PHYSCALE: PHYSICALLY GROUNDED OBJECT SIZE CORRECTION IN TEXT-TO-IMAGE GENERATION

Anonymous authors
Paper under double-blind review



Figure 1: **Relative-scale failures from modern image generators and corrections by PhyScale.** Each pair shows the original model output (left) and our corrected result (right), with more plausible physical scale while preserving the scene. Zoom in for best viewing.

ABSTRACT

Generated images can appear photorealistic while depicting objects at implausible relative physical sizes. Correcting these errors requires distinguishing physical size from its depth- and pose-dependent appearance. We introduce PhyScale, a framework that formulates size correction as geometric inference and planning, separating physical-size changes from their image-space realization. Using typical real-world object dimensions from a knowledge base and estimated scene geometry, it computes size, depth, and placement commands for an image editor controlled through spatial rotary positional embeddings. To learn the associated appearance changes, we adapt the editor on only 2,468 paired renders from controlled 3D object transformations. The resulting editor achieves the best target-fidelity and identity-preservation metrics among the compared editors on prescribed edits. Combined with our geometric inference and planning front end, it delivers aggregate GenScale correction performance comparable to a state-of-the-art correction pipeline at approximately one-twentieth of its estimated inference cost.

1 INTRODUCTION

Modern text-to-image generators can synthesize photorealistic scenes (Rombach et al., 2022; Esser et al., 2024), yet render familiar objects at implausible relative physical sizes (Fig. 1). This failure is

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.

systematic rather than anecdotal: GenScale (Li et al., 2026) documents a mean-regression pattern in which physically small objects are rendered too large and large objects too small by state-of-the-art image generators (OpenAI, 2026; Google DeepMind, 2026b; xAI, 2026; Wu et al., 2025; Black Forest Labs, 2025). We aim to correct these disproportions in existing images while preserving object identities and scene context.

Correcting the object sizes in generated images, however, requires distinguishing physical size from apparent size. Under perspective, a small nearby object and a larger distant object can have the same projected extent (Hartley & Zisserman, 2004), while pose further affects it through foreshortening. Size errors must therefore be estimated in the context of scene geometry. Even with perfect estimates, a feasible correction is not guaranteed: enlarging an object in place can cause collisions or move it outside the frame, so the object may instead need to be moved farther away to limit its projected footprint in the image. The physical change and its feasible image-space realization are thus distinct but coupled decisions.

Existing spatial controllers (Zhang et al., 2023; Li et al., 2023) and geometry-aware editors, including ObjectMover (Yu et al., 2025), PhyEdit (Xu et al., 2026), and the concurrent work RoPEMover (Oztas et al., 2026), provide mechanisms for prescribed spatial edits, but rely on user-specified layouts, regions, or transformations rather than inferring the required physical correction. ReScale (Li et al., 2026) addresses this decision stage with a vision-language model (VLM) that diagnoses size errors and plans local resizing while considering depth and scene constraints. Compared with VLM-based diagnosis and planning, explicit geometric computation makes the correction decisions directly inspectable and can be substantially faster and cheaper at inference. This raises two questions: **(1) Can physical-size corrections be computed explicitly from estimated scene geometry? (2) Can an image editor faithfully execute the resulting correction commands?**

To address these questions, we introduce PhyScale, a framework connecting size measurement, planning, and editing through explicit geometric commands (Fig. 2). Given canonical object dimensions and estimated scene geometry, a pose-aware observer estimates size errors by fitting scaled canonical geometry to the observed projections. A geometric planner then allocates corrections across objects and searches over depth and lateral placement under scene constraints and the editor's calibrated operating limits. The resulting decisions are inspectable and require no user-provided layouts, boxes, or resize factors.

To execute the resulting commands, we encode the planned apparent size and destination by modifying spatial rotary positional embeddings (RoPE) in Qwen-Image-Edit (Wu et al., 2025). This control specifies the projection, but the editor must also learn the associated appearance changes. We therefore adapt it with LoRA (Hu et al., 2022) on only 2,468 paired renders from controlled 3D object transformations using Kubric (Greff et al., 2022). Only the image editor is adapted, without real-image editing pairs; the geometric estimation and planning stages require no task-specific training.

This division yields a reusable image editor with the best target-fidelity and identity-preservation metrics among the compared editors on prescribed edits. The complete system achieves aggregate GenScale (Li et al., 2026) correction performance comparable to ReScale at approximately one-twentieth of its estimated inference cost.

In summary, our contributions are three-fold:

- We formulate **relative physical-size correction** as an inference-and-planning problem that separates physical-size changes from their depth-dependent image-space realization and connects measurement, correction allocation, and constrained placement through executable commands.
- We develop a **data-efficient RoPE-controlled image editor** that couples positional control with 3D rendered supervision, learning faithful geometric editing from only 2,468 rendered pairs.
- We build an **efficient, composable correction system** with an inspectable geometric front end that requires no VLM-based correction decisions or task-specific training and can be replaced independently of the image editor.

2 Related Work

Geometry-controlled image generation and editing. Spatial controllers guide synthesis with 2D maps or regions (Zhang et al., 2023; Li et al., 2023; Zhang et al., 2025) or rendered 3D lay-

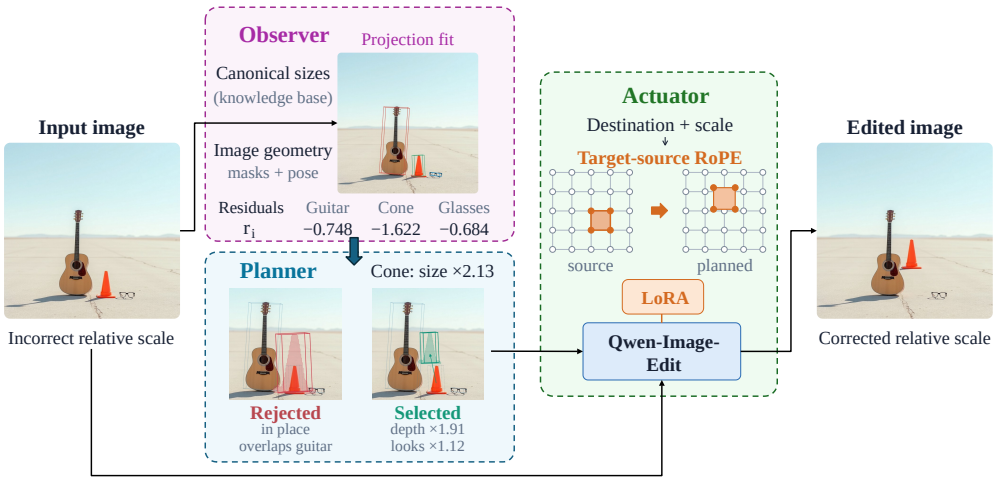

Figure 2: **Overview of PhyScale.** The observer combines instance masks, estimated scene geometry and pose, and knowledge-base dimensions to estimate physical-size residuals. The planner allocates size corrections and searches over depth and placement under scene constraints and the editor's operating limits. The actuator executes the resulting plan by modifying the edited object's source-token RoPE coordinates in an adapted Qwen-Image-Edit model.

outs (Agrawal et al., 2026), while language-mediated systems infer layouts or refine outputs (Lian et al., 2024; Wu et al., 2024). Image editors use lifted features or reconstructed geometry (Pandey et al., 2024; Yenphraphai et al., 2024), video priors (Yu et al., 2025; Wu et al., 2026), visual 3D controls (Wang et al., 2026; Xu et al., 2026), or positional encodings (Bai et al., 2026; Sella et al., 2026). To learn scene effects, prior work uses rendered (Michel et al., 2023; Zhang et al., 2026) or captured counterfactual pairs (Winter et al., 2024). Concurrent RoPEMover (Oztas et al., 2026) combines RoPE warping, depth conditioning, and synthetic-to-real fine-tuning for relocation. Our actuator couples positional control with size–depth correction, using only rendered counterfactuals for task-specific adaptation.

