KRIDRAG: SPATIALLY AUTOCORRELATED CONFLICT-FREE DRAG-BASED IMAGE EDITING

Anonymous authors
Paper under double-blind review

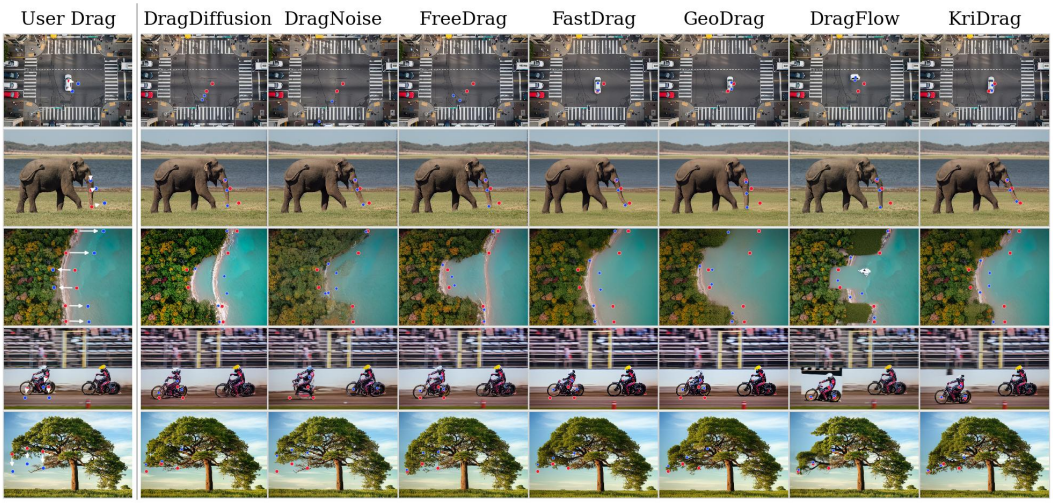

Figure 1: **KriDrag.** For drag-based image editing, user drags are shown by white arrows from red handles to blue targets. Compared with DragDiffusion (Shi et al., 2024), DragNoise (Liu et al., 2024), FreeDrag (Ling et al., 2024), FastDrag (Zhao et al., 2024), GeoDrag (Pu et al., 2026), and DragFlow (Zhou et al., 2026b), our KriDrag enables coherent image editing that more faithfully reflects the user's intended spatial manipulations.

ABSTRACT

Drag-based image editing provides an intuitive interface for spatial image editing through a small number of user-specified point movements. Existing methods primarily rely on interpolation to construct the displacement field, paying insufficient attention to the role of explicit spatial modeling. Moreover, the displacement fields induced by multiple drags are directly aggregated, often overlooking their mutual conflicts. In this work, we introduce KriDrag, a spatially autocorrelated and conflict-free framework for drag-based image editing. KriDrag constructs a Covariance-Guided Displacement Field, where each displacement influences its surrounding positions through spatial autocorrelation. For multiple drags, KriDrag constructs a continuous and smooth displacement field that preserves coherent motion across neighboring regions while suppressing conflicting displacements between drags. Experiments on DragBench and ReD-Bench demonstrate the efficiency and effectiveness of our method for drag-based image editing, particularly on multi-drag benchmarks.

1 INTRODUCTION

With the rapid development of text-to-image models, image editing has become increasingly important, as the images initially generated from text prompts may not fully satisfy users' requirements, necessitating further interactive refinement. While text-driven image editing (Kulikov et al., 2025)

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

enables semantic manipulation, it provides limited and indirect control over spatial changes, making fine-grained geometric editing less intuitive.

Drag-based image editing (Zhao et al., 2024) provides a direct and intuitive interface for spatial manipulation. Users specify one or more handle-target pairs to indicate the desired displacement, and image editing model moves the corresponding image content toward the target locations while preserving the surrounding background. Early approaches typically perform iterative latent optimization (Shi et al., 2024), feature tracking (Pan et al., 2023; Ling et al., 2024; Cui et al., 2024), or semantic propagation (Liu et al., 2024) to realize point-based manipulation. Subsequent methods improve editing efficiency through direct latent manipulation (Zhao et al., 2024), learned conditional generation (Shin et al., 2024; Shi et al., 2025), and reusing intermediate diffusion representations (Hou et al., 2024; Yang et al., 2025). Other works further reduce the ambiguity of point-based controls through region-based mechanisms (Zhou et al., 2026a; Yin et al., 2026; Sui et al., 2026). However, existing methods do not explicitly consider spatial dependencies among drag points, making the resulting displacement field potentially ambiguous and inconsistent.

For drag-based image editing, the main difficulty arises when multiple drag directions are combined without considering where each drag should take effect. Combining all drag directions into a single overall motion can cause spatially separated drags with different directions to interfere with each other, even though each drag should primarily affect its own local region. Instead, the influence of each drag should first be determined according to its spatial relationship with the query location, after which the relevant displacements can be combined locally. This allows spatially separated drags to remain independently effective, while nearby compatible drags can contribute jointly. Therefore, effective multi-drag editing requires resolving spatial influence before combining displacement directions, rather than treating all drags as a single global motion.

This motivates us to explicitly model spatial dependencies among drag points and construct a spatially coherent displacement field. To this end, we introduce KriDrag, which assigns each drag point a covariance-controlled spatial influence and combines it with query-dependent relative weighting to determine the contribution of each drag at every image location. We adopt Gaussian covariance to provide a smooth, direction-neutral spatial influence whose length scale directly controls the extent of each drag. KriDrag preserves the displacement specified by each drag point during spatial influence estimation and combines the resulting contributions only at the query location, thereby maintaining coherent motion across neighboring regions while reducing conflicts between competing drags. This dense displacement field can then be integrated into a standard latent diffusion relocation and reconstruction pipeline.

Contributions: 1) New Insight for Drag-Based Editing: We innovatively bridge Kriging interpolation with drag-based image editing, enabling spatially aware displacement modeling to address image transformations under multiple user drags. **2) Efficient and Effective Framework:** We introduce KriDrag, a framework that constructs a Gaussian spatial influence field for each drag and uses query-dependent relative handle weighting to produce a dense displacement field through local aggregation. **3) Focusing on Multi-Drag Benchmarks:** We construct benchmarks for multi-drag-based image editing, establish representative baselines, and demonstrate the effectiveness of our approach through extensive experiments.

2 Related Work

Drag-Based Image Editing Drag-based image editing enables intuitive spatial manipulation by allowing users to specify the desired motion of image content through point or region dragging. DragGAN (Pan et al., 2023) introduces point-based manipulation through motion supervision and point tracking, and DragDiffusion (Shi et al., 2024) adapts this optimization framework to diffusion models. To improve the dragging process, FreeDrag (Ling et al., 2024) uses adaptive template matching and line search, while GoodDrag (Zhang et al., 2025) coordinates dragging and denoising. DragonDiffusion (Mou et al., 2024b) and DiffEditor (Mou et al., 2024a) guide editing through feature correspondences, and DragLoRA (Xia et al., 2025) optimizes LoRA adapters at test time. To reduce inference cost, InstantDrag (Shin et al., 2024) and LightningDrag (Shi et al., 2025) train networks conditioned on drag instructions. Other methods implement edits through image warping or region-level control: Inpaint4Drag (Lu & Han, 2025) combines bidirectional warping with inpainting, RegionDrag (Lu et al., 2024) uses source-to-target region correspondences, and

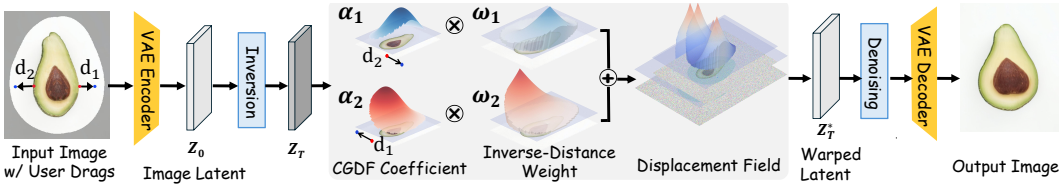

Figure 2: **Overview of KriDrag**. Given an input image and user-specified drags, our approach computes the Covariance-Guided Displacement Field (CGDF) coefficients and inverse-distance weights in the spatial coordinates of the latent space, and combines them with the user drags to construct a dense displacement field. The displacement field is then used to warp the noisy latent features, followed by denoising to produce the target image features.

