Towards Source-Aware Object Swapping with Initial Noise Perturbation

Supplementary Material

1. Cost Analysis

1.1. Training Data Size Analysis

Benefiting from high-quality pseudo pairs and source-aware training, SourceSwap attains strong performance with only **40K** images for fine-tuning. In contrast, prior learning-based baselines require orders of magnitude more data (Table 1). Paint-by-Example [16] uses roughly **190M** image samples; AnyDoor [3] and MimicBrush [2] rely on **hundreds of thousands** of images plus video frames; IMPRINT [13] and ObjectMate [15] further scale to **millions** of multi-view images, which are expensive to collect and curate. These data sources not only impose substantial acquisition cost but also depend on modalities beyond single images (videos or multi-view captures). By comparison, SourceSwap achieves strong performance using only about **40K** single-image samples.

We further conduct an ablation study on the amount of training data, as illustrated in Fig. 1. When the training data increases to about 40K, the model already produces high-quality outputs, while further increasing it to 50K brings only negligible improvement. Hence, we use about 40K samples for training throughout our experiments.

Table 1. Training data scale of learning-based baselines. Prior methods rely on large and expensive datasets (often videos or multi-view images), whereas SourceSwap requires only 40K single images for fine-tuning.

Baselines	# Training Data	Data Type
Paint-by-Example [16]	$\sim 190000k$	Images
AnyDoor [3]	$\sim 410k$	Images & Videos
MimicBrush [2]	$\sim 10100k$	Images & Videos
ObjectMate [15]	$\sim 600000k$	Multi-view Images
IMPRINT [13]	$\sim 1627k$	Multi-view Images & Videos
Ours	$\sim 40k$	Images

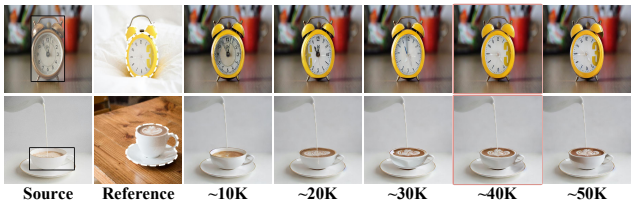

Figure 1. Ablation on training size. Performance saturates at **40K** samples, indicating that SourceSwap learns effectively from small-scale single-image training.

1.2. Inference Time Analysis

We report per-sample inference time together with three categories of comparison methods, as shown in Tab. 2. All methods are evaluated on a single 40 GB NVIDIA A100 GPU. Test-time tuning methods take the most time, primarily because a personalized DreamBooth [11] model must be trained during inference. Additionally, PhotoSwap [4] and SwapAnything [5] depend on precise inversion results, and the 50-step DDIM inversion together with the 500-step null-text optimization [8] introduces significant runtime overhead. InstantSwap [17] uses score distillation to bypass the inversion process, yet its single-image optimization still leads to a higher inference cost compared with learning-based approaches. For tuning-free methods, TIGIC [7] employs a 20-step DDIM inversion, but its three-branch architecture requires three U-Nets to run simultaneously, which limits the overall speed. DiptychPrompt [12] adopts the FLUX [6] with an additional ControlNet-Inpainting module [1], but the 40GB GPU is insufficient to hold all parameters, causing frequent memory swapping between GPU and CPU and thus limiting inference speed. Learning-based methods require only a forward sampling process, which makes them inherently faster during inference. Paint-by-Example, AnyDoor, and MimicBrush all use 50 diffusion steps per inference. In contrast, SourceSwap performs only 20 steps per inference. *Even with our 2-round iterative refinement, the total cost is merely 4.41 seconds per sample, faster than the most lightweight baseline (Paint-by-Example at 6.33s).*

2. Analysis of Iterative Refinement

Our source-aware design enables *iterative refinement*, in which the output of one inference round is fed back as the input for the next. The influence of the refinement rounds k is summarized in Tab. 3 and visualized in Fig. 2.

As shown in Fig. 2, one round of inference captures the overall structure of the reference object but may not fully reproduce its surface color. With $k = 2$, color consistency and appearance fidelity improve noticeably. Increasing the rounds to $k = 3$ or $k = 4$ yields only marginal gains. Tab. 3 further confirms this trend: object fidelity generally increases with k and peaks at $k = 3$. When increasing to $k = 4$, DreamSim slightly degrades, which may stem from accumulated compression artifacts introduced by repeated VAE encoding and decoding [14]. To balance inference efficiency (see Sec. 1.2) and performance, we report results with $k = 2$ in the main paper.