Physical size reasoning and correction. Choosing these edits for size correction requires distinguishing physical dimensions from apparent extent. Object-size knowledge supports commonsense reasoning (Bagherinezhad et al., 2016) and monocular depth estimation (Wu et al., 2018). In generation, ASQL regularizes summed entity attention using LLM-derived size orderings (Jayasekara et al., 2026). For existing images, ReScale (Li et al., 2026) uses a multimodal agent to diagnose size errors and plan contact-preserving resizing while considering depth and scene constraints. PhyScale instead estimates size deviations through pose-aware projection fitting and plans size changes, depth, and placement under scene and actuator constraints. This provides an inspectable geometric alternative to VLM decisions while keeping the actuator interchangeable.

3 Method

Approach overview. PhyScale separates geometric correction decisions from learned image editing through the three-stage pipeline in Fig. 2. We first formulate relative physical-size correction and define size–depth edit commands (Sec. 3.1). The **observer** estimates size errors by fitting knowledge-base object dimensions to image measurements under estimated scene geometry and pose (Sec. 3.2). The **planner** uses these estimates to choose size corrections, depth, and lateral placement under prioritized scene and editor constraints (Sec. 3.3). The **actuator** executes the resulting image-space transformation through spatial rotary positional embeddings (Sec. 3.4). The geometric decisions are explicitly computed and inspectable; only the actuator receives task-specific training.

3.1 Problem Formulation & Preliminaries

Relative physical size. Given a generated image I, target and reference object names T and R, and a knowledge base of typical real-world object dimensions, we seek a corrected image I' with

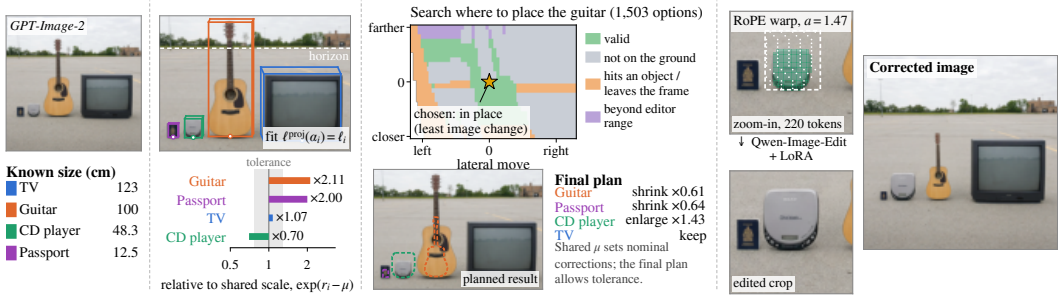

Figure 3: **Running example for the three method stages.** (a) The observer fits each object's knowledge-base cuboid to its image projection and estimates its size residual. (b) The planner allocates size corrections and searches depth–lateral placements under prioritized constraints; the colored map shows candidate placements for the guitar. (c) The actuator converts the selected projection into a RoPE warp and edits an enlarged local crop when the target is small.

plausible relative physical sizes while preserving object identities and scene context. The system binds the names to image instances and infers geometric changes without user-provided destination boxes or resize factors.

Let S_i denote a representative linear physical dimension of object i (e.g., its height), d_i its camera-frame depth, and ℓ_i its projected linear extent in pixels. Under local weak perspective (Hartley & Zisserman, 2004),

$$\ell_i \approx \kappa_i \frac{f S_i}{d_i}, \tag{1}$$

where f is the focal length in pixels and κ_i captures shape- and pose-dependent foreshortening. For two objects T and R in the same image, the focal length cancels:

$$\log \frac{\ell_T}{\ell_R} \approx \log \frac{S_T}{S_R} - \log \frac{d_T}{d_R} + \log \frac{\kappa_T}{\kappa_R}. \tag{2}$$

Projected size ratios therefore confound physical size with depth, shape, and pose. We use these relations to parameterize image edits.

Size–depth edits. For intended size S_i' and depth d_i', we define the size–depth command

$$\mathbf{c}_i = (c_{s,i}, c_{d,i}), \qquad c_{s,i} = \log \frac{S_i'}{S_i}, \qquad c_{d,i} = \log \frac{d_i'}{d_i}. \tag{3}$$

Positive $c_{s,i}$ enlarges the object; positive $c_{d,i}$ moves it farther away. For a fixed camera and an identity- and pose-preserving edit with approximately unchanged foreshortening, Eq. 1 gives the predicted apparent scaling factor

$$a_i = \exp(c_{s,i} - c_{d,i}) \approx \frac{\ell_i'}{\ell_i}. \tag{4}$$

Thus $(c_{s,i}, 0)$ changes size at fixed depth, whereas $(0, c_{d,i})$ changes depth at fixed size. Equal commands $c_{s,i} = c_{d,i}$ give $a_i = 1$, approximately preserving projected extent while allowing changes in position, occlusion, and contact. This freedom allows a physical-size correction without forcing an in-place resize. Appendix A gives the full derivation.

3.2 Pose-Aware Physical Observer

Physical size from inverse projection. The observer estimates size by fitting a uniform scale to a cuboid, rather than reconstructing detailed 3D shape. For instance i, let M_i be its mask, $\mathbf{D}_i = (L_i, W_i, H_i)$ its knowledge-base length, width, and height, and Q_i the eight corners of the corresponding ground-anchored cuboid. Flat and elongated objects use orientation R_i from a frozen pose estimator (You et al., 2026); compact objects use an upright orientation. The pose estimator is

used only for orientation: its predicted metric size and translation are not used as measurements, and its axis-length ratios only determine the correspondence between predicted axes and (L_i, W_i, H_i).

For ground-supported objects, we approximate the contact point by the horizontal center of the mask's bounding box at its lowest occupied row. We use fixed camera intrinsics and camera height, with tilt set by the estimated horizon; the horizon is fit from MoGe-2 depth predictions on segmented ground pixels (Wang et al., 2025). Intersecting the camera ray through the contact point with the ground plane gives position g_i. Appendix D.1 gives the camera parameters, horizon fitting, and validity checks.

Let Π be the camera projector and $\mathrm{extent}(\cdot)$ return the longest side of the axis-aligned 2D bounding box enclosing a mask or projected points. For a trial scale $\alpha > 0$ applied uniformly to the cuboid, we measure

$$\begin{aligned} \ell_i &= \mathrm{extent}(M_i), \\ \ell_i^{\mathrm{proj}}(\alpha) &= \mathrm{extent}\big(\Pi(g_i + \alpha R_i Q_i)\big). \end{aligned} \tag{5}$$

With the camera, orientation, and ground position held fixed during fitting, the only unknown is α. We obtain α_i from

$$\ell_i^{\mathrm{proj}}(\alpha_i) = \ell_i, \tag{6}$$

using bisection over a scale interval where projected extent increases with α and brackets the observed extent. If no such interval exists, the observer abstains.

Let S_i^{kb} be the knowledge-base value of the representative dimension S_i defined in Sec. 3.1 (e.g., H_i when S_i is height). Uniform scaling gives $\widehat{S}_i = \alpha_i S_i^{\mathrm{kb}}$, so the log-size residual is

$$r_i = \log \frac{\widehat{S}_i}{S_i^{\mathrm{kb}}} = \log \alpha_i. \tag{7}$$

Here $r_i = 0$ agrees with the knowledge-base size; positive and negative values indicate larger and smaller estimates. Relative size is correct when objects have the same residual. In Fig. 3(a), the fitted scales show the guitar and passport as relatively large, the CD player as relatively small, and the TV close to the shared scale used by the planner.

