In-context Region-based Drag: Drag Any Region to Any Shape

Jiacheng Sui†, Tianyu Hao†, Bingjie Gao, Li Niu*, and Guangtao Zhai

Shanghai Jiao Tong University
jcsui01@sjtu.edu.cn, htianyu429@gmail.com, whynothaha@sjtu.edu.cn,
ustcnewly@sjtu.edu.cn, zhaiguangtao@sjtu.edu.cn

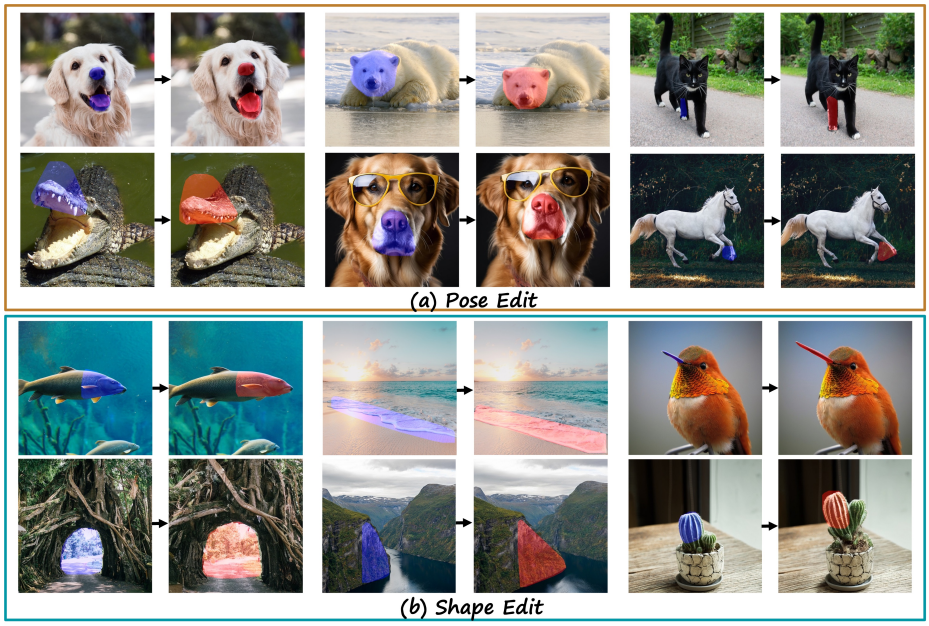

Fig. 1: Region-based Drag aims to transform the source region (blue mask) to align with the target region (red mask). Our In-Context Region-based Drag (ICRDrag) method supports fine-grained geometric editing like pose or shape adjustment.

Abstract. Diffusion models have shown promise in drag-style editing. Previous works mainly focus on point-based drag, which is inherently ambiguous. This paper focuses on region-based drag and introduces a novel In-Context Region-based Drag (ICRDrag) method. Under the in-context learning framework, ICRDrag consumes a source image, a source region mask, and a target region mask, producing the target dragged image. Built upon the basic in-context learning model, we introduce two novel attention regularization: 1) image-mask attention consistency to

† Equal contribution.
* Corresponding author.

ensure that a target region attends to similar source regions for image and mask modalities; 2) source-target attention correspondence to ensure the mutual correspondence between source and target regions. To facilitate region-based drag, we also construct Paired Region Dataset (PRD), a large-scale dataset with paired masks and images. Extensive experiments show that ICRDrag significantly outperforms existing methods in both quantitative metrics and user studies, achieving superior editing accuracy and visual fidelity. The dataset, code, and model are available at `https://github.com/bcmi/ICRDrag-Region-Drag-Editing`.

Keywords: In-context Learning · Diffusion Models · Drag-style Editing

1 Introduction

Diffusion models [15, 21, 22, 37, 49, 56, 60] have achieved remarkable success in diverse image generation and editing tasks, among which drag-style image editing [1, 4, 5, 7, 14, 16, 19, 20, 24, 27, 30, 32, 33, 35, 36, 39, 41, 43, 46–48, 52–54, 57–59, 62, 63, 65, 66] aims to drag partial regions in the image according to user-specified conditions. Based on the dragging condition, dragging task can be categorized into point-based drag and region-based drag.

In point-based drag, users provide pairs of source and target points. The source points in the image are expected to be dragged to the target points. However, this task suffers from inherent ambiguity. As discussed in RegionDrag [32], with limited point pairs, multiple plausible outcomes may exist, which are often misaligned with user intent. For instance, dragging a source point on a face towards a target point out of the face could mean either changing the facial orientation or stretching the face wider. Furthermore, due to the extreme sparsity of point-pair conditions, point-based methods have insufficient editing precision, that is, the source points can hardly be exactly dragged to the target points. Region-based drag [28] addresses such ambiguity by using dense spatial conditions: users provide a source region mask (original location/shape) and a target region mask (desired location/shape), offering denser and more precise control that substantially reduces ambiguity.

Compared with point-based drag, there are very few works on region-based drag. EditGAN [28] performs editing by optimizing the latent code to align with the target mask, but the interaction between image content and mask structure remains shallow — the mask primarily serves as a loss function rather than being deeply integrated into the generation process. RegionDrag [32] adopts a copy-paste strategy in latent space by copying features from the source region and pasting them into the target region guided by the masks. This approach often leads to inconsistent boundaries where the pasted content does not seamlessly blend with the surrounding area, and it struggles with complex shape deformations that require more than simple feature transplantation.

The above limitations motivate us to explore a deeper integration of image and mask information into region-based drag. Recent advances in In-Context

Learning (ICL) [6, 10, 12, 18, 45, 51] have shown that diffusion models exhibit inherent in-context capabilities, opening opportunities to treat structural cues like masks as conditioning context. Inspired by this, we propose In-context Region-based Drag (ICRDrag), built upon a DiT-based foundation model [67]. ICRDrag takes a source image, a source mask, and a target mask as unified context to synthesize the target image in a single forward pass.

Under this framework, we introduce two novel attention regularization. The first one is Image-Mask Attention Consistency (IMAC) regularization. We assume that a target region should attend to similar source regions for image and mask modalities, which enforces that the visual generation process is grounded on the spatial structures defined by the masks. The second one is Source-Target Attention Correspondence (STAC) regularization. We assume that the related regions in the source image and target image should mutually attend to each other, which reinforces the mutual correspondence between source and target regions. Additionally, we propose a novel two-stage training strategy that progressively increases task difficulty: the model first learns from complete region masks, then adapts to incomplete region masks containing only partial editing regions. This curriculum learning approach better simulates real-world sparse user inputs while ensuring stable training.

To the best of our knowledge, there is no large-scale dataset for region-based drag. Therefore, we construct Paired Region Dataset (PRD) from video dataset OpenVid [38], containing paired images and masks. Using SemanticSAM [25] and SAM2 [44], we extract multi-granularity segmentation masks with consistent labels and sample incomplete masks via optical flow. PRD training set contains 287,153 paired samples. Additionally, we construct PRDBench, a benchmark of 1,000 manually verified samples with both mask and point annotations.