DragFlow (Zhou et al., 2026b) introduces region-level affine supervision for diffusion-transformer features. For edits involving multiple handles, an additional consideration is how their displacements jointly determine motion across the editable region. FastDrag (Zhao et al., 2024) blends handle-specific warpage vectors with inverse-distance weights, whereas GeoDrag (Pu et al., 2026) incorporates geometric cues and assigns each editable location to its nearest handle. In contrast, our method combines covariance-based spatial responses with inverse-distance weighting to control the spatial influence of individual handles when composing the dense displacement field.

Training-Free Image Editing Training-free image editing leverages pretrained text-to-image models to follow target prompts while preserving the source content. SDEdit (Meng et al., 2021) adds noise to an input image and denoises it with a pretrained diffusion model, balancing input fidelity and image realism. For real image inversion, Null-text Inversion (Mokady et al., 2023) optimizes unconditional text embeddings to improve reconstruction while keeping model weights fixed. Attention control and feature injection are widely used to preserve source content during editing. Prompt-to-Prompt (Hertz et al., 2022) controls cross-attention maps, Plug-and-Play (Tumanyan et al., 2023) injects source spatial features and self-attention, and MasaCtrl (Cao et al., 2023) retrieves source content and textures through mutual self-attention. For diffusion transformers, Stable Flow (Avrahami et al., 2025) injects attention features into vital layers, while KV-Edit (Zhu et al., 2025) uses a KV cache to preserve background tokens. For localized image editing, DiffEdit (Couairon et al., 2022) automatically generates masks by comparing predictions conditioned on different text prompts, while LEDITS++ (Brack et al., 2024) uses implicit masking to support multiple simultaneous edits. Recent inversion-free methods avoid explicitly inverting the source image into noise. InfEdit (Xu et al., 2024) combines consistency sampling with attention control, and FlowEdit (Kulikov et al., 2025) constructs an ODE that directly maps between source and target distributions. FIA-Edit (Yang et al., 2026) combines frequency-domain feature interaction with feature injection to improve background preservation in inversion-free editing. Unlike these works that primarily focus on text-driven image editing, our approach focuses on drag-based image editing.

3 Method

3.1 Overview

For drag-based image editing, given an image I and a set of handle-target pairs $\mathcal{D} = \{(h_i, t_i)\}_{i=1}^{K}$, a binary editing mask M, the desired displacement for each pair is defined as $d_i = t_i - h_i$. Our goal is to invert I into a noise-corrupted latent z_T and construct a dense displacement field v over the editable latent region Ω. With the resulting dense displacement field v, latent features are warped via forward mapping:

$$z_T^*(p + v(p)) \leftarrow z_T(p), \qquad p \in \Omega. \tag{1}$$

The pipeline of our approach is shown in Figure 2. For the partially noised latent, interpolation fills the unassigned latent positions within the editable region, followed by diffusion reconstruction to generate the edited image (Zhao et al., 2024; Pu et al., 2026). Higher noise levels enable more flexible image editing. Meanwhile, as the number of diffusion reconstruction steps increases, the generated images also have higher flexibility and reconstruction quality.

3.2 Covariance-Guided Displacement Field

To illustrate our approach, we consider single-drag-based image editing, where $K = 1$. We study how the displacement affects surrounding positions through spatial autocorrelation. Unlike FastDrag, which uses a geometric construction based on the enclosing mask circle (Zhao et al., 2024), we construct a Covariance-Guided Displacement Field (CGDF) centered at each handle, with a length scale controlling its spatial extent.

For the handle position h_i, we consider a zero-mean scalar Gaussian process A_i and independently condition it on a unit-valued observation at h_i with noise variance ϵ. Using a normalized Gaussian covariance, the resulting response and displacement field are given by

$$k_\ell(p, h_i) = \exp\left(-\frac{\|p - h_i\|_2^2}{2\ell^2}\right), \qquad \alpha_i(p) = \frac{k_\ell(p, h_i)}{1 + \epsilon}, \qquad v_i(p) = \alpha_i(p) d_i, \tag{2}$$

where response coefficient $\alpha_i(p)$ determines the displacement strength, while d_i specifies its direction and magnitude. A smaller ℓ concentrates motion near the handle, whereas a larger ℓ spreads it farther. The radial response is orientation-agnostic and decays smoothly without a hard cutoff, independently of the enclosing mask circle, while editing remains restricted to Ω. For each handle position h_i, $\alpha_i(p)$ peaks at $1/(1 + \epsilon)$, and each displacement field can be computed independently.

3.3 Multi-Drag Displacement Field

For multi-drag-based image editing, where $K \geq 2$, a position may be influenced by multiple drags. Using inverse-distance weighting (Zhao et al., 2024), we scale each response coefficient $\alpha_i(p)$ as

$$q_i(p) = \frac{1}{\|p - h_i\|_2 + \delta}, \qquad w_i(p) = \frac{q_i(p)}{\sum_j q_j(p)}, \qquad r_i(p) = w_i(p)\alpha_i(p). \tag{3}$$

where $s(p)$ is the total response coefficient, computed as a weighted average of the $\alpha_i(p)$, such that $0 \leq s(p) \leq 1$. The coefficient $s(p)$ controls the attenuation of the local displacement mixture and decreases when all handles have weak influence.

The combined weights $r_i(p)$ are normalized as $\beta_i(p) = r_i(p)/s(p)$ to represent the relative contributions of the displacement vectors d_i. The normalized coefficient $\beta_i(p)$ accounts for both inverse-distance weighting and the spatial response.

The displacement field is then computed as

$$v(p) = \sum_i r_i(p) d_i = s(p) \sum_i \beta_i(p) d_i, \qquad \sum_i \beta_i(p) = 1, \tag{4}$$

where $s(p)$ controls the overall strength of the displacement combined from multiple handles, while $\beta_i(p)$ determines the relative contribution of each handle. The resulting displacement magnitude additionally depends on the lengths and directions of these vectors.

Algorithm 1 summarizes the editing procedure. Given an image I, a binary editing mask M, and handle-target pairs $\mathcal{D}$, we invert I to obtain z_T. The handle and target points are mapped to the latent grid, and M is downsampled using nearest-neighbor interpolation to define the editable region Ω. For each $p \in \Omega$, we evaluate the spatial responses $\alpha_i(p)$ using Eq. (2) and the distance scores $q_i(p)$ using Eq. (3). The scores are normalized across all handles, and the corresponding displacement fields are combined to obtain the final displacement field.

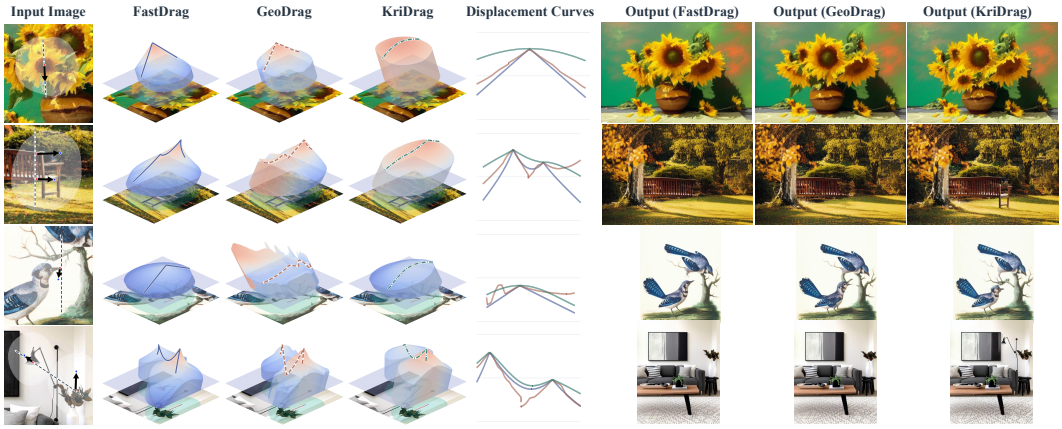

Figure 3: **Visualization of displacement fields and corresponding editing results.** The first column shows the input images with user-provided drag points. For better visualization of the drags, cropped regions containing the drag interactions are displayed. The second to fourth columns visualize the corresponding 3D displacement fields of FastDrag (Zhao et al., 2024), GeoDrag (Pu et al., 2026) and KriDrag (ours), where our approach generates smoother spatial variations without abrupt changes around the drag points. The fifth column compares the displacement curves along a fixed cross-section of the displacement fields. Together with the 3D visualizations in the second to fourth columns, these curves demonstrate the superior smoothness and spatial coherence of our displacement field. The sixth to eighth columns present the corresponding image editing results. Our method achieves more accurate drag-driven semantic transformations while better preserving the background content of the original images.

