OrionEdit: Bridging Reference and Source Images for Generalized Cross-Image Editing

Supplementary Material

7. Extended discussion

This section provides additional discussion of the method in Section 3.2. We begin by supplementing the preliminaries referenced in earlier sections, then analyze why standard orthogonality constraints are ineffective, and finally present further analysis of the proposed approach.

7.1. Preliminaries

Classic LoRA. Let $W \in \mathbb{R}^{d\times k}$ denote a pre-trained linear projection mapping from $\mathbb{R}^k$ to $\mathbb{R}^d$. Low-Rank Adaptation (LoRA) [21] enables efficient fine-tuning by constraining parameter updates to a learnable low-dimensional subspace. The adapted weight is defined as:

$$W' = W + \Delta W, \quad \Delta W = AB^\top, \tag{13}$$

where $A \in \mathbb{R}^{d\times r}$ and $B \in \mathbb{R}^{k\times r}$ are trainable low-rank matrices, and $r \ll \min(d, k)$ denotes the adaptation rank. This formulation substantially reduces trainable parameters while maintaining model expressiveness. For adaptation tasks involving multiple concepts, let $\mathcal{C} = c_1, c_2, \ldots, c_m$ denote the set of target concepts. A direct extension assigns an independent LoRA module to each concept, producing a set of concept-specific updates $\Delta W^{(i)}{}_{i=1}^{m}$:

$$\Delta W^{(i)} = A^{(i)}B^{(i)\top}, \Delta W = \sum_{i=1}^{m} \Delta W^{(i)}. \tag{14}$$

Nevertheless, the unconstrained linear aggregation of these low-rank increments may cause *concept entanglement* and *catastrophic forgetting*, as their respective subspaces often overlap within the shared parameter space.

Orthogonal constraint. To mitigate cross-concept interference, each concept-specific update can be constrained within an independent subspace, with mutual orthogonality enforced across subspaces. Formally, the orthogonality condition can be defined either in the weight space or directly on the low-rank factors as:

$$(A^{(i)})^\top A^{(j)} = 0, \quad (B^{(i)})^\top B^{(j)} = 0, \quad \forall, i \neq j, \tag{15}$$

or equivalently,

$$\langle \Delta W^{(i)}, \Delta W^{(j)} \rangle_F = 0, \quad \forall, i \neq j, \tag{16}$$

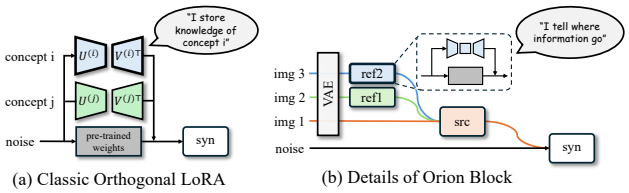
where $\langle \cdot, \cdot \rangle_F$ denotes the Frobenius inner product. Each low-rank increment $\Delta W^{(i)}$ acts as a compact representation of the i-th concept's adaptation signal. These constraints encourage independent parameter updates across LoRA modules, maintaining decorrelated and disentangled representations during adaptation. In practice, subspace independence is typically encouraged through orthogonality regularization or related orthogonalization techniques, formulated as:

$$\begin{aligned} \mathcal{L}_{\text{orth}}^{\Delta W} &= \lambda \sum_{i<j} \left\langle \Delta W^{(i)}, \Delta W^{(j)} \right\rangle_F^2 \\ &= \lambda \sum_{i<j} \left\| (\Delta W^{(i)})^\top \Delta W^{(j)} \right\|_F^2 . \end{aligned} \tag{17}$$

where λ controls the strength of the regularization term. The overall optimization objective becomes:

$$\mathcal{L} = \mathcal{L}_{\text{task}} + \mathcal{L}_{\text{orth}}^{\Delta W}, \tag{18}$$

which promotes mutually orthogonal adaptation directions within the low-rank subspace and preserves subspace independence throughout training.



(a) Classic Orthogonal LoRA (b) Details of Orion Block

Figure 9. **Information-flow regulation in OrionEdit** that enforces reverse-causal and unidirectional propagation from reference to source to synthesis.

7.2. Why standard orthogonality is not used

Standard orthogonal adaptation has shown clear benefits in early LoRA-based *concept customization*, promoting feature disentanglement and preserving subject identity. In this paradigm (Fig. 9(a)), low-rank updates act as **knowledge containers** that store separable concepts to reduce interference. However, this design also introduces several limitations that merit further discussion:

1. **Additional regularization overhead.** Classical orthogonal methods require explicit loss terms to enforce subspace independence, typically formulated as $\mathcal{L}_{\text{orth}}^{\Delta W}$ in Eq. 17. These additional objectives incur extra computational cost and may introduce gradient interference with the main task loss $\mathcal{L}_{\text{task}}$. Moreover, imperfect orthogonality ($A^{(i)\top}A^{(j)} \neq 0$) inevitably leads to residual information leakage across subject subspaces.