	Test-time Tuning		
	PhotoSwap	InstantSwap	SwapAnything
Time	128.85s(+751.97s)	23.48s(+751.97s)	127.39s(+751.97s)
	Tuning-free		
	TIGIC	DiptychPrompt	
Time	12.84s	124.63s	
	Learning-based		
	Paint-by-Example	AnyDoor	MimicBrush
Time	6.33s	11.01s	8.47s
	Ours (1 Inf.)	Ours (2 Inf.)	Ours (3 Inf.)	Ours (4 Inf.)
Time	2.28s	4.41s	6.64s	8.96s

Table 2. Inference time comparison across test-time tuning, tuning-free, and learning-based methods. For test-time tuning methods, we also report (in parentheses) the time required to train the personalized DreamBooth model. For our method, we report 1–4 rounds of inference and adopt 2 rounds as the final setting. Learning-based methods are substantially faster, and SourceSwap achieves the best inference speed among them.

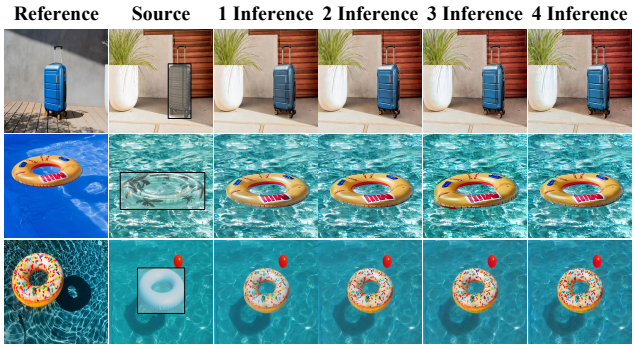

Figure 2. Qualitative effect of iterative refinement. One round ($k = 1$) captures the overall structure but misses surface colors; two rounds ($k = 2$) significantly improve color consistency, and further rounds ($k = 3, 4$) yield only marginal visual gains.

	$k = 1$	$k = 2$	$k = 3$	$k = 4$
DreamSim	0.4653	0.4428	0.4366	0.4405

Table 3. Effect of iterative refinement rounds k on object fidelity. As k increases, fidelity improves and peaks at $k = 3$, while $k = 4$ shows a slight drop in DreamSim.

3. Analysis of Mask Type

During source-aware training, the source mask M_s is expanded into a bounding-box mask M_{box}, which is then concatenated to the input of the denoising U-Net as a condition. This expansion relaxes the contour constraints imposed by M_s, allowing the resulting object to better preserve attributes of the reference object. To analyze the impact of mask shape on the results, we train two variants using M_s and M_{box} respectively, as shown in Fig. 3. Strictly constraining the object shape with M_s can lead to undesirable deformations, for example, thinning of a bear's arms, distorted leg proportions, and unnatural folding of a towel. In contrast, using M_{box} provides sufficient spatial guidance while allowing the model to adapt the final object shape according to the reference. We further quantify this effect using DreamSim: the variant trained with M_s yields 0.4600, worse than the 0.4428 obtained with M_{box}, confirming the advantage of relaxed shape constraints.

