OPRO: Orthogonal Panel-Relative Operators for Panel-Aware In-Context Image Generation

Supplementary Material

Figure 1. **Qualitative comparison on subject-driven image generation.** Results are shown on the DreamBooth [6] test set under a three-panel protocol. For each subject, two reference images sampled from a four-shot support set occupy the first two panels, and the third panel is synthesized from a fully masked target canvas. We compare LoRA-only fine-tuning of ICEdit [11] with the same model modulated by OPRO.

This supplementary material provides additional empirical results, analyses, and implementation details that complement the main manuscript. Section A evaluates OPRO on subject-driven image generation in a three-panel setting. Section B presents a RoPE-aligned block-diagonal parameterization and its formal derivation, and Section C provides a detailed analysis of the zero-initialization strategy. Section D analyzes the computational overhead of OPRO. Section E presents additional qualitative results and examines inference-time scalability. Section F reports ablation studies on instructional image editing, and Section G summarizes the complete hyperparameter settings used in the experiments.

A. Subject-Driven Image Generation with OPRO

To assess the scalability of OPRO beyond the two-panel setting in the main manuscript, we evaluate subject-driven image generation in a **three-panel layout** leveraging the DreamBooth [6] test dataset. For each subject, we construct a four-shot support set and randomly sample two reference images to populate the first two panels. The third panel serves as a fully masked target canvas. We adopt ICEdit [11] as the base model and integrate OPRO as a lightweight panel-relative adaptation module. This task places a stronger emphasis on cross-panel subject consistency because the target panel must be synthesized from scratch while aggregating subject cues from multiple reference panels. Optimization proceeds for 2,000 steps with Adam [3] at a learning rate of 1×10^{-4}.

Table 1. **Quantitative comparison on subject-driven image generation.** Results are reported on a subset of DreamBooth using a three-panel layout with two reference panels and one fully masked target panel. OPRO consistently improves ICEdit on both DINO and CLIP-I. Higher is better in all cases.

Method	DINO (↑)	CLIP-I (↑)
ICEdit [11] (LoRA-only)	0.5828	0.7376
ICEdit [11] + OPRO	**0.6192**	**0.7724**

Table 1 shows that OPRO improves ICEdit on both DINO and CLIP-I, with absolute gains of 0.0364 and 0.0348, respectively. Figure 1 further illustrates more faithful preservation of subject appearance and more coherent synthesis of the target panel than the LoRA-only baseline.

B. RoPE-Aligned Block-Diagonal Parameterization

This section complements Section 3.2 of the main manuscript by detailing the relationship between OPRO and orthogonal relative positional encodings [5, 8, 10]. As briefly discussed in the main text, OPRO admits an additional compositional interpretation around the frozen positional operator. We first derive this general orthogonal-relative form and then present a RoPE-aligned block-

diagonal specialization, which yields the panel-relative phase-shift interpretation. This specialization is introduced for analysis and intuition; the trainable parameterization used in the main experiments is the low-rank Lie exponential parameterization in Section 3.3 of the main manuscript.

General orthogonal-relative form. For completeness, the frozen position-aware vectors are expressed in matrix form. Let $q_i, k_j \in \mathbb{R}^{d_h}$ denote the content vectors before the frozen positional transform, and let the backbone positional mechanism be represented by an orthogonal operator $R(\mathbf{x}) \in \mathrm{SO}(d_h)$:

$$\tilde{q}_i = R(\mathbf{x}_i) q_i, \qquad \tilde{k}_j = R(\mathbf{x}_j) k_j.$$

Assume that $R(\mathbf{x})$ satisfies the relative-position property

$$R(\mathbf{x}_i)^\top R(\mathbf{x}_j) = R(\mathbf{x}_j - \mathbf{x}_i).$$

Applying OPRO gives

$$\hat{q}_i = U_{p(i)} \tilde{q}_i, \qquad \hat{k}_j = U_{p(j)} \tilde{k}_j,$$

and therefore

$$\langle \hat{q}_i, \hat{k}_j \rangle = q_i^\top R(\mathbf{x}_i)^\top U_{p(i)}^\top U_{p(j)} R(\mathbf{x}_j) k_j.$$

This expression shows that OPRO preserves the frozen positional operator while inserting a learnable panel-relative orthogonal factor $U_{p(i)}^\top U_{p(j)}$.

