SCALINGEDIT: SCALING REAL-WORLD 3D OBJECT EDITING WITH SELF-SUPERVISED TRAINING

Anonymous authors
Paper under double-blind review

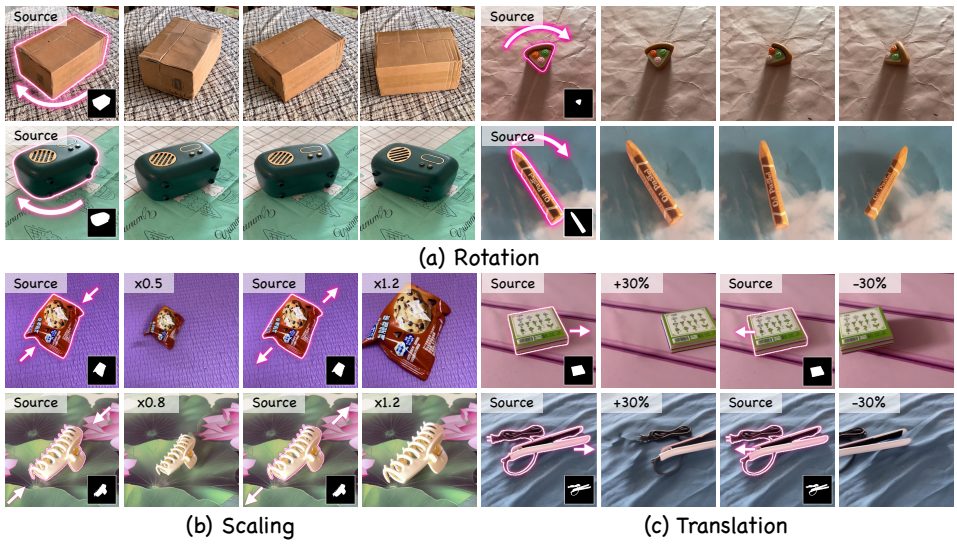


Figure 1: **Results of ScalingEdit**. ScalingEdit demonstrates precise and consistent 3D object manipulation on real-world images, supporting geometric transformations including (a) object rotation, (b) object scaling, and (c) object translation.

ABSTRACT

3D object editing aims to rotate, scale, and translate a target object while preserving its identity and surrounding scene. Scaling up this capability is hindered by the scarcity of real-world editing pairs with geometric supervision. Existing multi-view videos offer diverse object observations and camera poses, but camera motion changes both the object viewpoint and the background, creating a mismatch with scene-preserving editing. To address these challenges, we introduce **ScalingEdit**, a self-supervised framework that turns these observations into training data for geometry-controlled object editing. Specifically, we formulate object editing as a foreground transformation within a shared scene context. We construct scene-aligned training pairs by compositing the source foreground onto the completed target-view background while retaining the unmodified target image as supervision. This construction aligns training with the scene-preserving editing task and enables learning from real multi-view observations without dedicated captures of object manipulation. A geometry-conditioning module encodes relative camera pose and image-plane displacement to provide explicit control over object rotation, scaling, and translation. Experiments on ObjectMover-A and our RotationBench demonstrate improvements over the evaluated baselines in appearance fidelity and geometric accuracy. Our data-scaling experiments further demonstrate the potential to improve geometric object editing by increasing real-world multi-view supervision. Videos are available on our project page.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Recent advances in generative models have enabled flexible image editing through text (Brooks et al., 2023), masks (Yang et al., 2023), layouts (Alzayer et al., 2025), and reference images (Chen et al., 2024). Within this evolving landscape, 3D object editing is an important task that enables explicit control over the spatial configuration of objects in images (Pandey et al., 2024; Sajnani et al., 2025). Given a source image, a mask identifying the target object, and a desired geometric transformation, the task is to generate an edited image in which the object is rotated, scaled, or translated as specified, while preserving its identity and the surrounding scene. Such fine-grained control allows users to specify the orientation, size, and placement of an object relative to its surroundings, making it possible to adjust object alignment, spacing, and composition with precision. This capability supports applications such as product visualization, content creation, and scene design, where users need to explore and refine object configurations without repeatedly capturing or manually reconstructing the scene (Ruan et al., 2026). However, scaling up such training is challenging because real-world editing pairs are difficult to obtain. Such pairs must depict the same object before and after a specified transformation while preserving the surrounding scene. This scarcity makes training data construction a central challenge in scaling geometry-controlled object editing.

To address this data scarcity, existing methods obtain such supervision through two main routes. One line of work relies on 3D assets and rendering pipelines to construct paired training data with explicitly controlled geometric transformations (Michel et al., 2023; Yu et al., 2025; Ruan et al., 2026). Although rendering offers precise geometric control, the limited diversity of assets and discrepancies in materials, lighting, and scene composition can create a substantial domain gap with real-world images. Another line of work starts from real images or videos and uses estimated depth, pose, or reconstructed object geometry to construct geometrically annotated training pairs (Alzayer et al., 2025; Cheng et al., 2025; Ruan et al., 2026; Xu et al., 2026). However, capturing dedicated before-and-after editing pairs is expensive, while converting existing observations into reliable editing supervision often requires reconstruction, pose estimation, or additional rendering. As illustrated in Figure 2, models trained on synthetic data quickly saturate as the dataset grows, while real multi-view supervision continues to yield improvements, underscoring the need for scalable real-world data. These limitations make scalable training data a central bottleneck for real-world 3D object editing.

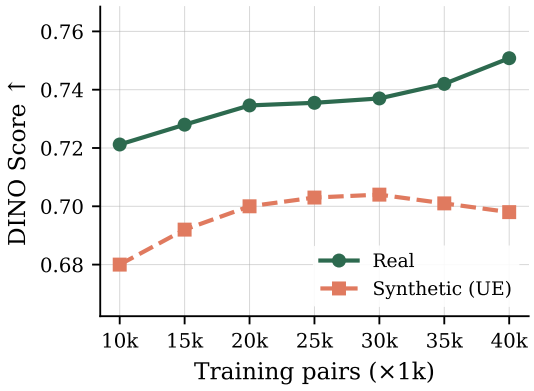


Figure 2: **Scaling real vs. synthetic training data.** Models trained on increasing amounts of real multi-view data (MVImgNet) show consistent improvements in DINO Score, whereas synthetic data (Unreal Engine) saturates early and slightly declines at larger scales.