We train our model on PRD, and evaluate on PRDBench and DragBench [32], comparing against both region-based and point-based methods. Extensive experiments show ICRDrag significantly outperforms existing methods in editing accuracy, visual realism, and detail preservation. Our contributions are threefold: 1) We propose ICRDrag, an in-context learning framework for region-based drag; 2) We introduce two novel attention regularization and a novel curriculum training strategy; 3) We construct the PRD dataset to advance the research on region-based drag.

2 Related Work

2.1 Point-based Dragging

Existing point-based dragging methods can be categorized into two types based on their editing strategies. Some existing methods perform iterative, step-by-step editing to gradually transform the source point to the target point. Methods like DragGAN [41], DragDiffusion [47], SDE-Drag [40], FreeDrag [29], CLIPDrag [20], StableDrag [9], EasyDrag [16], DragNoise [30], GoodDrag [63], AdaptiveDrag [5], FlowDrag [24], DirectDrag [27] and DragLoRA [54] fall into this type. DragGAN

pioneers motion supervision and point tracking using StyleGAN, while DragDiffusion introduces diffusion models into this task. Subsequent works improve point tracking (e.g., EasyDrag, StableDrag), optimization strategies (e.g., DragNoise, GoodDrag), or enhance semantic understanding (e.g., AdaptiveDrag). Other approaches such as DragonDiffusion [36], DragAPart [26], FastDrag [64], LightningDrag [46], LucidDrag [8], InstantDrag [48], GeoDrag [30], AttentionDrag [58], LazyDrag [59], ContextDrag [14] and Inpaint4Drag [31] edit images in a single forward pass. These methods incorporate innovations like classifier guidance, latent warping functions, large vision-language models, and optical flow prediction for conditioning the editing process.

2.2 Region-based Dragging

EditGAN [28] initiates an interactive paradigm of region-based drag, in which both source and target region masks are employed as control signals. The source region mask indicates the original shape or position of the object to be edited, while the target region mask specifies the desired shape or position after editing. Similar to EditGAN [28], Pixel-Guided Diffusion [34] inherits the interactive setting. Both Pixel-Guided Diffusion and EditGAN perform editing by predicting segmentation masks and optimizing the latent code to align with the target region mask. Differently, RegionDrag [32] adopts a copy-and-paste strategy in the latent space to achieve region-based dragging. Notably, RegionDrag draws inspiration from point-based dragging methods. It first transforms the editing regions into a set of point pairs, and then performs the latent copy-and-paste operation accordingly. More recently, DragFlow [66] proposes a training-free method built upon Diffusion Transformer (DiT) architecture, which addresses the poor performance of conventional point-based supervision on fine-grained DiT features by introducing region-level affine supervision.

3 Problem Definition

The user provides a source image $\boldsymbol{I}_s$ and the corresponding source region mask $\boldsymbol{M}_s$ that specifies the regions to be edited. In this mask, users can specify multiple editing regions, where each one is assigned with a unique label ID. This mask can be obtained using segmentation models such as SAM [23], or drawn by users. The user also provides a target region mask $\boldsymbol{M}_g$, which reflects the modifications to the source mask. In this mask, the label IDs of the editing regions should remain consistent with those in the source region mask. The goal of region-based drag is to generate a target edited image $\boldsymbol{I}_e$ from source image $\boldsymbol{I}_s$, based on the transformation defined by source/target region masks. $\boldsymbol{I}_e$ is expected to approach the ground-truth target image $\boldsymbol{I}_g$.

For notational convenience, we use $\hat{\boldsymbol{I}}_s, \hat{\boldsymbol{M}}_s, \hat{\boldsymbol{I}}_g, \hat{\boldsymbol{M}}_g \in \mathbb{R}^{H\times W\times D}$ to denote the latent codes of $\boldsymbol{I}_s, \boldsymbol{M}_s, \boldsymbol{I}_g, \boldsymbol{M}_g$, in which H, W, D are the height, width, channel number of latent code respectively. The objective is to learn a mapping function

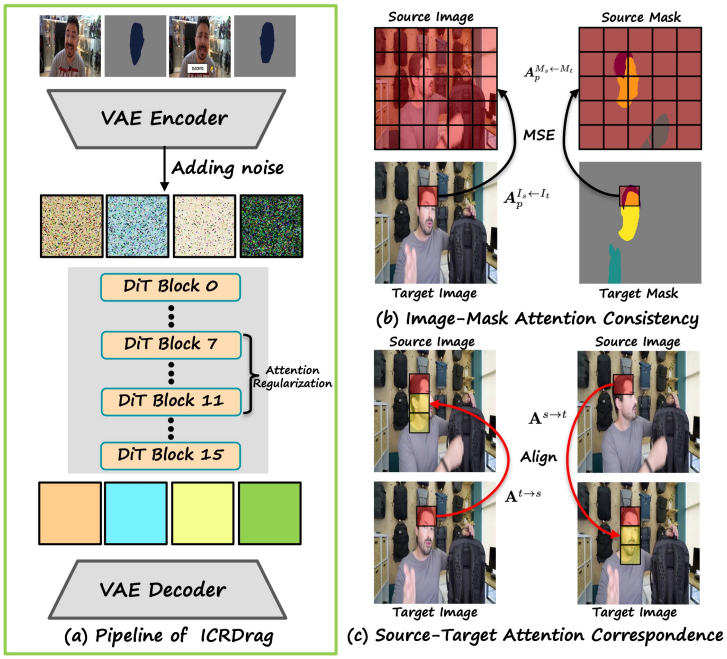

Fig. 2: (a) The overall pipeline of ICRDrag. (b) Image-Mask Attention Consistency. For one patch in the target image, its attention over the source image should mirror the attention of the corresponding patch in the target mask over the source mask. (c) Source-Target Attention Correspondence. If a target patch attends to a source patch, that source patch should also attend back to the same target patch.

$\mathcal{F}$ from $\{\hat{\boldsymbol{I}}_s, \hat{\boldsymbol{M}}_s, \hat{\boldsymbol{M}}_g\}$ to $\hat{\boldsymbol{I}}_g$:

$$\mathcal{F}_{\boldsymbol{\theta}}(\hat{\boldsymbol{I}}_s, \hat{\boldsymbol{M}}_s, \hat{\boldsymbol{M}}_g) : \mathbb{R}^{3\times H\times W\times D} \rightarrow \mathbb{R}^{H\times W\times D}, \tag{1}$$

where $\boldsymbol{\theta}$ represents the model parameters.

4 ICRDrag

In Section 4.1, we will introduce the in-context learning framework for region-based drag. In Section 4.2 and Section 4.3, we will present two novel attention regularization: image-mask attention consistency and source-target attention correspondence. In Section 4.4, we will introduce our two-stage curriculum training strategy.