Table 3. **Detailed quantitative results on multi-image generation:** Multi-Character (left) and Multi-Object (right). OrionEdit achieves competitive reference-based synthesis performance across all metrics, demonstrating strong multi-image fusion capability without relying on a source branch.

Multi-character generation						Multi-object generation					
Model	Aesthetic↑	DPG↑	DINOv3↑	SigLip-I↑	CLIP-T↑	Model	Aesthetic↑	DPG↑	DINOv3↑	SigLip-I↑	CLIP-T↑
UNO [74]	5.91	84.13	0.733	0.632	0.295	UNO [74]	5.81	82.44	0.755	0.767	0.313
OmniGen [77]	5.90	82.70	0.783	0.697	0.276	OmniGen [77]	5.49	70.00	0.719	0.706	0.306
OmniGen2 [72]	5.94	84.15	**0.790**	0.731	0.289	OmniGen2 [72]	5.37	92.67	0.780	0.840	0.306
Xverse [8]	**6.13**	83.34	0.771	0.673	0.291	Xverse [8]	5.83	86.89	0.768	0.757	0.307
DreamOmni2-gen [75]	5.84	**92.78**	0.723	0.762	0.299	DreamOmni2-gen [75]	5.80	92.90	**0.795**	0.746	0.299
GPT-4o [26]	5.86	80.29	0.768	0.750	0.290	GPT-4o [26]	5.47	**98.67**	0.767	**0.864**	**0.315**
Qwen-image-edit [71]	5.89	89.43	0.790	0.695	**0.309**	Qwen-image-edit [71]	5.87	91.83	0.736	0.717	0.315
OrionEdit-qwen	5.98	87.90	0.763	0.791	0.309	OrionEdit-qwen	5.82	93.30	0.767	0.791	0.313
OrionEdit-flux	5.95	84.17	0.760	**0.805**	0.304	OrionEdit-flux	**5.89**	91.11	0.764	0.789	0.302

Table 4. **Quantitative results on single-image generation.** OrionEdit maintains strong single-reference generative performance, confirming that its cross-image editing design does not compromise standard generation quality.

Single-character generation					
Model	Aesthetic↑	DPG↑	DINOv3↑	SigLip-I↑	CLIP-T↑
UNO [74]	6.33	82.48	0.692	0.732	0.286
OmniGen [77]	6.12	80.29	0.642	0.736	0.269
OmniGen2 [72]	6.13	84.13	0.730	0.760	0.293
Xverse [8]	6.43	83.34	0.699	0.759	0.287
DreamOmni2-gen [75]	6.47	85.52	0.751	0.749	0.292
GPT-4o [26]	**6.49**	**94.15**	**0.765**	**0.769**	0.302
Qwen-image-edit [71]	6.36	91.63	0.713	0.760	0.300
Flux-Kontext-dev [32]	6.45	88.24	0.746	0.765	0.291
OrionEdit-qwen	6.35	92.23	0.760	0.761	**0.310**
OrionEdit-flux	6.43	89.62	0.748	0.743	0.300

2. **Incompatibility with zero-initialized fine-tuning.** Orthogonality constraints commonly impose unit-norm conditions on the factor matrices (e.g., $A^{(i)\top}A^{(i)} \approx I$, $B^{(i)\top}B^{(i)} \approx I$), which are inherently incompatible with zero-initialized LoRA fine-tuning that assumes $\Delta W^{(i)} = A^{(i)}B^{(i)\top} = 0$ at the beginning of training. This constraint violates the desirable *zero-perturbation* property, i.e., $W_{\text{adapted}} = W_{\text{pretrained}}$ when $\Delta W^{(i)} = 0$, and may cause unintended shifts in the pretrained model's representation space before meaningful adaptation occurs.
3. **Outdated feature retention mechanism.** Early mechanisms for preserving subject-specific features commonly relied on LoRA-based *concept customization*, where models were fine-tuned on a small set of images representing a particular concept. However, modern models are capable of achieving zero-shot feature preservation from a single reference image, thereby diminishing the necessity of orthogonality-based concept customization.