Our key observation is that 3D object editing can be decomposed into two complementary components: preserving the scene context and transforming the foreground object. This decomposition suggests that useful supervision need not come from explicitly captured object manipulations. Existing large-scale multi-view datasets developed for 3D reconstruction already contain real observations of the same object from different viewpoints (Yu et al., 2023; Han et al., 2024). By selecting one frame as the source and another as the target, we obtain paired observations of the same object under different viewpoints related by a relative camera transformation. This raises a natural question: can existing multi-view data be retargeted into self-supervised training data for 3D object editing? The central challenge is that camera motion changes both the object observation and the background, whereas object editing requires transforming the object within a preserved scene.

Motivated by this observation, we introduce **ScalingEdit**, a self-supervised generative framework for scaling real-world 3D object editing. We develop a scene-aligned training strategy that reorganizes existing multi-view observations into editing supervision, without requiring dedicated captures of object manipulation. By retaining real target observations as supervision, this strategy allows the model to learn from their rich appearance and object–scene interactions. For fine-grained geomet-

ric control, we construct a geometry-conditioning module that encodes the relative transformation between source and target observations and integrates it with visual and textual conditions to guide the generative model. Experiments show that ScalingEdit improves geometric controllability, object appearance consistency, and scene preservation over the evaluated baselines. Our scaling study further shows that the largest-data training configuration achieves the best results (Figure 2), highlighting the potential of scaling real-world multi-view supervision. Overall, our contributions are summarized as follows.

- We propose **ScalingEdit**, a scene-aligned self-supervised approach with a geometry-conditioning module that encodes relative camera pose and image-plane displacement, enabling fine-grained object rotation, scaling, and translation while preserving object identity and scene context.
- We develop a scalable pipeline that converts real-world multi-view videos into scene-aligned editing pairs. By constructing aligned inputs while retaining unmodified target observations, the pipeline provides geometric and appearance supervision without dedicated captures of object manipulation.
- We evaluate translation on ObjectMover-A and introduce **RotationBench** for cross-view object rotation. Experiments demonstrate improvements over the evaluated baselines in geometric controllability, object appearance consistency, and scene preservation. We further conduct a scaling study, highlighting our framework's potential to achieve better performance with increasing amounts of real-world training data.

2 Related Work

2.1 3D-Aware Object Editing

3D-aware object editing aims to manipulate the position, scale, or 3D orientation of a target object while preserving its identity and maintaining consistency with the surrounding scene. Existing methods broadly approach this problem through learned generative priors or explicit geometric constraints. ObjectMover (Yu et al., 2025) exploits the spatiotemporal prior of video generation models to handle perspective changes, shadows, and occlusions during object relocation, but such generative approaches mainly rely on implicit spatial reasoning and offer limited control over precise 3D transformations. A complementary line of work explicitly incorporates geometric information into the editing process. Diffusion Handles (Pandey et al., 2024) lifts diffusion features onto a depth-derived 3D surface and manipulates them in 3D, while GeoDiffuser (Sajnani et al., 2025) directly incorporates geometric transformations into diffusion attention. OBJECT 3DIT (Michel et al., 2023) learns geometry-aware manipulation from procedurally generated 3D editing pairs, and more recent methods such as BlenderFusion (Chen et al., 2025) and GeoEdit (He et al., 2026) employ explicit 3D representations, geometric proxies, or graphics pipelines to provide stronger structural guidance. Ctrl&Shift (Ruan et al., 2026) further demonstrates that relative camera-pose conditioning can provide explicit geometric control within a generative framework without maintaining an explicit 3D representation at inference time. Despite substantial progress in geometric controllability, precise object manipulation still commonly relies on explicit 3D intermediates, synthetic geometric supervision, or specialized data-construction pipelines; obtaining scalable supervision from real-world observations that directly matches the scene-preserving editing setting remains a fundamental challenge.

2.2 Scalable Learning from Real-World Videos

As model capacity continues to grow, learning from large-scale real-world videos has become an increasingly important direction in 3D vision and generative modeling. MonoNeRF (Tian et al., 2023) demonstrates that dynamic 3D representations can be learned across scenes from monocular videos, reducing the dependence on scene-specific optimization. In generative novel-view synthesis, methods such as Generative Camera Dolly (Van Hoorick et al., 2024) obtain accurate camera-control supervision from synthetically generated dynamic multi-view scenes, but the scale and visual diversity of such training data remain constrained by available 3D assets and rendering systems. More recently, Scaling4D (Cai et al., 2026) shows that overcoming this data bottleneck does not necessarily require continuously expanding synthetic datasets; instead, it reformulates video novel-view synthesis as a correspondence-guided generation problem, allowing the geometric information naturally available in large-scale monocular videos to serve as self-supervision and yielding consistent

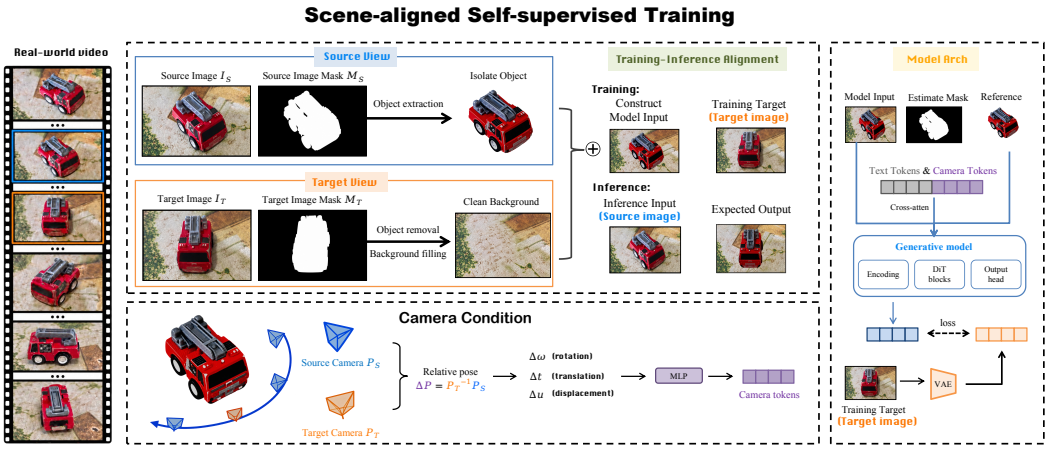


Figure 3: **Overview of ScalingEdit.** ScalingEdit converts real-world multi-view observations into scene-aligned editing supervision while retaining the original target image as real supervision. Relative camera pose is encoded as camera tokens to enable controllable 3D object transformations.

improvements as the amount of real training data increases. These developments suggest that an effective route toward scalable geometric learning is to reformulate the target task so that supervision already latent in real-world videos becomes directly usable. However, such scalable learning paradigms have primarily been explored for scene representation and novel-view synthesis, while 3D-aware object editing introduces the additional requirement of changing local object geometry while preserving the surrounding scene, for which scalable real-world training remains largely underexplored.

