## Supplementary Information

We conduct more details and results on Dragonite and organize the supplementary material in the following manner. We begin with more implementation details in Sec. A, followed by additional experiments and ablation study in Sec. B. We then provide a more comprehensive quantitative comparison in Sec. C and provide additional evaluation metrics in Sec. D. Subsequently, we include methodology details in Sec. E and conclude with more qualitative results in Sec. F.

## A. More Implementation Details

**Generated Drag Prompt.** To generate descriptive drag prompts for drag-based image editing operations, we employ GPT-4o, one of the most used multimodal model with visual understanding capabilities. For each image in the DragBench dataset, we generate corresponding drag prompts that capture the semantic intent of the edit operation. The GPT-4o prompt template used for generating drag prompt is structured as follows:

```
You are tasked with generating a caption for an image that visually
represents a drag operation.  The image features two distinct elements:  a
red dot and a blue dot.  The red dot indicates the starting point, while
the blue dot marks the destination after the drag.  Additionally, a
highlighted area shows the editable region involved in the operation.
Task:  Describe in a single, clear sentence the significant visual change
that occurred as a result of the drag operation, focusing on the movement
from the red dot to the blue dot within the editable region.  Your caption
should succinctly convey the transformation in a way that is easily
understandable and directly correlates with the visual representation.
Objective:  Your caption should help viewers quickly grasp the essence of
the change without needing to analyze the image in detail.  This will
enhance the accessibility and effectiveness of visual content in
presentations or educational materials.
```

**Components in Ablation Study.** In the ablation study presented in Tab. 2, we evaluate the impact of different components in Dragonite. The components are as follows: Modality, which refers to the use of multimodal inputs (image and drag prompt) in our method, "Geometric+Text" in the table refers to the use of both geometric and drag prompts, "Geometric" refers to the drag prompt set as null string. MAI, which stands for the use of our proposed Mean Adjusted Interpolation, while for the "No MAI" setting, we use the random sample noise interpolation method. Finally, the LSC just simply refers to the use of the LSC module in our method.

## B. Additional Experiments

**Mean Adjusted Interpolation.** We conduct an additional quantitative experiment to evaluate the effectiveness of mean adjusted interpolation in Dragonite. We compare different interpolation methods as mentioned in the main paper (Sec. 4.4), including: maintaining the original value, filling the position with zeros, filling with random sample noise, and bilateral nearest neighbor interpolation. These methods are denoted as "Original", "Zero", "Random", and "BNNI", respectively, and will be compared to our proposed Mean Adjusted Interpolation, denoted as "MAI". The results are shown in the left side of the Fig.1, our method achieves the best performance on mean distance (MD) compare to other methods, and achieves the second best performance on IF, only behind the "Origin" method, which is not a practical method in real-world applications. This demonstrates the effectiveness of our proposed Mean Adjusted Interpolation in both quantitative and qualitative aspects.

**Effect of Inversion Steps.** We analyze the trade-off between editability and reconstruction fidelity by varying the number of DDIM inversion/sampling steps $t \in \{4, 6, 8, 10, 12, 14, 20, 30\}$ and measuring the resulting MD and 1-LPIPS metrics. As shown in the right side of Fig. 1, 10 steps achieves the optimal balance: MD remains relatively low at 32.38 while maintaining high image fidelity (1-LPIPS = 0.86). Fewer steps (4-6) yield lower fidelity scores (1-LPIPS < 0.84), indicating poor reconstruction quality despite achieving lower MD. Beyond 10 steps, MD increases sharply, peaking at 36.79 for 14 steps. These results validate our choice of $t = 10$ as the sweet spot for high-quality drag-based editing.