Algorithm 1 KriDrag for Drag-Based Image Editing

Input: Image I, handle-target pairs $\mathcal{D}$, number of drag K, binary editing mask M
Output: Edited image I'

$z_T \leftarrow$ DDIM INVERSION(ENCODE(I))
Map $\mathcal{D}$ to the latent grid $\{(h_i, t_i)\}_{i=1}^{K}$, calculate displacement $d_i \leftarrow t_i - h_i$
$\Omega \leftarrow \{p : \text{DOWNSAMPLENEAREST}(M)(p) = 1\}$
Compute spatial response maps α_i using Eq.(2)
Compute normalized inverse-distance weight maps w_i using Eq.(3)
Construct $v(p) \leftarrow \sum_{i=1}^{K} w_i(p)\alpha_i(p)d_i$ ▷ Eq.(4)
$z_T^* \leftarrow \text{FORWARDRELOCATE}(z_T, v, \Omega)$ ▷ Eq. (1)
$z \leftarrow \text{INTERPOLATEUNFILLED}(z_T^*, \Omega)$
for τ in the denoising schedule **do**
    $z \leftarrow$ DDIM SAMPLING(z, τ)
end for
return $I' \leftarrow \text{DECODE}(z)$

3.4 DISCUSSION OF KRIDRAG

The displacement field determines not only where each latent feature moves, but also how neighboring features move relative to one another. For the relocation map $T(p) = p + v(p)$, two neighboring positions satisfy

$$T(q) - T(p) = (q - p) + [v(q) - v(p)]. \tag{5}$$

Abrupt displacement variations may distort local structures during latent relocation, while smoothly varying fields preserve spatial coherence. KriDrag controls such variations through covariance-guided responses and continuous multi-handle aggregation. Using $v(p) = s(p)\sum_i \beta_i(p)d_i$, the displacement variation can be written as

$$v(q) - v(p) = [s(q) - s(p)] \sum_i \beta_i(p)d_i + s(q) \sum_i [\beta_i(q) - \beta_i(p)]d_i, \tag{6}$$

where the first term corresponds to the spatial change of overall response strength, while the second term captures the variation of relative handle contributions. Gaussian responses provide smooth spatial transitions around handle points, and continuous weighting avoids abrupt changes between different drag motions.

Excessively large displacement magnitudes can relocate local latent features far from their original positions, thereby disrupting the spatial correspondence between neighboring features. The subsequent diffusion denoising process may further amplify these distortions, often compromising the structural and semantic information of the original image. Typical failure cases include the disappearance of the main subject or unintended semantic reconstruction. As shown in Figure 3, the displacement field produced by FastDrag (Zhao et al., 2024) exhibits a pronounced peak around each user-provided drag point, indicating highly concentrated and abrupt spatial displacement. Such large local variations can induce substantial distortions in neighboring latent features and consequently lead to degraded editing fidelity. Our method mitigates this issue by producing a spatially smooth displacement field that avoids excessively large local displacements and preserves the structural and semantic information of the original image.

4 Experiments

We evaluate KriDrag on two popular benchmarks, DragBench and ReD-Bench. In addition to the standard drag-based image editing setting, we establish new benchmarks for multi-drag editing. Both quantitative and qualitative analyses are conducted to demonstrate the effectiveness of KriDrag.

4.1 Experimental Setup

Datasets DragBench (Shi et al., 2024) contains 205 images and 349 handle-target pairs, comprising 108 single-drag editing instances and 97 multi-drag editing instances with 241 handle-target pairs. ReD-Bench (Zhou et al., 2026b) contains 120 images and 288 handle-target pairs, comprising 29 single-drag editing instances and 91 multi-drag editing instances with 259 handle-target pairs.

Metrics and Baselines We adopt Mean DIFT Distance (MD) (Tang et al., 2023), Image Fidelity (IF) (Zhang et al., 2018) and DAI-1/10/20 (Zhang et al., 2025) as evaluation metrics. MD measures the distance between the requested target points and their corresponding diffusion features, averaged over all handle-target pairs. IF is defined as $1 - \mathrm{LPIPS}_{\mathrm{AlexNet}}$ and averaged over editing instances. DAI-1/10/20 measures the target-patch difference at radii of 1, 10, and 20 pixels. These scores are first averaged over handles within each editing instance and then across instances. Lower MD and DAI indicate better localization, while higher IF indicates greater similarity to the source image. We compare KriDrag with DragDiffusion (Shi et al., 2024), DragNoise (Liu et al., 2024), FreeDrag (Ling et al., 2024), FastDrag (Zhao et al., 2024), GeoDrag (Pu et al., 2026), and DragFlow (Zhou et al., 2026b).

Implementation Details Our implementation follows the latent-space relocation and reconstruction pipeline of FastDrag (Zhao et al., 2024). The only difference lies in the construction of the dense displacement field. For KriDrag, we use Gaussian covariance with $\ell = 12$, an inversion strength of 0.5, and 20 diffusion steps, including 10 steps for inversion and 10 steps for reconstruction. Experiments are conducted on a single NVIDIA RTX 4090.

4.2 Comparison with Baselines

Results of Drag-Based Image Editing Table 1 reports the overall results on DragBench and ReD-Bench. Among the compared methods, KriDrag obtains the lowest MD and DAI-1/10/20 on both benchmarks, with MD values of 23.23 on DragBench and 26.98 on ReD-Bench, compared with 27.86 for GeoDrag and 30.53 for DragFlow, respectively. Compared with FastDrag and GeoDrag, which also construct displacement fields for editing, the proposed method achieves higher IF and lower MD and DAI on both benchmarks, improving image fidelity alongside editing accuracy. Without per-image iterative optimization, our approach takes approximately 2 s per image on average.

Table 1: Quantitative comparison on DragBench and ReD-Bench. Lower MD and DAI indicate better editing accuracy, and higher IF indicates better image fidelity. Time is the average processing time per image unless otherwise noted, and Mem is peak GPU memory. Best and second-best quality scores within each benchmark are bold and underlined.

	Method	MD↓	DAI_1 ↓	DAI_{10} ↓	DAI_{20} ↓	IF↑	Training	Time (s)	Mem (GB)
DragBench	DragDiffusion (Shi et al., 2024)	36.94	0.248	0.232	0.215	0.889	✓	89.60	13.28
	DragNoise (Liu et al., 2024)	34.63	0.234	0.219	0.203	0.902	✓	157.29	12.85
	FreeDrag (Ling et al., 2024)	31.83	0.276	0.244	0.220	**0.905**	✓	194.88	13.56
	FastDrag (Zhao et al., 2024)	31.73	0.197	0.184	0.164	0.860	✗	2.77	2.59
	GeoDrag (Pu et al., 2026)	27.86	0.191	0.170	0.153	0.847	✗	1.76	4.41
	KriDrag (Ours)	**23.23**	**0.139**	**0.114**	**0.106**	0.877	✗	2.05	2.59
ReD-Bench	DragDiffusion (Shi et al., 2024)	39.92	0.267	0.242	0.231	0.900	✓	93.63	13.81
	DragNoise (Liu et al., 2024)	35.92	0.254	0.240	0.226	0.894	✓	159.84	13.38
	FreeDrag (Ling et al., 2024)	33.73	0.270	0.225	0.216	**0.901**	✓	194.51	14.15
	FastDrag (Zhao et al., 2024)	33.89	0.235	0.209	0.205	0.857	✗	2.40	2.81
	GeoDrag (Pu et al., 2026)	37.51	0.205	0.200	0.199	0.853	✗	1.73	4.57
	DragFlow (Zhou et al., 2026b)	30.53	0.176	0.154	0.156	0.889	✓	847.65	34.36
	KriDrag (ours)	**26.98**	**0.175**	**0.149**	**0.147**	0.871	✗	2.23	2.81

Table 2: Comparison of multi-handle drag-based image editing on the benchmarks of ReD-Bench and DragBench. All comparison methods are reproduced by us. Since DragFlow (Zhou et al., 2026b) requires region-level annotations, we only reproduce its results on ReD-Bench.