RoPE-aligned block-diagonal specialization. To obtain a closed-form phase interpretation, we consider a stronger specialization in which U_p is restricted to the same block-diagonal $\mathrm{SO}(2)$ basis as RoPE. Let d_h be even and write

$$R(\mathbf{x}) = \mathrm{diag}\Big(R^{(1)}(\theta_1(\mathbf{x})), \ldots, R^{(d_h/2)}(\theta_{d_h/2}(\mathbf{x})) \Big),$$

where each $R^{(k)}(\theta) \in \mathrm{SO}(2)$ is a 2×2 rotation. We parameterize

$$U_p = \mathrm{diag}\Big(R^{(1)}(\phi_{p,1}), \ldots, R^{(d_h/2)}(\phi_{p,d_h/2}) \Big).$$

Because $R(\mathbf{x})$ and U_p are block-diagonal rotations acting on the same two-dimensional channel pairs, they commute:

$$U_p R(\mathbf{x}) = R(\mathbf{x}) U_p.$$

Hence,

$$\begin{aligned} \langle \hat{q}_i, \hat{k}_j \rangle &= q_i^\top R(\mathbf{x}_i)^\top U_{p(i)}^\top U_{p(j)} R(\mathbf{x}_j) k_j \\ &= q_i^\top R(\mathbf{x}_i)^\top R(\mathbf{x}_j) U_{p(i)}^\top U_{p(j)} k_j \\ &= q_i^\top R(\mathbf{x}_j - \mathbf{x}_i) U_{p(i)}^\top U_{p(j)} k_j. \end{aligned}$$

Moreover,

$$U_{p(i)}^\top U_{p(j)} = \mathrm{diag}\Big(R^{(1)}(\phi_{p(j),1} - \phi_{p(i),1}), \ldots, R^{(d_h/2)}(\phi_{p(j),d_h/2} - \phi_{p(i),d_h/2}) \Big),$$

so the effective angle of the k-th block is

$$\theta_k(\mathbf{x}_j - \mathbf{x}_i) + \phi_{p(j),k} - \phi_{p(i),k}.$$

Therefore, in this RoPE-aligned block-diagonal specialization, OPRO injects a learnable panel-relative phase offset into each frequency block.

Validation on Compositional Reasoning Task Table 2 summarizes the performance of the block-diagonal parameterization implementation (OPRO-BD) applied to the two-stage compositional reasoning task. With only a minimal parameter overhead equal to the number of panels ($P = 4, 9, 16$), OPRO-BD demonstrates improvements for orthogonal positional encodings in the 3×3 and 4×4 panel experiments.

C. Detailed Analysis of Zero Initialization Strategy

In this section, we provide a detailed analysis of the zero-initialization strategy. We first formally prove that our parameterization guarantees non-degenerate gradients.

Recall from Section 3.4 in manuscripts that for each panel p we parameterize the orthogonal operator as

$$U_p = \exp(A_p), \qquad A_p = L_p R_p^\top - R_p L_p^\top,$$

where $L_p, R_p \in \mathbb{R}^{d_h \times r}$ are learnable parameters and $\exp(\cdot)$ denotes the matrix exponential. We initialize

$$L_p = \mathbf{0}, \qquad R_p \sim \mathcal{N}(0, \sigma^2),$$

so that $A_p = \mathbf{0}$ and $U_p = I$ at step 0. Thus the OPRO operator has no effect on the pre-trained model at initialization, while still admitting a non-degenerate gradient, as we show below.