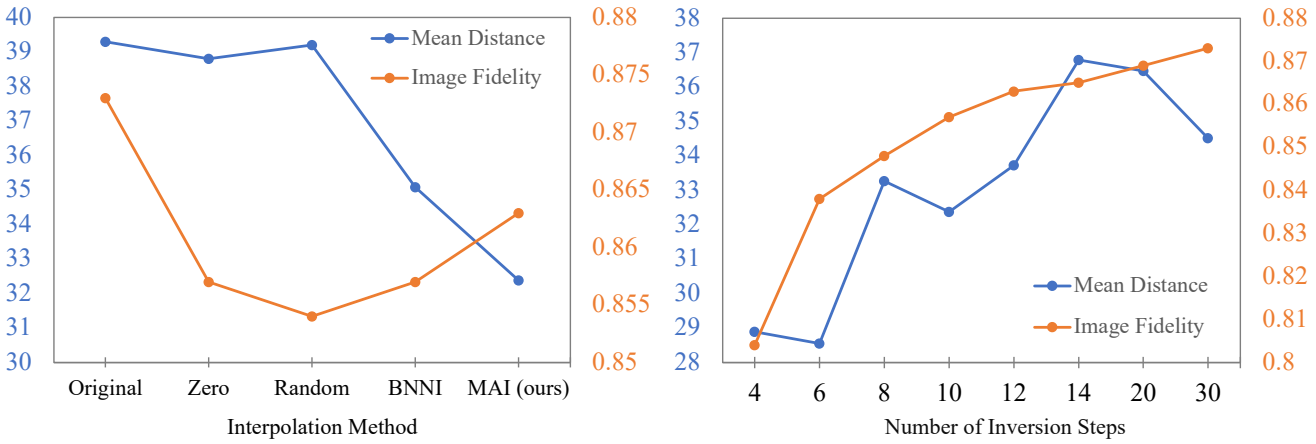


Figure 1. **Left.** Quantitative comparison of different interpolation methods compare to our proposed Mean Adjusted Interpolation (MAI). **Right.** Ablation study on number of inversion steps in terms of quantitative metrics.

**Text Fuse Coefficient.** We conduct an additional ablation study on the text fuse coefficient as discussed in the main paper (Eq.7), where the drag operations and the text prompt are in conflict, as shown in Fig. 2. When $\lambda$ is less than 100, geometric guidance dominates, and semantic guidance has minimal impact on the outcome. As $\lambda$ increases beyond 100, semantic guidance gradually becomes more evident in the solution. At $\lambda = 500$, the two forms of guidance begin to conflict, causing confusion in the editing results. By the time $\lambda$ reaches 800, semantic guidance starts to dominate. When $\lambda$ reaches over 1,000, semantic influence completely overrides geometric guidance. For instance, despite the drag operation intended to lengthen the dog's tongue, the semantic prompt "A dog with a shorter tongue" ultimately prevails, resulting in a visibly shorter tongue.

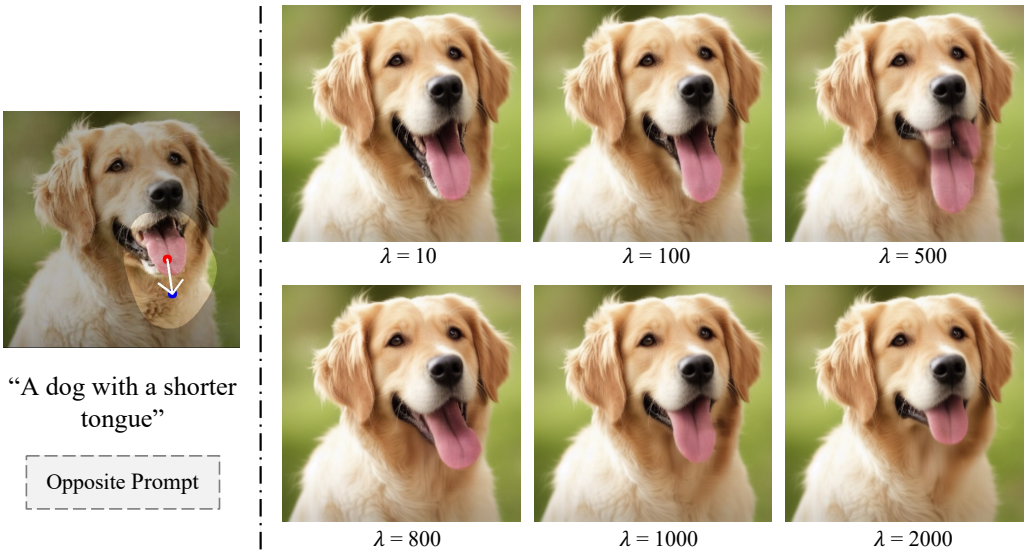


Figure 2. Ablation study on the text fuse coefficient $\lambda$ when drag operations and text prompts are conflicting. The results show how the text affects the editing results when the text fuse coefficient $\lambda$ is set to different values.