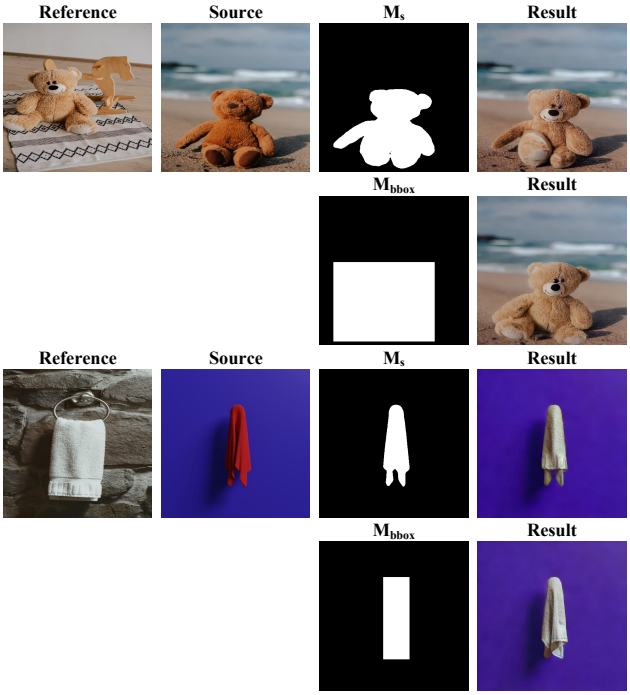

Figure 3. Comparison between the precise source mask M_s and the bounding-box mask M_{box}. We train two variants by conditioning the denoising U-Net on M_s and M_{box}, respectively. Strictly constraining the object shape with M_s often causes undesirable deformations, whereas M_{box} relaxes the shape constraint and enables the object to retain attributes from the reference.

4. Limitations

Although SourceSwap performs robustly in diverse scenarios, its accuracy depends on the localization quality of the mask. Off-the-shelf segmentation tools such as Grounded SAM [10] tend to segment all objects in the region (*e.g.*, several lemons in Fig. 4). In such cases, the model receives ambiguous control signals and may replace every segmented instance rather than the intended one. Interestingly, even under this challenging scenario, SourceSwap

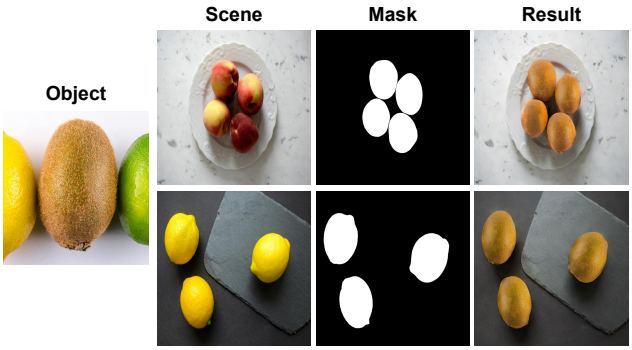

Figure 4. Failure case caused by ambiguous localization. Segmentation tools such as Grounded SAM [10] tend to extract all instances (*e.g.*, multiple lemons), leading the model to swap every segmented object rather than the intended one.

preserves scene consistency and produces visually coherent swaps, indicating that the learned alignment behavior remains stable despite incorrect masks. Nevertheless, resolving this limitation requires more precise target specification. A promising direction is to develop mask-free or instance-aware variants that rely on lightweight point-based interactions (*e.g.*, a single click) to explicitly identify the target object, thereby mitigating ambiguity introduced by multi-instance segmentation.

5. Applications

Our source-aware formulation allows SourceSwap to support a range of applications without any additional retraining, including subject-driven refinement, multi-object swapping, and face swapping. Given a suboptimal result produced by another method, we simply treat it as the new source image and keep the reference unchanged. The denoising U-Net refines object appearance and restores fine details. As shown in Fig. 5, our refinement significantly improves the output quality over the original AnyDoor result. Because each swap operates independently and requires no model-specific tuning, SourceSwap can sequentially replace multiple objects in a single scene. In Fig. 6 and the main text, we also present the results of multi-subject swapping. Multiple objects within a single scene are sequentially replaced with three different reference objects. Each object blends harmoniously into the background. Moreover, by replacing the reference and source with two different faces, SourceSwap is also capable of performing face swapping, as presented in the main text.

6. User Preference

To evaluate human preference, we conducted a user study using 153 randomly selected results from our method and all competing methods. Participants rated each result on

