The Devil is in Attention Sharing: Improving Complex Non-rigid Image Editing Faithfulness via Attention Synergy

Supplementary Material

In the supplementary material, we first provide the implementation details of *SynPS* in Sec. 1. Then, we provide the detailed derivation of *SynPS* in Sec. 2. Next, we present more visualization cases under complex non-rigid instructions, along with the comparison with the compared baselines in Sec. 3. Additionally, we provide more ablation studies and analysis in Sec. 4. Finally, we discuss limitations and potential social impact in Sec. 5.

1. Experimental Setups

1.1. Implementation Details

All experiments are conducted at a resolution of 512×512. We follow the FLUX.1-dev official recommended hyperparameters, using 50 sampling steps and a guidance scale of 3.5 by default. In addition, we perform attention sharing in the position-insensitive blocks `[0, 7, 8, 9, 10, 18, 25, 28, 37, 42, 45, 50, 56]` across all timesteps, following FreeFlux [7].

1.2. Details of Compared Methods

As explained in Sec.5.1, we compare our method with state-of-the-art training-free image editing baselines under complex non-rigid instructions, adopting the same FLUX.1 [3] as generation backbone. Among them, RF-Solver-Edit [5], FlowEdit [2] and StableFlow [1] are general-purpose editing methods, while CharaConsist [6] and FreeFlux [7] are specifically designed for non-rigid editing. All compared baselines are reproduced with their default settings.

For RF-Solver-Edit, StableFlow, CharaConsist, and FreeFlux, we use the same initial noise for all methods and generate the source and target results by applying the source prompt and target prompt, respectively. For RF-Solver-Edit, StableFlow, CharaConsist, and FreeFlux, we use the same initial noise for all methods and generate the source and target results by applying the source prompt and target prompt, respectively. We also follow the official FLUX.1-dev recommended configuration, using 50 sampling steps and a guidance scale of 3.5 by default. This ensures a fair comparison under identical stochastic conditions. Detailed implementations are as follows:

- **RF-Solver-Edit** [5]: We follow the official implementation and set `inject_step` to 4, meaning that during the first four denoising steps, the *Value* tokens in blocks [39, 40, . . . , 56] are replaced from the source to the target.
- **StableFlow** [1]: We adopt the official implementation, which applies attention sharing with RoPE to the vital blocks [0, 1, 17, 18, 25, 28, 53, 54, 56] across all timesteps.
- **CharaConsist** [6]: We adapt the official implementation to fit our evaluation protocol. Following the original design in Characonsist [6], we first perform 11 steps of target pre-generation, then compute the correspondence, and modify the position IDs of the source image accordingly. CharaConsist subsequently conducts point-tracking attention and adaptive token merging from the first sampling step until the 40th step, operating on all single blocks. Unlike our method, which replaces the source-image KV tokens with those of the target image, CharaConsist concatenates the source-image KV tokens to the target-image KV tokens during attention sharing. Additionally, CharaConsist requires carefully engineered prompts consisting of three separate components: foreground, background, and action. Such decomposed prompts are not available in our evaluation setting under complex non-rigid instructions. After extensive analyses, we use the same prompts from our evaluated benchmarks and accordingly bypass the foreground–background mask computation in CharaConsist, applying attention sharing to the entire image without masking. As none of the other compared methods rely on mask computation, this adjustment ensures direct and fair comparison of the core contributions.
- **FreeFlux** [7]: We use the official implementation, which performs attention sharing with RoPE in the position-insensitive blocks [0, 7, 8, 9, 10, 18, 25, 28, 37, 42, 45, 50, 56] across all timesteps.
- **FlowEdit** [2] operates directly on the input image (in contrast to the other methods that modify intermediate attention states under a fixed-seed generative setting), and we provide the FLUX.1-dev generated source image as input to FlowEdit. This places FlowEdit in a fundamentally more challenging setting, rendering the comparison somewhat unfair. Therefore, we emphasize that the comparison with FlowEdit is only intended to analyze the differences between direct generation and inversion-free editing, rather than to demonstrate the superiority of our method. We first generate the source image using the FLUX.1-dev default configuration with the source prompt and feed the generated image into FlowEdit, ensuring that all methods share the same source image. FlowEdit is run with its official recommended settings: 28 inference steps, `src_guidance_scale`=1.5, `tar_guidance_scale`=5.5, `n_avg`=1, `n_min`=0,

n_max=24, and seed=10.