	ReD-Bench					DragBench				
Method	MD↓	DAI_1 ↓	DAI_{10} ↓	DAI_{20} ↓	IF↑	MD↓	DAI_1 ↓	DAI_{10} ↓	DAI_{20} ↓	IF↑
DragDiffusion (Shi et al., 2024)	40.14	0.302	0.258	0.244	0.892	36.35	0.238	0.234	0.204	0.903
DragNoise (Liu et al., 2024)	35.20	0.267	0.246	0.231	0.885	33.89	0.236	0.222	0.195	0.907
FreeDrag (Ling et al., 2024)	33.75	0.304	0.244	0.230	**0.895**	29.33	0.257	0.240	0.203	**0.923**
FastDrag (Zhao et al., 2024)	32.09	0.255	0.211	0.204	0.847	33.38	0.227	0.206	0.180	0.886
GeoDrag (Pu et al., 2026)	35.81	0.205	0.199	0.196	0.842	28.09	0.194	0.177	0.152	0.869
DragFlow (Zhou et al., 2026b)	32.05	0.202	0.169	0.167	0.876			—		
KriDrag (ours)	**26.24**	**0.197**	**0.166**	**0.162**	0.860	**24.50**	**0.158**	**0.144**	**0.134**	0.894

Results of Multi-Drag-Based Image Editing Table 2 summarizes results of multi-drag-based image editing. The proposed chieves the lowest MD and all three DAI scores, with MD values of 24.50 on DragBench and 26.24 on ReD-Bench, compared with 28.09 for GeoDrag and 32.05 for DragFlow, respectively. On the ReD-Bench, the MD advantage over DragFlow increases from 3.55 pixels overall to 5.81 pixels on multi-handle cases. Higher IF than FastDrag and GeoDrag is also achieved on both multi-handle subsets, indicating improved editing accuracy while preserving source-image fidelity. These results demonstrate its effectiveness in handling multiple drag motions simultaneously while maintaining consistent displacement across different handles.

4.3 Ablations and Analysis

Ablation of Displacement Field We remove the spatial responses to obtain $v = \sum_i w_i d_i$ (denoted as w_i only), or remove the allocation weights to obtain the unweighted sum $v = \sum_i \alpha_i d_i$ (denoted as α_i only). Table 3 reports the ablation results under both the normal setting and the multi-drag setting. Removing either component leads to lower IF performance in both settings. In particular, removing the allocation weights results in a degraded MD score, indicating that the weighting strategy contributes to more accurate displacement estimation. Although removing the spatial responses appears to achieve a better MD score, it does not actually perform effective drag-based image editing, as verified by the visual results.

Figure 4 presents visual comparisons of our ablated methods. Both KriDrag and 'α_i only' follow the semantic intent of user drags, while 'w_i only' fails to perform drag-based editing and instead shifts the entire selected object along the dragging direction. Combined with the quantitative results in Table 3, these results suggest that the MD metric has limited capability in evaluating drag



Figure 4: **Qualitative results of ablation studies.** The top row shows the user drags, with red points denoting handle points and blue points denoting target points. The w_i only variant tends to produce globally motion, whereas the α_i only variant exhibits weaker drag alignment. Combining the two terms, KriDrag more reliably reaches the desired drag locations while preserving semantic consistency and the spatial structure of surrounding regions.

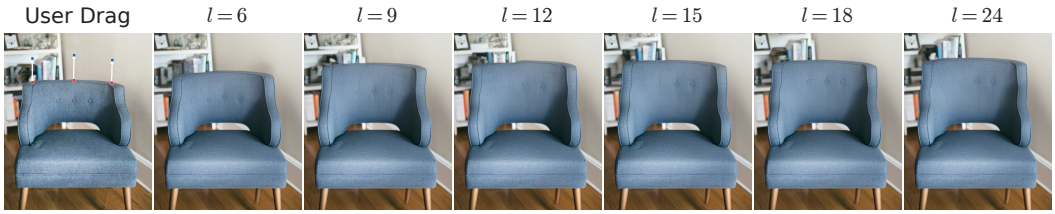

Figure 5: **Qualitative results of KriDrag with different Gaussian size** l. In this multi-drag example, small kernel sizes limit the deformation of the chair backrest toward the targets, whereas overly large sizes introduce broader deformation and noticeable changes to the background bookshelf.

editing accuracy, as it may favor methods that perform simple region translation rather than achieve meaningful content deformation.

Table 3: Ablation results of key components in the displacement field on the Drag-Bench.

Set	Method	MD↓	DAI_1 ↓	DAI_{10} ↓	DAI_{20} ↓	IF↑
Full	α_i only	31.50	0.177	0.142	0.128	0.862
	w_i only	18.59	0.122	0.107	0.099	0.826
	KriDrag	23.23	0.139	0.114	0.106	0.877
Multi	α_i only	37.00	0.238	0.204	0.180	0.862
	w_i only	21.27	0.151	0.142	0.130	0.872
	KriDrag	24.50	0.158	0.144	0.134	0.894

Table 4: Sensitivity of covariance kernel size ℓ on the DragBench under multi-drag setting.

Setting	MD↓	IF↑	DAI-1↓	DAI-10↓	DAI-20↓
$\ell = 6$	32.2398	0.9168	0.1941	0.1780	0.1617
$\ell = 9$	26.3336	0.9029	0.1668	0.1579	0.1424
$\ell = 12$	24.5186	0.8940	0.1580	0.1438	0.1339
$\ell = 15$	23.2025	0.8899	0.1548	0.1422	0.1304
$\ell = 18$	24.5019	0.8852	0.1500	0.1414	0.1289
$\ell = 24$	23.8565	0.8811	0.1544	0.1373	0.1259

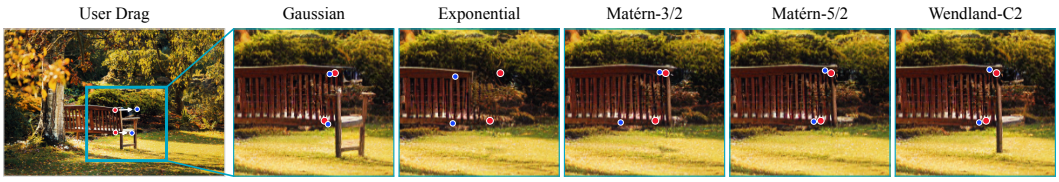

Figure 6: **Visualizations of drag-based image editing using different covariance kernels.** In this example, two drags extend the right end of the bench. Red and blue points denote the handle and target points, respectively. For better visualization, only the edited regions corresponding to the input regions are displayed. Our method with Gaussian kernel achieves accurate target alignment while preserving visual fidelity and object structure.

Table 5: Comparison of drag-based editing with different kernels on the Drag-Bench.

Kernel	MD↓	IF↑
Gaussian	**23.2348**	0.8768
Exponential	33.2010	**0.9019**
Matérn-3/2	26.7740	0.8883
Matérn-5/2	26.0119	0.8843
Wendland-C2	24.1399	0.8788

Table 6: Controlled experiments with similar IF scores on the multi-drag setting of DragBench. Gaussian denotes our approach, Constant denotes the variant with constant spatial response weights $\alpha_i = 1$, and FastDrag (Ling et al., 2024) denotes the variant using its geometric response.