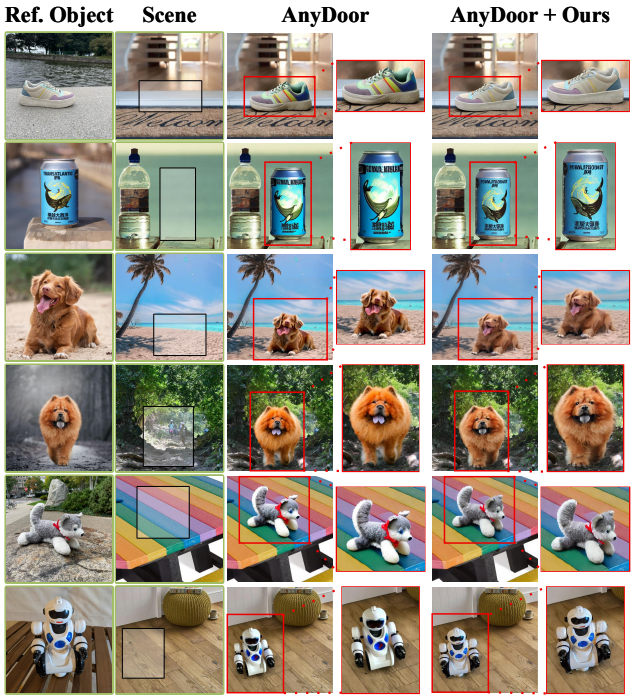

Figure 5. SourceSwap is capable of refining suboptimal outputs from other subject-driven generation methods, without additional retraining. It consistently improves the outputs of AnyDoor through its refinement capability.

Figure 6. SourceSwap is capable of handling multi-object swapping. In this example, all three pillows are successfully swapped and blend naturally with the scene.

a five-point Likert scale (1–5) across three aspects: *object fidelity* (object details), *scene fidelity* (scene preservation), and *object–scene harmony*. When rating object fidelity details, the reference image was shown, and participants were asked to rate how similar the reference object and the result appeared. When rating scene preservation, the source image and result image were presented, and participants were asked to rate how similar the backgrounds were. For object-scene harmony, participants rated how naturally the object blended into the background. An example questionnaire is shown in Fig. 7. In total, 19 volunteers were recruited, yielding 2,907 answers, as summarized in Tab. 4. Our

method achieves the highest scores across all three aspects, reflecting strong alignment with human perception. Competing methods either underperform across the board or excel only in one or two aspects, indicating limited balance among fidelity, preservation, and harmony. Notably, the human-judged harmony trend matches the MLLM-based evaluation in Table 2 of the main paper, where our method ranks first, followed by DiptychPrompt [12].

Table 4. User study results on three aspects (object fidelity (object details), scene fidelity (background preservation), and object–scene harmony) using a five-point Likert scale (1–5, higher is better).

Baselines	Object Fidelity ↑	Background Fidelity ↑	Harmony ↑
Paint-by-Example [16]	1.63	3.37	2.25
AnyDoor [3]	2.74	2.41	2.11
MimicBrush [2]	2.20	3.12	2.44
TIGIC [7]	1.73	1.95	1.53
Diptych Prompting [12]	2.62	2.68	3.15
PhotoSwap [4]	2.26	3.23	1.96
InstantSwap [17]	2.68	1.47	1.92
SwapAnything [5]	2.39	3.72	2.37
Ours	**3.98**	**3.82**	**4.13**

7. SourceBench Benchmark Details

7.1. Overview

SourceBench is a high-quality benchmark designed specifically for object swapping. It complements DreamEditBench by providing higher-resolution, real-world photographs with richer object categories and more complex object–scene interactions. SourceBench contains a comparable number of pairs to DreamEditBench, but significantly more categories and object instances under more challenging conditions.

7.2. Data Collection

Images are collected from Pexels and Unsplash, which provide high-quality, real-world photos under permissive licenses. We discard low-resolution, heavily compressed, or strongly stylized images and retain only images with a minimum resolution of 700×525 pixels. SourceBench uses single-view images only; no videos or multi-view sequences are involved, so all methods are evaluated purely from still images.

7.3. Data Filtering Strategy

To make the benchmark informative for evaluating object swapping, we adopt a simple but targeted filtering strategy when selecting background scenes and foreground objects:

- For scene images, we avoid images whose backgrounds are overly uniform (*e.g.*, blank walls or nearly textureless backdrops), since such scenes provide little structure for testing object–scene harmony.
- For reference images, we avoid choosing objects that are themselves almost textureless or visually trivial, and instead prefer objects with meaningful appearance and geometry (*e.g.*, mugs, books, electronics, tools).
- Some images are used in multiple pairs, acting as source in one pair and reference in another, or providing different objects in different pairs. This reuse increases diversity of roles and contexts without inflating the dataset size, as illustrated in Fig. 8.

7.4. Pair Construction and Statistics

Each pair in SourceBench consists of a source image, a reference image, and their corresponding masks. The source provides the scene and the object to be replaced, while the reference specifies the target appearance. We manually construct pairs to cover a wide range of everyday categories and interaction patterns, including hand–object interactions, objects placed on furniture, and partially occluded objects. Overall, SourceBench includes 54 categories, 577 distinct object instances, and 1,554 source–reference pairs, offering a more diverse and challenging testbed than DreamEditBench with a similar number of pairs.