## C. More Quantitative Comparison.

Below, we provide a more quantitative comparison with drag-based image editing methods on DragBench. These methods can be categorized into three types: optimization-based (Optimize), training-based (Trainable), and training-free (Train-free) approaches. We will evaluate our method's performance against each of these categories.

| Method | Venue | Modality | MD ↓ | 1-LPIPS ↑ | Runtime ↓ | Prep. ↓ | Category |
|---|---|---|---|---|---|---|---|
| EasyDrag | CVPR'24 | Geometric | 38.46 | **0.91** | 47.22 | 1 min (LoRA) | Optimize |
| GoodDrag | ICLR'25 | Geometric | **25.13** | 0.87 | 76.7 | 1 min (LoRA) | Optimize |
| DragText | WACV'25 | Geometric | 32.97 | 0.87 | **46.29** | 1 min (LoRA) | Optimize |
| AdaptiveDrag | - | Geometric | 35.75 | 0.86 | 72.63 | 1 min (LoRA) | Optimize |
| InstantDrag | SIGGRAPH'24 | Geometric | 31.51 | 0.87 | **1.57** | ✗ | Trainable |
| LightningDrag | ICML'25 | Geometric | **29.95** | **0.88** | 2.18 | ✗ | Trainable |
| SDEDrag | ICLR'24 | Geometric | 44.48 | **0.91** | 65.7 | 1 min (LoRA) | Train-free |
| Dragonite (ours) | – | Geometric+Text | **32.38** | 0.86 | **3.85** | ✗ | Train-free |

Table 1. Quantitative comparison with computationally heavy methods on DragBench.

**Optimization-based Methods.** As shown in Tab. 1, GoodDrag stands out in the optimization-based category with the lowest MD score of 25.13, but this performance comes at a substantial computational cost, requiring an average of 76.7 seconds per image and each image requires additional preparation time, such as LoRA fine-tuning. This significant tradeoff is worth consider in practical applications. In contrast, our proposed Dragonite method requires only 3.85 seconds per image without any additional preparation overhead, representing an approximately 2,000% improvement in computational efficiency compared to GoodDrag. Despite this significant reduction in processing time, our method achieves competitive performance with an MD score of 32.38 and a 1-LPIPS score of 0.86, demonstrating that our approach strikes a superior balance between computational efficiency and generation quality.

**Training-based Methods.** Training-based methods, such as InstantDrag and LightningDrag, offer faster inference times compared to optimization-based approaches. As shown in Tab. 1, InstantDrag and LightningDrag achieve impressive MD scores of 31.51 and 29.95, respectively, with inference times of 1.57 and 2.18 seconds per image. However, these methods require extensive training on large datasets, which can be resource-intensive and time-consuming. InstantDrag trains on large-scale video data for 5 days on 8 RTX A6000 GPUs, while LightningDrag trains on 220k pairs of video frame. These computational resource requirements create significant barriers to entry and are particularly challenging for researchers who have limited computational resources. In contrast, our Dragonite method does not require any training and can be applied directly to any pre-trained diffusion model. We achieves competitive performance with only a 0.8 MD differences compared to InstantDrag and running just 1.6 seconds slower than LightningDrag, while eliminating the need for any training. This highlights the advantage of our approach in being more accessible and practical for wider usage, particularly for those with limited computational resources.

**Training-free Methods.** Compared to other training-free methods, Dragonite demonstrates superior performance, achieving an MD score of 32.38 versus SDEDrag's 44.48. Moreover, Dragonite shows remarkable efficiency, processing each image in just 3.85 seconds without requiring any preparation overhead. In contrast, SDEDrag requires 65.7 seconds per image and requires a 1-minute LoRA fine-tuning step for each image. This represents a 17-fold improvement in processing speed while simultaneously delivering better performance metrics. These substantial gains in both accuracy and efficiency highlight the effectiveness of our training-free approach.