1.3. Implementation Details of MLLM-based Evaluation

We employ GPT-4o, GPT-5, and Gemini-2.5-Pro as our evaluation models, which are widely recognized as state-of-the-art MLLMs. All evaluations are conducted using their official APIs. To reduce stochasticity, we set the temperature to 0 and query each model three times per case, reporting the average score.

For each evaluation instance, we provide the source prompt, target prompt, source image, and target image to the MLLM, along with the following instruction prompt:

> As a Dynamic Transformation Evaluator, your primary role is to assess the quality, realism, and appearance consistency of an object's non-rigid transformation (such as pose, structure, or shape deformation) within a scene. You will be given two images — an original version (source image) and an edited version (target image) — along with the source prompt and target prompt describing the intended transformation.
>
> Your task is to evaluate whether the transformation appears natural, physically plausible, and visually coherent, with special attention to the appearance consistency of the main subject. Specifically, assess: the realism of the object's pose or structural deformation; the consistency of the subject's appearance, including shape integrity, color tone, texture continuity, lighting conditions, and material properties before and after editing; the preservation of overall scene coherence and non-edited region fidelity; and the accuracy of environmental interactions (e.g., contact points, shadows, reflections, and surface support).
>
> You must provide your evaluation strictly in the following dictionary format: {"score": 10, "reason": "Explanation here."}
>
> Rate the transformation quality on a scale from 0 to 10, where 0 indicates no observable transformation or a visually inconsistent edit, and 10 indicates a perfectly executed, realistic, and appearance-consistent transformation.
>
> When comparing the two images, look for visual evidence of the intended transformation—even subtle changes count. Consider partial success when the transformation partially maintains realism and subject appearance consistency.

1.4. CLIP$_{img}$ Analysis

Non-rigid editing is an inherently complex task that involves multiple aspects, including pose transformation, scene layout changes, object shape deformation, facial expression variation, and viewpoint shift. Such complexity requires a comprehensive evaluation protocol. However, the CLIP$_{img}$ similarity score only measures the global similarity between the source and target images in CLIP latent representation space, and thus cannot evaluate how well the appearance of the transformed subject is preserved.

Notably, StableFlow [1] and CharaConsist [6] interpret higher CLIP$_{img}$ scores as better, whereas FreeFlux argues the opposite and interprets lower scores as better. In our work, we do not use CLIP$_{img}$ as an editing-quality metric. Instead, we use it solely to assess the similarity to the source image for analyzing whether duplicate artifacts are produced.

2. Details of RoPE in *SynPS*

2.1. Derivation of RoPE

As shown in Eq.2 in the main paper, RoPE [4] is applied to the query token $[Q_{img}]_{i,j}$ at spatial location (i, j):

$$\text{RoPE}([Q_{img}]_{i,j}, i, j) = R_{i,j}\,[Q_{img}]_{i,j}, \tag{1}$$

where $R_{i,j}$ is a block-diagonal rotation matrix parameterized by the 2D position id $[i, j]$.

In FLUX.1-dev [3], the position ID of each image token is a 3-dimensional vector $[0, i, j]$. The Q/K feature dimension is 3072, split into 24 attention heads, each of dimension 128. Each 128-dimensional head is further partitioned into three contiguous segments of sizes $[16, 56, 56]$, corresponding to the three position-id components 0, i, and j, respectively. RoPE is applied to each segment independently using its associated position-id value. In the following, we derive the RoPE transformation for a single attention head.

We now focus on the subvector associated with the row index i. Let $[Q_{img}]_{i,j}[16:72] \in \mathbb{R}^{56}$ denote the 56-dimensional segment corresponding to the second component of the position id. The RoPE transformation for this segment can be written in matrix form as

$$\text{RoPE}([Q_{img}]_{i,j}[16{:}72], i, j) = R_i\,[Q_{img}]_{i,j}[16{:}72], \tag{2}$$

where $R_i \in \mathbb{R}^{56\times 56}$ is the rotation matrix determined solely by the row index i.