3 METHOD

We introduce **ScalingEdit**, a self-supervised framework for learning controllable 3D object editing from real-world multi-view videos. We first formulate the supervision mismatch that arises when camera-induced viewpoint changes are directly used to train scene-preserving object editing. We then address this mismatch from two aspects. In Sec. 3.2, we reorganize real multi-view observations into scene-aligned training pairs whose background remains fixed across the input and target. In Sec. 3.3, we convert the relative camera relation between the paired observations into an explicit control signal and inject it into the generative model, enabling a unified interface for rotation, scaling, and translation.

3.1 PROBLEM FORMULATION

Given a source image I_S, a mask M_S identifying the object of interest, and a geometric condition $\mathbf{g}$, our goal is to transform the object while preserving its identity and the surrounding scene. Let $\mathcal{C}(B, F)$ denote an image composed of a background B and an object foreground F. The desired editing process is

$$\mathcal{C}(B, F_S) \xrightarrow{\mathbf{g}} \mathcal{C}(B, F_T), \tag{1}$$

where F_S and F_T represent the source and desired object states. Learning this mapping requires paired supervision that captures the same object under different geometric states within a shared scene context. The pair must therefore specify both the object transformation to be learned and the scene content to be preserved.

Real-world multi-view videos provide a sequence of observations of the same object, represented as $\{\mathcal{C}(B_i, F_i)\}_{i=1}^N$, together with camera poses $\{P_i\}_{i=1}^N$, where B_i and F_i denote the background and object foreground observed in frame i. For a static object observed by a moving camera, these observations capture changes in the object's appearance and orientation relative to the camera, as well as changes in the surrounding scene. They therefore provide view-induced object variation and

relative-pose supervision without directly satisfying the shared-scene structure required for editing. The challenge is to retain this observed variation and its associated relative camera transformations while reorganizing the observations into supervision for scene-preserving editing.

3.2 Scene-Aligned Self-Supervision

Our goal is to exploit the view-induced object variation observed in real-world multi-view videos to construct paired supervision for scene-preserving geometric editing. Such supervision requires two properties to hold jointly: the pair must retain a view-induced change in object appearance associated with a relative geometric condition, while maintaining a common scene context across the input and target. Raw multi-view observations couple these factors through camera motion and therefore do not directly satisfy this requirement. We construct the desired supervision by separating the source object state from its original scene context and recombining it with the target-side background, thereby retaining the observed cross-view variation while removing the need to predict the accompanying global scene change.

Foreground extraction and background completion. As illustrated in Fig. 3, we randomly sample two frames from an object-centric video as the source image I_S and target image I_T, together with their object masks M_S, M_T and camera extrinsics P_S, P_T. The masks can be obtained through foreground segmentation, while camera extrinsics can be recovered through multi-view geometry estimation, such as structure from motion, to characterize the relative geometric relation between observations. The source image supplies the object's initial appearance and orientation, whereas the target image supplies the desired object observation and its scene context. We process them through two complementary branches: the source branch extracts $F_S = \mathcal{E}(I_S, M_S)$, where $\mathcal{E}$ denotes mask-based foreground extraction; the target branch removes the object indicated by M_T and completes the exposed region to obtain $B_T = \mathcal{H}(I_T, M_T)$, where $\mathcal{H}$ denotes object removal and background completion. The resulting F_S and B_T respectively provide the object state to be transformed and the scene in which that transformation is supervised.

Scene-aligned input composition. We merge the two branches by compositing the source foreground onto the completed target-side background:

$$\tilde{I} = \mathcal{C}(B_T, F_S), \qquad (\text{input}, \text{target}) = (\tilde{I}, I_T), \tag{2}$$

where $\mathcal{C}$ denotes foreground–background composition. The constructed input and target share the target-view scene context while depicting the object in its source and target states, respectively. This establishes the same scene-preserving relationship required at inference: a real input image supplies the scene, and the generated output should retain that scene while transforming the object according to the geometric condition. As illustrated in Fig. 3, training and inference therefore share a common editing structure, even though the training input is constructed and the inference input is a real image. Only the conditioning input is synthesized; the supervision target remains the unmodified observation I_T, allowing target appearance and object–scene interactions to be learned directly from real imagery.

3.3 Camera-Conditioned Editing Model

Geometry condition encoding. Fine-grained object manipulation requires an explicit geometric condition that specifies how the source object should transform. The camera extrinsics associated with multi-view observations provide a basis for constructing this condition. Given source and target camera-to-world transformations P_S and P_T, we compute their relative transformation as

$$\Delta P = P_T^{-1} P_S = \begin{bmatrix} \Delta R & \Delta t \\ 0 & 1 \end{bmatrix}. \tag{3}$$

We represent ΔR using a three-dimensional rotation vector $\Delta\omega$, and normalize the three-dimensional translation Δt by the sequence scale to obtain $\widetilde{\Delta t}$ (Appendix A.1). We combine these quantities with normalized image-plane displacement $\Delta u = (\Delta u_x, \Delta u_y)$, yielding the eight-dimensional descriptor $\mathbf{g} = [\Delta\omega; \widetilde{\Delta t}; \Delta u]$. These components describe complementary aspects of the requested edit: relative rotation specifies orientation changes, relative camera translation provides cues for changes in apparent scale, and image-plane displacement specifies horizontal and

vertical relocation. Together, these components provide explicit control cues for rotation, scaling, and translation.

Conditioning architecture. We augment a pretrained image generation model supporting visual and text conditioning with an explicit geometry-conditioning pathway for fine-grained object control. A lightweight camera encoder maps the geometric descriptor into camera tokens $Z_{\text{cam}} = f_{\text{cam}}(\mathbf{g})$, which are appended to the text tokens to form the joint conditioning sequence $Z_{\text{cond}} = [Z_{\text{text}}; Z_{\text{cam}}]$. This sequence is injected into the generative model through cross-attention. In parallel, the visual conditioning pathway receives a context image I_{ctx}, an object reference I_{ref}, and an editing mask $\hat{M}$. During training, $I_{\text{ctx}} = \tilde{I}$ is the scene-aligned input constructed in Sec. 3.2, while the reference is extracted from the source observation. The context and reference provide the scene and object appearance, and the editing mask specifies the region to be manipulated. Together, the visual and geometric conditions determine the content to preserve and the transformation to apply.

