also present qualitative results in Fig. 4.2. DragDiffusion and FreeDrag has difficulty in accurately dragging corresponding contents to designated target locations. Although DragonDiffusion and DiffEditor can better recognize handle points and achieve more precise point movement, they tend to introduce artifacts, thereby reducing image quality. In contrast, LucidDrag demonstrates superior drag control by understanding potential intentions and providing semantic guidance. Furthermore, LucidDrag generates images with greater fidelity due to the explicit quality guidance it provides.

Object moving We also conduct experiments on the object moving task, which can be seen as a special task of drag-style manipulation [45]. We compare our method with DragonDiffusion [45] and DiffEditor [46]. Following DragonDiffusion [45], we select 20 editing samples as the test set. We calculate the CLIP distance between the edited results and the target description. Besides, inspired by [72], we utilize Large Multimodal Models to evaluate the overall performance, denoted as the LMM score. The results are shown in Tab. 2. Our approach achieves higher CLIP scores and LMM scores, demonstrating the promising performance of our method.
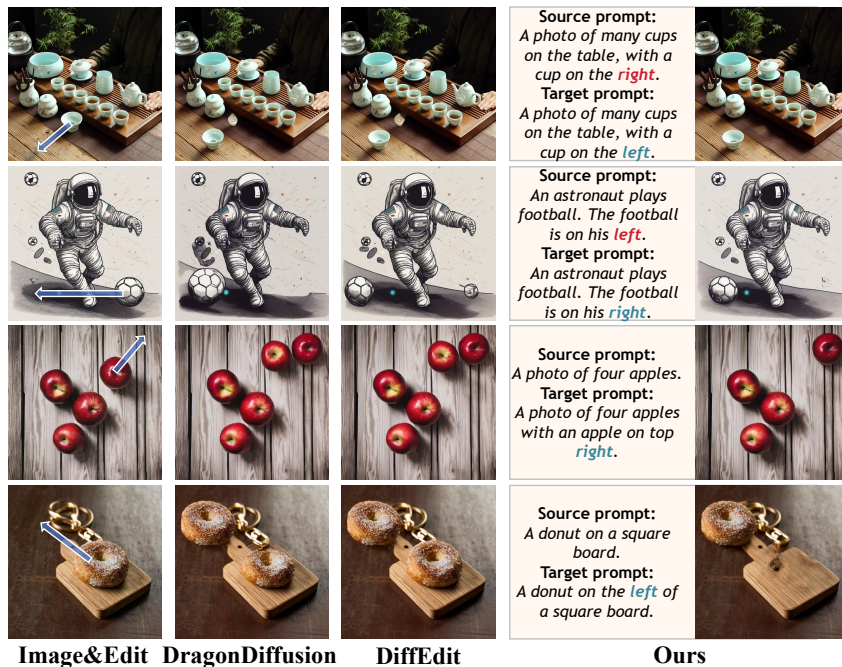
Figure 4: Qualitative comparison between our LucidDrag and other methods in object moving.

8

Qualitative comparisons of the object moving task are shown in Fig. 4. Although the comparison methods can precisely recognize and generate objects in the target locations, our method shows better performance in image quality benefiting from our novel framework. On the one hand, the quality guidance constrains the distribution of the generated images, avoiding artifacts and unexpected changes during generation. For example, in the case of the water cup on the lower left, the human hands of comparison methods are deformed. Differently from prior work, our method achieves better image fidelity. On the other hand, the intention predicted by our intention reasoner can provide semantic guidance, thus preventing the object from reappearing in the source location. For example, in the case of the doughnut on the lower right, the doughnut of comparison methods still appears in its original position. Conversely, our method successfully moves the doughnut to the target position.

Table 2: Compairisons of object moving.

	DragonDiffusion	DiffEdit	Ours
CLIP-score (↑)	0.255	0.257	0.260
LMM-score (↑)	6.25	6.5	7.25

4.3 Ablation Study

Intention planner LucidDrag introduces the intention reasoner for intention understanding. In one respect, this module enhances semantic-aware capability during image editing. In another respect, it provides semantic guidance during generation, prompting accurate dragging of handle points to target positions. To substantiate our claims, we present an ablation study of the intention reasoner. We present qualitative experiments in Fig. 5. For example, one possible semantic intention of the case in the first row of Fig. 5 is to reduce the size of the wheel. When the intention reasoner is removed, *i.e.*, *w/o Intention*, the model has difficulty in understanding the semantic intention, limiting its dragging ability. Alternatively, our approach provides a strong semantic understanding of intentions, enabling precise dragging of objects to their target positions. Quantitative results are presented in Tab 3. Employing the Intention planner yields a 3.20% performance improvement in Mean Distance and a 0.61% performance improvement in GScore, further demonstrating the necessity of semantic understanding and the effectiveness of the intention planner.

Quality guidance One of the major differences between our LucidDrag and the previous [45, 50, 46, 53] is that we explicitly introduce quality guidance to improve the image quality of the edited image. To verify the effectiveness of quality guidance, ablation studies are conducted both qualitatively and quantitatively. Specifically, we denote the ablation study of removing quality guidance as *w/o Quality*. As depicted in Fig. 5, removing quality guidance leads to reduced image quality, evidenced by unexpected changes in the front wheel of the bicycle (the red iron bracket) and artifacts in the spinning top. Ablating reasoned intention reduces the model's ability to perceive the target shape. For example, after removing the intention reasoner, there are some distortions of the structure. Additionally, Tab. 3 demonstrates that ablating quality guidance not only degrades image quality but also impacts edit precision. This may because of the fact that the unexpected artifacts hinder the dragging performance.