Spatial	IF↑	MD↓	$DAI_1 \downarrow$	$DAI_{10} \downarrow$	$DAI_{20} \downarrow$
Gaussian	0.89398	**25.76**	**0.1581**	**0.1437**	**0.1338**
Constant	0.89434	26.74	0.2561	0.2209	0.1887
FastDrag	0.89425	29.33	0.1856	0.1641	0.1496

Sensitivity of Covariance Kernel Size The Gaussian kernel size l determines how far each handle's displacement propagates into the surrounding region. As shown in Table 4, increasing l from 6 to 15 reduces MD from 32.24 to 23.20, while IF decreases from 0.9168 to 0.8899. Small kernel sizes concentrate displacement around the handles and may limit the drag displacement toward the target points, whereas larger kernel sizes propagate deformation over a broader region. Further visualizations of the editing images are shown in Figure 5. We set $l = 12$ to balance target alignment and source-image preservation.

Effect of Different Covariance Kernels We compare Gaussian, Exponential, Matérn-3/2, Matérn-5/2, and Wendland-C2 kernels in our editing pipeline in Table 5. Gaussian kernel achieves the lowest MD score of 23.23 on the full DragBench benchmark, followed by Wendland-C2 with an MD score of 24.14. Figure 6 illustrates the difference between kernels. Gaussian kernel moves the bench toward both target points more accurately, whereas kernels of Exponential and Matérn-3/2 fail to fully achieve the desired drag displacements.

Comparison of Spatial Responses with Similar Visual Fidelity Since MD and IF exhibit a trade-off between editing accuracy and visual fidelity, we conduct controlled experiments, as shown in Table 6, to investigate the impact of different spatial responses. By maintaining comparable IF scores, we compare the MD performance of different approaches. Our approach achieves the lowest MD and DAI scores, demonstrating more accurate drag-based image editing.

5 Conclusion

In this work, we introduce KriDrag, a spatially autocorrelated conflict-free drag-based image editing approach. By modeling spatial dependencies among user-provided handles through covariance-controlled spatial influence and local aggregation of handle-specific motions, KriDrag constructs coherent dense displacement fields for accurate multi-handle editing. Experiments show that KriDrag enables more precise object-level spatial transformation while preserving visual fidelity, demonstrating the importance of explicitly modeling spatial correlations in drag-based image editing.
Limitations: Without explicit textual semantic constraints, our method may not always accurately interpret the user's intended drag operations. Moreover, our approach primarily alleviates conflicts among multiple drag points rather than fully resolving them.

AI USE STATEMENT

In this work, we used generative AI tools for language editing and assistance with code development. All AI-assisted content, mathematical derivations, experimental implementations, and generated text were reviewed and verified by the authors. We take responsibility for the final content of this work, including text, claims, code, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not introduce new training data or collect human-subject data. The proposed method inherits potential risks and biases from the underlying pretrained generative models.

REPRODUCIBILITY STATEMENT

We provide the complete mathematical formulation, optimization procedure, and implementation details required to reproduce the proposed method. Hyperparameter settings, evaluation protocols, and additional experimental results are provided in the main paper and appendix. Code will be released upon publication.

REFERENCES

Omri Avrahami, Rinon Gal, Gal Chechik, Ohad Fried, Dani Lischinski, Arash Vahdat, and Weili Nie. Diffuhaul: A training-free method for object dragging in images. In *SIGGRAPH Asia Conference Papers*. Association for Computing Machinery, 2024.

Omri Avrahami, Or Patashnik, Ohad Fried, Egor Nemchinov, Kfir Aberman, Dani Lischinski, and Daniel Cohen-Or. Stable flow: Vital layers for training-free image editing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7877–7888. IEEE, 2025.

Manuel Brack, Felix Friedrich, Katharia Kornmeier, Linoy Tsaban, Patrick Schramowski, Kristian Kersting, and Apolinário Passos. Ledits++: Limitless image editing using text-to-image models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8861–8870, 2024.

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 *IEEE/CVF International Conference on Computer Vision*, pp. 22503–22513. IEEE, 2023.

Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. *arXiv preprint arXiv:2210.11427*, 2022.

Yutao Cui, Xiaotong Zhao, Guozhen Zhang, Shengming Cao, Kai Ma, and Limin Wang. Stabledrag: Stable dragging for point-based image editing. In *European Conference on Computer Vision*, pp. 340–356, 2024.

Weiran Guang, Xiaoguang Gu, Mengqi Huang, and Zhendong Mao. Dragin3d: Image editing by dragging in 3d space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21502–21512, 2025.

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

Xingzhong Hou, Boxiao Liu, Yi Zhang, Jihao Liu, Yu Liu, and Haihang You. Easydrag: Efficient point-based manipulation on diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8404–8413, 2024.

Meng-Ting Jhong, Tai-Ming Huang, Shang-Fu Chen, Wen-Huang Cheng, and Kai-Lung Hua. Dragonite: Single-step drag-based image editing with geometric-semantic guidance. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3057–3066, 2026.

Gwanhyeong Koo, Sunjae Yoon, Younghwan Lee, Ji Woo Hong, and Chang D. Yoo. Flowdrag: 3d-aware drag-based image editing with mesh-guided deformation vector flow fields. In *International Conference on Machine Learning*, volume 267, pp. 31467–31485, 2025.

Vladimir Kulikov, Matan Kleiner, Inbar Huberman-Spiegelglas, and Tomer Michaeli. Flowedit: Inversion-free text-based editing using pre-trained flow models. In *IEEE/CVF International Conference on Computer Vision*, pp. 19721–19730. IEEE, 2025.

Ruining Li, Chuanxia Zheng, Christian Rupprecht, and Andrea Vedaldi. Dragapart: Learning a part-level motion prior for articulated objects. In *European Conference on Computer Vision*, pp. 165–183, 2024.

Pengyang Ling, Lin Chen, Pan Zhang, Huaian Chen, Yi Jin, and Jinjin Zheng. Freedrag: Feature dragging for reliable point-based image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6860–6870, 2024.

Haofeng Liu, Chenshu Xu, Yifei Yang, Lihua Zeng, and Shengfeng He. Drag your noise: Interactive point-based editing via diffusion semantic propagation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6743–6752, 2024.

Jingyi Lu and Kai Han. Inpaint4drag: Repurposing inpainting models for drag-based image editing via bidirectional warping. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 18304–18313, 2025.

Jingyi Lu, Xinghui Li, and Kai Han. Regiondrag: Fast region-based image editing with diffusion models. In *European Conference on Computer Vision*, volume 15075, pp. 231–246, 2024.

Minxing Luo, Wentao Cheng, and Jian Yang. Rotationdrag: Point-based image editing with rotated diffusion features. In *European Conference on Computer Vision*, pp. 231–246, 2024.

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

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

Chong Mou, Xintao Wang, Jiechong Song, Ying Shan, and Jian Zhang. Diffeditor: Boosting accuracy and flexibility on diffusion-based image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8488–8497, 2024a.

Chong Mou, Xintao Wang, Jiechong Song, Ying Shan, and Jian Zhang. Dragondiffusion: Enabling drag-style manipulation on diffusion models. In *International Conference on Learning Representations*, pp. 31620–31631, 2024b.

Shen Nie, Hanzhong Guo, Cheng Lu, Yuhao Zhou, Chenyu Zheng, and Chongxuan Li. The blessing of randomness: SDE beats ODE in general diffusion-based image editing. In *International Conference on Learning Representations*, pp. 41763–41799, 2024.

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

Xinyu Pu, Hongsong Wang, Jie Gui, and Pan Zhou. Dragging with geometry: From pixels to geometry-guided image editing. In *International Conference on Learning Representations*, pp. 151958–151984, 2026.

Rahul Sajnani, Jeroen Vanbaar, Jie Min, Kapil D. Katyal, and Srinath Sridhar. Geodiffuser: Geometry-based image editing with diffusion models. In *Proceedings of the Winter Conference on Applications of Computer Vision*, pp. 472–482, 2025.