Training objective and inference. We retain the pretrained model's flow-matching objective (Lipman et al., 2023), using the unmodified target image I_T as supervision. Let $z_0 = \mathcal{V}(I_T)$ be its latent encoding and $z_\tau = (1-\tau)z_0 + \tau\epsilon$ the noisy latent, where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$. Given the combined visual and text–geometry conditions c, the model is optimized with $\mathcal{L}_{\text{FM}} = \mathbb{E}[w(\tau)\|v_\theta(z_\tau, \tau; c) - (\epsilon - z_0)\|_2^2]$, where timestep sampling and weighting follow the pretrained model. The editing mask serves as a visual condition rather than a loss mask. At inference, the real source image replaces the constructed input, and the requested transformation is encoded through the same geometry-conditioning pathway to generate the edited image without a target observation. A user-specified scale ratio controls the estimated target mask only; it does not alter the eight-dimensional pose descriptor.

4 Experiments

4.1 Implementation Details

We build ScalingEdit on Wan2.1 (Team Wan et al., 2025) and fine-tune it with rank-32 LoRA. Each image-editing sample is packaged in the temporal input format expected by the video backbone. Training uses 40,000 source–target image pairs from 2,000 objects from 100 MVImgNet2.0 (Han et al., 2024) categories (20 objects per category and 20 pairs per object). Images are resized to 768×432. We train for 62,500 steps using AdamW with a learning rate of 5×10^{-5}, weight decay of 1×10^{-4}, and 625 warmup steps. Additional training details are in Appendix C.

4.2 RotationBench

ObjectMover-A evaluates scene-preserving object translation, but does not directly test whether a model can transform an object across viewpoints. To address this limitation, we introduce **RotationBench**, a cross-view geometric editing benchmark constructed from images collected online. It contains 200 source–target pairs from 50 categories disjoint from training, with each pair providing a source image, object mask, target view, and relative camera transformation recovered by COLMAP. The benchmark is designed to evaluate both appearance preservation and controllable geometric transformation. Detailed construction and evaluation protocols are provided in Appendix B.2.

4.3 Comparison with Baselines

4.3.1 Quantitative Comparison

ObjectMover-A. We evaluate scene-preserving translation on 200 zero-shot examples from ObjectMover-A (Yu et al., 2025), comparing against OBJECT 3DIT (Michel et al., 2023), Paint by Example (Yang et al., 2023), AnyDoor (Chen et al., 2024), MagicFixup (Alzayer et al., 2025), and DragAnything (Wu et al., 2024). We report full-image PSNR, DINO-Score, CLIP-Score, and DreamSim under the same evaluation protocol for all methods (Appendix B.3). ScalingEdit leads on all four metrics (Table 1), with a 5.59 dB PSNR margin over the strongest baseline. Scene-aligned supervision trains the model to preserve the background while relocating the object.



(a) ObjectMover-A



(b) RotationBench

Figure 4: **Qualitative comparisons on the two benchmarks.** (a) Red boxes in the Source and GT columns indicate the source and target object locations. ScalingEdit relocates the object while preserving its identity and a clean background. (b) Methods transform the source object to a target pose observed from another viewpoint. ScalingEdit best matches the ground truth while preserving texture and fine details.

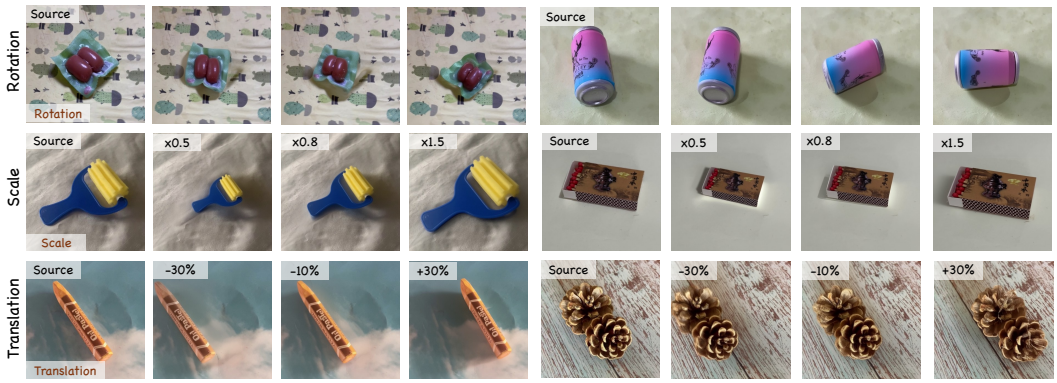


Figure 5: **Geometric control of ScalingEdit.** Given a source image and a requested transformation, ScalingEdit produces smooth object rotation, scaling ($0.5\times$ to $1.5\times$), and translation ($\pm 10\%$ to $\pm 30\%$ of the image width) while keeping the background and illumination fixed. These continuous transformations have no ground truth and are shown as a capability demonstration.

RotationBench. We compare against OBJECT 3DIT, DragAnything, VACE (Jiang et al., 2025), and Nano-Banana (Google DeepMind, 2025). We report PSNR, DINO-Score, CLIP-Score, DreamSim,

Table 1: Results on ObjectMover-A (200 examples). Best and second-best results are **bolded** and underlined. The Rotation and Scaling columns indicate whether a method provides an explicit control interface for the corresponding transformation.

Method	PSNR ↑	DINO ↑	CLIP ↑	DSim ↓	Rotation	Scaling
OBJECT 3DIT (Michel et al., 2023)	19.93	0.4681	0.8621	0.4364	✓	✓
Paint by Ex. (Yang et al., 2023)	23.43	0.6236	0.8728	0.3471	×	×
AnyDoor (Chen et al., 2024)	22.40	0.5789	0.8672	0.3937	×	×
MagicFixup (Alzayer et al., 2025)	22.96	0.5812	0.8630	0.3998	×	×
DragAnything (Wu et al., 2024)	16.44	0.4434	0.8610	0.4785	×	×
Ours	**29.02**	**0.7368**	**0.9229**	**0.2389**	✓	✓

Table 2: Results on RotationBench. The first four metrics are computed on all 200 pairs. Best and second-best results are **bolded** and underlined.

Method	PSNR ↑	DINO ↑	CLIP ↑	DSim ↓	Obj. IoU ↑	Pose MAPE ↓
OBJECT 3DIT (Michel et al., 2023)	19.58	0.3351	0.7115	0.5972	0.07	299.87%
DragAnything (Wu et al., 2024)	20.75	0.5957	0.8101	0.3383	0.51	81.26%
VACE-1.3B (Jiang et al., 2025)	20.66	0.6955	0.8549	0.2805	0.58	66.86%
Nano-Banana (Google DeepMind, 2025)	20.47	0.7126	0.8467	0.3114	0.35	104.27%
Ours	**22.38**	**0.7508**	**0.8614**	**0.2245**	**0.71**	**28.01%**

Object IoU, and Pose MAPE. ScalingEdit leads on all six metrics (Table 2), with a 1.63 dB PSNR margin over the strongest baseline. Its Object IoU of 0.71 and Pose MAPE of 28.01% indicate more accurate object placement and orientation. Baseline adaptation details are provided in Appendix B.1.