3.3 Size and Placement Planner

The observer estimates relative-size errors; the planner decides how to distribute the correction and where edited objects should be placed.

Allocating size corrections. After resizing object i by $c_{s,i}$, its residual becomes $r_i + c_{s,i}$. We parameterize the correction by a shared post-edit residual μ:

$$c_{s,i} = \mu - r_i. \tag{8}$$

Thus the nominal residuals agree, recovering the knowledge-base size ratios. Setting $\mu = r_R$ keeps reference R unchanged; varying μ distributes the correction across objects. A final tolerance refinement may reduce these commands. In Fig. 3(b), the resulting plan keeps the TV unchanged, shrinks the guitar and passport, and enlarges the CD player.

Searching for a valid placement. For each μ, Eq. 8 fixes $c_{s,i}$; we then search each edited object's depth command $c_{d,i}$ and horizontal displacement Δx_i, while vertical placement follows the ground plane. The projector predicts the destination footprint and apparent scale $a_i = \exp(c_{s,i} - c_{d,i})$.

Candidates are compared by prioritized constraints: support and collision first; then framing, overlap/depth ordering, and target-token support; then the actuator's calibrated scaling range. Ties are broken by image change, motion, and the number of edited objects, so lower-priority improvements cannot override higher-priority violations.

We search discrete μ values on a coarse depth–lateral grid and refine the selected solution on a finer grid; multi-object placement proceeds greedily from larger to smaller source areas. Figure 3(b) visualizes the guitar search: invalid candidates violate support, collision/frame, or editor-range constraints, while the selected in-place candidate yields the least image change among valid choices. A final refinement removes unnecessary resizing within the relative-size tolerance. Appendix D.2 gives the grids, tolerances, tie-breaking, and fallback rules.

3.4 PHYSICAL ROPE ACTUATOR

Encoding the planned edit. We encode the target's planned projection in Qwen-Image-Edit (Wu et al., 2025) through spatial rotary positional embeddings (RoPE), a control mechanism also explored in concurrent RoPEMover (Oztas et al., 2026). For one target, we omit the object index and use apparent scale $a = \exp(c_s - c_d)$ and the planned destination center.

Let $\mathbf{m}_{\text{src}}$ and $\mathbf{m}_{\text{dst}}$ be the source and destination centers on the latent grid, and let $\mathcal{J}_T$ contain source-image tokens selected by target mask M_T. We replace each source-token coordinate $\mathbf{p}_j$ by

$$\mathbf{p}'_j = \begin{cases} \mathbf{m}_{\text{dst}} + a(\mathbf{p}_j - \mathbf{m}_{\text{src}}), & j \in \mathcal{J}_T, \\ \mathbf{p}_j, & j \notin \mathcal{J}_T. \end{cases} \tag{9}$$

We recompute spatial RoPE only in the source-image stream; other coordinates and token features remain unchanged. Figure 3(c) shows the corresponding $a = 1.47$ warp and a zoomed crop providing 220 target tokens. The resulting positional field ρ is the sole geometric condition: depth affects both apparent scale and destination center, while commands producing the same field are indistinguishable to the actuator. Because the interface depends only on the projected transform, it can also accept requests from another front end.

Supervision from 3D scene edits. RoPE specifies the requested projection but not the associated changes in visibility, contact, or shadows. We therefore scale and reposition targets in 3D scenes and re-render them with Kubric/Blender (Greff et al., 2022), keeping the camera, reference object, and environment fixed. Our 2,468 validated pairs from 624 scanned assets cover pure and joint size–depth edits, approximately iso-apparent edits, and identity pairs (Fig. 4); no real-image editing pairs are used.

During training, the centers in Eq. 9 come from rendered masks, and the measured destination-to-source extent ratio a^{render} replaces analytic a. Partially occluded destinations use amodal masks. Appendix C gives data construction details.

Parameter-efficient adaptation. We freeze the pretrained editor and train only rank-16 LoRA parameters θ (Hu et al., 2022). With source latent z_{src}, noisy destination latent x_t, and flow-matching target v_t, we minimize the base objective (Lipman et al., 2023):

$$\mathcal{L}_{\text{act}} = \mathbb{E}\left[\|v_\theta(x_t, t; z_{\text{src}}, \rho) - v_t\|_2^2\right], \tag{10}$$

with no geometric, perceptual, or region-specific auxiliary loss. Identity pairs use unchanged images and positional fields, providing preservation supervision. Appendix B gives architecture and optimization details.

Inference for small targets. At inference, ρ uses the planner's destination and analytic scale a. Targets with insufficient source-token support are edited in an enlarged crop containing the source, destination, and surrounding scene, then blended back into the image. The system abstains if no crop satisfies the geometry and token-support requirements. Appendix B gives implementation details.



Figure 4: **Geometrically validated training pairs from 3D scenes.** Each pair shows a source image (left) and the destination re-rendered after the prescribed object transformation (right), with the camera, reference object, and environment fixed.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

We evaluate our complete correction method and ablate it by studying its three stages (diagnosis, placement, and editing) separately. GenScale tests correction across generators; scenes with known geometry test size estimation and placement; real before-and-after photographs test prescribed editing. Matched training controls assess the contributions of positional commands and 3D supervision. Human judgments complement automatic metrics for placement, editing quality, and scene effects, while the efficiency study compares the computational cost of geometric and VLM-based correction (Appendix E).

We adapt Qwen-Image-Edit-2511 (Wu et al., 2025; Qwen Team, 2025a) with rank-16 LoRA on 2,468 rendered pairs for ten epochs at 768×768. Inference uses 40 steps, guidance 1, and RoPE control for the first 20 steps (Appendix B).

4.2 END-TO-END RELATIVE-SIZE CORRECTION

On GenScale (Li et al., 2026), PhyScale reduces relative-size error across seven generators and achieves aggregate performance close to ReScale (Table 1; Fig. 5). Re-scoring both methods together on 700 shared sources also detects no difference; Appendix E.2 reports scoring and weighting checks.

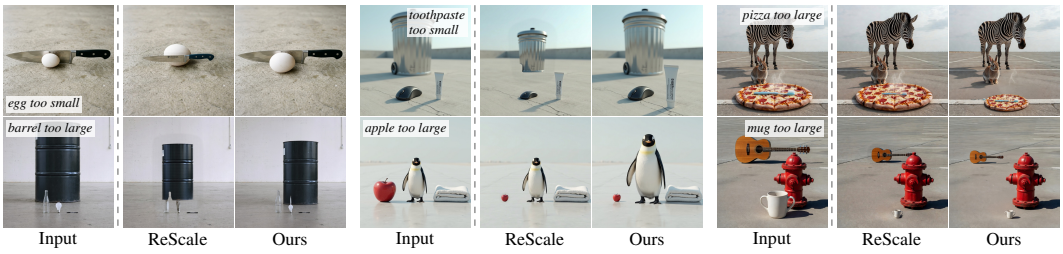

Figure 5: **End-to-end size correction.** Input / ReScale / PhyScale triplets; annotations mark size errors.

Table 1: **GenScale correction.** Mean relation error $|s-3|$ (↓), where $s \in \{1,\ldots,5\}$ is the judge's relative-size score; 2,800 images, 8,001 relations. Unjudgeable relations receive error 2.