Image & Edit Full Implementation w/o Quality Guidance w/o Intention Reasoner

Figure 5: Visualization of ablation study.
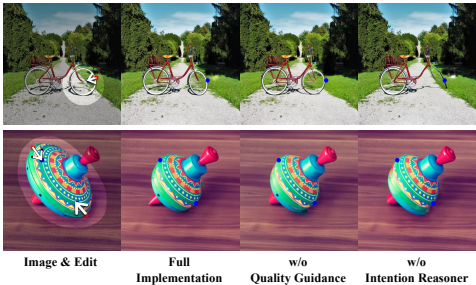
Table 3: Quantitative result of ablation study.

	Full Implementation (Ours)	w/o Intention Reasoner	w/o Quality Guidance
Mean Distance (↓)	**20.46** ± 0.77	23.66 ± 0.73	22.13 ± 0.64
GScore (↑)	**7.37**	6.76	6.47

9

5 Conclusion

In this work, we identify the limitations of previous drag editing models in understanding semantic intentions and generating high-quality edited images. In response, we design a novel framework called LucidDrag, which involves an intention reasoner to clarify possible intentions and a collaborative guidance sampling mechanism that incorporates explicit semantic guidance and quality guidance. LucidDrag excels in: **i)** adequate understanding of semantic intention, improving semantic perception ability and diversity of the generated images; **ii)** enhanced dragging performance, including improved drag accuracy and image quality. Extensive results show the efficiency of our approach and the potential for further advancements in semantic-aware drag-based editing.

6 Limitations

Although our method is capable of achieving semantic-aware drag-based editing without the need for training, there are some limitations. **i)** Complex objects are challenging to drag, and unexpected deformation can occur over long distances. This may be attributed to difficulties in comprehending the intricate nature of the object or inaccurate object tracking. In future work, we will investigate the potential for further improvements in performance by utilizing more powerful image generation models and incorporating comprehensive intention understanding. **ii)** Hyperparameters involved in the editing process will affect the editing results. In future work, we intend to utilize LLM as an agent to automatically determine model hyperparameters to enhance the editing performance.

Acknowledgement This research is sponsored by National Natural Science Foundation of China (Grant No. 62306041), Beijing Nova Program (Grant No. Z211100002121106, 20230484488), CCF-DiDi GAIA Collaborative Research Funds (Grant No. 202414), and Beijing Municipal Science & Technology Commission (Z231100007423015).

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Yuval Alaluf, Omer Tov, Ron Mokady, Rinon Gal, and Amit Bermano. Hyperstyle: Stylegan inversion with hypernetworks for real image editing. In *CVPR*, 2022.

[3] Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *CVPR*, 2023.

[4] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, 2023.

[5] 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. In *ICCV*, 2023.

[6] Minghao Chen, Iro Laina, and Andrea Vedaldi. Training-free layout control with cross-attention guidance. In *WACV*, 2024.

[7] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *URL: https://lmsys.org/blog/2023-03-30-vicuna/*, 2023.

[8] Gabriele Corso, Yilun Xu, Valentin De Bortoli, Regina Barzilay, and Tommi S Jaakkola. Particle guidance: non-iid diverse sampling with diffusion models. In *ICLR*, 2023.

[9] Xing Cui, Zekun Li, Peipei Li, Yibo Hu, Hailin Shi, Chunshui Cao, and Zhaofeng He. Chatedit: Towards multi-turn interactive facial image editing via dialogue. In *EMNLP*, 2023.

[10] Xing Cui, Zekun Li, Pei Pei Li, Huaibo Huang, and Zhaofeng He. Instastyle: Inversion noise of a stylized image is secretly a style adviser. *arXiv preprint arXiv:2311.15040*, 2023.

[11] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. In *NeurIPS*, 2024.

[12] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *NeurIPS*, 2021.

[13] Wenkai Dong, Song Xue, Xiaoyue Duan, and Shumin Han. Prompt tuning inversion for text-driven image editing using diffusion models. In *ICCV*, 2023.

[14] Y Endo. User-controllable latent transformer for stylegan image layout editing. In *Computer Graphics Forum*, 2022.

10

[15] Dave Epstein, Allan Jabri, Ben Poole, Alexei Efros, and Aleksander Holynski. Diffusion self-guidance for controllable image generation. In *NeurIPS*, 2023.

[16] Tsu-Jui Fu, Wenze Hu, Xianzhi Du, William Yang Wang, Yinfei Yang, and Zhe Gan. Guiding instruction-based image editing via multimodal large language models. In *ICLR*, 2024.

[17] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit Haim Bermano, Gal Chechik, and Daniel Cohen-or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *ICLR*, 2022.

[18] Hanan Gani, Shariq Farooq Bhat, Muzammal Naseer, Salman Khan, and Peter Wonka. Llm blueprint: Enabling text-to-image generation with complex and detailed prompts. In *ICLR*, 2023.

[19] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *CACM*, 2020.