4.3.2 Qualitative Comparison

ObjectMover-A. Figure 4(a) compares object translation results. Several baselines leave residual traces, distort object identity, or produce blurred copies, whereas ScalingEdit relocates the object while preserving its appearance and the surrounding scene. **RotationBench.** Figure 4(b) evaluates cross-view editing. Translation-based methods fail to apply the requested pose change, while other baselines alter object texture or geometry. ScalingEdit most closely matches the target view and preserves fine appearance details. **Geometric control.** Figure 5 further shows smooth and consistent responses to continuous rotation, scaling, and translation commands, while the background remains fixed. These examples have no ground truth and are presented as a capability demonstration.

4.4 Scaling Study

We examine the scalability of ScalingEdit, i.e., whether it continues to benefit from larger amounts of real multi-view supervision. We train the same architecture on seven configurations from 10k to 40k training pairs (in increments of 5k), while keeping the inference and evaluation protocol fixed. All configurations are evaluated under the RotationBench protocol of Sec. 4.2. The 40k model achieves the best results across all metrics, gaining 0.65 dB in PSNR and improving DINO Score from 0.7212 to 0.7508 over the 10k baseline, with no clear sign of saturation. This confirms that our scene-aligned self-supervision effectively converts additional real-world observations into improved geometric controllability and generation quality.

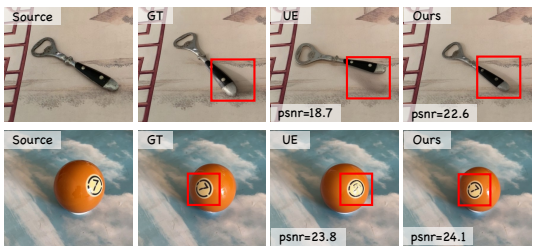


Figure 6: Real data vs. UE synthetic data on RotationBench.

Real vs. synthetic data. To further understand *why* real-world supervision scales effectively, we compare models trained on the same amount of data from two sources: real multi-view videos

(MVImgNet) and synthetic renderings produced by Unreal Engine (UE). Figure 6 shows qualitative results on RotationBench. The UE-trained model tends to lose fine surface details such as specular highlights and printed textures, and produces less accurate shadows—artifacts attributable to the domain gap between rendered and real-world imagery. In contrast, the model trained on real data faithfully preserves these appearance cues, confirming that real multi-view supervision provides richer and more transferable geometric editing signals than synthetic alternatives.

4.5 Ablation Study

We ablate three components on RotationBench (Table 3): the geometry-conditioning module, the rotation vector $\Delta\omega$, and the object reference image. Removing geometry conditioning causes the largest Pose MAPE increase, confirming it as the primary driver of geometric control. Removing $\Delta\omega$ increases Pose MAPE from 28.01% to 49.37%, indicating that the rotation vector provides complementary geometric information beyond Δt and Δu. Removing the reference image yields the largest DINO drop while retaining reasonable pose accuracy, showing its role is primarily in appearance preservation. Figure 7 illustrates these complementary failure modes: without geometry conditioning, the model largely ignores the requested rotation and retains the source pose; without the reference image, the model applies the correct transformation but loses object appearance—for example, the earplug's orange surface becomes desaturated and its shape turns amorphous. The full model produces both the correct pose and faithful appearance.

Table 3: Ablation study on RotationBench.

Variant	PSNR ↑	DINO ↑	Pose MAPE ↓
w/o Geometry	20.94	0.7035	72.58%
w/o $\Delta\omega$	21.83	0.7294	49.37%
w/o Reference	21.56	0.6742	34.26%
Full	**22.38**	**0.7508**	**28.01%**

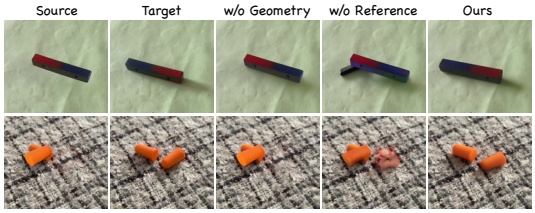


Figure 7: **Qualitative ablation on RotationBench.** Removing geometry conditioning (w/o Geometry) leads to incorrect object pose, while removing the reference image (w/o Reference) degrades appearance fidelity. The full model preserves both.

5 Conclusion

We introduced **ScalingEdit**, a self-supervised framework for scalable 3D object editing from real-world multi-view videos. Our central idea is to transform naturally occurring multi-view observations into scene-preserving editing supervision by separating camera-induced scene variation from the object-state transformation of interest. Based on this formulation, ScalingEdit combines scene-aligned real-image supervision with explicit relative-pose conditioning, enabling controllable object rotation, scaling, and translation without requiring explicit 3D reconstruction at inference time. Experiments on real-world object manipulation and cross-view editing demonstrate strong geometric controllability, background preservation, and appearance consistency, while the scaling study shows that performance continues to improve as more real-world training data is introduced. We hope this work provides a practical direction toward scaling geometry-aware visual editing by directly exploiting the rich geometric supervision already available in real-world videos.

Limitation and discussion. Our framework has two main limitations. First, the current data construction assumes that the target object remains static across the input video, such that cross-view changes can be attributed to camera motion. Videos containing independently moving objects violate this assumption and cannot be directly used for training; extending our framework to such data would require disentangling object motion from camera motion. Second, although our explicit geometric representation enables precise and continuous control, manually specifying transformation parameters is not sufficiently user-friendly. An important future direction is to translate intuitive user interactions, such as dragging, rotating, or scaling an object directly in the image, into the corresponding geometric control signals.

AI USE STATEMENT

Generative AI assistance was used to writing, translation, literature searches, reference formatting, and discussion of method presentation and reported results.

REFERENCES

Hadi Alzayer, Zhihao Xia, Xuaner (Cecilia) Zhang, Eli Shechtman, Jia-Bin Huang, and Michael Gharbi. Magic fixup: Streamlining photo editing by watching dynamic videos. *ACM Transactions on Graphics*, 2025. doi: 10.1145/3750722. URL https://doi.org/10.1145/3750722.

Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18392–18402, 2023. doi: 10.1109/CVPR52729.2023.01764.

Hongrui Cai, Junjie Luo, Zhihong Fu, Shengnan Zhu, Jiawei Wen, Wanquan Feng, Songtao Zhao, and Qian He. Scaling4D: Pushing the frontier of video novel view synthesis through large-scale monocular videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://rainbowrui.github.io/scaling4d/.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9650–9660, 2021. doi: 10.1109/ICCV48922.2021.00951.