4.1 Overall Architecture

Our proposed method ICRDrag is model-agnostic and applicable across different DiT architectures. As shown in Figure 2(a), we build our model upon Next-DiT [67] due to its strong generative performance across multiple modalities,

and adapt it to region-based drag. The latent codes $\hat{\boldsymbol{I}}_s, \hat{\boldsymbol{M}}_s, \hat{\boldsymbol{M}}_g$ obtained via VAE encoder serve as conditional inputs and remain noise-free throughout both training and inference. The target image latent $\hat{\boldsymbol{I}}_g$ is the only variable that undergoes noising and denoising. The model predicts a velocity field, which is used to denoise the target latent, and the final denoised latent is passed through the VAE decoder to synthesize the edited image.

Modality-specific LoRAs. Images and masks have intrinsically different properties: images are rich in texture and fine-grained details, while masks are sparse and encode only spatial structure information. Processing them through shared parameters risks feature confusion, that is, the mask's structural representations may contaminate the image pathway, leading to the loss of details and overly smooth generations. To address this issue, we incorporate separate LoRA [17] modules into the feed-forward networks: one for image tokens and another for mask tokens. This allows each modality to learn representations suited to its own properties without cross-modality interference.

Training. At each training iteration, we sample a timestep $t \sim \text{LogNorm}(0, 1)$ [11], along with Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$. The conditional inputs $\{\hat{\boldsymbol{I}}_s, \hat{\boldsymbol{M}}_s, \hat{\boldsymbol{M}}_g\}$ remain in their noise-free state. Only the target image $\hat{\boldsymbol{I}}_g$ is corrupted by adding noise:

$$\hat{\boldsymbol{I}}_g^t = (1-t)\hat{\boldsymbol{I}}_g + t\boldsymbol{\epsilon}. \tag{2}$$

The velocity field for the target image is defined as $\boldsymbol{u} = \hat{\boldsymbol{I}}_g - \boldsymbol{\epsilon}$. The training objective is pushing the predicted velocity field towards the target velocity $\boldsymbol{u}$ via the flow-matching loss:

$$\mathcal{L}_{\text{flow}} = \mathbb{E}\left[\left\|\boldsymbol{v}_{\boldsymbol{\theta}}(t, \hat{\boldsymbol{I}}_s, \hat{\boldsymbol{M}}_s, \hat{\boldsymbol{I}}_g^t, \hat{\boldsymbol{M}}_g) - \boldsymbol{u}\right\|^2\right], \tag{3}$$

where $\boldsymbol{\theta}$ represents the model parameters.

Inference. During inference, our goal is to generate the target edited image $\boldsymbol{I}_e$ conditioned on $\{\boldsymbol{I}_s, \boldsymbol{M}_s, \boldsymbol{M}_g\}$. We initialize the latent code of the target image $\hat{\boldsymbol{I}}_e^T$ with random Gaussian noise at the starting timestep T. The conditional inputs remain noise-free throughout the denoising process. At each denoising step t (from T down to 0), the model predicts the velocity field to update $\hat{\boldsymbol{I}}_e^t$ towards the clean target image. After completing all denoising steps, the final latent $\hat{\boldsymbol{I}}_e^0$ is passed through the VAE decoder to obtain the edited image $\boldsymbol{I}_e$.

4.2 Image-Mask Attention Consistency

We assume that a target region should attend to similar source regions for image and mask modalities, leading to Image-Mask Attention Consistency (IMAC) regularization. Specifically, for each patch in the target image, its attention over the source image to gather visual features should mirror the attention of the

corresponding patch in the target mask over the source mask to understand the spatial structures. Such alignment ensures that the visual generation process is grounded on the spatial structures defined by the masks.

As shown in Figure 2(b), the source and target masks contain the spatial structural information for hair, face, and arm. During model training, it could be easily learned that the hair (*resp.*, face, arm) patch in the target mask should attend to the hair (*resp.*, face, arm) patch in the source mask. By aligning image attention with mask attention, the generation of hair (*resp.*, face, arm) patch in the target image could better gather the visual features from the hair (*resp.*, face, arm) patch in the source image.

Formally, let $\mathcal{P}_g$ denote the set of patches in the target image that lie within the target region mask $\boldsymbol{M}_g$. For each patch $p \in \mathcal{P}_g$, we extract its corresponding token from the target image branch and use it as a query to compute attention over all patches in the source image, yielding the attention map $\boldsymbol{A}_p^{I_s \leftarrow I_g} \in \mathbb{R}^{N_s}$, where N_s is the number of patches in the source image. Similarly, we take the token of the corresponding patch in the target mask branch (*i.e.*, the patch at the same spatial location p) as a query to compute attention over all patches in the source mask, yielding the attention map $\boldsymbol{A}_p^{M_s \leftarrow M_g} \in \mathbb{R}^{N_s}$. We enforce consistency between these two attention maps by minimizing their discrepancy:

$$\mathcal{L}_{\text{imac}} = \sum_{p \in \mathcal{P}_g} \left\| \boldsymbol{A}_p^{I_s \leftarrow I_g} - \boldsymbol{A}_p^{M_s \leftarrow M_g} \right\|^2. \tag{4}$$

4.3 Source-Target Attention Correspondence

We assume that the related regions in the source image and target image should mutually attend to each other, leading to the Source-Target Attention Correspondence (STAC) regularization. As shown in Figure 2(c), if a target patch attends to a source patch, then that source patch should also attend back to the same target patch. Such mutual reinforcement ensures that the model establishes consistent correspondences between the source and target images, which is essential for accurate spatial transformations such as object movement, resizing, and articulation.

Formally, let $\mathcal{P}_g$ be the set of target image patches within the target region mask, and $\mathcal{P}_s$ be the set of source image patches within the source region mask. We consider two attention matrices $\mathbf{A}^{g \to s}$ and $\mathbf{A}^{s \to g}$. $\mathbf{A}^{g \to s} \in \mathbb{R}^{|\mathcal{P}_g| \times |\mathcal{P}_s|}$ denotes the attention from target patches to source patches, where each entry $\mathbf{A}_{ij}^{g \to s}$ is the attention value from target patch i (query) to source patch j. $\mathbf{A}^{s \to g} \in \mathbb{R}^{|\mathcal{P}_s| \times |\mathcal{P}_g|}$ denotes the attention from source patches to target patches, where each entry $\mathbf{A}_{ji}^{s \to g}$ means the attention value from source patch j (query) to target patch i.

We assume that if target patch i strongly attends to source patch j, then source patch j should also strongly attend to target patch i. Such mutual correspondence can be captured by the diagonal entries in the product $\mathbf{A}^{g \to s}\mathbf{A}^{s \to g} \in \mathbb{R}^{|\mathcal{P}_g| \times |\mathcal{P}_g|}$. In particular, for the target patch i, the diagonal entry $(\mathbf{A}^{g \to s}\mathbf{A}^{s \to g})_{ii} = \sum_j \mathbf{A}_{ij}^{g \to s}\mathbf{A}_{ji}^{s \to g}$ sums up the product of its mutual attentions over all source

patches. When maximizing $(\mathbf{A}^{g\to s}\mathbf{A}^{s\to g})_{ii}$, the source patches that the target patch i attends to are enforced to attend back to the target patch i. Therefore, we aim to minimize the following loss function:

$$\mathcal{L}_{\text{stac}} = -\operatorname{Trace}\left(\mathbf{A}^{g\to s}\mathbf{A}^{s\to g}\right), \tag{5}$$

where $\operatorname{Trace}(\cdot)$ denotes the trace operator. Minimizing the negative trace encourages the model to maximize the diagonal entries, thereby enforcing mutual correspondence between source and target patches.