7.3. Discussion on OrionEdit

As shown in Fig. 9(b), rather than adopting standard orthogonal LoRA adaptation, OrionEdit applies orthogonality in a different manner suited for cross-image editing. Instead of storing visual concepts within low-rank parameters, orthogonality here acts as an **information dispatcher**, it organizes image latents into separate branches and constrains how information propagates across them during generation. A key component enabling this behavior is the proposed *symmetric mapping*. Each branch is associated with a fixed orthonormal basis A_i, defining a subspace $S_i = \text{span}(A_i)$. Given $A_i^\top A_i = I$, we denote the projector onto this subspace as

$$P_i = A_i A_i^\top. \tag{19}$$

The update for branch i is parameterized as

$$\Delta W_i = A_i B_i A_i^\top. \tag{20}$$

Using the orthonormality condition of A_i, we obtain

$$\begin{aligned} \Delta W_i &= A_i B_i A_i^\top \\ &= A_i (A_i^\top A_i) B_i (A_i^\top A_i) A_i^\top \\ &= P_i \, \Delta W_i \, P_i. \end{aligned} \tag{21}$$

Right multiplication by P_i restricts the input to S_i, while left multiplication confines the output to the same subspace. Consequently, the update becomes *closed* on S_i ($\Delta W_i : S_i \to S_i$), ensuring subspace consistency without requiring additional orthogonality regularization losses. This symmetric structure enables OrionEdit to separate branch-wise feature updates while remaining compatible with zero-initialized LoRA training. Through this design, OrionEdit effectively overcomes three key challenges:

1. **Mitigation of gradient interference.** The interference in the parameter space is mitigated as (i) $A^{(i)}$ and $(A^{(i)})^\top$ inherently restrict updates to orthogonal subspaces, eliminating the need for any additional orthog-

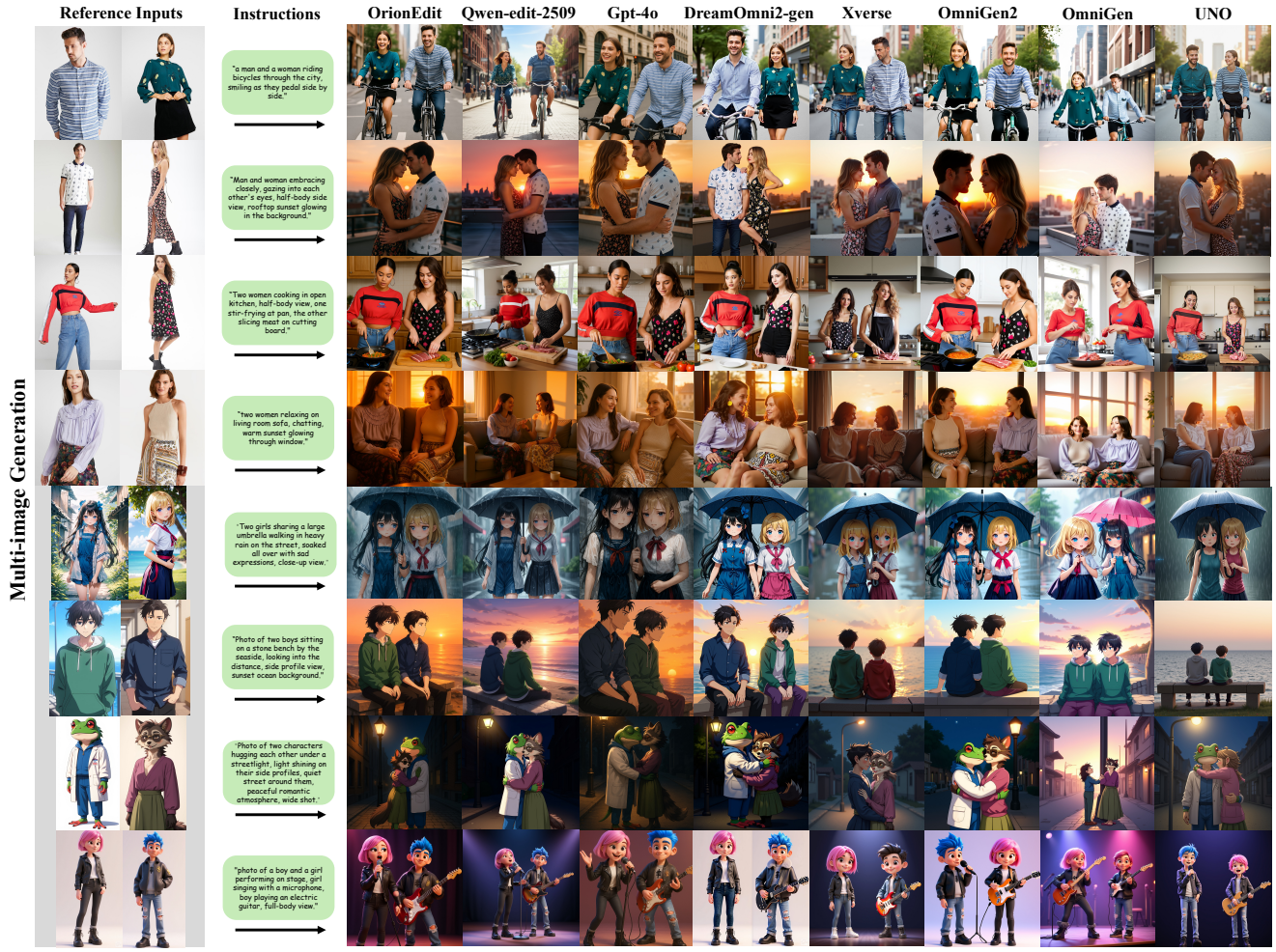


Figure 10. **Qualitative results on the multi-image generation** task using *semantically similar* reference images. OrionEdit produces distinct and identity-consistent character renderings, preserving fine-grained appearance features while mitigating the texture artifacts and concept blending observed in baseline methods. The results demonstrate the effectiveness of the proposed *Orion Block* in disentangling multi-reference information and generating coherent compositions.