Jiacheng Chen, Ramin Mehran, Xuhui Jia, Saining Xie, and Sanghyun Woo. Blenderfusion: 3d-grounded visual editing and generative compositing. *arXiv preprint arXiv:2506.17450*, 2025. URL https://arxiv.org/abs/2506.17450.

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*, pp. 6593–6602, 2024. doi: 10.1109/CVPR52733.2024.00630.

Yen-Chi Cheng, Krishna Kumar Singh, Jae Shin Yoon, Alexander Schwing, Liangyan Gui, Matheus Gadelha, Paul Guerrero, and Nanxuan Zhao. 3D-Fixup: Advancing photo editing with 3d priors. In *Proceedings of the SIGGRAPH Conference Papers*. Association for Computing Machinery, 2025. doi: 10.1145/3721238.3730695.

Stephanie Fu, Netanel Tamir, Shobhita Sundaram, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip Isola. Dreamsim: Learning new dimensions of human visual similarity using synthetic data. In *Advances in Neural Information Processing Systems*, volume 36, pp. 50742–50768, 2023.

Google DeepMind. Nano banana: Gemini 2.5 flash image. https://deepmind.google/models/gemini/nano-banana/, 2025.

Xiaoguang Han, Yushuang Wu, Luyue Shi, Haolin Liu, Hongjie Liao, Lingteng Qiu, Weihao Yuan, Xiaodong Gu, Zilong Dong, and Shuguang Cui. MVImgNet2.0: A larger-scale dataset of multi-view images. *ACM Transactions on Graphics*, 43(6), 2024. doi: 10.1145/3687973. URL https://doi.org/10.1145/3687973.

Yi He, Jiangming Wang, Xinyu Wang, Mark Fong, Songchun Zhang, Yuxuan Xue, Hai-Tao Zheng, and Yue Ma. Geoedit: Geometry-aware object editing via dual-branch denoising. In *European Conference on Computer Vision*, 2026. URL https://arxiv.org/abs/2606.30003.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. VACE: All-in-one video creation and editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17191–17202, 2025. URL https://arxiv.org/abs/2503.07598.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4015–4026, 2023. doi: 10.1109/ICCV51070.2023.00371. URL https://openaccess.thecvf.com/content/ICCV2023/html/Kirillov_Segment_Anything_ICCV_2023_paper.html.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 12888–12900. PMLR, 2022.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. In *Proceedings of the European Conference on Computer Vision*, pp. 38–55, 2024. doi: 10.1007/978-3-031-72970-6_3. URL https://www.ecva.net/papers/eccv_2024/papers_ECCV/html/6319_ECCV_2024_paper.php.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Oscar Michel, Anand Bhattad, Eli VanderBilt, Ranjay Krishna, Aniruddha Kembhavi, and Tanmay Gupta. OBJECT 3DIT: Language-guided 3d-aware image editing. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Karran Pandey, Paul Guerrero, Matheus Gadelha, Yannick Hold-Geoffroy, Karan Singh, and Niloy J. Mitra. Diffusion handles: Enabling 3d edits for diffusion models by lifting activations to 3d. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7695–7704, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 8748–8763. PMLR, 2021.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos, 2024. URL https://arxiv.org/abs/2408.00714.

Tianhe Ren and Shuo Shen. Grounded SAM 2: Ground and track anything in videos. GitHub repository, IDEA-Research, 2024. URL https://github.com/IDEA-Research/Grounded-SAM-2.

Penghui Ruan, Bojia Zi, Xianbiao Qi, Youze Huang, Rong Xiao, Pichao Wang, Jiannong Cao, and Yuhui Shi. Ctrl&Shift: High-quality geometry-aware object manipulation in visual generation. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2602.11440.

Rahul Sajnani, Jeroen Vanbaar, Jie Min, Kapil D. Katyal, and Srinath Sridhar. Geodiffuser: Geometry-based image editing with diffusion models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 472–482, 2025. doi: 10.1109/WACV61041.2025.00056.

Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4104–4113, 2016.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. URL https://arxiv.org/abs/2503.20314.

Fengrui Tian, Shaoyi Du, and Yueqi Duan. Mononerf: Learning a generalizable dynamic radiance field from monocular videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17903–17913, 2023. doi: 10.1109/ICCV51070.2023.01641.

Dmitry Tochilkin, David Pankratz, Zexiang Liu, Zixuan Huang, Adam Letts, Yangguang Li, Ding Liang, Christian Laforte, Varun Jampani, and Yan-Pei Cao. TripoSR: Fast 3D object reconstruction from a single image. *arXiv preprint arXiv:2403.02151*, 2024.

Basile Van Hoorick, Rundi Wu, Ege Ozguroglu, Kyle Sargent, Ruoshi Liu, Pavel Tokmakov, Achal Dave, Changxi Zheng, and Carl Vondrick. Generative camera dolly: Extreme monocular dynamic novel view synthesis. In *European Conference on Computer Vision*, 2024. URL https://gcd.cs.columbia.edu.

Weijia Wu, Zhuang Li, Yuchao Gu, Rui Zhao, Yefei He, David Junhao Zhang, Mike Zheng Shou, Yan Li, Tingting Gao, and Di Zhang. Draganything: Motion control for anything using entity representation. In *European Conference on Computer Vision*, pp. 331–348, 2024. doi: 10.1007/978-3-031-72670-5_19.

Ruihang Xu, Dewei Zhou, Xiaolong Shen, Fan Ma, and Yi Yang. Phyedit: Towards real-world object manipulation via physically-grounded image editing, 2026. URL https://arxiv.org/abs/2604.07230. Accepted to ACM Multimedia 2026.

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*, pp. 18381–18391, 2023.

Xianggang Yu, Mutian Xu, Yidan Zhang, Haolin Liu, Chongjie Ye, Yushuang Wu, Zizheng Yan, Chenming Zhu, Zhangyang Xiong, Tianyou Liang, Guanying Chen, Shuguang Cui, and Xiaoguang Han. MVImgNet: A large-scale dataset of multi-view images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9150–9161, 2023. doi: 10.1109/CVPR52729.2023.00883.

Xin Yu, Tianyu Wang, Soo Ye Kim, Paul Guerrero, Xi Chen, Qing Liu, Zhe Lin, and Xiaojuan Qi. Objectmover: Generative object movement with video prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17682–17691, 2025.

Jixin Zhao, Zhouxia Wang, Peiqing Yang, and Shangchen Zhou. Precise object and effect removal with adaptive target-aware attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://arxiv.org/abs/2505.22636.