Besides the basic flow-matching loss $\mathcal{L}_{\text{flow}}$ in Eqn. 3, we add two auxiliary losses $\mathcal{L}_{\text{imac}}$ in Eqn. 4 and $\mathcal{L}_{\text{stac}}$ in Eqn. 5, leading to the following total loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{flow}} + \lambda_1\mathcal{L}_{\text{imac}} + \lambda_2\mathcal{L}_{\text{stac}}, \tag{6}$$

where λ_1 and λ_2 are hyper-parameters.

Both auxiliary losses are computed using attention maps extracted from the $7^{th} - 11^{th}$ layers of the transformer (we elaborate on the rationale for selecting these layers to design the losses in Section 6.5), normalized via softmax. For $\mathcal{L}_{\text{imac}}$, we directly compute MSE on two attention maps. For $\mathcal{L}_{\text{stac}}$, we average attention across multiple heads before computing the trace of the product of bidirectional attention matrices.

4.4 Two-stage Curriculum Training Strategy

In real-world applications, the source and target region masks may be very sparse, in which only a few regions are separated out. We refer to the region masks with all semantic regions separated out as complete region mask, and the region masks with partial regions separated out as incomplete region mask (see Section 5 for detailed illustration).

Compared with complete region mask, dragging based on incomplete region masks poses a greater challenge for the model, due to the following reasons. Incomplete region masks contain much sparser and less information than complete region masks. In some scenarios, drag-style editing may imply changes beyond the explicitly separated regions. Therefore, the model must also learn to infer the necessary adjustments in those unseparated regions.

Following the routine of curriculum learning [3,13,42,50], we design a training strategy that progressively increases task difficulty, guiding the model to learn from easy to hard setting. To reduce the difficulty of model training and ensure smoother optimization process, we design a two-stage training strategy.

Training stage 1: We obtain the latent codes of source image $\hat{\boldsymbol{I}}_s$, target image $\hat{\boldsymbol{I}}_g$, source complete region mask $\hat{\boldsymbol{M}}'_s$, and target complete region mask $\hat{\boldsymbol{M}}'_g$ via VAE encoder. The conditional inputs ($\hat{\boldsymbol{I}}_s$, $\hat{\boldsymbol{M}}'_s$, $\hat{\boldsymbol{M}}'_g$) remain noise-free throughout training. Only the target image latent $\hat{\boldsymbol{I}}_g$ is corrupted by adding noise at a randomly sampled timestep t.

Training stage 2: This stage is similar to Training Stage 1 and the key difference lies in region masks. Instead of using complete region masks as input, we

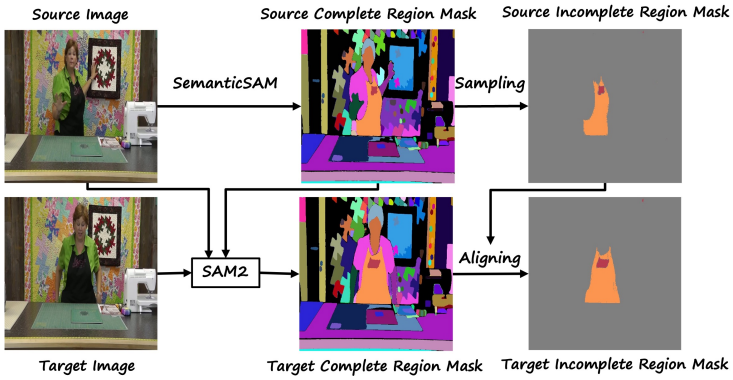

Fig. 3: Paired Region Dataset construction. We leverage SemanticSAM [25] and SAM2 [44] to generate fine-grained segmentation masks. Incomplete region masks are then sampled based on estimated optical flow combined with the watershed algorithm.

construct incomplete region masks $\boldsymbol{M}_s, \boldsymbol{M}_g$ by sampling specific regions from the complete $\boldsymbol{M}'_s, \boldsymbol{M}'_g$. We only retain the sampled regions while filling the remaining regions with gray value. To enhance the model robustness to potentially inaccurate user-provided region masks, we randomly apply dilation to the sampled regions.

5 Paired Region Dataset (PRD)

To the best of our knowledge, there is no existing large-scale dataset tailored to region-based drag. Therefore, we construct the Paired Region Dataset (PRD) consisting of two components: a large-scale training set for model training, and a high-quality benchmark for model evaluation.

Training set construction. We construct the training set based on the OpenVid [38] video dataset, which contains one million high-quality video clips accompanied by expressive captions.

Considering that users may edit an image at different levels of granularity (*e.g.*, coarse-grained face mask as a whole versus fine-grained part masks including ears, nose, and eyes), it is necessary to obtain multi-granularity segmentation masks while ensuring consistency in terms of segmentation labels, that is, the same region should have the same segmentation label across the source and target images. To achieve this goal, we use SemanticSAM [25] to extract multi-granularity segmentation masks from the source image, and then feed both the segmentation mask and the target image into SAM2 [44] to obtain the corresponding segmentation mask for the target image. In this way, we acquire source images $\boldsymbol{I}_s$, source complete region masks $\boldsymbol{M}'_s$, target images $\boldsymbol{I}_g$, and target complete region masks $\boldsymbol{M}'_g$.

To fulfill the requirements of the second-stage training, we need to obtain incomplete region masks. We first compute the optical flow between the source image $\boldsymbol{I}_s$ and the target image $\boldsymbol{I}_g$ using UniMatch [55]. Based on the estimated optical flow, we apply the watershed algorithm to sample 1–5 keypoints as source points. The corresponding target points are then obtained by adding the optical flow vectors to the source points. We extract the regions in the source complete region mask $\boldsymbol{M}_s'$ containing the start points, and those in the target complete region mask $\boldsymbol{M}_g'$ containing the end points, leading to incomplete region masks $\boldsymbol{M}_s, \boldsymbol{M}_g$.

In total, we obtain 287,153 tuples of source image, source region mask, target image, target region mask for the training set.

Benchmark construction. Besides the large-scale training set, a high-quality benchmark is essential for model evaluation. Therefore, we construct a benchmark of 1,000 manually refined and annotated samples through the following procedure.

Data selection: We apply the same pipeline as constructing the training set to the remaining raw data that are not included in the training split. From the processed candidate pairs, we select samples to ensure diversity in object categories, editing types (*e.g.*, position change, resizing, shape deformation, pose adjustment), and complexity levels.