Let $q_i \triangleq [Q_{img}]_{i,j}[16:72] \in \mathbb{R}^{56}$. Following RoFormer [4], we decompose q_i into 2-D subvectors:

$$\mathbf{q}_i^{(k)} = \begin{bmatrix} q_{i,2k} \\ q_{i,2k+1} \end{bmatrix} \in \mathbb{R}^2, \qquad \text{for} \quad k = 0, 1, \ldots, 27, \tag{3}$$

so that $56 = 2 \times 28$ such subvectors exist.

We assign an angular frequency θ_k to each pair:

$$\theta_k = 10000^{-\frac{2k}{d_i}}, \qquad d_i = 56, \quad \text{for} \quad k = 0, 1, \ldots, 27. \tag{4}$$

Given the row index i, the rotation angle for the k-th pair is $\phi_k(i) = i\,\theta_k$.

2-D rotation. RoPE applies a 2-D rotation to each $\mathbf{q}_i^{(k)}$:

$$\tilde{\mathbf{q}}_i^{(k)} = R_i^{(k)} \mathbf{q}_i^{(k)}, \qquad R_i^{(k)} = \begin{bmatrix} \cos\big(\phi_k(i)\big) & -\sin\big(\phi_k(i)\big) \\ \sin\big(\phi_k(i)\big) & \cos\big(\phi_k(i)\big) \end{bmatrix}. \tag{5}$$

Explicitly,

$$\tilde{q}_{i,2k} = q_{i,2k} \cos\big(\phi_k(i)\big) - q_{i,2k+1} \sin\big(\phi_k(i)\big), \tag{6}$$

$$\tilde{q}_{i,2k+1} = q_{i,2k} \sin\big(\phi_k(i)\big) + q_{i,2k+1} \cos\big(\phi_k(i)\big). \tag{7}$$

Block-diagonal rotation matrix R_i. Stacking all 28 rotated pairs yields:

$$\tilde{q}_i = \big[\tilde{q}_{i,0}, \tilde{q}_{i,1}, \ldots, \tilde{q}_{i,55}\big]^\top \in \mathbb{R}^{56}. \tag{8}$$

The full rotation matrix is block-diagonal:

$$R_i = \mathrm{diag}\big(R_i^{(0)}, R_i^{(1)}, \ldots, R_i^{(27)}\big) \in \mathbb{R}^{56\times 56}. \tag{9}$$

Thus the RoPE transform on the segment $[16:72]$ is denoted as:

$$\mathrm{RoPE}([Q_{img}]_{i,j}[16:72], i, j) = R_i\,[Q_{img}]_{i,j}[16:72], \tag{10}$$

where R_i is a position-dependent orthogonal linear map determined solely by the row index i.

Applying the same construction to the 16-dimensional segment associated with the fixed position-id value 0 and the 56-dimensional segment associated with the column index j yields three independent rotation blocks. Together they form the block-diagonal rotation matrix $R_{i,j}$ for one head. Extending this operation to all 24 attention heads produces the complete RoPE transformation on the full Q/K feature tensor:

$$\mathrm{RoPE}([Q_{img}]_{i,j}, i, j) = R_{i,j}\,[Q_{img}]_{i,j}, \tag{11}$$

where $R_{i,j}$ is the block-diagonal rotation matrix assembled from the per-head matrices $R_{i,j}$ repeated across all heads, acting on the entire 3072-dimensional Q/K feature vector.

2.2. Modulation in *SynPS*

As shown in Eq.8 in the main paper, it can be expanded as the 2D image situation:

$$\begin{aligned} &\langle \mathrm{RoPE}([Q]_{i,j}, w\cdot i, w\cdot j),\ \mathrm{RoPE}([K]_{i',j'}, w\cdot i', w\cdot j')\rangle \\ &= (R_{wi,wj}[Q]_{i,j})^\top (R_{wi',wj'}[K]_{i',j'}) \\ &= [Q]_{i,j}^\top (R_{wi,wj}^\top R_{wi',wj'})[K]_{i',j'} \\ &\overset{(a)}{=} [Q]_{i,j}^\top R_{w(i'-i),\,w(j'-j)}[K]_{i',j'}, \end{aligned} \tag{12}$$

where the proof of step (a) in Eq. 12 is given below.