We use SourceBench with the same evaluation protocol as in the main paper (DreamSim for object fidelity, local LPIPS for scene preservation, and MLLM/user studies for object–scene harmony), enabling fair comparison across benchmarks.

8. Multi-modal Large Language Model Evaluation Protocol

To assess the perceptual quality of object-scene harmony beyond conventional metrics, we employ a Two-Alternative Forced Choice (2AFC) evaluation protocol using ChatGPT-5 [9] as the judge. This approach leverages the advanced vision-language understanding capabilities of MLLMs to evaluate nuanced visual qualities such as lighting consistency, viewpoint alignment, and physical plausibility of object-background interactions.

Input Format. For each test case, we construct a concatenated image containing four sub-images arranged horizontally:

1. **Source Image** (I_s): The source scene with the object to be replaced.
2. **Reference Image** (I_r): The image containing the reference object to be swapped in.
3. **Generated Output 1**: The result produced by our SourceSwap method.
4. **Generated Output 2**: The result produced by a baseline method.

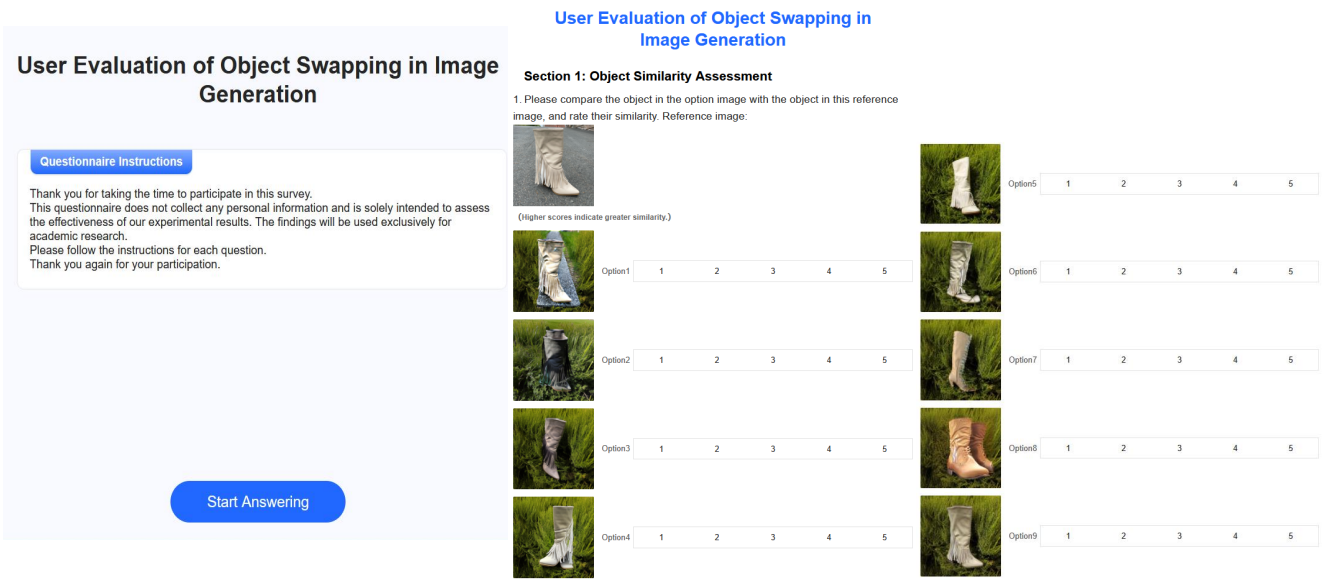

Figure 7. Example of the questionnaire used in the user study. Participants rate each result on object fidelity (object details), scene fidelity (scene preservation), and object–scene harmony using a five-point Likert scale.

The order of outputs (positions 3 and 4) is randomized to mitigate position bias in the MLLM's responses. We employ a sequential two-stage prompting approach to ensure thorough evaluation:

Stage 1: Description. The MLLM is first instructed to describe each of the two generated outputs:

> *"Imagine you're an expert in image generation. In the set of four images below, the first is the source image and the second is the reference image. Our goal is to replace the object in the source with the object from the reference. The last two images are the generated outputs. First, describe each of the last two images."*

This stage serves two purposes: (i) it ensures the model carefully examines both outputs before making a judgment, and (ii) it provides interpretable evidence for the subsequent selection.

