

QK-Edit: Revisiting Attention-based Injection in MM-DiT for Image and Video Editing

Supplementary Material

Algorithm 1: QK-Edit

Input: 1. T_list : timesteps to replace the Q_{Vision} and K_{Vision}
2. QK_list : record Q_{Vision}^{src-t} and K_{Vision}^{src-t} of every timestep t
Output: an edited latent embedding z^{tgt-0}

Part 1: Inversion to record QK from source

for $t = 1, \dots, T-1, T$ **do**

$Q^{src-t-1} = [Q_{Text}^{src-t-1}, Q_{Vision}^{src-t-1}]$
 $K^{src-t-1} = [K_{Text}^{src-t-1}, K_{Vision}^{src-t-1}]$
 $V^{src-t-1} = [V_{Text}^{src-t-1}, V_{Vision}^{src-t-1}]$
 $[Q_{Vision}^{src-t}, K_{Vision}^{src-t}], z^{src-t} = \text{Inversion}([Q^{src-t-1}, K^{src-t-1}, V^{src-t-1}], z^{src-t-1}, t-1)$

$QK_list.append([Q_{Vision}^{src-t}, K_{Vision}^{src-t}])$

end

Part 2: Performing editing with QK replacement

for $t = T, T-1, \dots, 1$ **do**

if t in T_list **do**

$[Q_{Vision}^{tgt-t}, K_{Vision}^{tgt-t}] = QK_list[t]$

$Q^{tgt-t} = [Q_{Text}^{tgt-t}, Q_{Vision}^{tgt-t}]$

$K^{tgt-t} = [K_{Text}^{tgt-t}, K_{Vision}^{tgt-t}]$

$V^{tgt-t} = [V_{Text}^{tgt-t}, V_{Vision}^{tgt-t}]$

$z^{tgt-t} = \text{Forward}([Q^{tgt-t}, K^{tgt-t}, V^{tgt-t}], z^{tgt-t-1}, t)$

end

Return z^{tgt-0}

Figure 9. The pseudo code of QK-Edit.

6. Pseudo Code

To illustrate the simplicity of our QK-Edit, we provide the pytorch-style pseudo code for our algorithm in Fig. 9. This pseudo code demonstrates the straightforward implementation of QK-Edit.

7. Additional Image Editing Results

Additional qualitative results for image editing are provided in Fig. 11. These examples further demonstrate QK-Edit’s ability to preserve fine-grained details, maintain structural consistency, and achieve precise semantic alignment across diverse editing scenarios.

8. Video Editing Results for High-Resolution

Please refer to the accompanying MP4 files for video editing results at higher resolutions, longer frames, and two different aspect ratios. Specifically, the two files contain results for videos with dimensions of 1280×720 with 129 frames (1280_720_129.mp4) and 720×1280 with 65

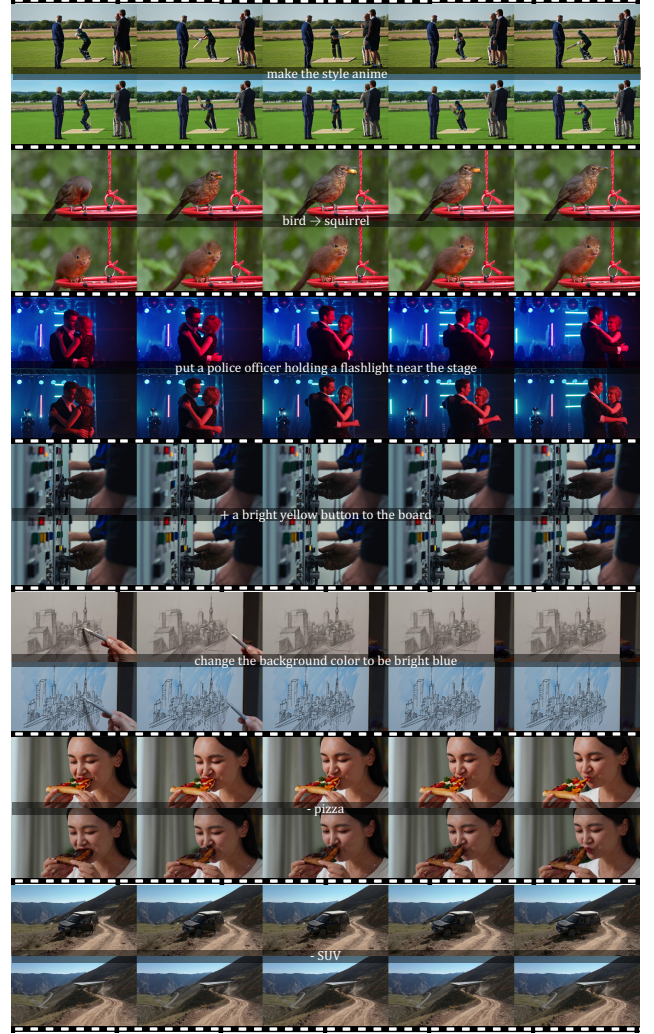


Figure 10. Failure cases and artifacts.

frames (720_1280_65.mp4), showcasing the flexibility of our method on both landscape and portrait formats.

