1. IMPLEMENTATION DETAILS

1.1. Iterative Editing

We show the implementation details of iterative editing in Algorithm 1, following some formulations in Prompt-to-Prompt [1].

**Algorithm 1** Algorithm for iterative editing.

**Input:** All prompts \( \{P^1, P^2, \ldots, P^K\} \), a random seed \( \xi \), and words \( \{(w^1, w^2), (w^2, w^3), \ldots, (w^{K-1}, w^K)\} \) specifying the editing region for local editing.

**Output:** All images \( x^1, x^2, \ldots, x^K \).

1. \( z^T \sim N(0, I) \), a unit Gaussian random variable with \( \xi \);
2. \( (z_1^T, z_2^T, \ldots, z_K^T) \leftarrow z_T \);
3. for \( t = T, T - 1, \ldots, 1 \) do
4. \( z_{k-1}^T, M_{k-1}^t \leftarrow DM(z_{k-1}^T, P_{k-1}^t, t, \xi) \);
5. for \( k = 2, \ldots, K \) do
6. \( z_{k-1}^t \leftarrow DM(z_{k-1}^T, P_k, t, \xi) \);
7. \( M_k^t \leftarrow DM(z_k^T, P_k, t, \xi) \);
8. \( \widetilde{M}_k^t \leftarrow Edit(M_k^{t-1}, M_k^t, t) \);
9. \( z_k^t \leftarrow DM(z_k^T, P_k, t, \xi) \{M \leftarrow \widetilde{M}_k^t \} \);
10. if local then
11. \( \alpha \leftarrow B(\widetilde{M}_{t,(wk)}^k) \cup B(\widetilde{M}_{t,(wk)}) \);
12. \( z_{k-1}^t \leftarrow (1 - \alpha) \odot z_{k-1}^t - \alpha \odot z_{k-1}^t \);
13. end if
14. end for
15. end for
16. \( x^1, x^2, \ldots, x^K = Decode(z_0^1, z_0^2, \ldots, z_0^K) \).

1.2. Training Details

We choose the pre-trained Stable Diffusion v1.4 in our pipeline. When sampling the original and the edited images, we run 50 inference steps with a classifier-free guidance scale of 7.5. During the optimization of strokes, we train 1000 iterations for each example. We use 96 strokes each of which includes 4 control points to represent the vector sketches. The stroke width is defined with a fixed value 1.0. In the stroke-level local editing scheme, we adopt cross-attention maps of resolution \( 16 \times 16 \) in up and down blocks in the diffusion model.

2. MORE RESULTS

We show more results including:

- Comparisons in Word Swap mode: Fig. 1;
- Comparisons in Prompt Refinement mode: Fig. 2;
- Attention Re-weighting mode: Fig. 3;
- Iterative editing: Fig. 4 and 5.

3. REFERENCES

Fig. 1. Comparisons with baseline methods in Word Swap mode.
Fig. 2. Comparisons with baseline methods in Prompt Refinement mode.
Fig. 3. Results in Attention Re-weighting mode.
"A painting of a squirrel eating a burger"

seed=25760

"squirrel" ➔ "rabbit"
"burger" ➔ "pumpkin"

seed=51355

"A kangaroo near a car"

"kangaroo" ➔ "deer"
"deer" ➔ "tiger"

seed=609

"A kangaroo near a car"

"kangaroo" ➔ "deer"
"deer" ➔ "tiger"

seed=53487

"An evening dress"

"... with sleeves"
"... and a belt"

seed=4804

"An evening dress"

"... with long sleeves"
"... and belt"

Fig. 4. Results of iterative editing.
Fig. 5. Results of iterative editing.