[20] Alexandros Graikos, Nikolay Malkin, Nebojsa Jojic, and Dimitris Samaras. Diffusion models as plug-and-play priors. In *NeurIPS*, 2022.

[21] Bohai Gu, Heng Fan, and Libo Zhang. Two birds, one stone: A unified framework for joint learning of image and video style transfers. In *ICCV*, 2023.

[22] Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *CVPR*, 2023.

[23] Wes Gurnee and Max Tegmark. Language models represent space and time. In *ICLR*, 2024.

[24] Cheng Han, James Chenhao Liang, Qifan Wang, MAJID RABBANI, Sohail Dianat, Raghuveer Rao, Ying Nian Wu, and Dongfang Liu. Image translation as diffusion visual programmers. In *ICLR*, 2023.

[25] Amir Hertz, Kfir Aberman, and Daniel Cohen-Or. Delta denoising score. In *ICCV*, 2023.

[26] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-or. Prompt-to-prompt image editing with cross-attention control. In *ICLR*, 2022.

[27] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

[28] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPSW*, 2021.

[29] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019.

[30] Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. Musiq: Multi-scale image quality transformer. In *ICCV*, 2021.

[31] Dongjun Kim, Yeongmin Kim, Se Jung Kwon, Wanmo Kang, and Il-Chul Moon. Refining generative process with discriminator guidance in score-based diffusion models. In *ICML*, 2023.

[32] Mingi Kwon, Jaeseok Jeong, and Youngjung Uh. Diffusion models already have a semantic latent space. In *ICLR*, 2023.

[33] Peipei Li, Yibo Hu, Ran He, and Zhenan Sun. Global and local consistent wavelet-domain age synthesis. *IEEE TIFS*, 2019.

[34] Peipei Li, Huaibo Huang, Yibo Hu, Xiang Wu, Ran He, and Zhenan Sun. Hierarchical face aging through disentangled latent characteristics. In *ECCV*, 2020.

[35] Peipei Li, Rui Wang, Huaibo Huang, Ran He, and Zhaofeng He. Pluralistic aging diffusion autoencoder. In *ICCV*, 2023.

[36] Peipei Li, Xiang Wu, Yibo Hu, Ran He, and Zhenan Sun. M2fpa: A multi-yaw multi-pitch high-quality dataset and benchmark for facial pose analysis. In *ICCV*, 2019.

[37] Long Lian, Boyi Li, Adam Yala, and Trevor Darrell. Llm-grounded diffusion: Enhancing prompt understanding of text-to-image diffusion models with large language models. *arXiv preprint arXiv:2305.13655*, 2023.

[38] Pengyang Ling, Lin Chen, Pan Zhang, Huaian Chen, and Yi Jin. Freedrag: Point tracking is not you need for interactive point-based image editing. In *CVPR*, 2024.

[39] Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B Tenenbaum. Compositional visual generation with composable diffusion models. In *ECCV*, 2022.

[40] Xuannan Liu, Pei Pei Li, Huaibo Huang, Zekun Li, Xing Cui, Weihong Deng, Zhaofeng He, et al. Fka-owl: Advancing multimodal fake news detection through knowledge-augmented lvlms. In *ACM MM*, 2024.

[41] Xuannan Liu, Zekun Li, Peipei Li, Shuhan Xia, Xing Cui, Linzhi Huang, Huaibo Huang, Weihong Deng, and Zhaofeng He. Mmfakebench: A mixed-source multimodal misinformation detection benchmark for lvlms. *arXiv preprint arXiv:2406.08772*, 2024.

[42] Grace Luo, Trevor Darrell, Oliver Wang, Dan B Goldman, and Aleksander Holynski. Readout guidance: Learning control from diffusion features. In *CVPR*, 2024.

[43] Jiafeng Mao, Xueting Wang, and Kiyoharu Aizawa. Guided image synthesis via initial image editing in diffusion model. In *ACM MM*, 2023.

[44] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *CVPR*, 2023.

[45] Chong Mou, Xintao Wang, Jiechong Song, Ying Shan, and Jian Zhang. Dragondiffusion: Enabling drag-style manipulation on diffusion models. In *ICLR*, 2023.

11

[46] Chong Mou, Xintao Wang, Jiechong Song, Ying Shan, and Jian Zhang. Diffeditor: Boosting accuracy and flexibility on diffusion-based image editing. In *CVPR*, 2024.

[47] Hyelin Nam, Gihyun Kwon, Geon Yeong Park, and Jong Chul Ye. Contrastive denoising score for text-guided latent diffusion image editing. In *CVPR*, 2024.

[48] Alexander Quinn 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. In *ICML*, 2022.

[49] OpenAI. Chatgpt. In *URL: https://openai.com/blog/chatgpt*, 2022.

[50] 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 SIG-GRAPH*, 2023.

[51] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

[52] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. In *NeurIPS*, 2022.

[53] 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 *CVPR*, 2024.

[54] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015.

[55] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2020.

[56] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2020.

[57] Qianrui Teng, Rui Wang, Xing Cui, Peipei Li, and Zhaofeng He. Exploring 3d-aware lifespan face aging via disentangled shape-texture representations. In *ICME*, 2024.