	Nano Banana 2	GPT-Image-2	Z-Image-Turbo	Grok Imagine	Qwen-Image 2512	FLUX.2	SD3.5 Large	All seven
Source	0.620	0.707	0.898	0.787	0.992	1.188	1.174	0.910
ReScale	**0.459**	0.557	0.682	**0.492**	**0.581**	0.824	0.948	0.649
Ours	0.472	**0.491**	**0.673**	0.514	0.664	**0.816**	**0.851**	**0.640**

To separate decision quality from editing, we pair our geometric and ReScale's Gemini front ends with both editors on a shared 196-image GenScale subset, randomly sampled from all seven generators to limit evaluation cost (Table 4). With our front end fixed, replacing ReScale's editor with ours improves correction. Gemini gives lower error point estimates at higher cost, but paired tests do not establish a front-end accuracy advantage (Appendix E.3).

Inference efficiency. On 30 shared GenScale inputs, PhyScale has lower summed processing time and estimated cost than ReScale (Table 2). Savings mainly come from replacing VLM diagnosis and planning with geometric computation. Times sum observation, planning, editing, and restoration; Appendix E.9 gives the breakdown and pricing assumptions.

Table 2: **Pipeline efficiency.** Mean summed stage time and estimated cost per image.

Method	Time (s)↓	Cost ($)↓
ReScale	98.0	0.696
PhyScale	**52.8**	**0.037**

4.3 OBSERVATION AND PLANNING

We construct 116 three-object tabletop scenes in Blender with ground-truth geometry, covering cases where in-place size correction causes truncation, occlusion, or undersized targets. Gemini-3.1-Flash-Lite-Image (Google DeepMind, 2026a) generates matched photorealistic counterparts, of which 109 pass consistency checks. Appendix E.5 details construction, filtering, and examples.

Table 3: **Observer accuracy (%).** Given true dimensions, identify the wrong-size object and predict its correction within 20%. Photoreal.: generated images.

Method	Blender	Photoreal.
Gemini 3.1 Pro	44.8	38.5
PhyScale	**56.9**	**44.0**

Table 4: **Component swap.** Size error (↓) and estimated cost per image; 196 sources.

Front end	Editor	Error	Cost ($)
Gemini 3.1 Pro	ReScale	0.701	0.731
PhyScale	ReScale	0.826	0.314
Gemini 3.1 Pro	PhyScale	**0.570**	0.499
PhyScale	PhyScale	0.654	**0.057**

Size diagnosis. Our observer has higher point-estimate accuracy than Gemini in both domains (Table 3). Diagnostics identify horizon estimation and the cuboid approximation as major error sources; ground-truth orientation yields no detected improvement. Appendix E.4 reports these diagnostics and compares a reconstruction-based observer.

Feasible placement. On this set, our planner and Gemini 3.1 Pro receive identical observations and a prescribed resize, choosing only depth and lateral position. Both use Blender to execute rendered-scene plans and our frozen editor for photorealistic scenes. Automatic checks test framing, target visibility and size, and neighbor preservation; five blinded raters compare placements. Our rendered-scene advantage is concentrated in occlusion cases. With the corrected horizon, photorealistic pass rates show no detected difference (the frozen-horizon run favors Gemini; Appendix E.5.3), while raters prefer our placements in both domains (Table 5; Fig. 6; Appendix E.5).

Table 5: **Planner stress tests.** Automatic pass rate (%, ↑); five-rater preference ($+1/0/-1$: ours/tie/Gemini). Support: full tabletop support.

Scenes	Ours	Gemini	Preference
Blender	**87.9**	80.2	+0.122
+ support	**75.9**	69.0	—
Photoreal.	46.8	**56.0**	+0.172

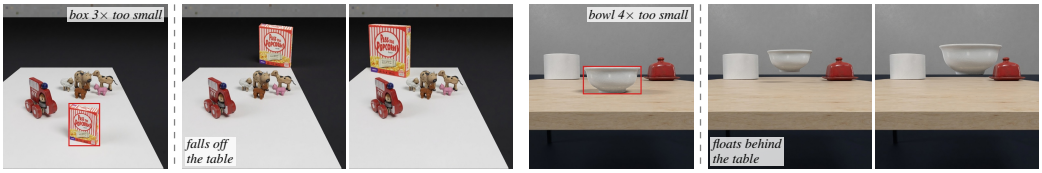

Figure 6: **Placement after resizing.** Gemini produces off-table or floating destinations in these examples.

4.4 Controlled Editing and Actuator Adaptation

ObjectMover (Yu et al., 2025) provides real before-and-after photographs for prescribed object movement and resizing. All editors receive the same 188 requests; ReScale uses only its editor. Target DINO (Caron et al., 2021) measures feature cosine similarity between edited and ground-truth destination object crops; DreamSim (Fu et al., 2023) measures their perceptual distance. Identity DINO uses the same similarity against source-object crops as an identity-preservation proxy. Our editor leads these target-fidelity and identity-preservation metrics (Table 6; Fig. 7).

Table 6: **Image editing on ObjectMover.** Text-box and RoPE controls start from the same Qwen-Image-Edit-2511 checkpoint and use identical training pairs and numbers of updates. Retrained controls are unranked.

	Qwen-Image-Edit-2511 (Qwen Team, 2025a)			PhyEdit (Xu et al., 2026)	ReScale (Li et al., 2026)	ObjectMover (Yu et al., 2025)	PhyScale	Retrained controls	
Metric	Crop + text	RoPE only	Structured					Text boxes	RoPE
Target DINO ↑	0.410	0.634	0.696	0.791	0.814	0.867	**0.899**	0.439	0.895
DreamSim ↓	0.622	0.399	0.326	0.216	0.177	0.140	**0.105**	0.620	0.108
Identity DINO ↑	0.379	0.561	0.653	0.738	0.786	0.813	**0.856**	0.365	0.856

For real-image edits, raters compare outputs for the same request and assess identity, naturalness, and overall preference. Their judgments favor our editor over PhyEdit (Xu et al., 2026) and Qwen but detect no significant difference from ObjectMover (Table 7; Appendix E.8).

Positional control. To test whether positional commands help beyond fine-tuning, we train two models with matched backbones, data, budgets, and aligned visual inputs. One uses RoPE; the other receives source and destination boxes in text. RoPE performs better on both ObjectMover and held-out renders (Table 6; Appendix E.6), so real-image transfer alone cannot explain the text-box gap. The aligned RoPE rerun shows no detected difference from our main editor.

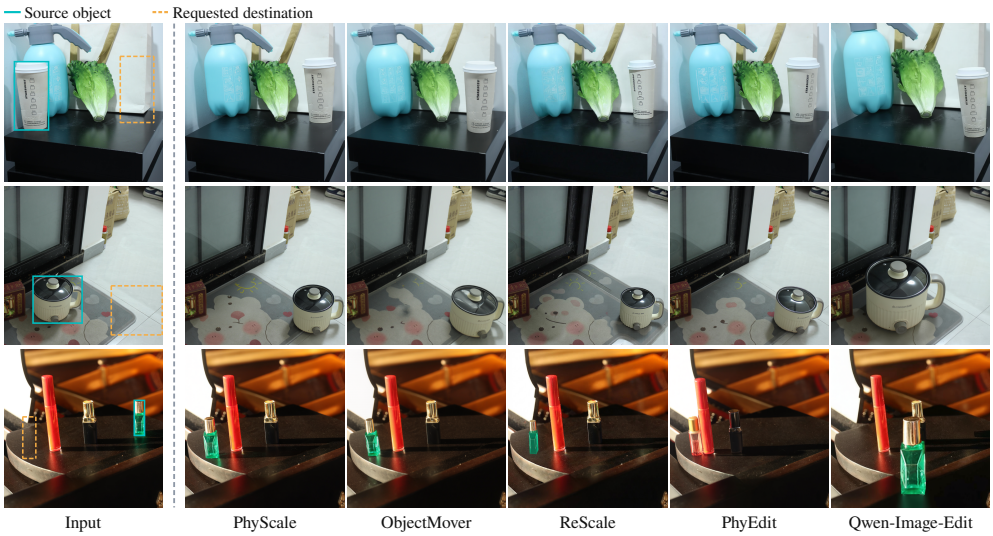

Figure 7: **Prescribed real-image edits.** Shared source/destination boxes; ReScale uses only its editor.