onalization losses; and (ii) the zero-initialized $B^{(i)}$ ensures $\Delta W^{(i)} = 0$ at the beginning of training, providing a stable and perturbation-free optimization process.

2. **Logically consistent feature propagation.** The proposed information-flow mask $\mathcal{M}(p, q)$ regulates inter-branch interactions in the feature space (Fig. 9). By enforcing a reverse-causal and unidirectional flow from *reference* to *source* to *synthesis*, it reduces crosstalk-induced semantic drift and preserves structural consistency during generation.
3. **Zero-shot cross-image editing.** With the above mechanisms, OrionEdit performs zero-shot editing on one source and multiple reference images without additional fine-tuning, enabling instant and instruction-free generation while preserving semantic fidelity and cross-image consistency.

8. Extended results

Detailed performance on generation tasks. While previous sections summarized model performance across various tasks, here we focus specifically on generation scenarios. Table 3 reports comprehensive results on multi-image fusion, covering both multi-character and multi-object generation. Since these tasks do not require a source branch, only reference-driven synthesis is evaluated. OrionEdit demonstrates competitive performance across all metrics, closely matching or surpassing several state-of-the-art multi-image generators. To ensure that the introduction of our cross-image editing framework does not compromise single-reference generative capability, Table 4 further benchmarks single-character generation. OrionEdit retains strong single-image performance, achieving balanced im-

Figure 11. **Qualitative results on multi-subject composition.** OrionEdit produces more coherent multi-subject compositions by maintaining visual–perspective consistency across references and preserving the source image layout, resulting in smoother and more stable subject integration.

provements in fidelity and semantic correspondence, confirming that the proposed design preserves generation quality while supporting robust multi-image fusion.

Additional visual results on generation. Fig. 10 provides additional qualitative comparisons on the *multi-image generation* task. We construct **similar-attribute** subsets (e.g., male–male, female–female, monsters), where the reference images share closely related semantics. This setting is particularly challenging because models often struggle to disentangle closely related visual concepts. As illustrated, OrionEdit delivers clear improvements in preserving identity-specific details such as clothing patterns, hairstyle structures, and overall appearance, while also avoiding the texture artifacts and blending inconsistencies commonly observed in baseline [71]. Beyond the editing scenario, these results further demonstrate that Orion Block **effectively decouples multiple reference features during generation**, reducing cross-reference interference and yielding visually coherent compositions.

Additional visual results on editing. As shown in Fig. 11 and Fig. 15, we provide supplementary qualitative results corresponding to the editing tasks discussed in Section 4.1, covering *multi-subject composition*, *style transfer* and *virtual try-on*. The comparisons show that the close-source model [26], while visually strong, often fails to maintain the structural layout of the source image. Qwen-Image-Editing [71], in contrast, tends to over-preserve the source image composition, resulting in pronounced "texture overlays" and a noticeable "pasted appearance". Other models exhibit similar patterns. OrionEdit, supported by the

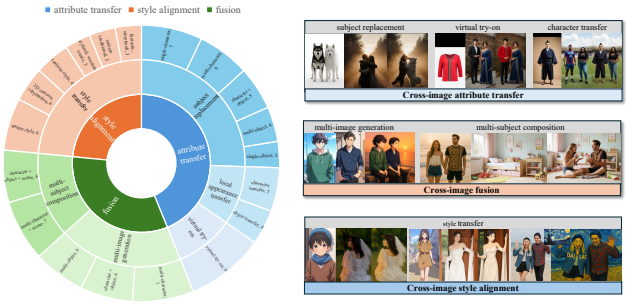


Figure 12. **Overview of OrionEditBench.** The benchmark includes three task families: cross-image attribute transfer, style alignment, and fusion, with diverse sub-tasks such as subject replacement, virtual try-on, and multi-subject composition.

information-flow mask, achieves stronger style alignment between reference and source images while preserving the source layout, enabling **reference attributes to merge naturally without introducing any texture artifacts** (while referring to reference subjects only through minimal descriptors (e.g., "a character", "a woman"), without exposing additional attributes). This effect is particularly evident in *virtual try-on* and *constrained editing* scenarios, such as transferring realistic garments onto stylized characters or placing realistic subjects into anime-style scenes, demonstrating the model's robustness in **editing tasks with diverse and mismatched visual styles**.

