EasyDrag: Efficient Point-based Manipulation on Diffusion Models

Supplementary Material

6. Algorithm for our pipeline

We present our pipeline in Algorithm 2.

7. Links to SD-1.5 fine-tuned variants.

We provide the links to the SD-1.5 fine-tuned variants used by us:

Majicmix Realistic V7:
https://civitai.com/models/43331/majicmix-realistic,
Interior Design Supermix:
https://civitai.com/models/28112/xsarchitectural-interiordesign-forxslora,
Dvarch:
https://civitai.com/models/8552/dvarch-multi-prompt-architecture-tuned-model.

8. More qualitative results

Comparison. To better demonstrate the effectiveness of our method, we present additional qualitative results in Fig. 12. As shown in the last line, the user wants to drag the left pillow upward while keeping the right one in place. However, DragDiffusion, fine-tuned with LoRA, learns features indicating that both pillows should align, preventing the desired dragging. On the other hand, in our method, with a provided mask encompassing a small portion of the right pillow, an additional area appears on the right side. In the case of our auto-generated mask, it accurately drags only the left pillow upward because the automatically generated mask precisely matches the dragging point.

Auto mask generation. The comparison in Tab. 1 reveals that our method achieves the highest image fidelity (IF), surpassing even the results obtained with masks. This is mainly because the automatically generated mask is very compact and changes with the variation of points. As shown in Fig. 13, the automatically generated mask region is very small, but it can achieve extensive dragging. This results in larger regions remaining unchanged compared to manually drawn masks. As IF calculates the similarity between the generated result and the original image, the outcomes in the case of automatically generated masks will be closer to the original image.

Reference guidance. In Fig. 6, we demonstrate the editing of out-of-distribution images. Here, we provide additional examples to explain the role of reference guidance in Fig. 11. Whether it's in-distribution (sofa) or out-of-distribution (robot), after dragging, the details of the entire image change. When reference guidance is small, the background can maintain consistency, but there may be residual flaws in the editing area. As we increase the reference guidance, the flaws gradually disappear. However, when the reference guidance is too large, the generated image may have some issues with color or background. This is mainly because excessive reference guidance can cause the generated image to go beyond the generating space of the original image. Therefore, an appropriate reference guidance is necessary (as mentioned in Sec. 4.3, in the range of 3 to 5).

Algorithm 2: Pipeline of EasyDrag

Input: Input image $\mathcal{I}$; Prompt embedding $\mathcal{C}$; Handle points $\boldsymbol{p}$; Target points $\boldsymbol{q}$;
Output: Output image $\mathcal{I}^*$;

Compute the intermediate results $z_t, ..., z_0$ using DDIM inversion over $\mathcal{I}$;
Initialize $\boldsymbol{p}^* \leftarrow \boldsymbol{p}, z_t^* \leftarrow z_t, \boldsymbol{p}^{**} \leftarrow next(\boldsymbol{p}^*, \boldsymbol{q})$;
Extract the intermediate features $\mathbf{F}_0$ from $\epsilon_\theta(z_t, t, \mathcal{C})$;
Initialize mask $\mathbf{M}$ with $\mathcal{L}_{cos}(\mathbf{F}_0(\boldsymbol{p}^{**}), \mathbf{F}_0(\boldsymbol{p}))$;
while $\boldsymbol{p}^*$ not reach $\boldsymbol{q}$ **do**
    Extract the intermediate features $\mathbf{F}$ from $\epsilon_\theta(z_t^*)$;
    for $j = 0, 1, ..., N-1$ **do**
        if $cos(\boldsymbol{F}(\boldsymbol{p}_j^{**}), \boldsymbol{F}_0(\boldsymbol{p}_j)) > cos(\boldsymbol{F}(\boldsymbol{p}_j^{*}), \boldsymbol{F}_0(\boldsymbol{p}_j))$ **then**
            $\boldsymbol{p}_j^* \leftarrow \boldsymbol{p}_j^{**}$;
            $\boldsymbol{p}_j^{**} \leftarrow next(\boldsymbol{p}_j^*, \boldsymbol{q}_j)$;
        end
    end
    if any point updated **then**
        Update mask $\mathbf{M}$ with $\mathcal{L}_{cos}(\mathbf{F}(\boldsymbol{p}^{**}), \mathbf{F}_0(\boldsymbol{p}))$;
    end
    $\mathcal{L} \leftarrow \mathcal{L}_{cos}(\mathbf{F}(\boldsymbol{p}^{**}), \mathbf{F}_0(\boldsymbol{p})) + \lambda \mathcal{L}_{cos}(\mathbf{F} * (1-\mathbf{M}), \mathbf{F}_0 * (1-\mathbf{M}))$;
    $z_t^* \leftarrow z_t^* - \eta \nabla_{z_t^*} \mathcal{L}$;
end
for $i = t, t-1, ..., 1$ **do**
    $\epsilon_i^* \leftarrow w\epsilon_\theta(z_i^*, z_i, i, \mathcal{C}) + (1-w)\epsilon_\theta(z_i^*, i, \mathcal{C})$;
    $z_{i-1}^* \leftarrow \sqrt{\frac{\alpha_{i-1}}{\alpha_i}} z_i^* + \sqrt{\alpha_{i-1}}\left(\sqrt{\frac{1}{\alpha_{i-1}} - 1} - \sqrt{\frac{1}{\alpha_i} - 1}\right)\epsilon_i^*$;
end
$\mathcal{I}^* \leftarrow Decoder(z_0^*)$;
return $\mathcal{I}^*$