[58] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[59] Shanu Vashishtha, Abhinav Prakash, Laliteh Morishetti, Kaushiki Nag, Yokila Arora, Sushant Kumar, and Kannan Achan. Chaining text-to-image and large language model: A novel approach for generating personalized e-commerce banners. *arXiv preprint arXiv:2403.05578*, 2024.

[60] Rui Wang, Hailong Guo, Jiaming Liu, Huaxia Li, Haibo Zhao, Xu Tang, Yao Hu, Hao Tang, and Peipei Li. Stablegarment: Garment-centric generation via stable diffusion. *arXiv preprint arXiv:2403.10783*, 2024.

[61] Zhizhong Wang, Lei Zhao, and Wei Xing. Stylediffusion: Controllable disentangled style transfer via diffusion models. In *ICCV*, 2023.

[62] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, 2022.

[63] Jinheng Xie, Yuexiang Li, Yawen Huang, Haozhe Liu, Wentian Zhang, Yefeng Zheng, and Mike Zheng Shou. Boxdiff: Text-to-image synthesis with training-free box-constrained diffusion. In *ICCV*, 2023.

[64] Serin Yang, Hyunmin Hwang, and Jong Chul Ye. Zero-shot contrastive loss for text-guided diffusion image style transfer. In *ICCV*, 2023.

[65] TaeHo Yoon, Kibeom Myoung, Keon Lee, Jaewoong Cho, Albert No, and Ernest Ryu. Censored sampling of diffusion models using 3 minutes of human feedback. In *NeurIPS*, 2024.

[66] Haoxuan You, Haotian Zhang, Zhe Gan, Xianzhi Du, Bowen Zhang, Zirui Wang, Liangliang Cao, Shih-Fu Chang, and Yinfei Yang. Ferret: Refer and ground anything anywhere at any granularity. *ICLR*, 2024.

[67] Yuqian Yuan, Wentong Li, Jian Liu, Dongqi Tang, Xinjie Luo, Chi Qin, Lei Zhang, and Jianke Zhu. Osprey: Pixel understanding with visual instruction tuning. In *CVPR*, 2024.

[68] Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. In *NeurIPS*, 2024.

[69] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023.

[70] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[71] Shu Zhang, Xinyi Yang, Yihao Feng, Can Qin, Chia-Chih Chen, Ning Yu, Zeyuan Chen, Huan Wang, Silvio Savarese, Stefano Ermon, et al. Hive: Harnessing human feedback for instructional visual editing. *arXiv preprint arXiv:2303.09618*, 2023.

[72] 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.

[73] Jing Zhao, Heliang Zheng, Chaoyue Wang, Long Lan, Wanrong Huang, and Wenjing Yang. Null-text guidance in diffusion models is secretly a cartoon-style creator. In *ACM MM*, 2023.

[74] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

12

A Appendix

This appendix contains additional details for the NeurIPS 2024 submission, titled "Localize, Understand, Collaborate Guide: Semantic-aware Dragging via Intention Reasoner." The appendix is organized as follows:

- §A.1 Additional Preliminaries of Diffusion Model.
- §A.2 Algorithm Pipeline of LucidDrag.
- §A.3 Instruct Prompts for LLM.
- §A.4 More Comparsions.
- §A.5 More Analysis.
- §A.6 Social Impacts.

A.1 Additional Preliminaries of Diffusion Model

DDIM In the inference time of a diffusion model, given an initial noise vector z_T, the noise is gradually removed by sequentially predicting the added noise for T steps. DDIM [55] is one of the efficient denoising approaches that follow a deterministic process, in contrast to the original stochastic diffusion process:

$$z_{t-1} = \sqrt{\frac{\alpha_{t-1}}{\alpha_t}} z_t + \left(\sqrt{\frac{1}{\alpha_{t-1}} - 1} - \sqrt{\frac{1}{\alpha_t} - 1} \right) \cdot \tilde{\varepsilon}_\theta, \tag{13}$$

where $\tilde{\varepsilon}_\theta$ is the estimated noise.

DDIM inversion [55, 44] refers to a technique that aims to transform an input image into a noise vector z_T conditioned on a given prompt or target representation. This process is accomplished by reversing the diffusion process, whereby the final noise sample z_T is returned to the initial z_0.

$$z_{t+1} = \sqrt{\frac{\alpha_{t+1}}{\alpha_t}} z_t + \left(\sqrt{\frac{1}{\alpha_{t+1}} - 1} - \sqrt{\frac{1}{\alpha_t} - 1} \right) \cdot \tilde{\varepsilon}_\theta. \tag{14}$$

Classifier-free guidance For text-based image generation using diffusion models, the classifier-free guidance technique is often employed to address the challenge of amplifying the effect of the text condition. To this end, Ho et al. [28] have presented the classifier-free guidance technique, whereby the prediction is also performed unconditionally and then extrapolated with the conditioned prediction. Specifically, the estimated noise in Eq. 2 is adjusted as follows:

$$\tilde{\varepsilon}_\theta(z_t, t, y, \varnothing) = \varepsilon_\theta(z_t, t, \varnothing) + w \cdot (\varepsilon_\theta(z_t, t, y) - \varepsilon_\theta(z_t, t, \varnothing)), \tag{15}$$

where $\varnothing = \psi("")$ is the embedding of a null text. $\varepsilon_\theta(z_t, t, y)$ represents the conditional predictions. w is the guidance scale parameter.