Notation Let $\mathcal{L}$ be a scalar loss and define the Frobenius inner product $\langle X, Y \rangle = \mathrm{tr}(X^\top Y)$. Write

$$G := \nabla_{U_p} \mathcal{L} \quad \text{and} \quad \tilde{G} := \mathrm{D}\exp^*_{A_p}[G],$$

where $\mathrm{D}\exp_{A_p}$ is the differential of the matrix exponential at A_p and $\mathrm{D}\exp^*_{A_p}$ is its adjoint with respect to the Frobenius inner product [1].

Proposition. 1 (Zero initialization identity mapping with non-degenerate gradient)**.** *Let* $U_p = \exp(A_p)$ *with*

$$A_p = L_p R_p^\top - R_p L_p^\top.$$

Table 2. Effect of the block-diagonal implementation of OPRO on top of LoRA ($r = 8$). We report the accuracy (%) of LoRA+OPRO-BD and the absolute change Δ (percentage points) compared to the LoRA baseline from Tab. 1 of the main manuscript.

	Panel 2×2		Panel 3×3		Panel 4×4	
Type	**+OPRO-BD**	**Δ**	**+OPRO-BD**	**Δ**	**+OPRO-BD**	**Δ**
APE	37.10	−0.90	23.60	−0.80	19.00	−0.50
RoPE[8]	45.80	−0.60	38.70	+2.50	32.50	+2.20
LieRE[5]	58.70	+0.60	36.20	+2.00	23.30	+0.40
ComRoPE[10]	57.90	−0.60	40.90	+3.10	29.80	+0.60

Then the gradients of $\mathcal{L}$ with respect to L_p and R_p are

$$\nabla_{L_p}\mathcal{L} = (\tilde{G} - \tilde{G}^\top)\, R_p, \qquad \nabla_{R_p}\mathcal{L} = (\tilde{G}^\top - \tilde{G})\, L_p.$$

In particular, at zero initialization ($A_p = \mathbf{0}$ and $L_p = \mathbf{0}$), we have $U_p = I$ and $\tilde{G} = G$, so

$$\nabla_{L_p}\mathcal{L} = (G - G^\top)\, R_p, \qquad \nabla_{R_p}\mathcal{L} = \mathbf{0}.$$

Thus, the operator is initially the identity, but optimization starts immediately through L_p, while R_p remains fixed at the first step.

Proof. By the chain rule and the expression for the differential of the matrix exponential [1], for any perturbation E we have

$$d\mathcal{L} = \langle G, \mathrm{D}\exp_{A_p}[dA_p]\rangle = \langle \tilde{G}, dA_p\rangle.$$

Differentiating $A_p = L_p R_p^\top - R_p L_p^\top$ gives

$$dA_p = dL_p\, R_p^\top + L_p\, dR_p^\top - dR_p\, L_p^\top - R_p\, dL_p^\top.$$

Substituting this into the inner product and applying the identity $\langle X, YZ^\top\rangle = \langle XZ, Y\rangle$, we expand $\langle \tilde{G}, dA_p\rangle$:

$$\begin{aligned}\langle \tilde{G}, dA_p\rangle &= \big\langle \tilde{G},\, dL_p\, R_p^\top + L_p\, dR_p^\top - dR_p\, L_p^\top - R_p\, dL_p^\top \big\rangle \\ &= \big\langle (\tilde{G} - \tilde{G}^\top)\, R_p,\, dL_p\big\rangle + \big\langle (\tilde{G}^\top - \tilde{G})\, L_p,\, dR_p\big\rangle.\end{aligned}$$

By the definition of the gradient with respect to the Frobenius inner product, this implies

$$\nabla_{L_p}\mathcal{L} = (\tilde{G} - \tilde{G}^\top)\, R_p, \qquad \nabla_{R_p}\mathcal{L} = (\tilde{G}^\top - \tilde{G})\, L_p.$$