9. OrionEditBench

As noted in Section 4, no existing dataset explicitly targets "reference–source–synthesis" triples for cross-image edit-

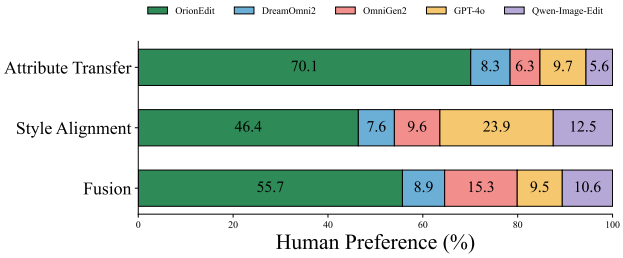


Figure 13. **User Study.** Human preference comparison across three editing tasks: Attribute Transfer, Style Alignment, and Fusion. Participants select the result that best satisfies the editing objective. OrionEdit receives the highest preference in all tasks, indicating superior fidelity and visual coherence compared with existing editing and generation models.

ing. To address this gap, we construct OrionEditBench, a benchmark built from synthetic reference–source pairs generated with Nano-Banana and GPT-4o. Each sample consists of a reference image, a source image, and a target synthesis objective. Figure 12 provides an overview of the benchmark. Following the taxonomy in Table 1, OrionEditBench organizes tasks into three primary categories: cross-image attribute transfer, cross-image fusion, and cross-image style alignment. Attribute transfer evaluates scenarios such as subject replacement, local appearance transfer, and virtual try-on. Fusion evaluates compositional generation tasks, including multi-image generation and multi-subject composition. Style alignment focuses on reference-guided stylization across diverse visual domains. Overall, OrionEditBench provides a diverse set of cross-image editing scenarios, enabling systematic evaluation of attribute transfer, style alignment, and compositional generation capabilities, Figure 16 shows a dataset preview.

10. User Study

In this section, we conduct a user study to evaluate human preferences for the generated editing results. To ensure an unbiased comparison, we build a lightweight internal website where outputs from different models are presented in randomized order. For each example, participants are shown results generated by five models (OrionEdit-qwen, DreamOmni2, OmniGen2, GPT-4o, and Qwen-Image-Edit-2509) under the same prompt and reference inputs, with model identities hidden, and are asked to select the image that best matches their preference. As shown in Fig. 13, OrionEdit consistently receives the highest human preference, obtaining 55.7%, 46.4%, and 70.1% of the votes across the evaluated settings. In comparison, the remaining baselines receive substantially lower scores, with GPT-4o performing relatively better in one setting (23.9%). These results suggest that OrionEdit more reliably preserves subject attributes while maintaining structural consistency during cross-image editing, leading to outputs more frequently preferred by human evaluators.

(a) Failure cases in composition and transfer

(b) Failure cases in character replacement

(c) Failure cases in style transfer

Figure 14. **Failure cases of OrionEdit.** (a) Viewpoint shifts when reference and source images contain conflicting perspectives. (b) Attribute drift in character replacement when generation and editing occur simultaneously. (c) Style deviation caused by biases in the training data.

11. Failure Cases

Although OrionEdit performs well across a wide range of tasks, certain challenging conditions can still lead to failure cases. In particular, the model may fail to fully preserve the spatial configuration of the *source image* (e.g., camera viewpoint) or subject-specific attributes such as clothing color and texture. In addition, biases in the training data may occasionally cause style fixation in style-transfer scenarios. Fig. 14 shows representative examples. As illustrated in (a), when the reference and source images exhibit noticeably different viewpoints, the synthesized result

may display a shifted camera perspective, likely due to the model reconciling conflicting visual cues. In (b), when generation and editing are performed simultaneously without additional semantic guidance, the character attributes may drift from the intended appearance. Finally, as shown in (c), because part of the training data is constructed using closed-source models such as GPT-4o, latent biases in these generated pairs may propagate into the training distribution, occasionally causing the model to deviate from the reference style in style-transfer tasks.

12. Limitations and future work

Limitations. OrionEdit demonstrates strong generalization across a wide range of cross-image editing tasks; however, several limitations remain. First, the framework decomposes multiple image branches into mutually orthogonal subspaces (see Eq. 3). While the fixed matrix A does not increase training complexity, the associated matrix operations introduce additional overhead during inference. Our experiments indicate that this overhead leads to an approximately 30% increase in inference latency when applied to the Flux-kontext-dev [33] and Qwen-image-edit-2509 [71] backbones. Second, the information-flow mask effectively regulates feature propagation, but in complex multi-reference scenarios it may impose overly restrictive constraints on fine-grained attributes, producing texture-like artifacts similar to those observed in the Qwen-image-edit-2509 model. Despite this issue, OrionEdit surpasses most open-source editing systems and achieves performance on par with commercial solutions, thereby providing a valuable reference for research on cross-domain and multimodal editing. Third, since part of the training data is constructed from AI-generated image pairs, potential biases in the synthetic data may propagate to the model, occasionally causing deviations in style-transfer tasks.