Classifier guidance The diffusion sampling process can be guided by a variety of energy functions, $g(z_t; t, y)$, which are not limited to probabilities derived from a classifier. Such energy functions may comprise, for instance, the approximate energy derived from another model [39], a similarity score derived from a CLIP model [48], time-independent energy in universal guidance [3], bounding box penalties on attention [6], or any attributes of the noisy images.

The combination of this additional guidance with *"classifier-free guidance"*[28] enables the generation of high-quality text-to-image samples that also possess low energy according to the energy function g:

$$\hat{\epsilon}_t = (1 + s)\epsilon_\theta(z_t; t, y) - s\epsilon_\theta(z_t; t, \emptyset) + \eta \nabla_{z_t} g(z_t; t, y), \tag{16}$$

where s represents the strength of the classifier-free guidance; v is an additional weight for the guidance provided by g. As with classifier guidance, we scale by σ_t to convert the score function to a prediction of ϵ_t. Our work contributes by identifying energy functions g that can be used to control the properties of objects and interactions between them.

13

A.2 Algorithm Pipeline of LucidDrag

To facilitate the understanding of our LucidDrag, we present the entire algorithm pipeline in Algorithm 1. The intention reasoner employs a locator and a reasoner to infer the underlying intentions. Subsequently, collaborative guidance is leveraged to generate images through the integration of semantic guidance, quality guidance, and editing guidance.

Algorithm 1: Proposed LucidDrag

---
1   **Require**:
2   pre-trained SD $\epsilon_{\boldsymbol{\theta}}$; image to be edited x_0; editing guidance steps n_1. quality guidance steps n_2 ($n_2 < n_1$).
3   **Intention Reasoner**:
4   (1) Locatize the interested region, $O = LVLM(x, P)$.
5   (2) Understand the possible intentions, $D = LLM(O, C, P)$.
6   (3) Sample source prompt and target prompt: p_j^s, p_j^t.
7   **Collaborative Guidance**:
8   (1) $z_0 = Encoder(x_0)$ (2) Invert z_0 to z_T^{gud}, $z_{t+1}^{gud} = DDIM\_inversion(z_t^{gud}, p_*^s)$. Then, build the memory bank.
9   (3) Initialize z_T^{gen} with z_T^{gud}.
10   **for** $t = T, \ldots, 1$ **do**
11   noise prediction: $\hat{\epsilon}_t = \epsilon_{\boldsymbol{\theta}}(z_t, t, c, c_{im})$;
12   **if** $T - t < n_1$ **then**
13   compute g_{edit} by Eq. 11;
14   **if** $T - t < n_2$ **then**
15   compute $g_{quality}$ by Eq. 8;
16   compute energy function by Eq. 12;
17   **else**
18   set energy function as g_{edit};
19   inject gradient guidance by Eq. 5;
20   compute z_{t-1} by Eq. 3;
21   $x_0 = Decoder(z_0)$;
22   **Output:** x_0

A.3 Instruct Prompts for LLM

Our approach employs the advanced spatial and reasoning capabilities of LLMs to infer potential intentions. The specific instructions provided in this work are presented below.

Instruct prompt for reasoning potential intentions.

```
You are a helpful assistant. Given the original description and
human-select drag information (including the description of the start point
and the drag direction), reason the intention of the human based on the
drag information, then generate a refined source prompt and target prompt.
The intention of the human is dragging some part of the image to make some
deformation (bigger, shorter, longer, open, closed, etc.) or change its
posture (looking left, looking right, looking sideways, looking closer,
etc.). The source prompt and target prompt should be similar and reflect
the difference before and after dragging. Notably, if the difference
between the source and target prompt is hard to describe, you can directly
set the source prompt and target prompt as the original description. For
example:
example 1:
INPUT:
Original description: a red motorcycle. Drag information 1: The start
point is the behind of the motorcycle. Direction is [[292, 276], [345,
```

14

```
275]].
OUTPUT:
Intention:  making the motorcycle longer.
Source prompt:  a short red motorcycle.
Target prompt:  a long red motorcycle.
example 2:
INPUT:
Original description:  a photo of a raccoon.  drag information 1:  The
start point is the nose of a squirrel.  Direction is [[297, 270], [347,
248]].
OUTPUT:
Intention:  make the raccoon look sideways.
Source prompt:  a photo of a raccoon looking forward.
Target prompt:  a photo of a raccoon looking sideways.
...
```

A.4   More Comparisons

Quantitative comparison   In the primary paper, Tab. 1 presents the Mean Distance and Gscore, averaged over all samples in the DragBench dataset. This supplementary section presents a detailed comparison of the results obtained for each individual category within the DragBench dataset.

The comparison in terms of Mean Distance is presented in Tab. 4, while the comparison in terms of GScore is provided in Tab. 5. As demonstrated in Tab. 4, the proposed LucidDrag method achieves the lowest mean distance, thereby confirming the superiority of our technique in accurately dragging the content to the target position.

About the image fidelity metric as reported in Tab. 5, our LucidDrag approach also demonstrates superior overall average performance. While methods such as DragDiffusion and FreeDrag achieve higher G-scores on some individual classes, this is likely due to their use of additional model fine-tuning to better fit the input image, which can be time-consuming.