Under zero initialization $A_p = \mathbf{0}$, the Jacobian of the exponential map is the identity, therefore $\tilde{G} = G$. With $L_p = \mathbf{0}$, the gradients simplify to:

$$\nabla_{L_p}\mathcal{L} = (G - G^\top)\, R_p, \qquad \nabla_{R_p}\mathcal{L} = \mathbf{0},$$

□

D. Computational Cost Analysis

We analyze the computational overhead of OPRO when integrated with FluxFill [2]. OPRO introduces additional orthogonal transformations within the attention layers. Specifically, at each step and layer, OPRO performs two 128×128 matrix-vector rotations, corresponding to queries and keys, across all tokens. The additional floating-point operations (ΔFLOPs) can be approximated as

$$\Delta\text{FLOPs} \approx N_{\text{panel}} \cdot N_{\text{head}} \cdot d_h^2 \cdot N_{\text{tokens}} \cdot N_{\text{layers}} \cdot N_{\text{steps}}, \quad (1)$$

where N_{tokens} denotes the number of tokens per panel. Substituting the configuration parameters detailed in Section 4.2 of the main manuscript ($N_{\text{panel}} = 2$, $N_{\text{head}} = 24$, $d_h = 128$, $N_{\text{tokens}} = 4{,}096$, $N_{\text{layers}} = 57$, $N_{\text{steps}} = 28$), the total additional computation amounts to approximately 10.3 TFLOPs. Furthermore, the cost of computing the matrix exponential is negligible (approximately 6.7 GFLOPs). Given the substantial computational budget of diffusion transformers, this theoretical overhead remains marginal.

E. Qualitative Results and Inference-Time Scalability

We present additional qualitative results generated by ICEdit [11] equipped with OPRO. Figure 2 demonstrates the versatility of OPRO across multiple instructional editing tasks. The provided examples illustrate the capability of the model to execute precise modifications, including object replacement, attribute alteration, text rendering, and global style transfer, while preserving the content of the original image.

Furthermore, Figure 3 details the inference-time scalability of OPRO by demonstrating compositional generation with multi-reference inputs. Specifically, we apply a model trained on a fixed two-panel layout to a three-panel configuration comprising two reference images. We enable this multi-reference inference by reusing the OPRO learned for the single-reference panel across both references, while applying the target operator to the generation panel. By assigning images to panels that share identical functional roles

Figure 2. **Qualitative results on diverse instructional editing tasks.** We demonstrate the versatility of OPRO across a broad spectrum of editing categories. The examples illustrate the model's capability to precisely follow instructions for object replacement, attribute modification, text rendering, and global style transfer, all while maintaining high fidelity to the original image content.

Table 3. **Ablation Studies on MagicBrush.** The table validates the design principles of OPRO on the ICEdit baseline (r=16). Relative to OPRO, breaking isometry (APB) or same-panel invariance (Asym-OPRO) reduces performance. Removing zero initialization preserves spatial alignment but lowers semantic consistency, as reflected by CLIP-I and DINO.

Method	Isometry	SP-Inv	L1 ↓	CLIP-I ↑	DINO ↑
LoRA (Baseline)	-	-	0.1189	0.8703	0.7706
+ APB	No	No	0.0966	0.8893	0.8196
+ Asym-OPRO	Yes	No	0.0988	0.8880	0.8151
+ OPRO (w/o Zero Init)	Yes	Yes	**0.0780**	0.8989	0.8510
+ OPRO (Ours)	**Yes**	**Yes**	0.0781	**0.9002**	**0.8531**

during inference, we achieve multi-reference compositional generation without requiring retraining.

F. Ablation Studies on Instructional Image Editing

Table 3 extends the ablation studies of the main manuscript to the MagicBrush dataset. Consistent with the results of the compositional reasoning task, violating isometry (APB) or same-panel invariance (Asym-OPRO) degrades performance across all metrics. Furthermore, omitting zero initialization (+ OPRO w/o Zero Init) achieves a spatial alignment error (L1) comparable to that of the proposed OPRO, yet yields lower semantic consistency scores (CLIP-I and DINO) than the proposed method. This semantic degradation aligns with the accuracy drop observed in the main manuscript, demonstrating that an identity mapping initialization is essential to preserve the visual priors of the pre-trained model.