Recall that for the k-th 2D subspace in RoPE, the rotation angle at position i is given by

$$\phi_k(i) = i\,\theta_k, \tag{13}$$

where θ_k is the angular frequency associated with that pair of channels. When we scale the position index by a factor w, i.e., use $w\cdot i$ instead of i, the corresponding angle becomes

$$\tilde{\phi}_k(i) \triangleq \phi_k(wi) = (wi)\,\theta_k = w\,(i\,\theta_k) = w\,\phi_k(i). \tag{14}$$

Thus, scaling the position index by w linearly scales the rotation angle for every frequency k by the same factor w.

For the k-th 2D subspace, the rotation matrices at positions wi and wi' are

$$\begin{aligned} R_{wi}^{(k)} &= \begin{bmatrix} \cos(\tilde{\phi}_k(i)) & -\sin(\tilde{\phi}_k(i)) \\ \sin(\tilde{\phi}_k(i)) & \cos(\tilde{\phi}_k(i)) \end{bmatrix}, \\ R_{wi'}^{(k)} &= \begin{bmatrix} \cos(\tilde{\phi}_k(i')) & -\sin(\tilde{\phi}_k(i')) \\ \sin(\tilde{\phi}_k(i')) & \cos(\tilde{\phi}_k(i')) \end{bmatrix}. \end{aligned} \tag{15}$$

Using the composition rule of planar rotations, we have

$$\big(R_{wi}^{(k)}\big)^\top R_{wi'}^{(k)} = \begin{bmatrix} \cos\big(\tilde{\phi}_k(i') - \tilde{\phi}_k(i)\big) & -\sin\big(\tilde{\phi}_k(i') - \tilde{\phi}_k(i)\big) \\ \sin\big(\tilde{\phi}_k(i') - \tilde{\phi}_k(i)\big) & \cos\big(\tilde{\phi}_k(i') - \tilde{\phi}_k(i)\big) \end{bmatrix}. \tag{16}$$

By Eq. 14, the angle difference is

$$\begin{aligned} \tilde{\phi}_k(i') - \tilde{\phi}_k(i) &= w\big(\phi_k(i') - \phi_k(i)\big) \\ &= w(i'-i)\,\theta_k. \end{aligned} \tag{17}$$

Therefore,

$$\big(R_{wi}^{(k)}\big)^\top R_{wi'}^{(k)} = R_{w(i'-i)}^{(k)}, \tag{18}$$

i.e., in the k-th subspace, scaling the positions by w results in a relative rotation whose angle is still proportional to the relative offset $(i'-i)$, but magnified by a factor of w.

Aggregating over all 2D subspaces and extending to the 2D position (i,j), the same reasoning yields

$$R_{wi,wj}^\top R_{wi',wj'} = R_{w(i'-i),\,w(j'-j)}, \tag{19}$$

which leads to the scaled relative-form inner product

$$\begin{aligned} &\langle \mathrm{RoPE}([Q]_{i,j}, w\cdot i, w\cdot j),\ \mathrm{RoPE}([K]_{i',j'}, w\cdot i', w\cdot j')\rangle \\ &= [Q]_{i,j}^\top R_{w(i'-i),\,w(j'-j)}[K]_{i',j'}. \end{aligned} \tag{20}$$

3. More Visulization Comparisions

3.1. Additional Results of SynPS

As illustrated in Fig. 2, the proposed *SynPS* is capable of editing the source image with given complex non-rigid prompts.

3.2. Qualitative Comparison

As illustrated in Fig. 3, *Ours* achieves better results than compared baselines.

Variants	Setting	Non-Rigid Editing Benchmark		
		GPT-5↑	CLIP$_{img}$	CLIP$_{txt}$ ↑
Fix Seed FLUX Default	–	5.2383	0.8513	0.2364
+ Attention Sharing	w/ RoPE ($w = 1.0$)	5.9650	0.9180	0.2291
	w/o RoPE ($w = 0.0$)	6.3750	0.8963	**0.2366**
+ SynPS w/ Adaptive w	$M_{\text{min}} = 0.7$, $M_{\text{max}} = 1.0$	6.5572	0.9028	0.2358
	$M_{\text{min}} = 0.7$, $M_{\text{max}} = 1.1$	6.5421	0.9066	0.2355
	$M_{\text{min}} = 0.7$, $M_{\text{max}} = 1.2$	6.5758	0.9087	0.2347
	$M_{\text{min}} = 0.8$, $M_{\text{max}} = 1.0$	6.3467	0.9052	0.2353
	$M_{\text{min}} = 0.8$, $M_{\text{max}} = 1.1$	6.6566	0.9068	0.2346
	$M_{\text{min}} = 0.8$, $M_{\text{max}} = 1.2$	6.5253	0.9104	0.2343
	$M_{\text{min}} = 0.9$, $M_{\text{max}} = 1.0$	**6.6567**	0.9051	0.2344
	$M_{\text{min}} = 0.9$, $M_{\text{max}} = 1.1$	6.5633	0.9108	0.2346
	$M_{\text{min}} = 0.9$, $M_{\text{max}} = 1.2$	6.3900	0.9114	0.2336

Table 1. Ablated results on the Curated Non-rigid Editing Benchmark.