Yujun Shi, Chuhui Xue, Jun Hao Liew, Jiachun Pan, Hanshu Yan, Wenqing Zhang, Vincent Y. F. Tan, and Song Bai. Dragdiffusion: Harnessing diffusion models for interactive point-based image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8839–8849, 2024.

Yujun Shi, Jun Hao Liew, Hanshu Yan, Vincent Y. F. Tan, and Jiashi Feng. Lightningdrag: Lightning fast and accurate drag-based image editing emerging from videos. In *International Conference on Machine Learning*, volume 267, pp. 54981–54993. PMLR, 2025.

Joonghyuk Shin, Daehyeon Choi, and Jaesik Park. Instantdrag: Improving interactivity in drag-based image editing. In *ACM SIGGRAPH Asia Conference Proceedings*, pp. 1–10, 2024.

Jiacheng Sui, Tianyu Hao, Bingjie Gao, Li Niu, and Guangtao Zhai. In-context region-based drag: Drag any region to any shape. In *European Conference on Computer Vision*, 2026.

Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion. In *Advances in Neural Information Processing Systems*, volume 36, pp. 1363–1389, 2023.

Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1921–1930. IEEE, 2023.

Siwei Xia, Li Sun, Tiantian Sun, and Qingli Li. Draglora: Online optimization of LoRA adapters for drag-based image editing in diffusion model. In *Proceedings of the International Conference on Machine Learning*, volume 267, pp. 68277–68291, 2025.

Sihan Xu, Yidong Huang, Jiayi Pan, Ziqiao Ma, and Joyce Chai. Inversion-free image editing with language-guided diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9454–9461. IEEE, 2024.

Biao Yang, Muqi Huang, Yuhui Zhang, Yun Xiong, Kun Zhou, Xi Chen, Shiyang Zhou, Huishuai Bao, Chuan Li, Feng Shi, and Hualei Liu. Attentiondrag: Exploiting latent correlation knowledge in pre-trained diffusion models for image editing. In *International Joint Conference on Artificial Intelligence*, pp. 2188–2196, 2025.

Kaixiang Yang, Boyang Shen, Xin Li, Yuchen Dai, Yuxuan Luo, Yueran Ma, Wei Fang, Qiang Li, and Zhiwei Wang. Fia-edit: Frequency-interactive attention for efficient and high-fidelity inversion-free text-guided image editing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 11613–11621, 2026.

Zixin Yin, Xili Dai, Duomin Wang, Xianfang Zeng, Lionel M. Ni, Gang Yu, and Heung-Yeung Shum. Lazydrag: Enabling stable drag-based editing on multi-modal diffusion transformers via explicit correspondence. In *International Conference on Learning Representations*, pp. 23223–23244, 2026.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018.

Zewei Zhang, Huan Liu, Jun Chen, and Xiangyu Xu. Gooddrag: Towards good practices for drag editing with diffusion models. In *International Conference on Learning Representations*, pp. 13785–13808, 2025.

Xuanjia Zhao, Jian Guan, Congyi Fan, Dongli Xu, Youtian Lin, Haiwei Pan, and Pengming Feng. Fastdrag: Manipulate anything in one step. In *Advances in Neural Information Processing Systems*, volume 37, pp. 74439–74460, 2024. doi: 10.52202/079017-2369.

Yuan Zhou, Junbao Zhou, Qingshan Xu, Kesen Zhao, Yuxuan Wang, Hao Fei, Richang Hong, and Hanwang Zhang. Dragnext: Rethinking drag-based image editing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 13818–13825, 2026a.

Zihan Zhou, Shilin Lu, Shuli Leng, Shaocong Zhang, Zhuming Lian, Xinlei Yu, and Adams Wai-Kin Kong. Dragflow: Unleashing DiT priors with region-based supervision for drag editing. In *International Conference on Learning Representations*, pp. 136308–136344, 2026b.

Tianrui Zhu, Shiyi Zhang, Jiawei Shao, and Yansong Tang. Kv-edit: Training-free image editing for precise background preservation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16607–16617. IEEE, 2025.

Supplementary Contents

A Additional Related Works **14**

B Theoretical Analysis **15**

B.1 Single-Observation Covariance Derivation 15

B.2 Covariance Properties 15

C Additional Experiments **16**

C.1 Effect of Local Aggregation 16

C.2 Effect of Partitioning 16

C.3 Effect of Handle Allocation 17

C.4 Additional Qualitative Results 17

A Additional Related Works

Training-Based Drag Editing DragGAN (Pan et al., 2023) introduces point-based manipulation through motion supervision and point tracking, and DragDiffusion (Shi et al., 2024) adapts this optimization framework to diffusion models. To improve the dragging process, FreeDrag (Ling et al., 2024) uses adaptive template matching and line search, while GoodDrag (Zhang et al., 2025) coordinates dragging and denoising. DragonDiffusion (Mou et al., 2024b) and DiffEditor (Mou et al., 2024a) guide editing through feature correspondences, and DragLoRA (Xia et al., 2025) optimizes LoRA adapters at test time. To reduce inference cost, InstantDrag (Shin et al., 2024) and LightningDrag (Shi et al., 2025) train networks conditioned on drag instructions. Other methods implement edits through image warping or region-level control: Inpaint4Drag (Lu & Han, 2025) combines bidirectional warping with inpainting, RegionDrag (Lu et al., 2024) uses source-to-target region correspondences, and DragFlow (Zhou et al., 2026b) introduces region-level affine supervision for diffusion-transformer features. For edits involving multiple handles, an additional consideration is how their displacements jointly determine motion across the editable region.

Training-Free Drag Editing Another line of work constructs edits directly from pretrained generative priors without task-specific fine-tuning. DiffUHaul repurposes internal diffusion features for object relocation, while AttentionDrag exploits latent correlations to propagate the requested motion (Avrahami et al., 2024; Yang et al., 2025). Inpaint4Drag repurposes an inpainting model through bidirectional warping, and SDE-Drag performs latent-space editing through an SDE-based diffusion process (Lu & Han, 2025; Nie et al., 2024). ContextDrag removes inversion through position-aligned context injection, whereas Dragonite combines geometric deformation and semantic cues in a single-step update (Jhong et al., 2026). FastDrag (Zhao et al., 2024) blends handle-specific warpage vectors with inverse-distance weights, whereas GeoDrag (Pu et al., 2026) incorporates geometric cues and assigns each editable location to its nearest handle. Our method, KriDrag, combines covariance-based spatial responses with inverse-distance weighting to control the spatial influence of individual handles when composing the dense displacement field. Rather than learning drag-specific parameters, these methods derive control from frozen generative features, explicit transformations, or sampling-time guidance. Methods that explicitly coordinate multiple handles or structured regions are discussed separately below.

Multi-handle Drag Editing Multiple handle–target pairs introduce a distinct composition problem: their influence regions can overlap, and opposing displacement vectors can weaken one another. FastDrag constructs a dense field by aggregating handle-specific warpage vectors with query-dependent inverse-distance weights, whereas LazyDrag and GeoDrag use nearest-handle, Voronoi-like hard partitions (Zhao et al., 2024; Yin et al., 2026; Pu et al., 2026). RotationDrag models rotational transformations from point controls, while DragAPart introduces part-level motion priors

for articulated objects (Luo et al., 2024; Li et al., 2024). Related geometry-aware editors use projected geometry, deformation flows, or three-dimensional structure to preserve object shape during controlled motion (Sajnani et al., 2025; Koo et al., 2025; Guang et al., 2025). KriDrag instead studies composition under the standard point interface by combining covariance-shaped supports with query-dependent weights to compose multiple handle-specific displacements into one dense field.

Region-Based Drag Editing A separate line of work represents the editing constraint at the region level or applies explicit geometric transformations to image regions. Inpaint4Drag combines bidirectional warping with inpainting, RegionDrag uses source-to-target region correspondences, ICRDrag maps paired regions to target shapes, and DragFlow introduces region-level affine supervision for diffusion-transformer features (Lu & Han, 2025; Lu et al., 2024; Sui et al., 2026; Zhou et al., 2026b). These methods replace purely point-wise control with structured region correspondences, warping, or affine transformations.