APPENDIX

A Detailed Implementation **13**

A.1 Camera Encoding 13

A.2 Training Data and Supervision 14

A.3 Model and Optimization 14

B Evaluation Details **14**

B.1 Baseline Adaptation 14

B.2 Benchmark Details 14

B.3 Evaluation Metrics 15

C Training and Inference Computation Cost **16**

A DETAILED IMPLEMENTATION

A.1 CAMERA ENCODING

Relative camera transformation. COLMAP (Schönberger & Frahm, 2016) provides world-to-camera extrinsics, denoted by $E_i = [R_i \mid t_i]$, such that a world point X is mapped to $R_i X + t_i$. The camera-to-world matrices in Eq. 3 are therefore $P_i = E_i^{-1}$, with homogeneous coordinates understood. For source and target observations, the source-to-target camera transformation is $\Delta P = E_T E_S^{-1}$, giving

$$\Delta R = R_T R_S^\top, \qquad \Delta t = t_T - \Delta R\, t_S. \tag{4}$$

Rotation: three dimensions. We convert ΔR into the rotation vector $\Delta\omega = \theta\mathbf{a}$, where $\mathbf{a}$ is the unit rotation axis and θ is the rotation angle in radians. For $0 < \theta < \pi$,

$$\theta = \arccos\left(\frac{\mathrm{tr}(\Delta R) - 1}{2}\right), \qquad \mathbf{a} = \frac{1}{2\sin\theta}\begin{bmatrix}\Delta R_{32} - \Delta R_{23}\\ \Delta R_{13} - \Delta R_{31}\\ \Delta R_{21} - \Delta R_{12}\end{bmatrix}. \tag{5}$$

The three entries are the components of the axis–angle rotation vector.

Translation: three dimensions. We normalize relative translation by a scale estimated from the entire sequence. Let $C_i = -R_i^\top t_i$ be the registered camera centers and let $\bar{C}$ be their coordinate-wise median. The sequence scale s_{seq} is the median of the non-negligible distances $\|C_i - \bar{C}\|_2$. The next three entries are

$$\widetilde{\Delta t} = \frac{\Delta t}{\max(s_{\mathrm{seq}}, \epsilon)}. \tag{6}$$

This normalization removes the arbitrary reconstruction scale while retaining translation magnitude relative to the camera trajectory.

Image-plane displacement: two dimensions. For each foreground mask, we average the horizontal and vertical coordinates of its foreground pixels to obtain the centroid $c_i = (c_{i,x}, c_{i,y})$. The final two entries encode the source-to-target centroid displacement in normalized image coordinates:

$$\Delta u_x = \frac{2(c_{T,x} - c_{S,x})}{W}, \qquad \Delta u_y = \frac{2(c_{T,y} - c_{S,y})}{H}. \tag{7}$$

Here W and H are the image width and height; when the mask dimensions differ, the preprocessing uses the larger width and height of the pair. Positive displacement denotes movement to the right or downward, respectively. For ObjectMover-A, which has no camera extrinsics, the first six entries are zero and the last two are computed from the supplied source and target bounding-box centers using the same normalization.

Vector assembly and token projection. We concatenate the three groups in the following order:

$$\mathbf{g} = [\Delta\omega_x, \Delta\omega_y, \Delta\omega_z, \widetilde{\Delta t}_x, \widetilde{\Delta t}_y, \widetilde{\Delta t}_z, \Delta u_x, \Delta u_y]^\top \in \mathbb{R}^8. \tag{8}$$

The camera encoder augments these eight values with sine and cosine features at 16 logarithmically spaced frequencies from 1 to 64, using phases $\pi f_k \mathbf{g}$. Concatenating the original vector with these Fourier features yields 264 values. A three-layer MLP with two 1,024-dimensional hidden layers, SiLU activations, and dropout 0.05 projects them to 8×4096 values, which are reshaped into eight camera tokens. These tokens are appended to the text-conditioning sequence. At inference, the user-specified scale ratio is used only to construct the estimated target mask. It is not converted into camera translation and does not modify the eight-dimensional pose descriptor.

A.2 Training Data and Supervision

Data organization. The training set contains 100 MVImgNet2.0 object categories, with 20 instances per category and 20 source–target image pairs per instance, for 2,000 objects and 40,000 training pairs. For frame index k in a video with frame rate FPS, its timestamp is $\tau_k = k/\mathrm{FPS}$. Eligible frames must have registered camera poses and nonempty foreground masks that remain nonempty after training-time resizing.

View-pair selection. For each object, we randomly sample 20 source–target image pairs from its available multi-view images. The relative camera poses of each pair provide the geometric conditions for training.

Scene-aligned supervision. Following Sec. 3.2, we extract the source foreground using M_S and use ObjectClear (Zhao et al., 2026) to remove the object indicated by M_T and complete the exposed target-side background. Compositing the source foreground onto this background yields the context image $\tilde{I}$. The source object also supplies the reference condition. ObjectClear is used only to construct the training input, while the supervision target remains the unmodified real observation I_T.

A.3 Model and Optimization

Model configuration. We use Wan2.1-14B as the editing backbone, with rank-32 LoRA (Hu et al., 2022) adaptation and a trainable camera encoder. We organize each image-editing sample in the temporal input format expected by Wan2.1. The camera encoder maps the eight-dimensional descriptor defined in Appendix A.1 to camera tokens, which are concatenated with the text tokens and supplied to cross-attention. The visual pathway receives the context image, source-object reference, and editing mask.

Optimization settings and training cost are detailed in Appendix C.

B Evaluation Details

B.1 Baseline Adaptation

On ObjectMover-A, OBJECT 3DIT uses BLIP-base (Li et al., 2022) captions, and DragAnything uses 25 frames with CFG 3.0. On RotationBench, OBJECT 3DIT takes 256×256 inputs with CFG 3.0 and a rotation angle derived from the relative camera rotation; DragAnything uses 320×576 inputs, 25 frames, CFG 3.0, and a linear-Gaussian trajectory from the source to the target object location; VACE-1.3B uses Moving-Anything bounding-box control under the official 1.3B defaults; Nano-Banana (Gemini 2.5 Flash Image) is prompted with the source image, the estimated target mask, and the NDC displacement together with the yaw/pitch differences. All methods use their released checkpoints and standard inference interfaces, and all receive the same source image, object mask, and target specification.

B.2 Benchmark Details

ObjectMover-A. ObjectMover-A (ObjMove-A in the original paper) (Yu et al., 2025) is a real-image benchmark for object translation containing 200 image sets captured by photographers. Each

set depicts the same scene with an object physically placed at two distinct locations and includes a clean background image without the object. The source image and object mask identify the object to translate, while the target-location mask specifies its destination. The photograph captured with the object at the target location serves as the ground-truth reference. The benchmark uses SAM-generated object masks (Kirillov et al., 2023) and tests whether an edit preserves object identity, removes the object from its original location, and produces a plausible result at the requested target location. We use all 200 examples for evaluation and none for training.