Verification and correction: For each candidate sample, annotators check the validity of each dragging scenario, including stable viewpoint, no object insertion or removal, and reasonable transformation magnitude. They further verify the consistency of source/target points and masks with the intended edit. Samples that do not meet these criteria are discarded or re-annotated.

Keypoint annotation and mask derivation: For samples that require refinement, annotators re-establish the editing correspondences by selecting a set of source points $\{q_s^k|_{k=1}^n\}$ and corresponding target points $\{q_g^k|_{k=1}^n\}$. Each point pair is constrained to share the same segmentation label to maintain semantic consistency. The number of point pairs n is in the range of $[1, 5]$. This design facilitates a fair comparison between point-based and region-based dragging approaches. Region masks are then automatically derived by extracting the regions containing the annotated points from the complete region masks $\boldsymbol{M}_s'$ and $\boldsymbol{M}_g'$, forming the incomplete region masks $\boldsymbol{M}_s$ and $\boldsymbol{M}_g$.

The final benchmark consists of 1,000 high-quality tuples, each containing: (1) source image $\boldsymbol{I}_s$ and target image $\boldsymbol{I}_g$; (2) source region mask $\boldsymbol{M}_s$ and target region mask $\boldsymbol{M}_g$; (3) A set of source points $\{q_s^k|_{k=1}^n\}$ and corresponding target points $\{q_g^k|_{k=1}^n\}$.

6 Experiment

6.1 Experimental Setting

Implementation details. During the training stage 1, we train the model for 60,000 steps with batch size 2 and learning rate 1×10^{-4}. While for training stage

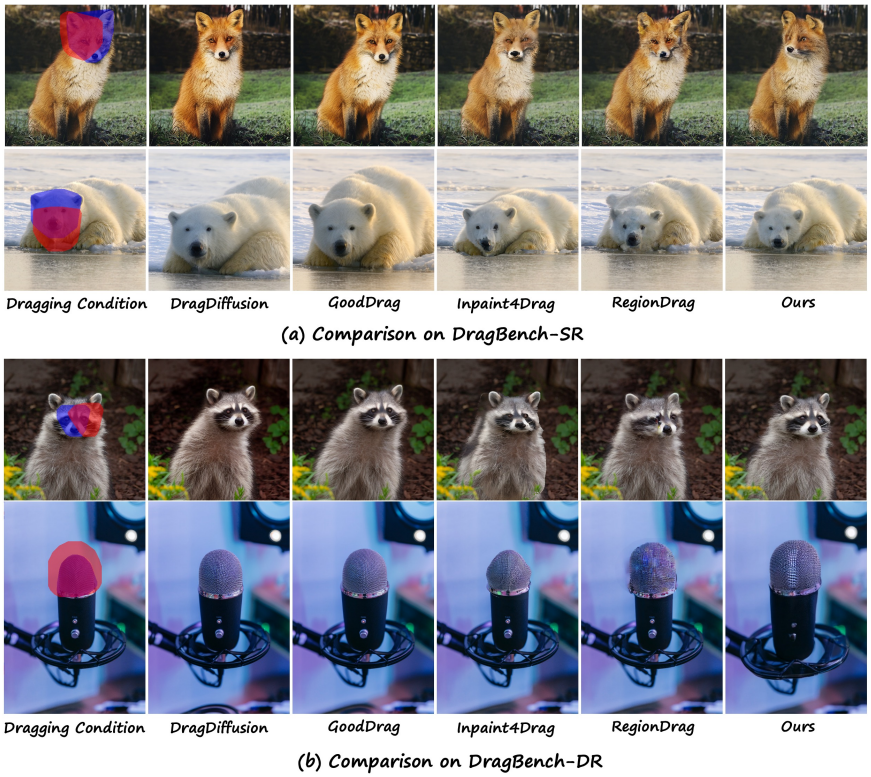

Fig. 4: Qualitative results on DragBench-SR and DragBench-DR [32]. In the "Dragging Condition" column, the blue mask indicates the source region, while the red mask indicates the target region.

2, we train the model for another 2,000 steps, with batch size 1 and learning rate 5×10^{-5}. More implementation details are left to supplementary.

Baseline. We compare our method against both region-based and point-based dragging approaches. For region-based drag, we compare with RegionDrag [32]. While EditGAN [28] and Pixel-Guided Diffusion [34] also fall into this category, they are less suitable for general-domain evaluation. EditGAN is built upon GAN architectures, and Pixel-Guided Diffusion relies on DDPMSegmentation [2]. Their design choices limit their applicability to general-domain images. Therefore, we focus our region-based comparison with those methods designed for general-purpose editing. For point-based dragging, we select representative methods that cover diverse technical approaches: DragDiffusion [47] as a pioneering diffusion-based method, SDE-Drag [40], GoodDrag [63] and DragLoRA [54] for its optimization-based improvements, and DiffEditor [35], FastDrag [64], Inpaint4Drag [31], as recent one-step editing approaches. For training-based baseline DiffEditor [35], we fine-tune it on our PRD training set to ensure fair comparison.

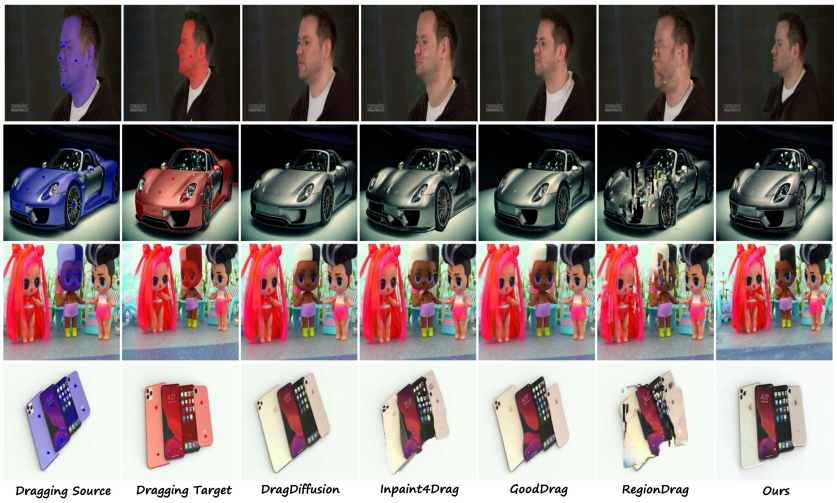

Fig. 5: Visual results on our PRD benchmark.

Table 1: Quantitative analysis on PRD benchmark.