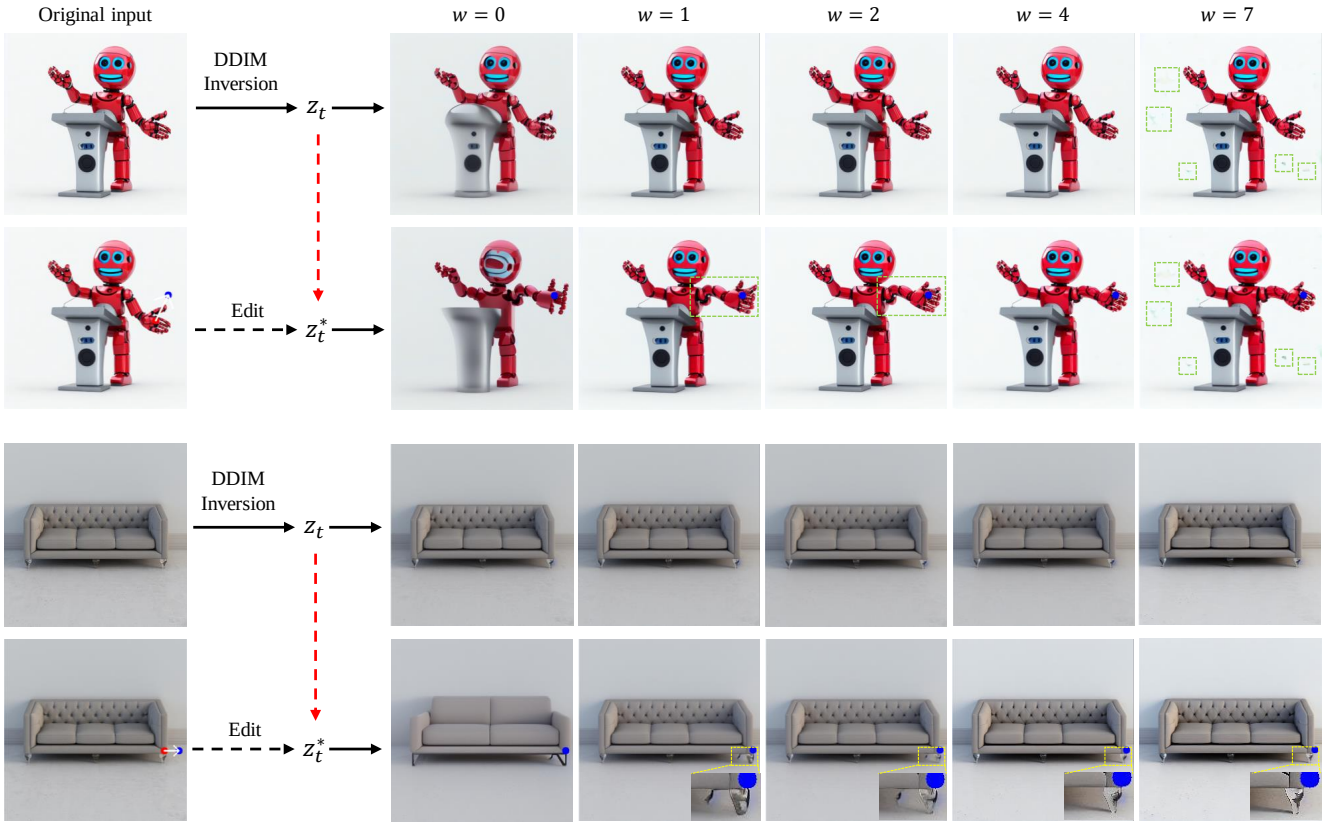

Figure 11. Effect of reference guidance. The edited image always cannot preserve consistency to the original image without reference guidance.