Despite comparable image fidelity, the competing methods are deficient in terms of dragging accuracy, as evidenced by the elevated Mean Distance values in Tab. 4. This demonstrates the superiority of the LucidDrag method in maintaining both high-quality image generation and precise content manipulation capabilities.

Table 4: **Comparisons of Dragging Accuracy (Mean Distance) on DragBench (↓).**

	Artworks	Landscape	City	Countryside	Animals	Head	Upper body	Full body	Interior	Other	Average
DragDiffusion	30.74	36.55	27.28	43.21	39.22	36.43	39.75	20.56	24.83	39.52	32.60
FreeDrag	31.16	31.92	28.87	33.57	32.12	39.02	34.46	19.75	23.27	36.11	30.40
DragonDiffusion	26.23	25.42	29.23	34.17	36.13	28.86	48.61	6.97	18.63	37.36	29.16
DiffEditor	24.41	27.52	34.94	38.42	25.62	24.60	24.77	6.81	17.71	35.32	26.01
Ours	**19.71**	**17.76**	**26.74**	**26.67**	**22.74**	**23.87**	**20.70**	**5.56**	**14.34**	**26.54**	**20.46**

Table 5: **Comparisons of Image Fidelity (GScore) on DragBench (↑).**

	Artworks	Landscape	City	Countryside	Animals	Head	Upper body	Full body	Interior	Other	Average
DragDiffusion	6.71	6.19	<u>7.19</u>	**6.91**	<u>7.10</u>	7.70	<u>7.70</u>	7.30	7.15	6.74	6.94
FreeDrag	<u>7.04</u>	**7.24**	6.88	6.37	7.06	**8.28**	**8.18**	<u>7.33</u>	<u>7.49</u>	<u>7.07</u>	<u>7.15</u>
DragonDiffusion	6.09	6.07	6.12	5.47	6.58	6.93	6.42	6.93	7.03	6.09	6.26
DiffEditor	6.90	6.58	6.656	6.58	6.47	6.22	6.32	6.52	6.78	6.58	6.62
Ours	**7.28**	<u>6.95</u>	**7.28**	<u>6.81</u>	**7.14**	<u>7.75</u>	7.54	**8.05**	**8.16**	**7.43**	**7.37**

15

Image&Edit **DragDiffusion** **FreeDrag** **DragonDiffusion** **DiffEditor** **Ours**

Figure 6: More qualitative comparison on content dragging.

16
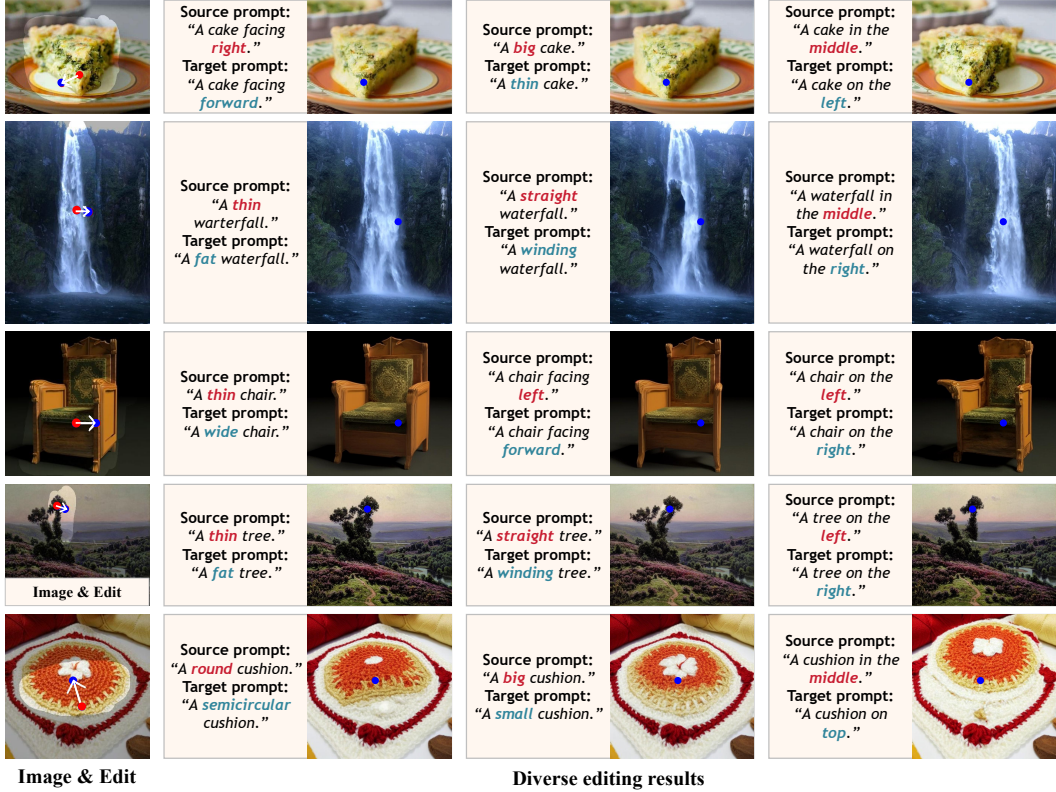
Figure 7: LucidDrag allows generating diverse results.