G. Complete Hyperparameter Settings

We provide detailed hyperparameter configurations used in our experiments. Tab. 4 summarizes the training settings for

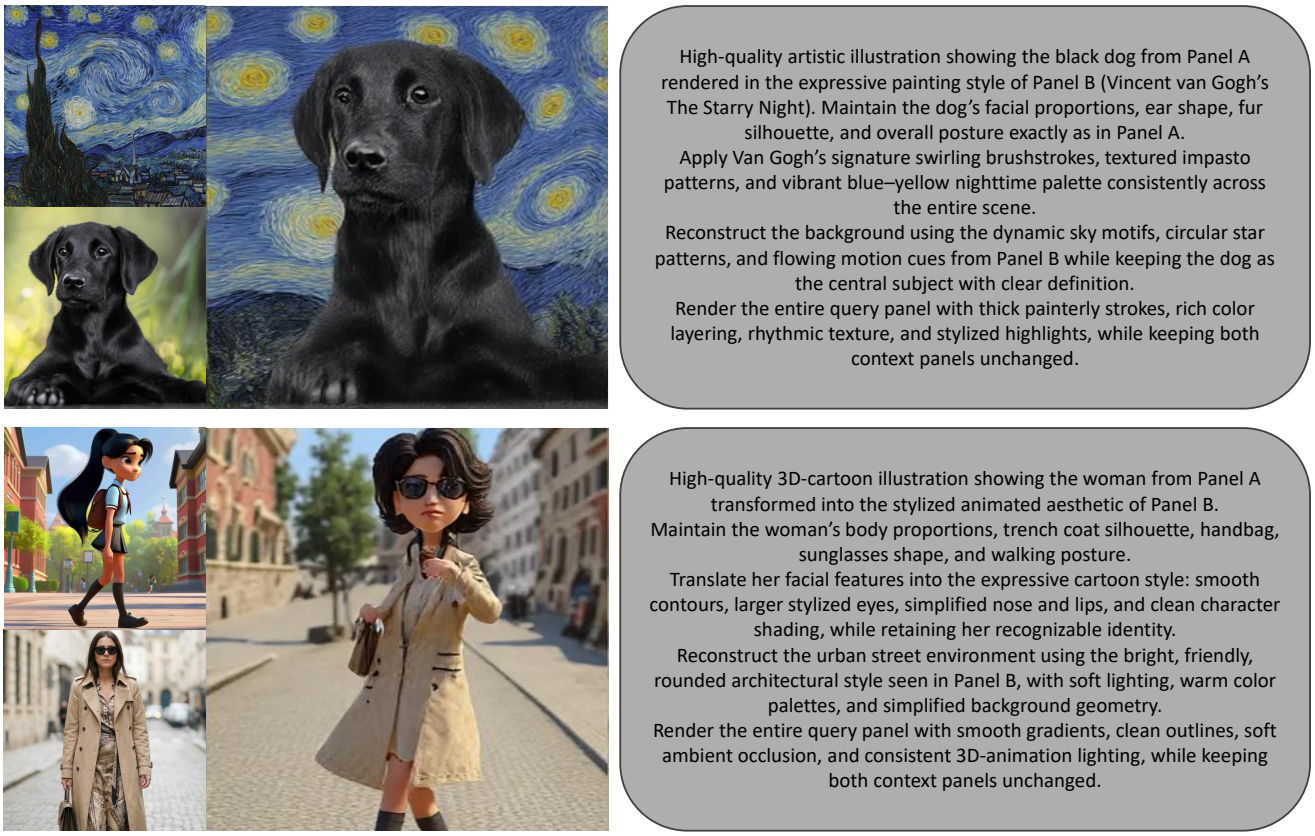

Figure 3. **Qualitative examples of multi-reference compositional generation.** We demonstrate the capability to integrate attributes from multiple context panels. The model synthesizes a new image by combining the *style* from the first panel and the *object* from the second panel. Note that this compositional ability emerges without explicit training on multi-reference layouts.

the instructional image editing baselines. Tab. 5 presents the optimization details for the two-stage compositional reasoning task.