Motion supervision. In Tab. 2, we compare the quantitative results of different "drag" method. We also present the qualitative comparison in Fig. 14.

9. Videos for dragging trajectories

In the supplementary material, we also provide the dragging trajectories in the "video" folder. It is important to note that in the actual usage of EasyDrag, intermediate results are not outputted, as denoising at each update of handle points can be time-consuming.

Figure 12. More qualitative comparison of our approach and DragDiffusion. Our method always performs better.

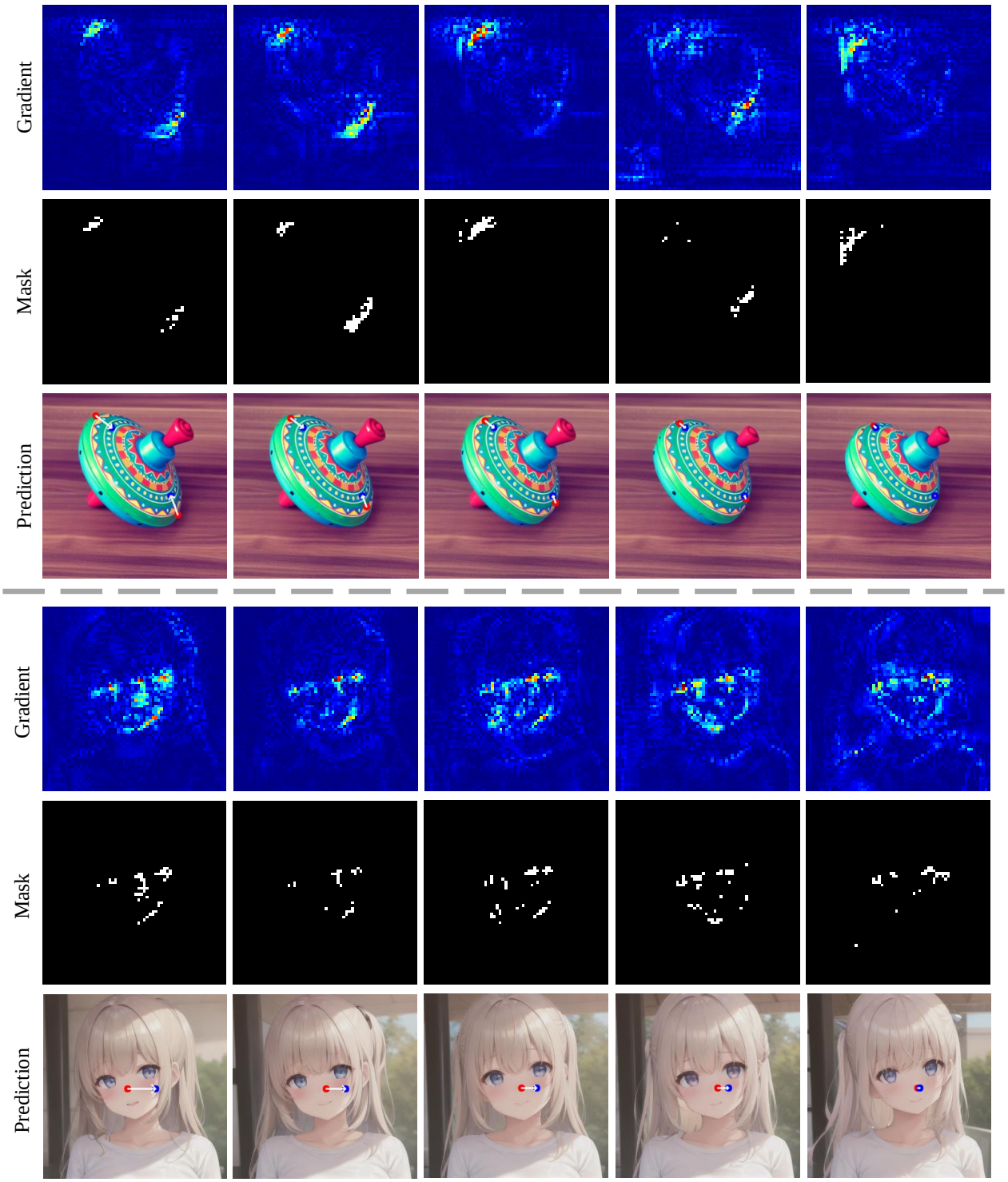

Figure 13. The changes in gradient, mask and prediction of z_t during optimization. The achievement of extensive dragging goals requires only a very small editable area.