Table 7: **User study on editing quality.** Descriptive identity and naturalness scores (0–4) and completed requests (%). ReScale† uses a separate aggregate.

	Qwen	PhyEdit	ReScale†	ObjectMover	PhyScale
Identity ↑	3.233	2.200	3.600	<u>3.767</u>	**3.861**
Naturalness ↑	1.627	2.417	3.100	<u>3.633</u>	**3.728**
Completed (%) ↑	2.0	31.2	<u>78.0</u>	74.0	**89.9**

Supervision and scene effects. We compare full-scene 3D re-rendering with source-object warping and compositing (2.5D), using the same editor, source scenes, and training budget. On 150 held-out renders, target DINO shows no detected difference, while silhouette IoU favors 2.5D. Three blinded raters assess physical plausibility (contact, shadows, and occlusion) on a fixed 90-request subset, preferring 3D despite lower completion (Appendix E.6.5). Figure 8 shows real-image examples.

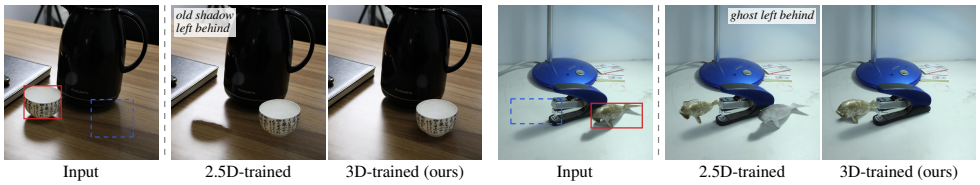

Figure 8: **Scene effects.** Real-image edits show old shadows or object remnants after 2.5D training.

5 Conclusion

We presented PhyScale, a framework that separates physical-size inference and planning from learned image editing. Its observer estimates size errors from knowledge-base dimensions and scene geometry, its planner selects size, depth, and placement under prioritized constraints, and its RoPE-controlled editor executes the resulting projection using limited training pairs. Across GenScale and controlled editing, PhyScale achieves correction performance comparable to ReScale while providing stronger prescribed-edit fidelity and substantially lower estimated inference cost. These results show that explicit geometric computation and data-efficient learned execution provide a practical alternative to VLM-based correction decisions.

Limitations and Future Work. PhyScale relies on several pretrained vision modules, so errors in segmentation, scene geometry, or pose can propagate into size estimation and planning. The observer further approximates objects with category-level dimensions and cuboids; our diagnostics identify horizon estimation and this shape approximation as major remaining error sources. More fundamentally, PhyScale corrects images after generation and assumes that relevant objects are visible and locally editable. Future work should incorporate metric size priors and scale-aware objectives directly into generative-model training, so plausible relative size is learned during generation rather than repaired afterward.

AI USE STATEMENT

The core research idea originated with the authors. We used generative AI tools to discuss and refine this idea, assist with implementing the method and experimental code, polish the manuscript, and prepare figures. Separately from this research assistance, generative models were used for image generation and editing, construction of photorealistic evaluation inputs, and offline model-based assessment, as described in Section 4 and Appendix E. Implementation checks and experimental validation procedures are documented in Appendices B–E. The authors reviewed the AI-assisted contributions incorporated into this work and take full responsibility for the final methods, results, claims, and artifacts.

REPRODUCIBILITY STATEMENT

Appendix A provides the derivation of the size–depth parameterization. Appendices B–D document the actuator architecture, training and inference settings, construction and validation of the rendered training pairs, and observer and planner implementations, including planner pseudocode. Appendix E describes the evaluation datasets and protocols, baseline configurations, scoring and statistical procedures, ablations, human studies, and runtime measurements and cost-estimation assumptions. Appendix F provides additional qualitative comparisons and the editing-study interface.

REFERENCES

Vaibhav Agrawal, Rishubh Parihar, Pradhaan Bhat, Ravi Kiran Sarvadevabhatla, and R. Venkatesh Babu. SeeThrough3D: Occlusion aware 3D control in text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Hessam Bagherinezhad, Hannaneh Hajishirzi, Yejin Choi, and Ali Farhadi. Are elephants bigger than butterflies? Reasoning about sizes of objects. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pp. 3449–3456, 2016.

Yunpeng Bai, Haoxiang Li, and Qixing Huang. Positional encoding field. In *International Conference on Learning Representations*, 2026.

Black Forest Labs. FLUX.2: Frontier visual intelligence. Official release announcement, November 2025. URL https://bfl.ai/blog/flux-2. Accessed September 25, 2026.

Berk Calli, Aaron Walsman, Arjun Singh, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M. Dollar. Benchmarking in manipulation research: Using the Yale–CMU–Berkeley object and model set. *IEEE Robotics & Automation Magazine*, 22(3):36–52, 2015. doi: 10.1109/MRA.2015.2448951.

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*, 2021.

Xingyu Chen, Fu-Jen Chu, Pierre Gleize, Kevin J. Liang, Alexander Sax, Hao Tang, Weiyao Wang, Michelle Guo, Thibaut Hardin, Xiang Li, Aohan Lin, Jia-Wei Liu, Ziqi Ma, Anushka Sagar, Bowen Song, Xiaodong Wang, Jianing Yang, Bowen Zhang, Piotr Dollár, Georgia Gkioxari, Matt Feiszli, and Jitendra Malik. SAM 3D: 3Dfy anything in images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7220–7232, 2026.

Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B. McHugh, and Vincent Vanhoucke. Google Scanned Objects: A high-quality dataset of 3D scanned household items. In *2022 International Conference on Robotics and Automation (ICRA)*, 2022. doi: 10.1109/ICRA46639.2022.9811809.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12606–12633, 2024.

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. Gemini 3.1 Flash-Lite Image. Model card, June 2026a. URL https://deepmind.google/models/model-cards/gemini-3-1-flash-lite-image/. Accessed September 25, 2026.

Google DeepMind. Gemini 3.1 Flash Image. Model card, February 2026b. URL https://deepmind.google/models/model-cards/gemini-3-1-flash-image/. Accessed September 25, 2026.

Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J. Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, et al. Kubric: A scalable dataset generator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3749–3761, 2022.

Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.

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.

Hirunima Jayasekara, Chuong Huynh, Yixuan Ren, Christabel Acquaye, and Abhinav Shrivastava. All-in-one conditioning for text-to-image synthesis. *arXiv preprint arXiv:2602.09165*, 2026.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4015–4026, 2023.

Lingyu Kong, Ruicheng Li, Ruicheng Wang, Sicheng Xu, Chengtang Yao, Jianfeng Xiang, and Jiaolong Yang. MoGe-3: Fine-detail monocular geometry estimation with self-guided sparse volumetric refinement. *arXiv preprint arXiv:2607.17967*, 2026.

Lingxiao Li, Max Whitton, Ledell Wu, and Boqing Gong. GenScale: A benchmark for relative object scale in image generation and editing. *arXiv preprint arXiv:2609.00525*, 2026.

Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. GLIGEN: Open-set grounded text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Long Lian, Boyi Li, Adam Yala, and Trevor Darrell. LLM-grounded diffusion: Enhancing prompt understanding of text-to-image diffusion models with large language models. *Transactions on Machine Learning Research*, 2024.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

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 *European Conference on Computer Vision*, pp. 38–55, 2024.

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*, 2023.

OpenAI. GPT-Image-2. OpenAI API model documentation, 2026. URL `https://developers.openai.com/api/docs/models/gpt-image-2`. Accessed September 25, 2026.

Ipek Oztas, Duygu Ceylan, Aybars Bugra Aksoy, and Aysegul Dundar. RoPEMover: Depth-aware object relocation via positional embeddings. *arXiv preprint arXiv:2606.27332*, 2026.

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*, 2024.

Qwen Team. Qwen-Image-Edit-2511. Hugging Face model card, 2025a. URL `https://huggingface.co/Qwen/Qwen-Image-Edit-2511`. Accessed September 25, 2026.

Qwen Team. Qwen-Image-2512. Hugging Face model card, 2025b. URL `https://huggingface.co/Qwen/Qwen-Image-2512`. Accessed September 25, 2026.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

Etai Sella, Yoav Baron, Hadar Averbuch-Elor, Daniel Cohen-Or, and Or Patashnik. LooseRoPE: Content-aware attention manipulation for semantic harmonization. In *ACM SIGGRAPH 2026 Conference Papers*. Association for Computing Machinery, 2026. doi: 10.1145/3799902.3811142.

Stability AI. Introducing Stable Diffusion 3.5. Official release announcement, October 2024. URL https://stability.ai/news-updates/introducing-stable-diffusion-3-5. Accessed September 25, 2026.

Feifei Wang, Shiyuan Yang, Xiaoyu Li, and Jing Liao. BoxCtrl: 3D-aware visual prompting for geometric image editing. In *ACM SIGGRAPH 2026 Conference Papers*. Association for Computing Machinery, 2026. doi: 10.1145/3799902.3811169.

Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. MoGe-2: Accurate monocular geometry with metric scale and sharp details. In *Advances in Neural Information Processing Systems*, volume 38, pp. 40377–40408, 2025.

Daniel Winter, Matan Cohen, Shlomi Fruchter, Yael Pritch, Alex Rav-Acha, and Yedid Hoshen. ObjectDrop: Bootstrapping counterfactuals for photorealistic object removal and insertion. In *Computer Vision – ECCV 2024*, volume 15135 of *Lecture Notes in Computer Science*, pp. 112–129. Springer, 2024. doi: 10.1007/978-3-031-72980-5_7.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-Image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

Jay Zhangjie Wu, Xuanchi Ren, Tianchang Shen, Tianshi Cao, Kai He, Yifan Lu, Ruiyuan Gao, Enze Xie, Shiyi Lan, Jose M. Alvarez, Jun Gao, Sanja Fidler, Zian Wang, and Huan Ling. ChronoEdit: Towards temporal reasoning for in-context image editing and world simulation. In *International Conference on Learning Representations*, 2026.

Tsung-Han Wu, Long Lian, Joseph E. Gonzalez, Boyi Li, and Trevor Darrell. Self-correcting LLM-controlled diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6327–6336, 2024.

Yiran Wu, Sihao Ying, and Lianmin Zheng. Size-to-depth: A new perspective for single image depth estimation. *arXiv preprint arXiv:1801.04461*, 2018.

xAI. Image generation. Grok API documentation, 2026. URL https://docs.x.ai/developers/model-capabilities/images/generation. Accessed September 26, 2026.

Ruihang Xu, Dewei Zhou, Xiaolong Shen, Fan Ma, and Yi Yang. PhyEdit: Towards real-world object manipulation via physically-grounded image editing. In *Proceedings of the 34th ACM International Conference on Multimedia*, 2026.

Jiraphon Yenphraphai, Xichen Pan, Sainan Liu, Daniele Panozzo, and Saining Xie. Image sculpting: Precise object editing with 3D geometry control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Yang You, Yi Du, Cole Harrison, and Leonidas Guibas. UniPose9D: Universal category-agnostic object pose estimation. *arXiv preprint arXiv:2607.09985*, 2026.

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*, 2025.

Z-Image Team, Huanqia Cai, Sihan Cao, Ruoyi Du, Peng Gao, Aiming Hao, Steven Hoi, Zhaohui Hou, Shijie Huang, Dengyang Jiang, Yuming Jiang, Xin Jin, Liangchen Li, Zhen Li, Zhong-Yu Li, David Liu, Dongyang Liu, Qilong Wu, Feng Yu, Zechao Zhan, Chi Zhang, Shifeng Zhang, Ruikai Zhou, and Shilin Zhou. Z-Image: An efficient image generation foundation model with single-stream diffusion transformer. *arXiv preprint arXiv:2511.22699*, 2025.

Hong Zhang, Zhongjie Duan, Xingjun Wang, Yingda Chen, and Yu Zhang. EliGen: Entity-level controlled image generation with regional attention. In *Proceedings of the 7th ACM International Conference on Multimedia in Asia*, pp. 1–7, 2025.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3836–3847, 2023.

Shuo Zhang, Wenzhuo Wu, Huayu Zhang, Jiarong Cheng, Xianghao Zang, Chao Ban, Hao Sun, Zhongjiang He, Tianwei Cao, Kongming Liang, and Zhanyu Ma. Geometric image editing via effects-sensitive in-context inpainting with diffusion transformers. In *International Conference on Learning Representations*, 2026.

Appendix

This appendix provides derivations, implementation and data details, and full experimental protocols and uncertainty estimates. Sections A–D supplement the method; Section E supplements the experiments. Section F provides additional visual comparisons and the user-study interface.

Contents

A Derivation of the Size–Depth Parameterization 16
B Actuator Implementation Details 16
B.1 Backbone and Parameter-Efficient Adaptation 17
B.2 RoPE Command Injection 17
B.3 Training Objective and Optimization 18
B.4 Inference 19
B.5 Object-Centric Resampling 19
C Training Pairs from 3D Scene Edits 19
C.1 Motivation and Rendering Setup 19
C.2 Scene and Command Distribution 19
C.3 Constraint-Driven Row Construction 20
C.4 Token-Support Constraint 20
C.5 Rendered Geometry and Materialization 21
C.6 Physical Certification 21
D Observer and Planner Details 22
D.1 Physical Observer 22
D.2 Size and Placement Planner Details 24
D.2.6 Planner Pseudocode 25
E Experimental Protocols and Additional Results 27
E.1 Implementation and Statistical Setup 27
E.2 GenScale Scoring and Sensitivity 27
E.3 Front-End–Actuator Swap 28
E.4 Observer Evaluation 29
E.4.1 Reconstruction-Based Alternative 29
E.4.2 Matched Horizon and Orientation Diagnostic 30
E.5 Planner Stress Tests 30
E.5.1 Construction and Automatic Evaluation 30
E.5.3 Photorealistic Transfer 32
E.5.5 Complete Stress-Test Galleries 34
E.6 Controlled Editing and Training Ablations 37
E.6.4 Textual versus Positional Commands 38
E.6.5 Held-Out 3D versus 2.5D Supervision Tests 39
E.6.6 Training Dynamics 41
E.6.7 Command and Geometric Diagnostics 43
E.7 Additional Component Contrasts 44
E.8 User Study on Editing Quality 45
E.9 Pipeline Time and Cost 46
E.10 Earlier Stress Tests and Scoring Diagnostics 46
F Additional Visualizations and User-Study Interface 48
F.1 GenScale Relative-Size Correction 48
F.2 Prescribed Object Movement and Resizing 49
F.3 Photorealistic Planner Comparisons 50
F.4 3D versus 2.5D Supervision 51
F.5 User-Study Interface for Editing Quality 52

A Derivation of the Size–Depth Parameterization

This section derives the command convention in Sec. 3.1. Physical size is denoted by S, camera-frame depth by d, projected linear extent by ℓ, and the apparent scaling factor by a. VAE latents retain the symbol z. All geometric equations below use the convention in which positive c_d moves an object farther from the camera.