References

[1] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ, 2008. 2, 3

[2] Black Forest Labs. FLUX.1 Fill [pro], 2024. 3

[3] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[4] Chaojie Mao, Jingfeng Zhang, Yulin Pan, Zeyinzi Jiang, Zhen Han, Yu Liu, and Jingren Zhou. Ace++: Instruction-based image creation and editing via context-aware content filling. *arXiv preprint arXiv:2501.02487*, 2025. 6

[5] Sophie Ostmeier, Brian Axelrod, Maya Varma, Michael Moseley, Akshay S Chaudhari, and Curtis Langlotz. Liere: Lie rotational positional encodings. In *Forty-second International Conference on Machine Learning*. 1, 3

[6] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22500–22510, 2023. 1

[7] Wensong Song, Hong Jiang, Zongxing Yang, Ruijie Quan, and Yi Yang. Insert anything: Image insertion via in-context editing in dit. *arXiv preprint arXiv:2504.15009*, 2025. 6

[8] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. 1, 3

[9] Shaojin Wu, Mengqi Huang, Wenxu Wu, Yufeng Cheng, Fei Ding, and Qian He. Less-to-more generalization: Unlocking more controllability by in-context generation. *arXiv preprint arXiv:2504.02160*, 2025. 6

[10] Hao Yu, Tangyu Jiang, Shuning Jia, Shannan Yan, Shunning Liu, Haolong Qian, Guanghao Li, Shuting Dong, and Chun Yuan. Comrope: Scalable and robust rotary position embedding parameterized by trainable commuting angle matrices. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 4508–4517, 2025. 1, 3

[11] Zechuan Zhang, Ji Xie, Yu Lu, Zongxin Yang, and Yi Yang. In-context edit: Enabling instructional image editing with in-

Table 4. **Hyperparameters for instructional image editing baselines.** All models are trained for 5,000 steps using the AdamW optimizer (weight decay 0.01, learning rate 1×10^{-4}) with a batch size of 8. We use bfloat16 precision and a constant learning rate schedule. Note that InsertAnything uses FluxPriorRedux for reference-image conditioning, while UNO adopts an in-context approach. We adapt OPRO in each self-attention layer.

Method	**Base Model**	**Pos. Encoding**	**LoRA Target Modules**	**LoRA Rank (r)**	**OPRO Rank (ρ)**
ICEdit [11]	FluxFill Dev	Global-canvas	Attention (`q,k,v,out`)	16	32
ACE++ [4]	FluxFill Dev	Global-canvas	Attn + MLP + Modulation	16	32
InsertAnything [7]	FluxFill Dev (+Redux)	Global-canvas	Attention Projections (`q,k,v,out`)	16	32
UNO [9]	Flux Dev	Per-panel	Attention Projections + MLP	256	32

Table 5. **Detailed hyperparameters for two-stage compositional reasoning.**

Hyperparameter	**Stage 1 (Pre-training)**	**Stage 2 (Fine-tuning)**
Optimization	Adam ($\beta_1 = 0.9, \beta_2 = 0.999$)	Adam ($\beta_1 = 0.9, \beta_2 = 0.999$)
Batch Size	256	256
Learning Rate	1×10^{-3} (Warmup+Cosine)	5×10^{-4} (Constant)
Weight Decay	0.05	0.05
Training Steps	50k	2k
Architecture / Adapter		
Patch Size	16×16	16×16
Positional Encoding	Learnable (APE/RoPE etc.)	Frozen
Adapter Rank	-	LoRA $r = 8$ / OPRO $\rho = (2, 4, 8)$

context generation in large scale diffusion transformer. *arXiv preprint arXiv:2504.20690*, 2025. 1, 3, 6