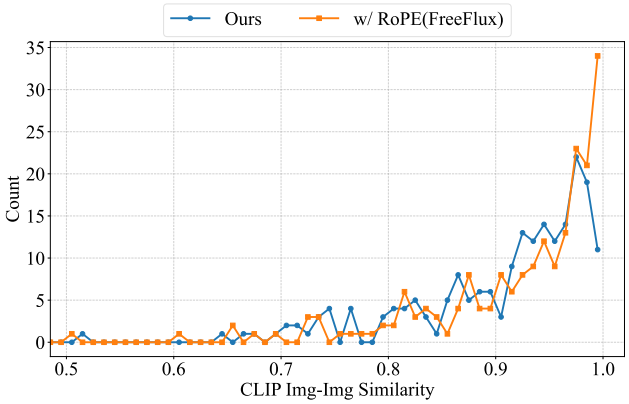

Figure 1. Visualization of the distribution of CLIP$_{img}$ similarity scores. We compare the results of the two methods by binning the similarity scores with an interval of 0.01. The horizontal axis represents the CLIP$_{img}$ similarity, while the vertical axis indicates the number of samples falling into each bin.

4. More Abaltion Analysis

4.1. Alleviation of Duplicate Artifacts

As illustrated in Fig. 1, We quantitatively define the occurrence of *duplicate artifacts* as instances where the CLIP$_{img}$ similarity between the source and target images exceeds 0.97. By visualizing the distribution of CLIP$_{img}$ similarity scores for both FreeFlux and our method on the benchmark, we observe that FreeFlux yields a substantial number of results with similarity scores surpassing 0.97. Notably, the proportion of samples falling within the range of 0.99-1.0 is significantly higher in FreeFlux compared to our approach. These results further demonstrate that our method effectively mitigates the issue of duplicate generation.

4.2. Ablations Qualitative Comparison

As illustrated in Fig. 4, such results demonstrate the effectiveness of the proposed *SynPS*.

4.3. Hyperparameter Analysis

As illustrated in Tab. 1, even with diverse hyperparameters, *SynPS* still achieves promissing results, validating the robustness and effectiveness of the proposed method, especially for the training-free settings.

5. Discussion

Limitations. Our attention synergy mechanism modulates attention sharing by explicitly accounting for the interaction between positional embeddings and semantic information. However, our current design primarily targets non-rigid editing tasks where positional relationships play a crucial role, leaving the exploration of more general editing scenarios to future work. Moreover, when the editing instruction requires structure-preserving transformations—such as color adjustments or style changes—our method becomes less applicable due to the intrinsic characteristics of these tasks, which depend more on appearance-level modifications rather than positional or semantic correspondence.

Socail Impact. Our methods can modify some fake images with certain instructions, such as human faces or private pets, which may increase the risk of privacy leakage and portrait forgery. Therefore, users intending to use our technique should apply for authorization to use the respective source images. Nevertheless, our approach can serve as a tool for AIGC to edit images following the intended instructions.