Method	MSE ↓	LPIPS↓	SSIM ↑	MD(RegionDrag)↓	MD(DragLoRA)↓
DragDiffusion [47]	0.0937	0.1836	0.5993	5.17	26.79
SDE-Drag [40]	0.1017	0.2018	0.5849	7.97	44.04
DiffEditor [35]	0.0959	0.1949	0.6071	23.45	31.19
FastDrag [64]	0.0962	0.2049	0.5862	6.01	31.22
Inpaint4Drag [31]	0.0972	0.1923	<u>0.6102</u>	4.24	23.62
GoodDrag [63]	<u>0.0902</u>	<u>0.1761</u>	0.6094	<u>3.70</u>	**18.01**
DragLoRA [54]	0.0933	0.1870	0.5974	4.62	23.96
RegionDrag [32]	0.0977	0.1944	0.6076	8.02	43.05
ICRDrag	**0.0735**	**0.1610**	**0.6284**	**3.66**	<u>22.34</u>

Dataset and metrics. We train our model on our Paired Region Dataset (PRD). We evaluate our proposed method on PRD benchmark and DragBench [32], which includes DragBench-DR and DragBench-SR. Note that PRD benchmark has ground-truth images while the other two do not have. On PRD benchmark, we adopt LPIPS [61], SSIM, MSE, and Mean Distance (MD) to measure the difference between editing result and ground-truth. On DragBench-DR and DragBench-SR, we conduct user study from three aspects: realism, fidelity, and region accuracy.

6.2 Experimental Result

Quantitative analysis. We quantitatively compare our method with existing drag-style editing methods on PRD benchmark. As shown in Table 1, our method

Table 2: User study results on DragBench.

Method	Realism↑	Fidelity↑	Region Accuracy↑
GoodDrag [63]	0.2876	0.2120	0.1276
RegionDrag [32]	0.2442	0.2314	0.3518
ICRDrag	**0.4682**	**0.5566**	**0.5206**

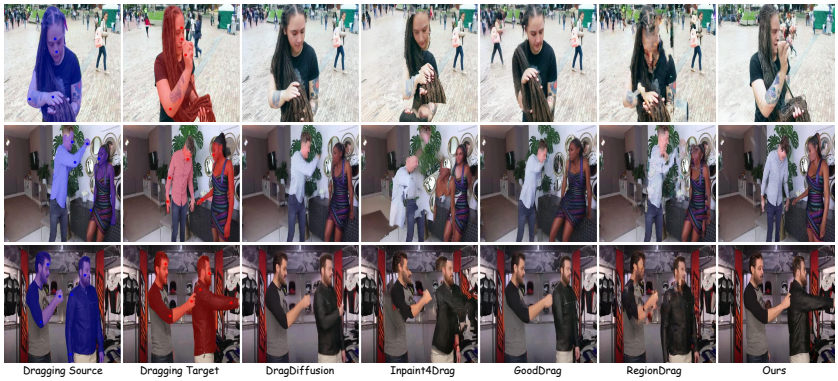

Fig. 6: Comparison with baselines on hard cases involving large topology changes, occlusion, and human limb repositioning.

consistently outperforms existing methods across most metrics. It achieves lower LPIPS and higher SSIM scores, indicating better visual fidelity and detail preservation. Additionally, lower MSE and lower MD indicate more accurate and controllable editing.

Qualitative analysis. Figure 4 compares our method ICRDrag with point-based and region-based methods on DragBench. ICRDrag achieves superior performance in editing accuracy, realism, artifact reduction, and global consistency. On the more challenging PRD dataset (Figure 5), ICRDrag demonstrates robust results across complex cases. In the third example of Figure 5, point-based methods introduce ambiguity by dragging the person toward the left side of the image, while region-based drag better aligns with the editing objective. Moreover, GoodDrag, a representative point-based method, exhibits noticeable degradation in editing fidelity and realism in complex scenes. In contrast, our ICRDrag consistently maintains high accuracy and visual quality. Additional hard-case examples and cross-dataset transfer results are provided in supplementary.

User study. We obtain images from DragBench and present original image, editing region masks, and the edited results from GoodDrag, RegionDrag and ICRDrag to 50 users. Participants were asked to choose the best result in terms of realism, fidelity, and region accuracy. The percentage of preferred results is summarized in Table 2. Since GoodDrag can only take point pairs as input, its region accuracy is much worse than other methods.

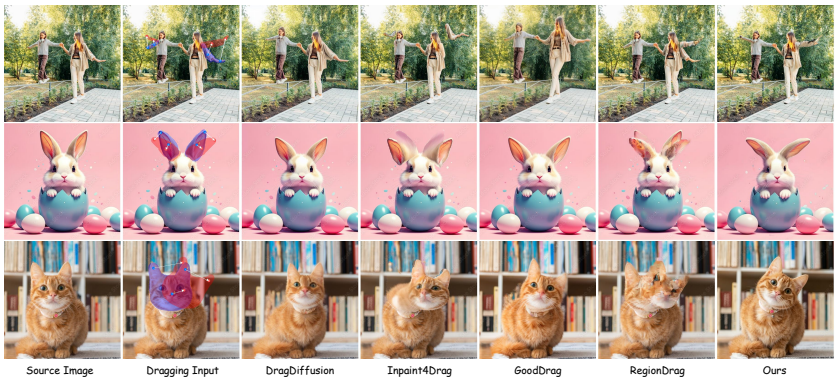

Fig. 7: Comparison with baselines on cross-dataset transfer examples collected from Adobe Stock.

Fig. 8: Visual results of ablation studies on our IMAC and STAC losses.

6.3 Hard Cases and Cross-dataset Transfer

We further provide qualitative comparisons on challenging non-rigid editing scenarios and out-of-distribution images, covering cases beyond simple translation or scaling. As shown in Figure 6, ICRDrag handles hard cases involving large topology changes, partial occlusion, and human limb repositioning. Figure 7 shows cross-dataset transfer results on Adobe Stock images. ICRDrag follows the source-target region masks and produces plausible edits, demonstrating its generalization beyond PRD and DragBench.

6.4 Ablation Study

To evaluate Image-Mask Attention Consistency (IMAC) loss and Source-Target Attention Correspondence (STAC) loss, we conduct ablation studies by enabling each loss separately during training. Figure 8 shows visual results on the PRD benchmark. Without IMAC, the model fails to accurately align edits with the target mask, causing structural distortions or boundary misalignment. Without STAC, the model struggles to preserve fine-grained source details, such as textures, patterns, and identity-specific features. The full model with both losses

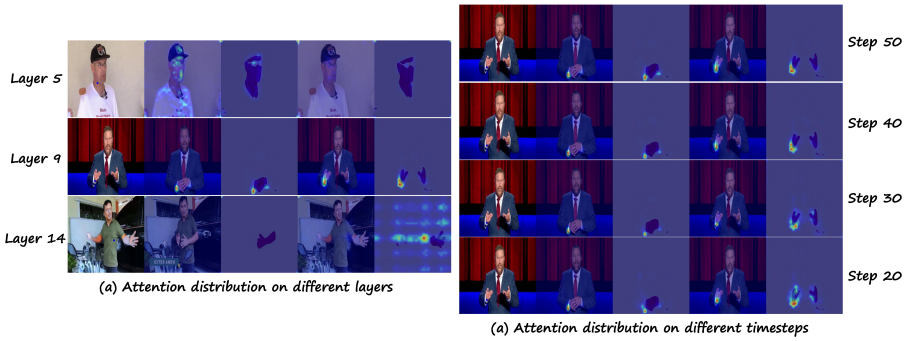

Fig. 9: (a) Visualization of attention maps for a target patch across different transformer layers (NextDiT has 16 layers in total). (b) Attention maps from a middle transformer layer at different denoising timesteps.

achieves the best results, producing edits that are both mask-aligned and visually faithful. Quantitative ablation results are provided in the supplementary material.