Branch limits and trade-offs. The number of branches is bounded by the feature dimension d and the rank r. Let $\bar{A} = [A^{(1)}, \ldots, A^{(m)}] \in \mathbb{R}^{d \times mr}$; orthogonality yields $\bar{A}^\top \bar{A} = I$, implying $mr \leq d$ and $m \leq \lfloor d/r \rfloor$. Assuming $d = 3072$ and a relatively large adaptation rank ($r = 512$), this leads to an upper bound of $m \leq 6$ branches. However, in practice the experiments are limited by memory constraints, and we only train models with at most 4 branches. Specifically, the training setup contains one synthesis branch, one source branch, and two reference branches. Despite this limitation during training, we observe that the model generalizes well to inference scenarios involving more than four branches (e.g., three different images in the reference branches). We hypothesize that the model learns a generalizable "branch routing" behavior under the low-branch setting, which can naturally extend to higher-branch configurations at inference time. It is also worth noting that, in practice, adding branches does not cause severe dimensional squeezing. The frozen matrices $A^{(i)}$ only perform subspace projections, while the effective representational capacity is learned in $B^{(i)}$.

Future work. This work focuses on cross-image editing with static reference images, yet the proposed framework can be extended to more dynamic settings, such as temporally consistent video editing or multi-reference fusion. As multimodal datasets continue to expand, OrionEdit is well positioned to serve as a foundational capability for controllable cross-domain visual transfer, supporting downstream applications in product design, film post-production, and interactive visual content creation.

13. More cross-image editing cases

As illustrated in Fig 17 and Fig. 18, we present additional results generated by OrionEdit, covering visual character replacement, style transfer, and virtual try-on sub-tasks.