Stage 2: Selection. Building upon the descriptions from Stage 1, the MLLM is then asked to make a preference judgment:

> *"Second, choose the better of the two outputs, judging by how naturally the reference object blends into the source image."*

The conversation history from Stage 1 is maintained to provide context for this decision. This two-stage approach encourages more deliberate and justified evaluations compared to direct preference queries.

Post-processing. To parse the responses from Stage 2, we query once more for each case to extract a definitive choice using the structured model output feature provided by the OpenAI API. We ask the model to parse the response from Stage 2 into a decision and confidence (0-1) tuple. The average confidence of all the responses is 0.939. Note that no image is provided in this parsing step; the decision and confidence are inferred solely from the text response.

For each baseline method, we compute the percentage of test cases where ChatGPT-5 preferred our SourceSwap method over the competitor. The results reported in the main paper represent these preference rates across all evaluated pairs for each benchmark.

9. More Visual Results

Fig. 9 and Fig. 10 present additional qualitative comparisons between SourceSwap and the baselines on DreamEditBench and SourceBench, respectively. The examples cover diverse object shapes, viewpoints, and interaction relationships, demonstrating the robustness of our method under various scenarios.

References

[1] Alimama-Creative. Flux.1-dev-controlnet-inpainting-beta. https://huggingface.co/alimama-creative/FLUX.1-dev-Controlnet-Inpainting-Beta, 2024. 1

[2] Xi Chen, Yutong Feng, Mengting Chen, Yiyang Wang, Shilong Zhang, Yu Liu, Yujun Shen, and Hengshuang Zhao. Zero-shot image editing with reference imitation. *Advances*

Figure 8. Examples of categories and source-reference pairs in SourceBench. Scene images feature rich backgrounds and complex interactions, while reference images provide diverse object appearances. Note that the same image may participate in multiple pairs.

in Neural Information Processing Systems, 37:84010–84032, 2024. 1, 4

[3] Xi Chen, Lianghua Huang, Yu Liu, Yujun Shen, Deli Zhao, and Hengshuang Zhao. Anydoor: Zero-shot object-level image customization. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 6593–6602, 2024. 1, 4

[4] Jing Gu, Yilin Wang, Nanxuan Zhao, Tsu-Jui Fu, Wei Xiong, Qing Liu, Zhifei Zhang, He Zhang, Jianming Zhang, HyunJoon Jung, et al. Photoswap: Personalized subject swapping in images. *Advances in Neural Information Processing Systems*, 36:35202–35217, 2023. 1, 4

[5] Jing Gu, Nanxuan Zhao, Wei Xiong, Qing Liu, Zhifei Zhang, He Zhang, Jianming Zhang, HyunJoon Jung, Yilin Wang, and Xin Eric Wang. Swapanything: Enabling arbitrary object swapping in personalized image editing. In *European Conference on Computer Vision*, pages 402–418. Springer, 2024. 1, 4

[6] Black Forest Labs. Flux.1-dev. https://bfl.ai/blog/24-08-01-bfl, 2024. 1

[7] Pengzhi Li, Qiang Nie, Ying Chen, Xi Jiang, Kai Wu, Yuhuan Lin, Yong Liu, Jinlong Peng, Chengjie Wang, and Feng Zheng. Tuning-free image customization with image and text guidance. In *European Conference on Computer Vision*, pages 233–250. Springer, 2024. 1, 4