B Theoretical Analysis

B.1 Single-Observation Covariance Derivation

For a scalar observation $y = A(h) + \eta$, with zero-mean field covariance k and observation-noise variance ϵ, consider a linear predictor ay for $A(p)$. The mean-squared error is

$$\mathcal{E}(a) = k(p,p) - 2ak(p,h) + a^2[k(h,h) + \epsilon]. \tag{S1}$$

Differentiating gives

$$a^* = \frac{k(p,h)}{k(h,h)+\epsilon}, \qquad \mathcal{E}(b) - \mathcal{E}(a^*) = [k(h,h)+\epsilon](b-a^*)^2 \geq 0. \tag{S2}$$

Eq. (2) therefore gives the minimum-error linear predictor under the specified covariance model. With the normalized Gaussian covariance, $k(h,h) = 1$, each per-drag predictor becomes a scaled radial profile centered at the handle. KriDrag constructs this predictor independently for every drag and clamps the resulting influence to $[0,1]$.

B.2 Covariance Properties

The covariance-guided response used in KriDrag is defined as

$$\alpha_i(p) = \frac{1}{1+\epsilon} \exp\left(-\frac{\|p-h_i\|_2^2}{2\ell^2}\right), \tag{S3}$$

where h_i is a user-provided handle point and ℓ controls the spatial extent of its influence. Let $s_i = \|p - h_i\|_2$. Then

$$0 < \alpha_i(p) \leq \frac{1}{1+\epsilon} \leq 1, \qquad \alpha_i(h_i) = \frac{1}{1+\epsilon}, \qquad \lim_{s_i \to \infty} \alpha_i(p) = 0. \tag{S4}$$

Moreover,

$$\frac{\partial \alpha_i(p)}{\partial s_i} = -\frac{s_i}{\ell^2}\alpha_i(p) \leq 0. \tag{S5}$$

Therefore, the spatial influence decreases monotonically with the distance from the corresponding handle point. A larger ℓ produces a broader spatial response, while a smaller ℓ leads to a more localized response. Since the Gaussian response is continuous with respect to p, the resulting displacement

$$v_i(p) = \alpha_i(p) d_i \tag{S6}$$

is continuous in p before masking and discrete latent relocation.

Table S1: Global and local aggregation on the multi-handle subsets of DragBench (97 cases, 241 pairs) and ReD-Bench (91 cases, 259 pairs). Both variants share the spatial responses, inverse-distance weights, and editing pipeline. DAI uses the per-handle definition in the evaluation protocol.

Dataset	Aggregation	MD↓	IF↑	DAI_1 ↓	DAI_{10} ↓	DAI_{20} ↓
DragBench	Global	31.03	0.9133	0.2323	0.2227	0.1914
	Local (KriDrag)	24.50	0.8940	0.1581	0.1437	0.1338
ReD-Bench	Global	28.76	0.8673	0.2528	0.2207	0.2016
	Local (KriDrag)	26.24	0.8603	0.1966	0.1656	0.1617

Table S2: Effect of partitioning on the multi-handle subsets of DragBench (97 cases, 241 pairs) and ReD-Bench (91 cases, 259 pairs). The hard-partition control uses nearest-handle assignment and regional length-scale recalibration; KriDrag uses soft local aggregation. Both variants share the relocation and reconstruction procedure. DAI follows the per-handle evaluation protocol.

Dataset	Strategy	MD↓	IF↑	DAI_1 ↓	DAI_{10} ↓	DAI_{20} ↓
DragBench	Hard partition	28.15	0.8768	0.1447	0.1167	0.0999
	KriDrag	24.50	0.8940	0.1581	0.1437	0.1338
ReD-Bench	Hard partition	27.03	0.8483	0.1684	0.1395	0.1333
	KriDrag	26.24	0.8603	0.1966	0.1656	0.1617

C Additional Experiments

C.1 Effect of Local Aggregation

We compare global and local aggregation on the multi-handle subsets of both benchmarks. Both variants use the same Gaussian responses $\alpha_i(p)$, inverse-distance weights $w_i(p)$, and total response $s(p) = \sum_i w_i(p)\alpha_i(p)$. The global control first combines the requested displacements $d_i = t_i - h_i$ into a shared vector $d_{\text{eff}} = \sum_i c_i d_i / (\sum_i c_i + 10^{-6})$, where c_i is the response α_i sampled at the latent-grid location nearest to its handle, and constructs $v_{\text{global}}(p) = s(p)d_{\text{eff}}$. KriDrag instead uses local aggregation, $v_{\text{local}}(p) = \sum_i w_i(p)\alpha_i(p)d_i$, allowing the relative handle contributions and resulting motion to vary with position. The relocation and reconstruction procedure is fixed.

Table S1 reports the results. Local aggregation reduces MD and all three DAI scores on both benchmarks, accompanied by lower IF. On DragBench, MD decreases from 31.03 to 24.50 and DAI-20 from 0.1914 to 0.1338; on ReD-Bench, the corresponding values decrease from 28.76 to 26.24 and from 0.2016 to 0.1617. These results support position-dependent aggregation for improving target alignment and target-patch agreement, with a corresponding trade-off in source-image fidelity.

C.2 Effect of Partitioning

We further examine whether each position should receive contributions from multiple handles or be assigned to a single handle. KriDrag uses inverse-distance weights to combine the displacement fields through local aggregation. The hard-partition control instead assigns each editable position to its nearest handle and recalibrates the Gaussian length scale within each region. Both variants use Gaussian spatial responses and the same relocation and reconstruction procedure. As shown in Table S2, KriDrag obtains lower mean MD and higher IF on both multi-handle benchmarks. On DragBench, MD changes from 28.15 with hard partitioning to 24.50 with KriDrag, while IF increases from 0.8768 to 0.8940. On ReD-Bench, the corresponding changes are 27.03 to 26.24 and 0.8483 to 0.8603. Hard partitioning gives lower DAI-1/10/20 on both benchmarks, indicating a different trade-off between target alignment and image fidelity. KriDrag retains contributions from multiple handles without hard region assignment, providing higher image fidelity and lower mean MD at the reported settings.

Table S3: Effect of handle allocation on 97 multi-handle DragBench cases with fixed Gaussian responses and editing pipeline. IDW uses normalized inverse-distance weights, while Equal weights assigns $w_i = 1/K$ to every handle.

Allocation	MD↓	IF↑	DAI_{20} ↓
IDW	24.5186	0.8940	0.1339
Equal weights	29.4186	0.9129	0.1978

C.3 Effect of Handle Allocation

We examine whether distance-dependent allocation remains useful when Gaussian responses already provide spatial attenuation. Keeping the Gaussian responses ($l = 12$) and the editing pipeline fixed, we compare inverse-distance weighting (IDW) with equal weights in $v(p) = \sum_i w_i(p)\alpha_i(p)d_i$. IDW assigns larger allocation weights to nearby handles, whereas equal weights sets $w_i(p) = 1/K$ for every handle. Both strategies retain the handle-specific Gaussian responses; equal allocation therefore does not imply equal final contributions.

Table S3 reports the results on 97 multi-handle DragBench cases. Compared with equal weights, IDW reduces mean MD from 29.42 to 24.52 and DAI-20 from 0.1978 to 0.1339, while IF decreases from 0.9129 to 0.8940. These results show that distance-dependent allocation improves target alignment even when Gaussian spatial attenuation is already present, with a corresponding trade-off in source-image fidelity.

C.4 Additional Qualitative Results

We visualize the drag-based image editing results and compare our KriDrag with DragDiffusion (Shi et al., 2024), DragNoise (Liu et al., 2024), FreeDrag (Ling et al., 2024), FastDrag (Zhao et al., 2024), GeoDrag (Pu et al., 2026), and DragFlow (Zhou et al., 2026b). Figure S1 shows visualizations of single-drag-based image editing results, while Figure S2 presents visualizations of multi-drag-based image editing results. The results show that our method achieves more accurate and spatially coherent image transformations than the compared methods.