6.5 Analysis of Attention Map

We further analyze the attention maps of ICRDrag across transformer layers and denoising timesteps, as shown in Figure 9. The attention maps reveal a progressive transition from source-content localization in early layers, to source-target context integration in middle layers, and finally to spatial refinement in late layers. They also remain relatively stable through the whole denoising process, suggesting that source-target and image-mask correspondences are established early and consistently guide generation. These observations motivate applying IMAC and STAC losses to the middle transformer layers (7–11) and activating them through the whole denoising process.

7 Conclusion

In this paper, we proposed In-context Region-based Drag (ICRDrag), an in-context learning framework for region-based drag equipped with Image-Mask Attention Consistency (IMAC) and Source-Target Attention Correspondence (STAC) regularization. We also constructed the Paired Region Dataset (PRD) to support research on drag-style editing. Experiments show that ICRDrag achieves superior accuracy, detail preservation, and realism, demonstrating the potential of large-scale in-context training for drag-style editing.

Acknowledgements

The work was supported by the National Natural Science Foundation of China (Grant No. 62471287).

References

1. Avrahami, O., Gal, R., Chechik, G., Fried, O., Lischinski, D., Vahdat, A., Nie, W.: Diffuhaul: A training-free method for object dragging in images. In: ACM SIGGRAPH Asia (2024)
2. Baranchuk, D., Voynov, A., Rubachev, I., Khrulkov, V., Babenko, A.: Label-efficient semantic segmentation with diffusion models. In: ICLR (2022)
3. Bengio, Y., Louradour, J., Collobert, R., Weston, J.: Curriculum learning. In: ICML (2009)
4. Cai, P., Liu, Z., Zhu, G., Niu, Y., Wang, J.: Auto draggan: Editing the generative image manifold in an autoregressive manner. In: ACMMM (2024)
5. Chen, D., Chen, B., Geng, Y., Bo, L.: Adaptivedrag: Semantic-driven dragging on diffusion-based image editing. arXiv preprint arXiv:2410.12696 (2024)
6. Chen, M., Du, J., Pasunuru, R., Mihaylov, T., Iyer, S., Stoyanov, V., Kozareva, Z.: Improving in-context few-shot learning via self-supervised training. In: ACL (2022)
7. Choi, G., Jeong, T., Hong, S., Hwang, S.J.: Dragtext: Rethinking text embedding in point-based image editing. In: WACV (2025)
8. Cui, X., Li, P.P., Li, Z., Liu, X., Zou, Y., He, Z.: Localize, understand, collaborate: Semantic-aware dragging via intention reasoner. In: NeurIPS (2024)
9. Cui, Y., Zhao, X., Zhang, G., Cao, S., Ma, K., Wang, L.: Stabledrag: Stable dragging for point-based image editing. In: ECCV (2025)
10. Dong, Q., Li, L., Dai, D., Zheng, C., Ma, J., Li, R., Xia, H., Xu, J., Wu, Z., Chang, B., Sun, X., Li, L., Sui, Z.: A survey on in-context learning. In: EMNLP (2024)
11. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al.: Scaling rectified flow transformers for high-resolution image synthesis. In: ICML (2024)
12. Gu, Y., Dong, L., Wei, F., Huang, M.: Pre-training to learn in context. In: ACL (2023)
13. Hacohen, G., Weinshall, D.: On the power of curriculum learning in training deep networks. In: ICML (2019)
14. He, H., Yan, P., Yi, Z., Zhong, W., Liu, Z., Tang, Y., Yang, H., Gai, K., Li, G., Jin, L.: Contextdrag: Precise drag-based image editing via context-preserving token injection and position-consistent attention. arXiv preprint arXiv:2512.08477 (2025)
15. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: NeurIPS (2020)
16. Hou, X., Liu, B., Zhang, Y., Liu, J., Liu, Y., You, H.: Easydrag: Efficient point-based manipulation on diffusion models. In: CVPR (2024)
17. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al.: Lora: Low-rank adaptation of large language models. In: ICLR (2022)
18. Huang, L., Wang, W., Wu, Z.F., Shi, Y., Dou, H., Liang, C., Feng, Y., Liu, Y., Zhou, J.: In-context lora for diffusion transformers. arXiv preprint arXiv:2410.23775 (2024)
19. Jiang, Z., Wang, Z., Chen, L.: CLIPDrag: Combining text-based and drag-based instructions for image editing. In: ICLR (2025)
20. Jiang, Z., Wang, Z., Chen, L.: Clipdrag: Combining text-based and drag-based instructions for image editing. In: ICLR (2025)
21. Karnewar, A., Vedaldi, A., Novotny, D., Mitra, N.J.: Holodiffusion: Training a 3d diffusion model using 2d images. In: CVPR (2023)