## D. Other Evaluation Metric

We provide additional comparisons using the DAI and GScore metrics proposed in GoodDrag, which offer a different perspective to evaluate drag editing performance. As shown in Tab. 2, DAI (Dragging Accuracy Index) calculates the mean squared error between image patches centered at the original starting point and the target point in the edited image, quantifying how accurately semantic content is transferred to the designated location. The patch radius $\gamma$ can be adjusted to control the extent of context, where smaller values focus on precise point-level measurements while larger values encompass broader contextual information. Lower DAI values indicate more accurate drag editing. As shown in Tab. 3, GScore (Gemini Score) leverages Large Multimodal Models to assess the naturalness and perceptual quality of edited images on a scale from 0 to 10, where higher scores indicate better quality. It is important to note that GScore provides relative rankings rather than absolute quality scores, comparing multiple methods within the same evaluation context. Following the prompt design in GoodDrag,

| Method | $\gamma = 1$ | $\gamma = 5$ | $\gamma = 10$ | $\gamma = 20$ |
|---|---|---|---|---|
| DragDiffusion | 0.1477 | 0.1439 | 0.1298 | 0.1146 |
| FastDrag | 0.1296 | 0.1324 | 0.1239 | 0.1156 |
| CLIPDrag | 0.1414 | 0.2400 | 0.2374 | 0.2195 |
| Dragonite (ours) | **0.1231** | **0.1294** | **0.1211** | **0.1130** |

Table 2. Evaluation of drag accuracy in terms of DAI. $\gamma$ corresponds to the patch radius. Lower values indicate more accurate drag editing.

| Method | GScore ↑ |
|---|---|
| DragDiffusion | 6.47 |
| FastDrag | 6.82 |
| CLIPDrag | **7.41** |
| Dragonite (ours) | <u>7.33</u> |

Table 3. Evaluation of image quality in terms of GScore. The GScore is on a relative scale from 0 to 10, with higher scores indicating better quality.

we can compare up to four methods simultaneously, where the Gemini model evaluates and ranks the edited results based on their perceptual quality with the original image as a reference.

## E. Methodology Details

**Semantic Guidance.** The core idea of semantic guidance is to find the direction of change in the latent space that corresponds to the desired semantic edit. We utilizes the CLIP encoder to extract text feature and the image feature, then we calculate the semantic loss by cosine similarity, as shown in Eq. 3 in the main paper. Here comes the main questions: How to find the direction of change in the latent space that corresponds to the desired semantic edit? To address this, we found that the Jacobian matrix is a fundamental tool in calculus for understanding how a function's output changes in response to small changes in its input. The Jacobian matrix provides the best linear approximation of a non-linear function near a specific point. By using the Jacobian, we can captures how the semantic loss changes with respect to tiny changes in the latent code. The Jacobian matrix represent the gradient of each output dimension with respect to each input dimension and its elements indicate the sensitivity of the semantic loss to changes in individual dimensions of the latent code. By calculating this matrix, we can determine the most effective direction in the latent space to move in order to minimize the semantic loss, thereby better aligning the synthesized image with the target text prompt.

**Angle-aware Fusion.** To effectively integrate both semantic and geometric guidance, we draw inspiration from ProGrad, which fuses gradients according to the angle between them. We adopt this angular-based strategy to address potential conflicts between the two forms of guidance. Specifically, the method relies solely on the vectorial relationship, thereby providing a principled yet computationally efficient mechanism for gradient alignment. When the angle between the vectors is acute ($\cos(\theta) > 0$), the gradients are oriented in similar directions. In this case, the geometric guidance vector ($V_g$) is refined by adding the orthogonal component of the semantic guidance vector ($V_s$). This approach preserves the primary geometric direction while simultaneously incorporating complementary semantic information. When the angle is obtuse or orthogonal ($\cos(\theta) \leq 0$), the gradients diverge or oppose each other. To prevent semantic guidance from interfering with the primary objective, the conflicting semantic component is suppressed. This ensures the preservation of geometric fidelity while avoiding detrimental deviations in the fused vector.