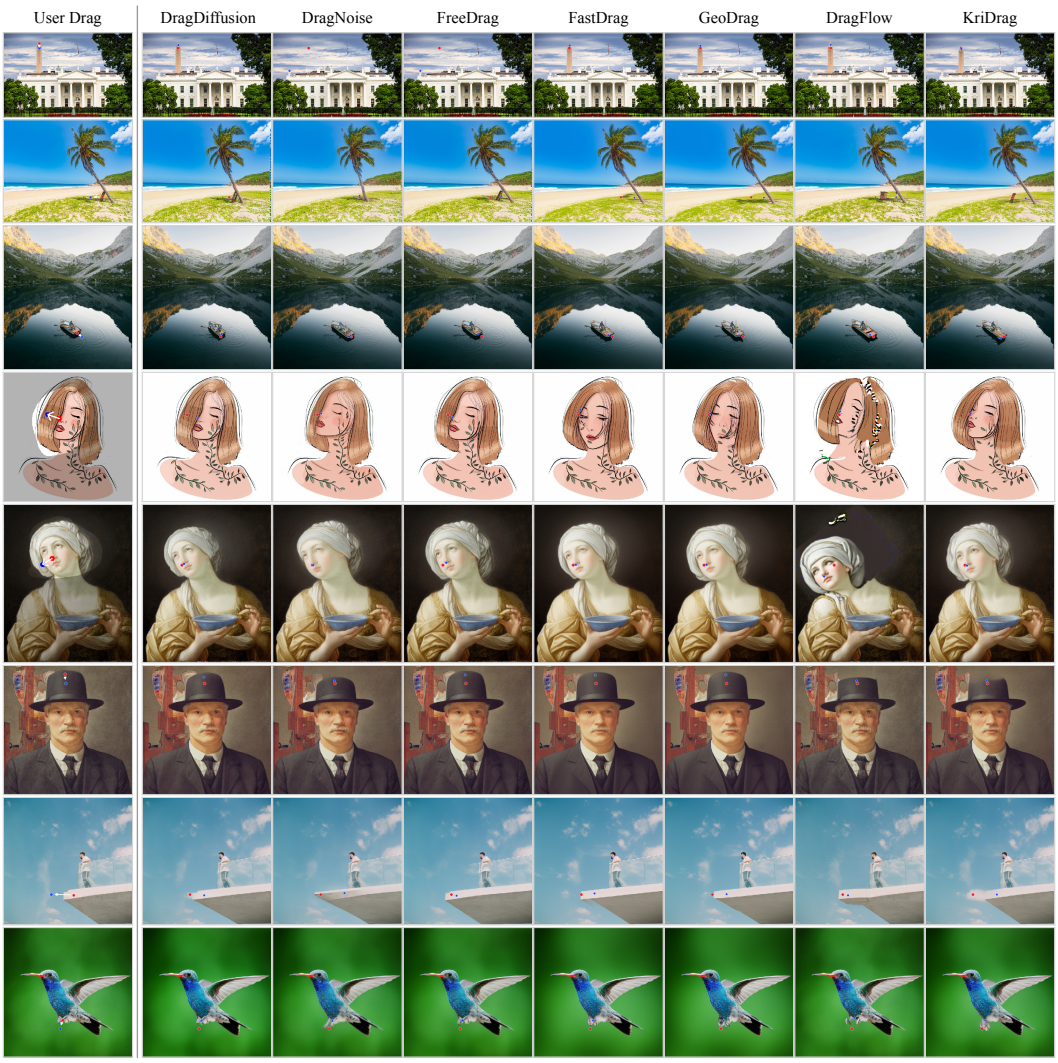

Figure S1: Qualitative comparison of single-handle drag editing on DragBench and ReD-Bench. Each row shows the same input and drag instruction evaluated by seven methods.

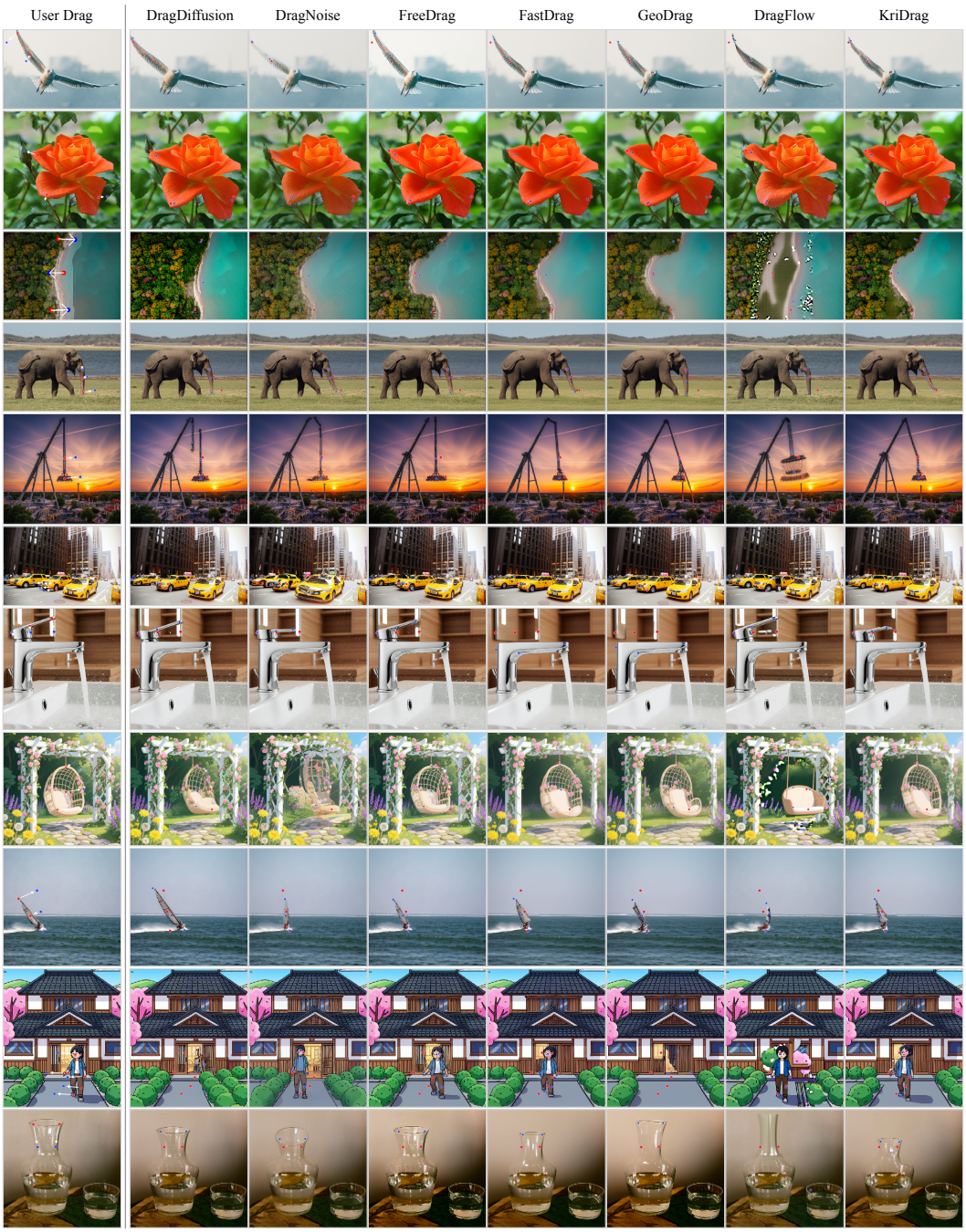

Figure S2: Additional seven-method qualitative comparison on selected multi-handle ReD-Bench cases. Each row uses the same input and drag instruction across methods. The thin gray rule separates the user drag from the edited outputs. In the input column, red and blue points denote source handles and target positions, respectively, and white arrows show the requested motion. In the output columns, red points mark requested targets and blue points show DIFT correspondences. Together with Figure 1, the examples cover rotations, translations, and articulated deformations across realistic and illustrated content.