More Qualitative Results   More comparison results for the content dragging task are presented in Fig. 6. These results, in conjunction with those presented in Fig. 4.2 of the main paper, substantiate the superiority of our method in terms of both drag-based editing performance and image quality maintenance, and also demonstrate the generality of our approach.

Subsequently, we present additional qualitative results that demonstrate the semantic awareness capability of our proposed method. As illustrated in Fig. 7, our approach is capable of generating a diverse range of output images that are highly faithful to the intentions that have been deduced.

A.5   More Analysis

Analysis of different LVLMs and LLMs in Intention Reasoner   We conduct experiments to examine the performance of different LVLMs and LLMs in the Intention Reasoner module. Specifically, we utilize Osprey [67] and Ferret [66] for LVLM and Vicuna [7], LLama3 [58], and GPT 3.5 [49] for LLM. We test various combinations, with Osprey+GPT3.5 being the default setting in our paper. As shown in Table 6, all combinations outperform the experiment without the Intention Reasoner, confirming its reliability. This reliability stems two-fold: the LVLMs are trained with large-scale point-level labeled data and can easily achieve point-level understanding [67]. Therefore, they can understand the user-given points without further fine-tuning. For the LLMs, state-of-the-art LLMs have been proven to possess strong spatial reasoning abilities [23], enabling them to deduce reasonable intentions.

Table 6: Results with different LVLMs and LLMs

	w/o Intention Reasoner	Ferret+ Vicuna	Ferret+ LLama3	Ferret+ GPT3.5	Osprey+ Vicuna	Osprey+ LLama3	Osprey+ GPT3.5 (Ours)
Mean Distance (↓)	23.66	22.49	21.96	20.65	20.84	20.48	**20.46**
GScore (↑)	6.76	7.12	7.11	7.35	7.27	7.13	**7.37**

17

Analysis of the confidence probabilities   We analyze the confidence probabilities of the intention reasoner. As shown in Fig. 8, the confidence probability reflects the quality of the output text in LLM. A higher confidence probability indicates that the intention of the output is more reasonable, leading to better editing results.
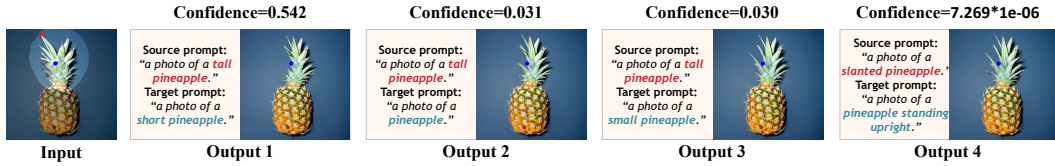
Figure 8: Analysis of confidence probabilities. A higher confidence probability indicates that the intention of the output is more reasonable, leading to better editing results.

Analysis of the guidance   Fig. 9 illustrates the evolution of gradient maps during the drag-based editing task at different time steps. The first row of the figure depicts the gradient maps produced by the editing guidance, g_{edit}, while the second row depicts the gradient maps generated by the quality guidance, $g_{quality}$. The visualizations presented demonstrate a process of gradual convergence. In particular, as the sampling progresses, the activation range of the gradient maps narrows progressively, gradually converging toward the respective editing areas.
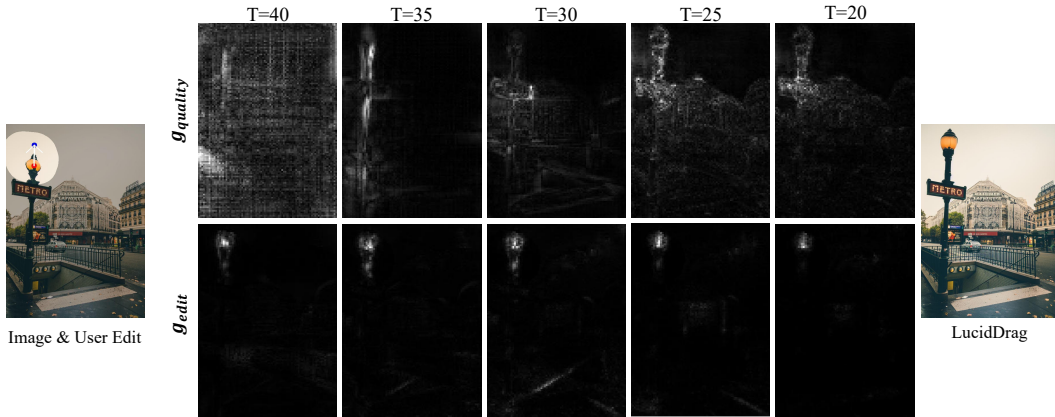
Figure 9: Visualization of the quality guidance an editing guidance.