RotationBench. RotationBench is constructed from images collected online. It contains 200 source–target pairs from 50 object categories, with four distinct object instances per category. Each pair provides source and target images, object masks, and the corresponding COLMAP reconstruction. The 50 evaluation categories are disjoint from the 100 training categories. Relative camera rotation angles range from 19.8° to 67.0°; 199 pairs fall in the medium rotation band and one in the small rotation band. This benchmark tests whether a model can transform an object to its target viewpoint while preserving object identity and scene consistency.

B.3 Evaluation Metrics

On ObjectMover-A, PSNR measures image-level reconstruction fidelity and scene preservation, while DINO-Score (Caron et al., 2021), CLIP-Score (Radford et al., 2021), and DreamSim (Fu et al., 2023) characterize the semantic and perceptual consistency of the manipulated object.

Image fidelity. For a generated image $\hat{I}$ and a reference image $I^\star$ with N scalar pixel values, PSNR is

$$\mathrm{MSE} = \frac{1}{N}\sum_{j=1}^{N}(\hat{I}_j - I^\star_j)^2, \qquad \mathrm{PSNR} = 10\log_{10}\frac{L^2}{\mathrm{MSE}}, \tag{9}$$

where L is the maximum pixel value. We compute PSNR over the full image, with higher values indicating greater pixel-level agreement.

Feature similarity and perceptual distance. For a prediction–reference image pair $(\hat{I}, I^\star)$ and visual encoder f, the feature score is cosine similarity,

$$S_f(\hat{I}, I^\star) = \frac{f(\hat{I})^\top f(I^\star)}{\|f(\hat{I})\|_2\,\|f(I^\star)\|_2}. \tag{10}$$

DINO and CLIP scores compare visual features after ℓ_2 normalization. CLIP-Score measures image–image similarity. DreamSim uses the pretrained ensemble and its supplied preprocessing. Higher similarity and lower DreamSim distance indicate better appearance agreement. We report the mean score across examples.

ObjectMover-A reference protocol. The original ObjectMover protocol (Yu et al., 2025) computes PSNR over the full generated image against the target photograph. DINO-Score, CLIP-Score, and DreamSim are computed separately on the source and target regions cropped from the generated and reference images, and the two region scores are averaged. This evaluates both removal at the original location and insertion at the destination.

Geometric metric computation. On RotationBench, we additionally compute two geometric metrics. To obtain Pose MAPE, we reconstruct a 3D mesh from the source image and mask using TripoSR (Tochilkin et al., 2024) and estimate object pose via coarse-to-fine silhouette fitting under orthographic projection. Pose MAPE reports the mean absolute percentage error over five pose parameters (yaw, pitch, scale, and 2D position) between the estimated and ground-truth deltas relative to the source pose. Object IoU is measured by segmenting the edited output with Grounded-SAM-2 (Ren & Shen, 2024; Liu et al., 2024; Ravi et al., 2024) and computing intersection-over-union against the ground-truth target mask. For the generated-object mask $\hat{M}$ and target mask M_T,

$$\mathrm{IoU}(\hat{M}, M_T) = \frac{|\hat{M} \cap M_T|}{|\hat{M} \cup M_T|}. \tag{11}$$

Let $\mathbf{q} = (\psi, \vartheta, s, u, v)$ collect the five pose parameters. For nonzero target changes, the absolute percentage-error term is $100|\widehat{\Delta q}_k - \Delta q_k^\star|/|\Delta q_k^\star|$.

C Training and Inference Computation Cost

Training cost. All images are resized to 768×432. We train the rank-32 LoRA for 62,500 optimization steps using AdamW (Loshchilov & Hutter, 2019) with a learning rate of 5×10^{-5} and weight decay of 1×10^{-4}; the learning rate is linearly warmed up for the first 625 steps. Unless otherwise specified, experiments use this configuration. We train on four NVIDIA H100 GPUs with a per-device batch size of eight and two gradient-accumulation steps, giving an effective batch size of 64. Training uses BF16 mixed precision and gradient checkpointing, and takes 30 hours in total. The learning rate remains constant after warm-up. The camera encoder is trained jointly with LoRA. The 10k- to 35k-pair scaling experiments use the same number of optimization steps.

Inference configuration. Inference uses 28 denoising steps with a guidance scale of 1.0. Each example generates a single image using one reference-prefix frame.

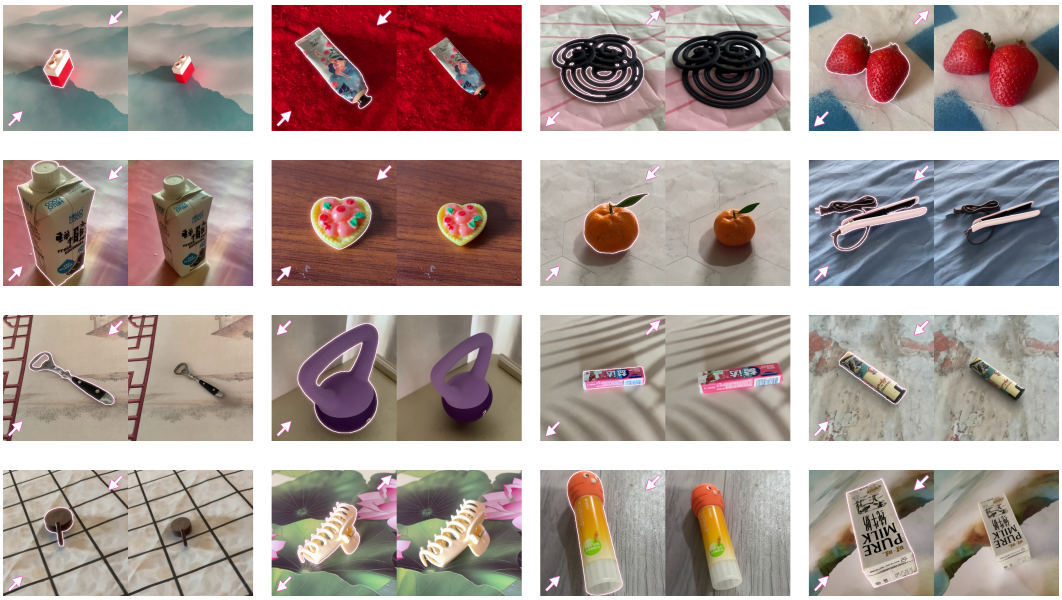
D Supplementary Results

D.1 Object Rotation



D.2 Object Scaling





D.3 OBJECT TRANSLATION