9. Failure Cases and Artifacts

As a training-free method, our performance is inherently bounded by the capabilities of the base model. We have discussed some failure cases in the Limitation section. More failure cases are illustrated in Fig. 10.

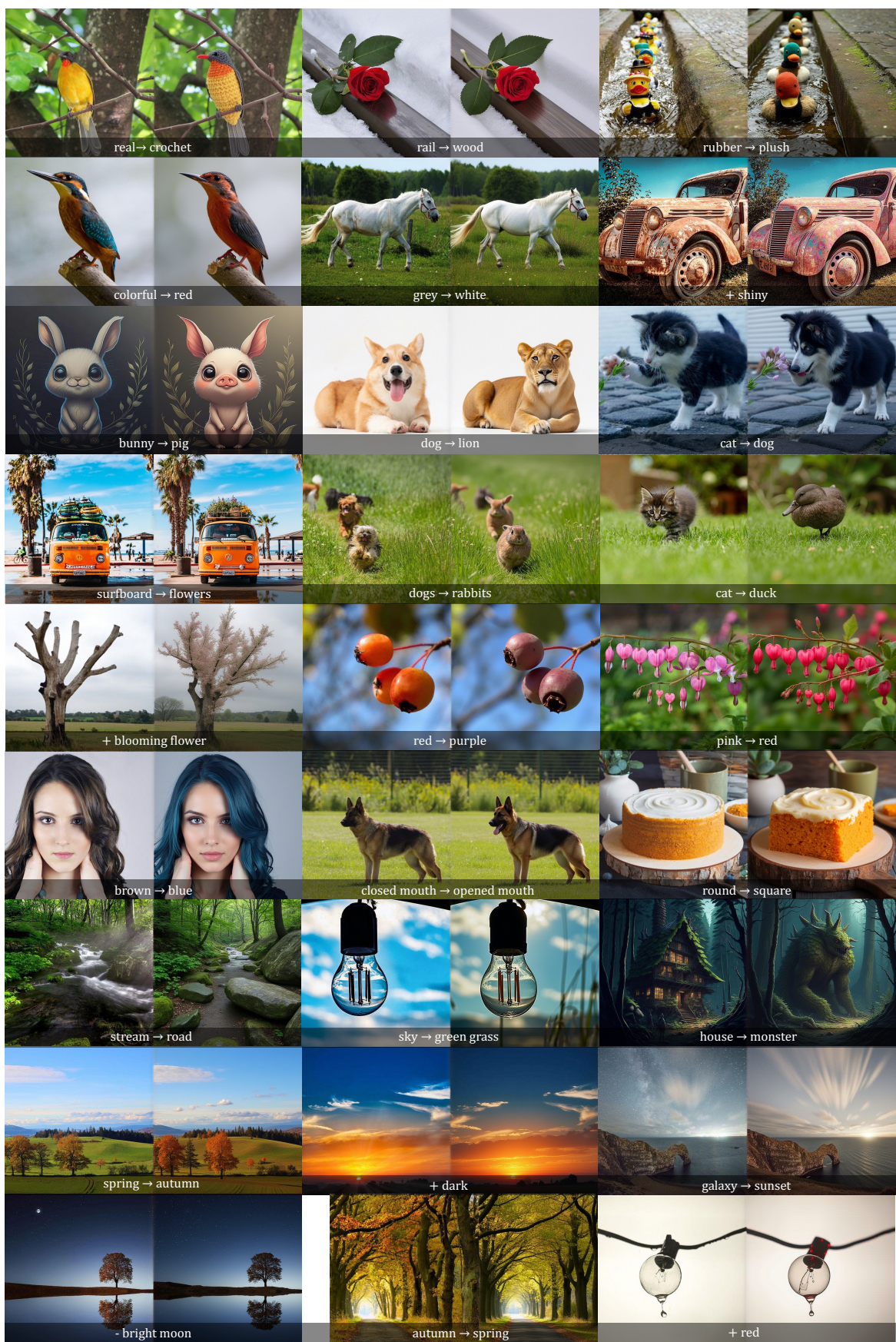


Figure 11. Additional image editing results.