Fig. 10 presents a visual analysis of the influence of the weight of quality guidance $w_{quality}$ on the image editing results. The figure illustrates that increasing the weights can strengthen the importance of their respective energy functions. However, as a greater degree of editing is more likely to result in distortion or artifacts in the image, there is a trade-off between the editing effects and the image quality. Consequently, in our design, we have set $w_{quality}$ as 1e-3 to avoid excessive constraints from the quality guidance, which could otherwise limit the effectiveness of the image editing process.
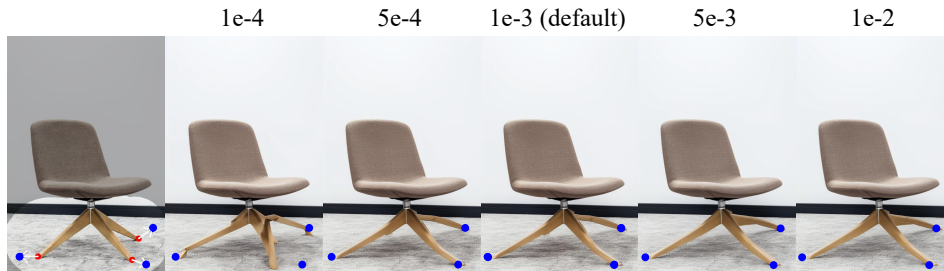
Figure 10: Analysis of the quality guidance weight.

18

The efficiency of different methods   We present the efficiency of different methods in Table 7. Our method has a relatively small inference time and comparable memory requirements.

Table 7: Efficiency of different methods.

	DragDiffusion	**FreeDrag**	**DragonDiffusion**	**DiffEditor**	**Ours**
Time (s) ↓	80	92	30	35	48
Memory (GB) ↓	12.8	13.1	15.7	15.7	15.8

Bad case   As illustrated in Fig. 11, some complex objects are challenging to drag and some unexpected deformation occurs when dragging over long distances. This may be attributed to difficulties in comprehending the intricate nature of the object or inaccurate object tracking.
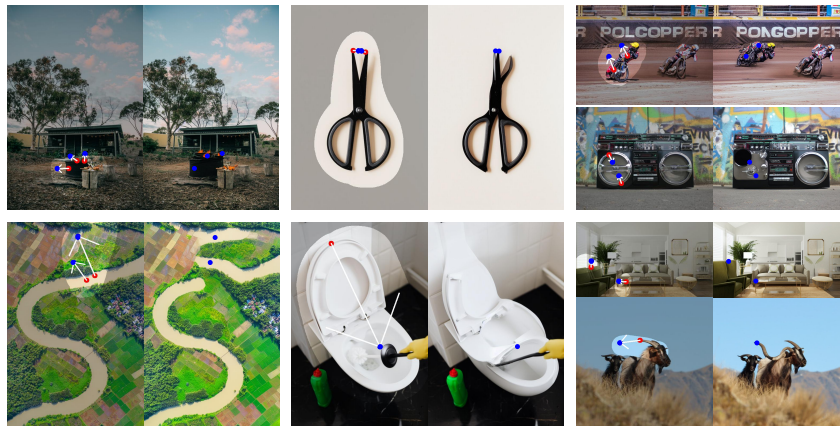
Figure 11: Bad case of our LucidDrag.

A.6   Social Impacts

The emerging LucidDrag technology has promising applications in image editing, content creation, and visual design, inspiring people to create art. However, the use of this technology requires a critical evaluation of potential negative consequences, such as the generation of false or misleading content and potential privacy violations. To address these concerns, the integration of robust illegal content identification models is a viable approach to mitigate these risks and ensure the responsible and ethical use of LucidDrag within the creative landscape.

19

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: In our abstract and introduction, we propose LucidDrag, a model that enables semantic-aware dragging via an LLM intention reasoner and enhances the ability to accurately drag the image with collaborative guidance. Extensive experiments in this paper demonstrate remarkable performance in generating high-quality images with semantic perception.

   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 in the main paper and provide bad cases of our methods corresponding with sufficient analysis in 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 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**

20

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 does not include theoretical results. Our contribution is that we consider the semantics during drag-based editing and design collaborative guidance to generate high-quality images.

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: We provide an algorithm pipeline in Sec. A.2 in the Appendix. Besides, we fully describe the experiment details, including the experimental setup and data set. This ensures that the contributions of the paper can be independently verified by other researchers.

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).

21

(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: [No]

Justification: Due to the time limitation, the code has not been prepared well. We will release it as soon as possible. Besides, researchers can reproduce our method based on the algorithm in Sec. A.2 of the Appendix.

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, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Detailed reporting of the training parameters for the quality discriminator, such as optimizer, learning rate, epochs, and batch size, facilitates replication of experimental results. Similarly, the specification of the key parameters of the diffusion process, including the base model, inversion steps, denoising steps, and classifier guidance strength, allows a thorough understanding of the implementation and allows comparisons with other diffusion-based approaches.

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?

22

Answer: [Yes]

Justification: We provide the mean and standard deviation of quantitative indicators in Tab. 1 and Tab. 4.

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 provide Experiments Compute Resources in Sec. 4.1

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: Our work and research fully comply with the ethical guidelines established by NeurIPS to ensure that the research process and its results have a responsible and controllable impact on society and the environment.

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).

23

10. **Broader Impacts**

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

Answer: [Yes]

Justification: We discuss the social impacts in Sec. A.6 of 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 responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [Yes]

Justification: We utilize the safety checker in the diffusion model to avoid generating unsafe iamges.

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: These assets are publicly released, properly licensed, and can be used for scientific research.

24

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: Although we plan to open source code, our code is not ready yet. We will provide a document with our code when it is released.

    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: Our work does not involve crowdsourcing nor research with 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?

25

Answer: [NA]

Justification: Our work does not involve crowdsourcing nor research with human subjects.

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.

26