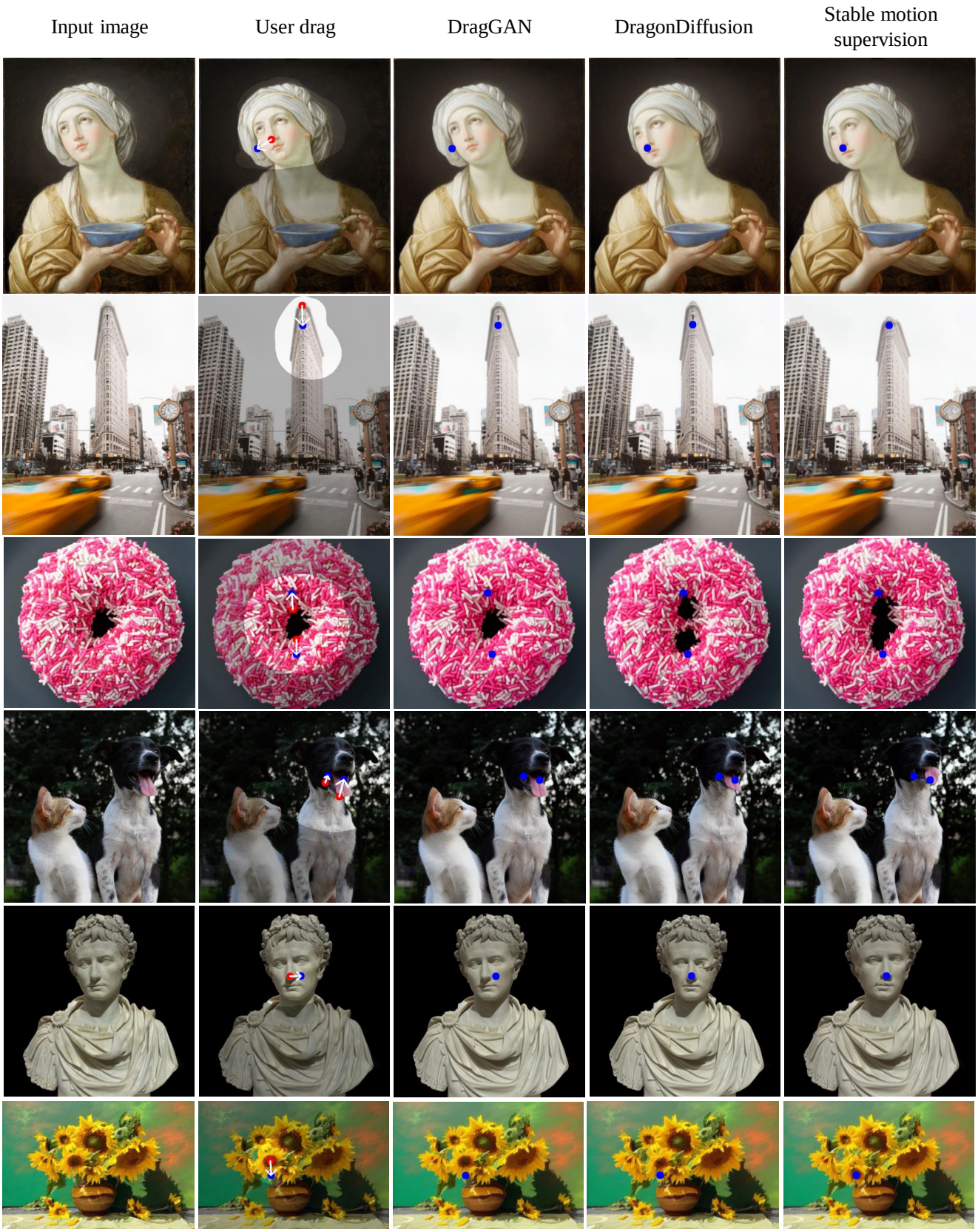

Figure 14. Different "drag" method. Our stable motion supervision achieves more accurate results.