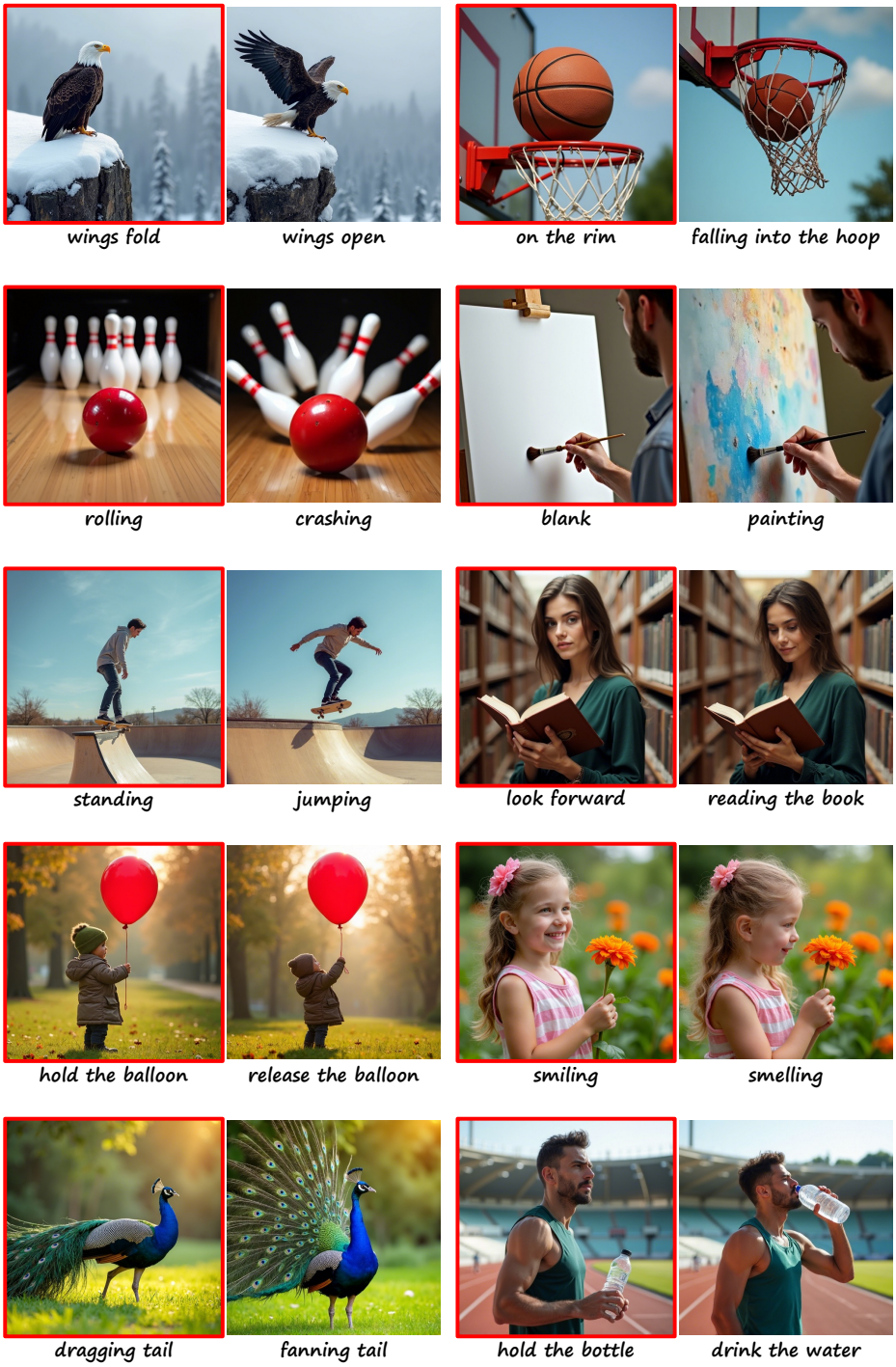

Figure 2. More editing results of *SynPS*.

References

[1] 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 *CVPR*, pages 7877–7888, 2025. 1, 2

[2] Vladimir Kulikov, Matan Kleiner, Inbar Huberman-Spiegelglas, and Tomer Michaeli. Flowedit: Inversion-free text-based editing using pre-trained flow models. In *ICCV*, pages 19721–19730, 2025. 1