## F. More Qualitative Results

Below, we provide additional qualitative results on the DragBench dataset.

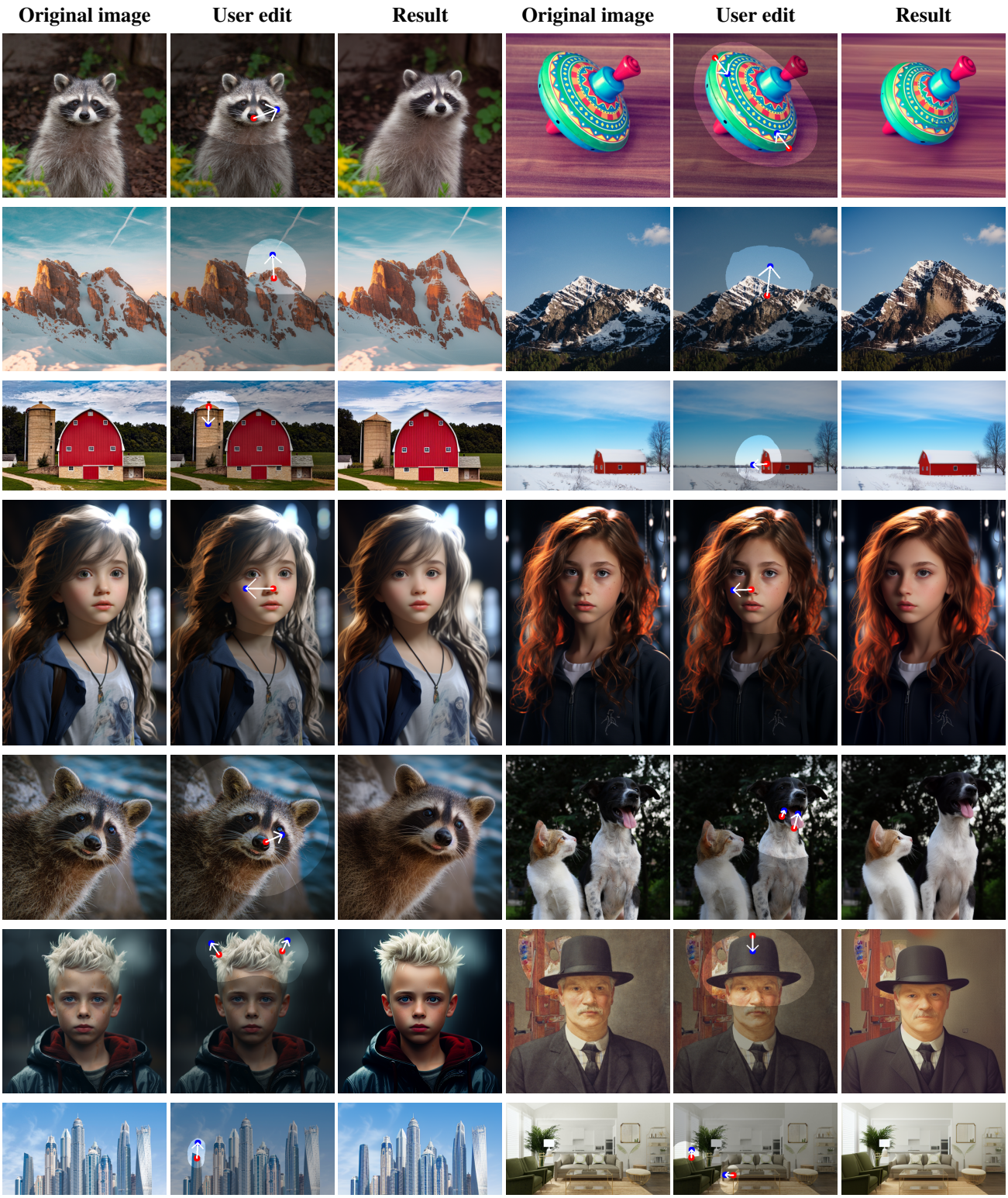


Figure 3. More qualitative results of Dragonite on the DragBench dataset.