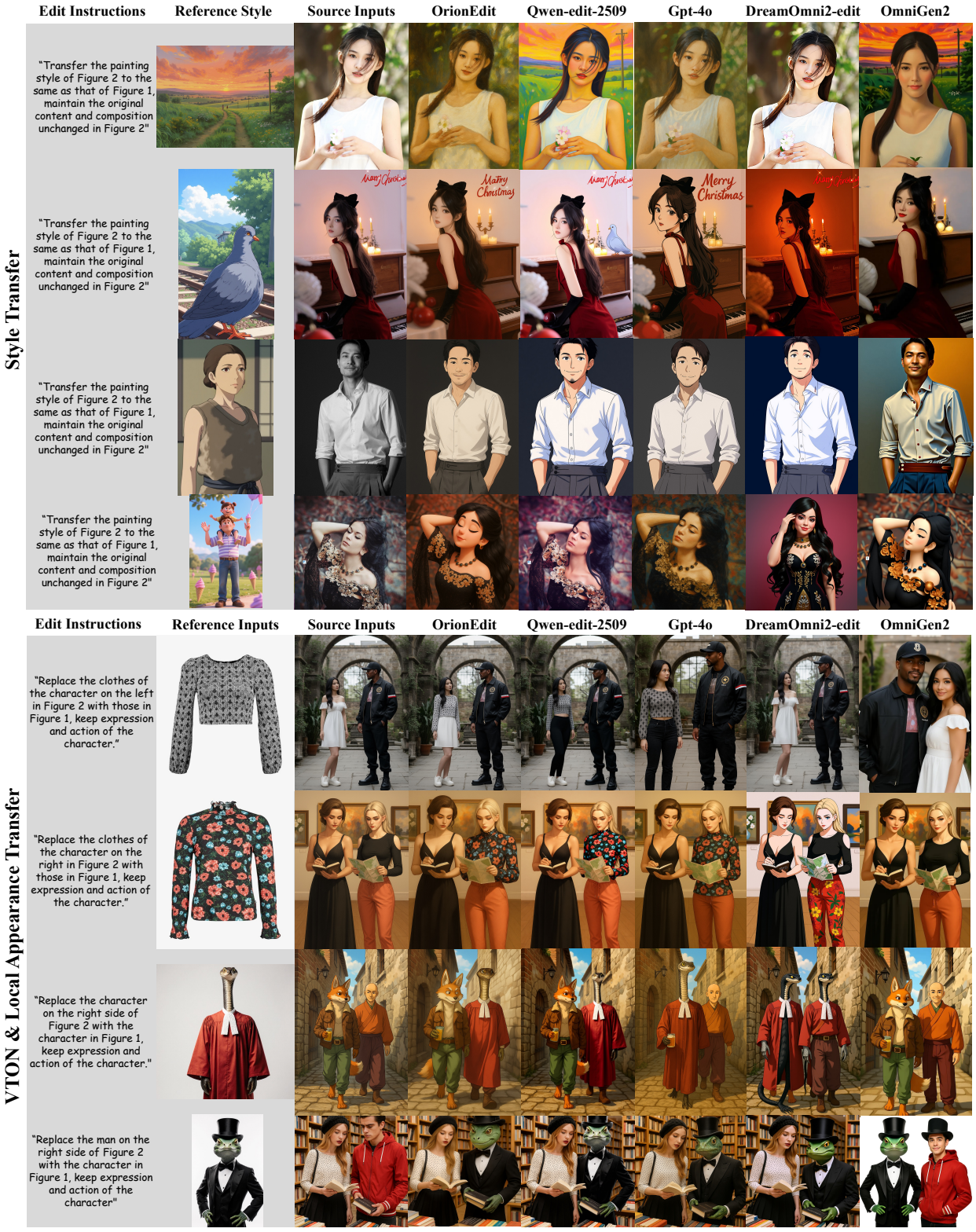


Figure 15. **Qualitative results on style transfer, virtual try-on and local appearance transfer.** The baseline models exhibit noticeable texture incompatibility in these tasks, often producing mismatched textures or unnatural overlays. OrionEdit achieves a more balanced outcome by preserving the structural layout of the source image while reducing texture artifacts.

Figure 16. **Examples from OrionEditBench**, illustrating representative reference–source pairs curated for evaluating cross-image editing performance. In each triplet, the left column shows one or multiple reference images, the middle column presents the source image, and the right column displays the corresponding target image.