Perspective projection. When an object's depth variation is small relative to its camera distance, the local weak-perspective approximation (Hartley & Zisserman, 2004) gives

$$\Delta x \simeq \frac{f}{d_i}\Delta X, \qquad \Delta y \simeq \frac{f}{d_i}\Delta Y, \tag{11}$$

and hence

$$\ell_i \approx \kappa_i \frac{f S_i}{d_i}, \tag{12}$$

where κ_i captures the shape, pose, viewpoint, and chosen extent measurement. Taking logarithms yields

$$\log \ell_i \approx \log S_i - \log d_i + \log f + \log \kappa_i. \tag{13}$$

For two objects viewed by the same camera, the focal length cancels:

$$\log \frac{\ell_T}{\ell_R} \approx \log \frac{S_T}{S_R} - \log \frac{d_T}{d_R} + \log \frac{\kappa_T}{\kappa_R}. \tag{14}$$

This relation exposes the ambiguity: the same apparent ratio can arise from different size, depth, and pose configurations. It does not itself estimate physical size. The observer instead fits knowledge-base geometry under its estimated camera and orientation (Sec. 3.2).

Physical commands. For an edited object, with the index omitted, define

$$\mathbf{c} = (c_s, c_d), \qquad c_s = \log \frac{S'}{S}, \qquad c_d = \log \frac{d'}{d}. \tag{15}$$

The physical update is therefore

$$S' = S\exp(c_s), \qquad d' = d\exp(c_d). \tag{16}$$

Subtracting Eq. (13) between destination and source gives

$$\log \frac{\ell'}{\ell} \approx c_s - c_d + \log \frac{\kappa'}{\kappa}. \tag{17}$$

Under approximately unchanged pose and foreshortening, $\kappa' \approx \kappa$, this reduces to

$$\log \frac{\ell'}{\ell} \approx c_s - c_d, \qquad a \triangleq \exp(c_s - c_d) \approx \frac{\ell'}{\ell}. \tag{18}$$

Thus $(c_s, 0)$ changes size at fixed depth, while $(0, c_d)$ changes depth at fixed size. Equal size and depth commands, $c_d = c_s$, preserve the predicted projected extent. For $c_s \neq 0$, the compensation ratio is

$$\beta \triangleq \frac{c_d}{c_s}, \tag{19}$$

with $\beta = 0$ for a pure size change and $\beta = 1$ for an iso-apparent edit. Preserving projected extent need not preserve position, visibility, contact, or the full image. During training, the actuator uses scale and destination measurements from rendered masks rather than assuming this local approximation is exact.

B Actuator Implementation Details

Throughout the appendix, V7 denotes the final actuator and its 2,468-pair training set, and C1 denotes the planner deployed on GenScale.

B.1 Backbone and Parameter-Efficient Adaptation

We instantiate the actuator with Qwen-Image-Edit-2511 (Wu et al., 2025). The pretrained transformer, text encoder, and VAE are frozen throughout actuator training. The source image is provided through the edit-image stream with its conditioning timestep set to zero (`zero_cond_t=True`).

We adapt the transformer with LoRA (Hu et al., 2022) of rank $r_{\mathrm{LoRA}} = 16$ and scaling $\alpha_{\mathrm{LoRA}} = 16$. For a pretrained linear layer W, the adapted layer is

$$Wx + \frac{\alpha_{\mathrm{LoRA}}}{r_{\mathrm{LoRA}}} BAx, \tag{20}$$

where A is initialized from $\mathcal{N}(0, 0.02^2)$ and B is initialized to zero. Consequently, the initial actuator is exactly equivalent to the pretrained editor before optimization.

LoRA modules are inserted into 12 linear projections in each transformer block:

$$\begin{aligned} &\{\texttt{to_q}, \texttt{to_k}, \texttt{to_v}, \texttt{to_out.0}\}, \\ &\{\texttt{add_q_proj}, \texttt{add_k_proj}, \texttt{add_v_proj}, \texttt{to_add_out}\}, \\ &\{\texttt{img_mlp.net.2}, \texttt{txt_mlp.net.2}, \texttt{img_mod.1}, \texttt{txt_mod.1}\}. \end{aligned} \tag{21}$$

This yields 117.97M trainable parameters while leaving the 20B-parameter backbone frozen. Backbone weights are stored in bfloat16; LoRA parameters and LoRA computation use FP32.

B.2 RoPE Command Injection

Qwen-Image represents image position using three-dimensional rotary positional embeddings. The RoPE head dimensions are divided across the frame and spatial axes (t, h, w) as

$$(n_t, n_h, n_w) = (16, 56, 56), \tag{22}$$

with RoPE base $b_{\mathrm{RoPE}} = 10{,}000$. This base is distinct from the learned LoRA parameters θ in Sec. 3.4. For coordinate value v and an axis of dimension n, we construct the corresponding rotary frequencies in the standard manner,

$$\omega_k(v) = v\, b_{\mathrm{RoPE}}^{-2k/n}, \qquad k = 0, \ldots, n/2 - 1. \tag{23}$$

Spatial grid coordinates are centered before conversion to frequencies. The frame coordinate distinguishes the two image streams: the noisy output stream uses frame index 0, while the source edit stream uses frame index 1. We modify only the spatial (h, w) coordinates of target tokens and keep the frame coordinate unchanged.

Target-token support. Let M_T be the source target mask and let the image resolution be R. Qwen-Image uses a spatial stride of $p = 16$, yielding a $G \times G$ image-token grid with $G = R/16$. We average-pool M_T over each 16×16 patch and select

$$\mathcal{J}_T = \left\{ j : \mathrm{AvgPool}_{16\times 16}(M_T)_j > 0.25 \right\}. \tag{24}$$

The same selection rule is used during training and inference. The effective number of command-carrying tokens is therefore

$$N_T = |\mathcal{J}_T| \approx \frac{|M_T|}{16^2}. \tag{25}$$

Spatial warp. For each selected token, let $\mathbf{p}_j = (h_j, w_j)$ denote its source-grid coordinate, $\mathbf{m}_{\mathrm{src}}$ the source target center, and $\mathbf{m}_{\mathrm{dst}}$ the destination center, all on the latent grid. At inference the planner supplies the destination; during training both centers come from rendered masks. From Sec. 3.1, the required apparent linear scale is

$$a = \exp(c_s - c_d). \tag{26}$$

We assign the target token the transformed coordinate

$$\mathbf{p}'_j = \mathbf{m}_{\mathrm{dst}} + a(\mathbf{p}_j - \mathbf{m}_{\mathrm{src}}), \qquad j \in \mathcal{J}_T, \tag{27}$$

and recompute its RoPE frequencies after the implementation's valid-grid clamp. Equation (27) is the affine field in Eq. 9 before this boundary safeguard. All non-target source tokens and all noisy-output

tokens retain their original coordinates. During training, a in this map is replaced by the measured rendered extent ratio a^{render}, as specified in Sec. 3.4; at inference it remains $\exp(c_s - c_d)$.

Because the source image occupies the second image stream, the modified frequencies are written only to the corresponding source-stream token positions. For a resolution R,

$$N_{\text{img}} = (R/16)^2, \tag{28}$$

and the source-stream target token j is located at offset $N_{\text{img}} + j$. We explicitly verify that the RoPE map remains inside the source stream and that the positional-embedding tensor contains exactly the expected two image streams.