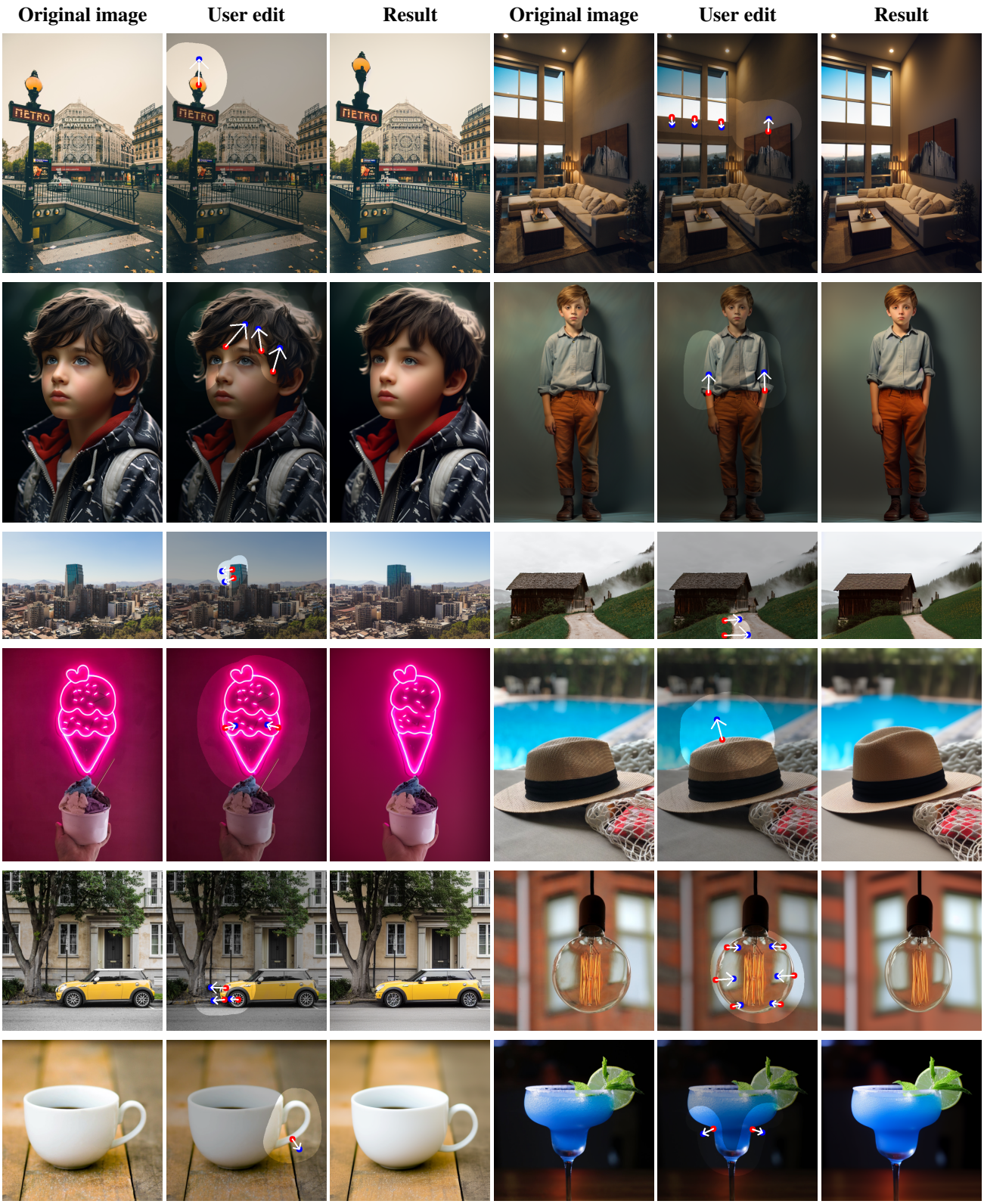


Figure 3. More qualitative results of Dragonite on the DragBench dataset.