22. Kim, G., Kwon, T., Ye, J.C.: Diffusionclip: Text-guided diffusion models for robust image manipulation. In: CVPR (2022)
23. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. In: ICCV (2023)
24. Koo, G., Yoon, S., Lee, Y., Hong, J.W., Yoo, C.D.: Flowdrag: 3d-aware drag-based image editing with mesh-guided deformation vector flow fields. In: ICML (2025)
25. Li, F., Zhang, H., Sun, P., Zou, X., Liu, S., Yang, J., Li, C., Zhang, L., Gao, J.: Semantic-sam: Segment and recognize anything at any granularity. arXiv preprint arXiv:2307.04767 (2023)
26. Li, R., Zheng, C., Rupprecht, C., Vedaldi, A.: Dragapart: Learning a part-level motion prior for articulated objects. In: ECCV (2024)
27. Liao, S.H., Chen, S.F., Huang, T.M., Cheng, W.H., Hua, K.L.: Directdrag: High-fidelity, mask-free, prompt-free drag-based image editing via readout-guided feature alignment. In: WACV (2026)
28. Ling, H., Kreis, K., Li, D., Kim, S.W., Torralba, A., Fidler, S.: Editgan: High-precision semantic image editing. In: NeurIPS (2021)
29. Ling, P., Chen, L., Zhang, P., Chen, H., Jin, Y.: Freedrag: Point tracking is not you need for interactive point-based image editing. arXiv preprint arXiv:2307.04684 (2023)
30. Liu, H., Xu, C., Yang, Y., Zeng, L., He, S.: Drag your noise: Interactive point-based editing via diffusion semantic propagation. In: CVPR (2024)
31. Lu, J., Han, K.: Inpaint4drag: Repurposing inpainting models for drag-based image editing via bidirectional warping. In: ICCV (2025)
32. Lu, J., Li, X., Han, K.: Regiondrag: Fast region-based image editing with diffusion models. In: ECCV (2024)
33. Luo, G., Darrell, T., Wang, O., Goldman, D.B., Holynski, A.: Readout guidance: Learning control from diffusion features. In: CVPR (2024)
34. Matsunaga, N., Ishii, M., Hayakawa, A., Suzuki, K., Narihira, T.: Fine-grained image editing by pixel-wise guidance using diffusion models. AI for Content Creation workshop at CVPR (2022)
35. Mou, C., Wang, X., Song, J., Shan, Y., Zhang, J.: Diffeditor: Boosting accuracy and flexibility on diffusion-based image editing. CVPR (2024)
36. Mou, C., Wang, X., Song, J., Shan, Y., Zhang, J.: Dragondiffusion: Enabling drag-style manipulation on diffusion models. In: ICLR (2024)
37. Mou, C., Wang, X., Xie, L., Wu, Y., Zhang, J., Qi, Z., Shan, Y., Qie, X.: T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. arXiv preprint arXiv:2302.08453 (2023)
38. Nan, K., Xie, R., Zhou, P., Fan, T., Yang, Z., Chen, Z., Li, X., Yang, J., Tai, Y.: Openvid-1m: A large-scale high-quality dataset for text-to-video generation. In: ICLR (2025)
39. Nguyen, T., Ojha, U., Li, Y., Liu, H., Lee, Y.J.: Edit one for all: Interactive batch image editing. In: CVPR (2024)
40. Nie, S., Guo, H.A., Lu, C., Zhou, Y., Zheng, C., Li, C.: The blessing of randomness: SDE beats ODE in general diffusion-based image editing. In: ICLR (2024)
41. Pan, X., Tewari, A., Leimkühler, T., Liu, L., Meka, A., Theobalt, C.: Drag your gan: Interactive point-based manipulation on the generative image manifold. In: ACM SIGGRAPH (2023)
42. Pentina, A., Sharmanska, V., Lampert, C.H.: Curriculum learning of multiple tasks. In: CVPR (2015)
43. Pu, X., Wang, H., Gui, J., Zhou, P.: Dragging with geometry: From pixels to geometry-guided image editing. In: ICLR (2026)

44. Ravi, N., Gabeur, V., Hu, Y.T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., Mintun, E., Pan, J., Alwala, K.V., Carion, N., Wu, C.Y., Girshick, R., Dollár, P., Feichtenhofer, C.: Sam 2: Segment anything in images and videos. arXiv preprint arXiv:2408.00714 (2024)
45. Shi, W., Min, S., Lomeli, M., Zhou, C., Li, M., Lin, X.V., Smith, N.A., Zettlemoyer, L., tau Yih, W., Lewis, M.: In-context pretraining: Language modeling beyond document boundaries. In: ICLR (2024)
46. Shi, Y., Liew, J.H., Yan, H., Tan, V.Y.F., Feng, J.: Lightningdrag: Lightning fast and accurate drag-based image editing emerging from videos. In: ICML (2025)
47. Shi, Y., Xue, C., Liew, J.H., Pan, J., Yan, H., Zhang, W., F. Tan, V.Y., Bai, S.: Dragdiffusion: Harnessing diffusion models for interactive point-based image editing. In: CVPR (2024)
48. Shin, J., Choi, D., Park, J.: InstantDrag: Improving Interactivity in Drag-based Image Editing. In: ACM SIGGRAPH Asia (2024)
49. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: ICML (2015)
50. Wang, X., Chen, Y., Zhu, W.: A survey on curriculum learning. TPAMI **44**(9), 4555–4576 (2021)
51. Wang, Z., Jiang, Y., Lu, Y., He, P., Chen, W., Wang, Z., Zhou, M., et al.: In-context learning unlocked for diffusion models. In: NeurIPS (2023)
52. Wang, Z., Peng, D., Chen, F., Yang, Y., Lei, Y.: Training-free dense-aligned diffusion guidance for modular conditional image synthesis. In: CVPR (2025)
53. Xia, B., Zhang, Y., Li, J., Wang, C., Wang, Y., Wu, X., Yu, B., Jia, J.: Dreamomni: Unified image generation and editing. arXiv preprint arXiv:2412.17098 (2024)
54. Xia, S., Sun, L., Sun, T., Li, Q.: Draglora: Online optimization of lora adapters for drag-based image editing in diffusion model. In: ICML (2025)
55. Xu, H., Zhang, J., Cai, J., Rezatofighi, H., Yu, F., Tao, D., Geiger, A.: Unifying flow, stereo and depth estimation. TPAMI (2023)
56. Xu, X., Wang, Z., Zhang, G., Wang, K., Shi, H.: Versatile diffusion: Text, images and variations all in one diffusion model. In: ICCV (2023)
57. Yan, Z., Ma, Y., Zou, C., Chen, W., Chen, Q., Zhang, L.: Eedit: Rethinking the spatial and temporal redundancy for efficient image editing. In: ICCV (2025)
58. Yang, B., Huang, M., Zhang, Y., Xiong, Y., Zhou, K., Chen, X., Zhou, S., Bao, H., Li, C., Shi, F., Liu, H.: Attentiondrag: Exploiting latent correlation knowledge in pre-trained diffusion models for image editing. In: IJCAI (2025)
59. Yin, Z., Dai, X., Wang, D., Zeng, X., Ni, L., YU, G., Shum, H.Y.: Lazydrag: Enabling stable drag-based editing on multi-modal diffusion transformers via explicit correspondence. In: ICLR (2026)
60. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: ICCV (2023)
61. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018)
62. Zhang, Y., Zhou, X., Zeng, Y., Xu, H., Li, H., Zuo, W.: Framepainter: Endowing interactive image editing with video diffusion priors. In: ICCV (2025)
63. Zhang, Z., Liu, H., Chen, J., Xu, X.: Gooddrag: Towards good practices for drag editing with diffusion models. In: ICLR (2025)
64. Zhao, X., Guan, J., Fan, C., Xu, D., Lin, Y., Pan, H., Feng, P.: Fastdrag: Manipulate anything in one step. In: NeurIPS (2024)
65. Zhou, Y., Zhou, J., Xu, Q., Zhao, K., Wang, Y., Fei, H., Hong, R., Zhang, H.: Dragnext: Rethinking drag-based image editing. In: AAAI (2025)

66. Zhou, Z., Lu, S., Leng, S., Zhang, S., Lian, Z., Yu, X., Kong, A.W.K.: Dragflow: Unleashing dit priors with region-based supervision for drag editing. In: ICLR (2026)
67. Zhuo, L., Du, R., Xiao, H., Li, Y., Liu, D., Huang, R., Liu, W., Zhu, X., Wang, F.Y., Ma, Z., Luo, X., Wang, Z., Zhang, K., Zhao, L., Liu, S., Yue, X., Ouyang, W., Qiao, Y., Li, H., Gao, P.: Lumina-next : Making lumina-t2x stronger and faster with next-dit. In: NeurIPS (2024)