[8] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and

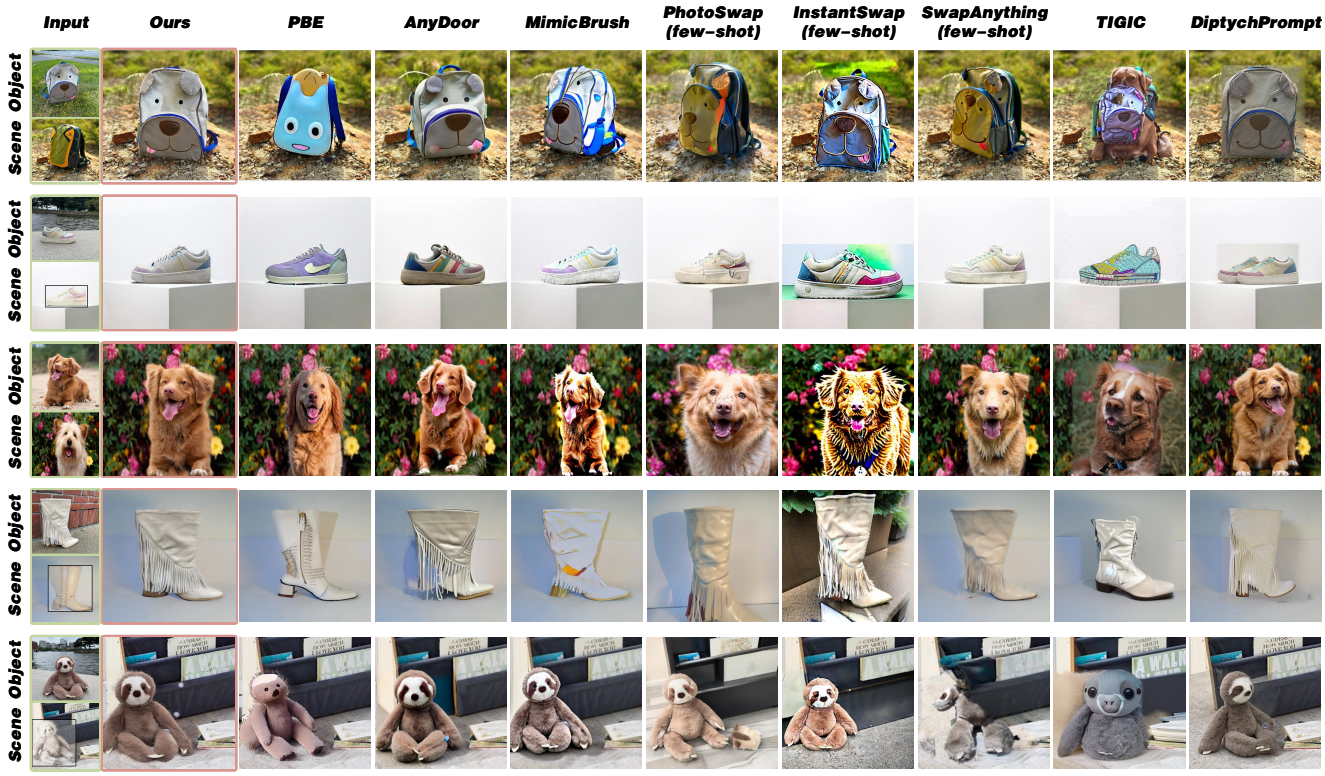

Figure 9. Qualitative comparison with baselines on DreamEditBench.

Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 6038–6047, 2023. 1

[9] OpenAI. Chatgpt5. https://openai.com/index/introducing-gpt-5/, 2025. 4

[10] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024. 2, 3

[11] 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

[12] Chaehun Shin, Jooyoung Choi, Heeseung Kim, and Sungroh Yoon. Large-scale text-to-image model with inpainting is a zero-shot subject-driven image generator. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 7986–7996, 2025. 1, 4

[13] Yizhi Song, Zhifei Zhang, Zhe Lin, Scott Cohen, Brian Price, Jianming Zhang, Soo Ye Kim, He Zhang, Wei Xiong, and Daniel Aliaga. Imprint: Generative object compositing by learning identity-preserving representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8048–8058, 2024. 1

[14] Yikai Wang, Chenjie Cao, Junqiu Yu, Ke Fan, Xiangyang Xue, and Yanwei Fu. Towards enhanced image inpainting: Mitigating unwanted object insertion and preserving color consistency. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 23237–23248, 2025. 1

[15] Daniel Winter, Asaf Shul, Matan Cohen, Dana Berman, Yael Pritch, Alex Rav-Acha, and Yedid Hoshen. Objectmate: A recurrence prior for object insertion and subject-driven generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16281–16291, 2025. 1

[16] Binxin Yang, Shuyang Gu, Bo Zhang, Ting Zhang, Xuejin Chen, Xiaoyan Sun, Dong Chen, and Fang Wen. Paint by example: Exemplar-based image editing with diffusion models. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 18381–18391, 2023. 1, 4