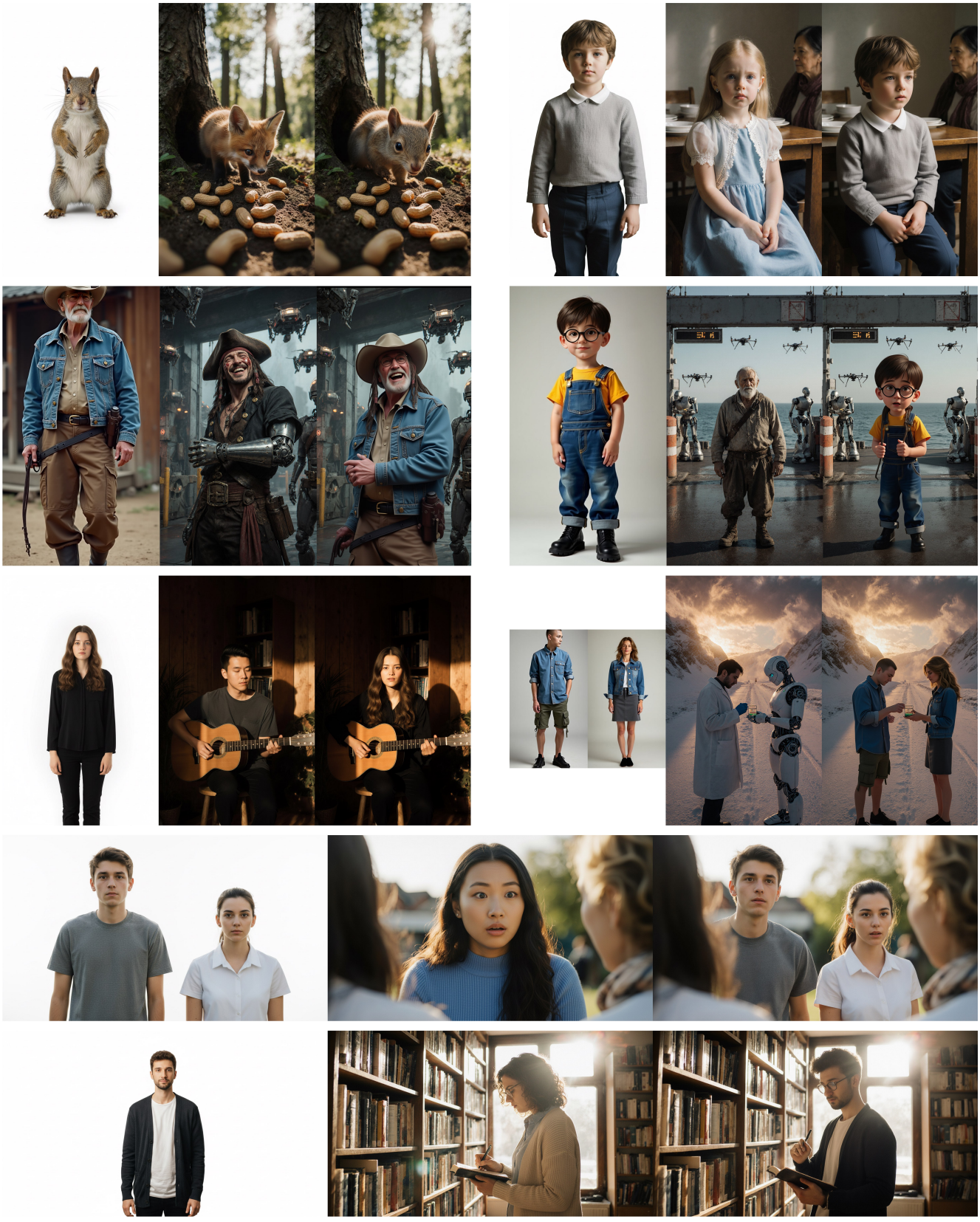


Figure 17. **More cross-image editing cases**. Each group consists of three parts: the left column shows the reference inputs, the middle column presents the source image to be edited, and the right column displays the final output generated by OrionEdit.

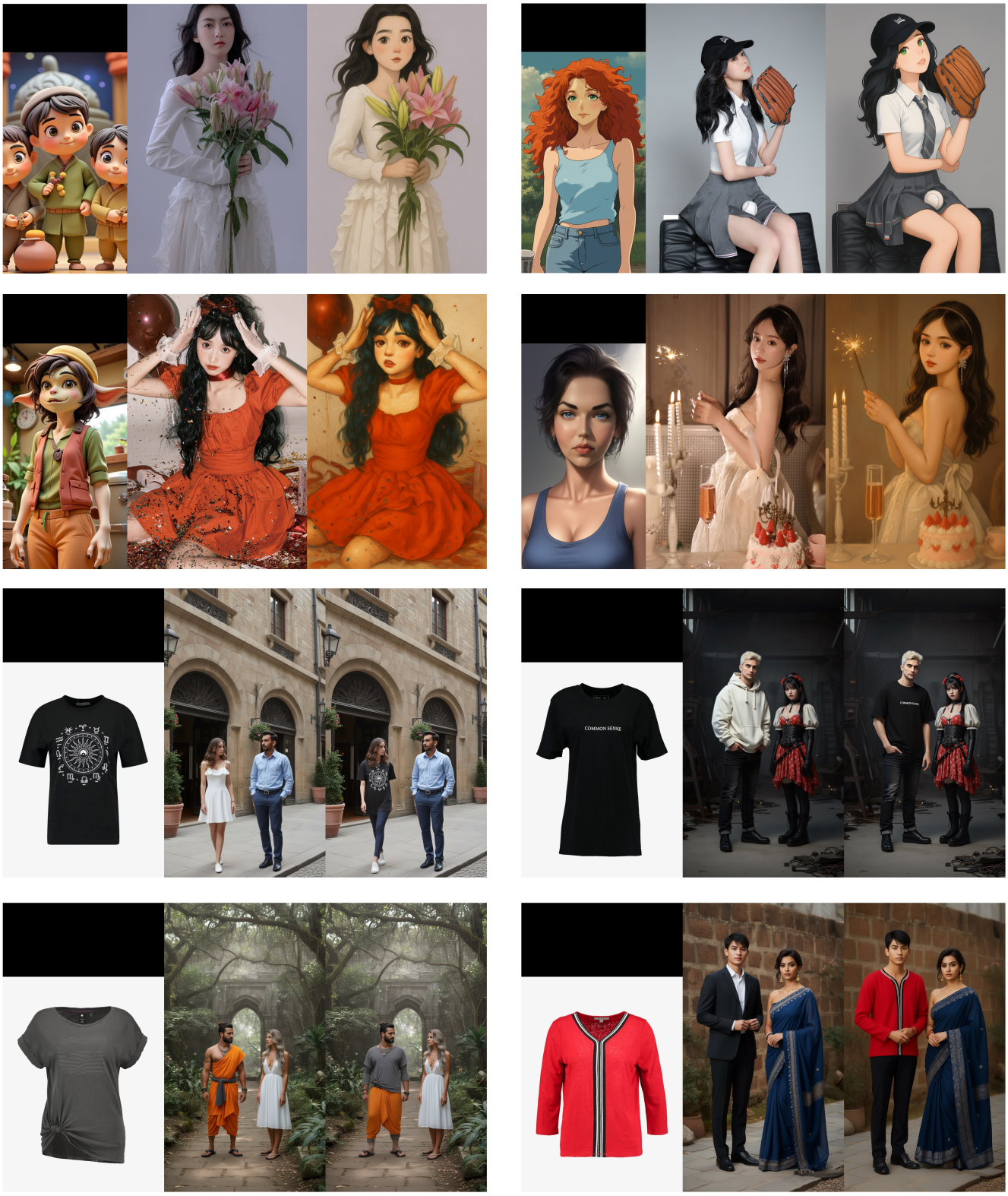


Figure 18. **More cross-image editing cases**. Each group consists of three parts: the left column shows the reference inputs, the middle column presents the source image to be edited, and the right column displays the final output generated by OrionEdit.