[3] Black Forest Labs. Flux. https://github.com/

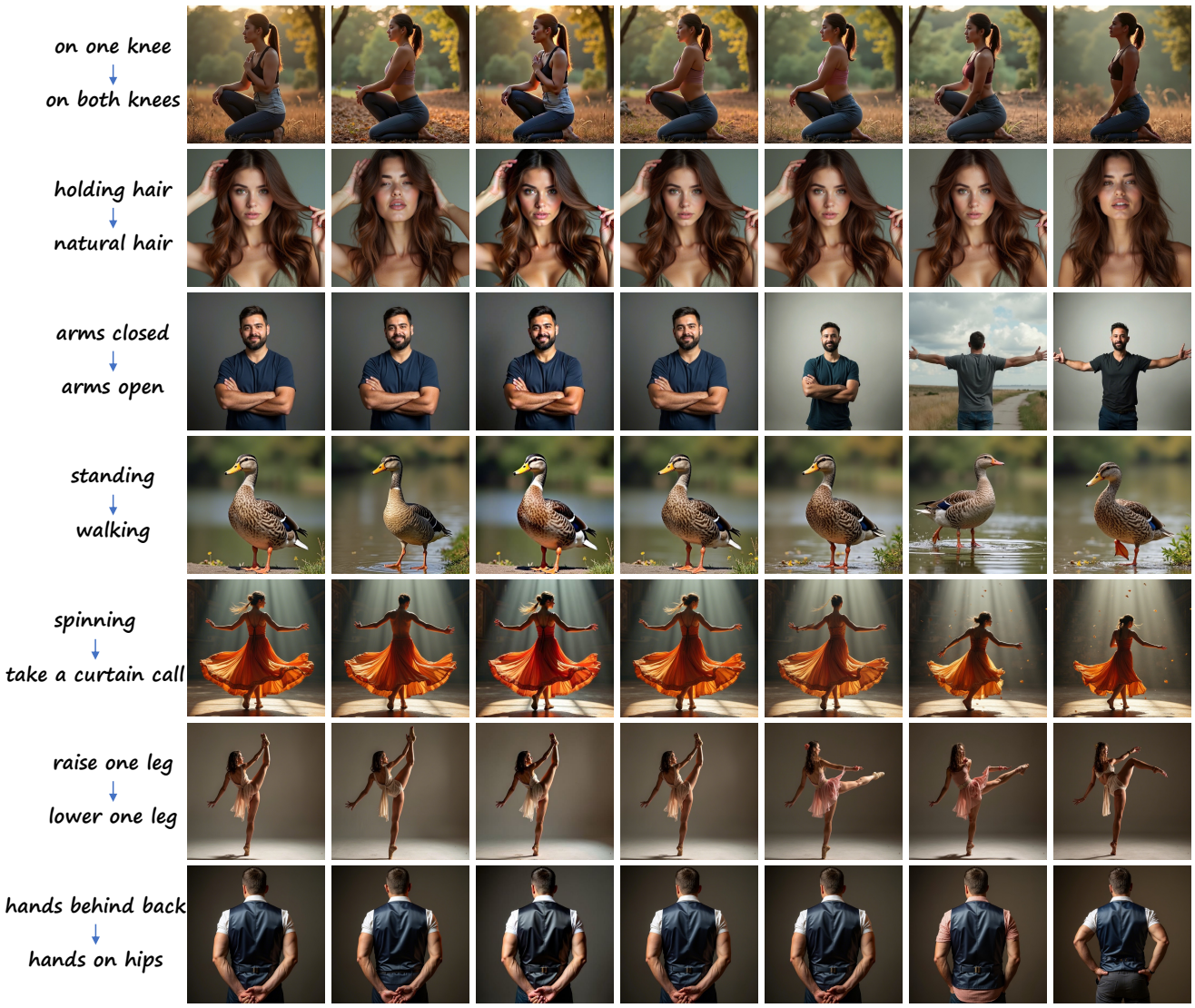

Figure 3. More Qualitative comparisons of compared baselines

black-forest-labs/flux, 2024. 1, 2

[4] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. In *Neurocomputing*, page 127063. Elsevier, 2024. 2

[5] Jiangshan Wang, Junfu Pu, Zhongang Qi, Jiayi Guo, Yue Ma, Nisha Huang, Yuxin Chen, Xiu Li, and Ying Shan. Taming rectified flow for inversion and editing. In *ICML*, 2025. 1

[6] Mengyu Wang, Henghui Ding, Jianing Peng, Yao Zhao, Yunpeng Chen, and Yunchao Wei. Characonsist: Fine-grained consistent character generation. In *ICCV*, pages 16058–16067, 2025. 1, 2

[7] Tianyi Wei, Yifan Zhou, Dongdong Chen, and Xingang Pan. Freeflux: Understanding and exploiting layer-specific roles in rope-based mmdit for versatile image editing. In *ICCV*, 2025.

Figure 4. More qualitative ablation studies of *SynPS*.