[17] Chenyang Zhu, Kai Li, Yue Ma, Longxiang Tang, Chengyu Fang, Chubin Chen, Qifeng Chen, and Xiu Li. Instantswap: Fast customized concept swapping across sharp shape differences. In *The Thirteenth International Conference on Learning Representations*, 2025. 1, 4

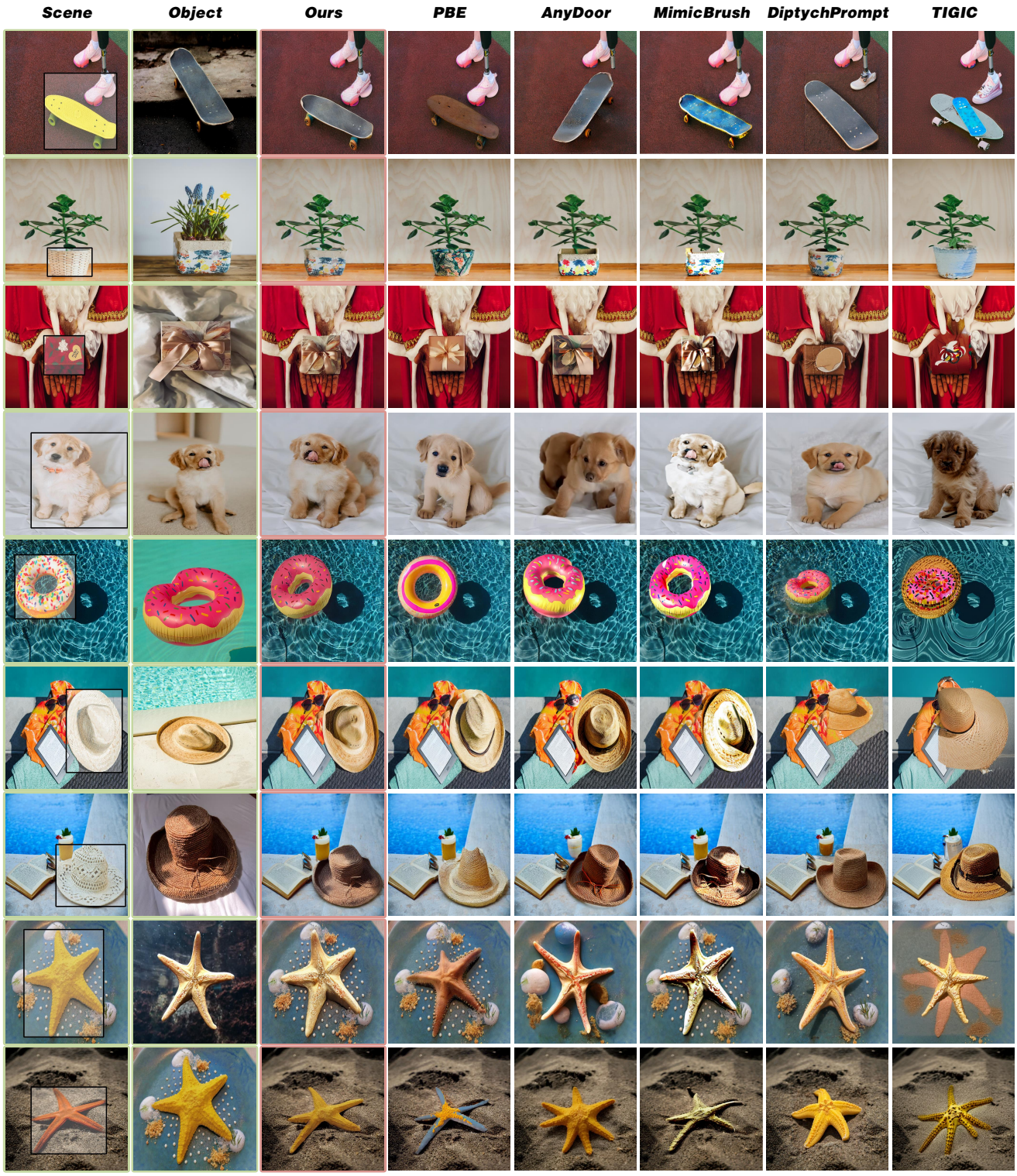

Figure 10. Qualitative comparison with baselines on SourceBench.