The resulting conditioning retains the target's visual content but associates its tokens with the destination geometry requested by the planner. The learned editor therefore resolves the disagreement between source appearance and modified positional structure through generative re-rendering rather than pixel-space transformation.

B.3 Training Objective and Optimization

For each certified pair, let I_{src} and I_{dst} denote the source and rendered destination images, and let z_{src} and z_{dst} be their VAE latents. We sample a training timestep uniformly from the base model's 1,000 diffusion timesteps, construct the noisy destination latent x_t, and use the same flow-matching target v_t as the pretrained editor. The only task-specific conditioning is the source image together with the warped positional field:

$$\mathcal{L}_{\text{act}} = \mathbb{E}_{I_{\text{src}}, I_{\text{dst}}, t, \epsilon}\left[\left\| v_\theta\left(x_t, t; z_{\text{src}}, \rho\right) - v_t \right\|_2^2\right]. \tag{29}$$

We use no segmentation, geometric-regression, perceptual, ranking, or region-weighted auxiliary loss.

The field ρ depends on the source mask, source and destination centers, and apparent scale, not on the two physical commands alone. As in Sec. 3.4, identical positional fields cannot distinguish different underlying physical edits. Identity examples use $I_{\text{dst}} = I_{\text{src}}$ exactly, $\mathbf{c} = (0, 0)$, and no lateral displacement. For these examples no RoPE transformation is applied. The 364 identity rows (14.75% of the V7 data) provide direct preservation supervision for the target, reference, and background.

We optimize the LoRA parameters with AdamW (Loshchilov & Hutter, 2019) using

$$\text{lr} = 10^{-4}, \qquad (\beta_1, \beta_2) = (0.9, 0.95), \qquad \text{weight decay} = 0,$$

and clip the global gradient norm to 1.0. The final model is trained at 768×768 resolution. Each of the 2,468 certified training rows is seen exactly ten times, for 24,680 sample exposures and 4,936 optimizer updates at global batch five. Within each epoch, rows are deterministically shuffled using a hash of the row identifier, epoch, and global seed, so every training row has identical exposure.

Our implementation uses functional DiffSynth forward calls rather than the standard `DDP.forward` path. We therefore explicitly average LoRA gradients across workers after backpropagation. Synchronization is checked both after the first update and at the final checkpoint by verifying cross-worker gradient and parameter agreement.

Checkpoints and conditioning prompts. The final checkpoint is taken after 24,680 sample exposures (4,936 optimizer updates). Training uses global batch five, deterministic row shuffling with seed zero, a constant learning rate without warmup, and the prompt "Move the {target} to a different position on the floor, adjusting its apparent size for perspective. Keep the {reference}, the floor and the lighting exactly unchanged." Deployment instead uses "Move or resize the {target} as indicated. Keep the {reference} and the background exactly unchanged."

Software and version records. The transformer is `Qwen/Qwen-Image-Edit-2511`; the text encoder and VAE are from `Qwen/Qwen-Image`, with the `Qwen/Qwen-Image-Edit` processor. The framework is DiffSynth-Studio at commit `744bf641d9ea02adba77b7407da039a053f2b74d`. Recorded evaluation revision prefixes are `6f3ccc0b56e4`, `75e0b4be04f6`, and `ac7f9318f633`, respectively. Pair certification checks the stored renders; the original auxiliary-visual-input cache and the subsequent per-sample alignment audit are documented in Appendix E.6.4.

B.4 INFERENCE

Unless otherwise stated, we run the Qwen-Image editor for 40 denoising steps with classifier-free guidance scale 1.0. The spatial RoPE modification is active during the first 20 denoising steps and disabled thereafter. The source image is supplied without automatic resizing and with `zero_cond_t=True`. GenScale uses deterministic source–command seeds; the ObjectMover and stress-test benchmarks use seed 42.

At inference time, the actuator receives the source mask, the planned destination, and (c_s, c_d). It rasterizes the source target onto the latent grid using Eq. (24), constructs the transformed coordinates using Eq. (27), and applies the resulting RoPE field to the source edit stream.

B.5 OBJECT-CENTRIC RESAMPLING

RoPE control is carried only by the source target's tokens. For small objects, deployed inference therefore selects the largest local crop covering both source and planned destination while providing at least 16 token cells for each where feasible. The entire crop is resized to 768×768, edited, resized back with Lanczos interpolation, and blended into the source with a six-pixel feather (Gaussian $\sigma = 3$). This is a local scene edit, not an isolated object cutout. Planned targets that disappear are restored with the recorded mask-premultiplied warp on the 1024 grid.

Earlier runs used a fixed-margin crop. For source box B_{src} and source–destination union B_{union}, its side length was

$$L = \mathrm{clip}(\max(m\,\mathcal{L}(B_{\mathrm{src}}), 1.6\,\mathcal{L}(B_{\mathrm{union}}))\,, 64, 512)\,, \tag{30}$$

with $m = 3$, or $m = 2$ for objects thinner than 1.5 token cells. The union is centered horizontally and placed at 40% of the crop height vertically to retain support context below it. This legacy margin rule is distinct from the later token-constrained crop search. The deployed token threshold is 16; the 20-token floor in Appendix C is a training-data certification threshold.

C TRAINING PAIRS FROM 3D SCENE EDITS

C.1 MOTIVATION AND RENDERING SETUP

A source–target pair suitable for physical actuation must distinguish physical size from depth even when their effects on apparent extent cancel. We therefore construct the final actuator supervision by modifying explicit 3D scene geometry and re-rendering the complete scene using Kubric/Blender (Greff et al., 2022). Source and destination images share scene identity, camera, environment, reference object, and rendering configuration; only the prescribed target transformation and associated scene interaction change.

For each target, we apply a world-space scale transformation corresponding to c_s and a camera-relative depth transformation corresponding to c_d. In addition, an independent lateral translation can be introduced along the camera-right direction. Because this direction is orthogonal to the viewing direction and camera up vector, it changes image-space horizontal placement without changing the requested depth variable. All destination images are then re-rendered, allowing perspective, occlusion, exposed surfaces, contact, shadows, and indirect illumination to respond to the transformation.

C.2 SCENE AND COMMAND DISTRIBUTION

The final V7 dataset is rendered at 768×768 resolution and contains 2,468 certified source–destination pairs. The candidate pool contains 802 Google Scanned Objects (Downs et al., 2022) assets, after holding out 231 of 1,033 available assets; the certified rows use 624 assets in 500 asset pairs. The construction limits each asset to at most two pair assignments.

To avoid tying the actuator to a single synthetic camera, we sample from a factorized camera grid comprising

- focal lengths $\{60, 75, 89.06, 110, 135\}$ mm;
- camera heights $\{0.15, 0.22, 0.32, 0.45, 0.62\}$ m;

- camera distances $\{1.6, 2.2, 3.0, 4.0\}$ m;
- horizontal camera offsets $\{-1, -0.5, 0, 0.5, 1\}$ m; and
- horizon locations $\{0.36, 0.42, 0.50\}H$.

The Cartesian product contains 1,500 camera configurations, of which 835 occur in certified rows. Camera tilt is restricted to at most 5°. We use 381 indoor and outdoor HDR environments after excluding pure-sky environments and limiting night scenes.

Objects are rendered at yaw rotations $\{0^\circ, 90^\circ, 180^\circ, 270^\circ\}$. Lateral destination motion is stratified into approximately 0–2, 2–8, and 8–40 latent-token bins; the realized maximum displacement is approximately 39 token cells.

Physical commands cover

$$c_s \in [-0.6, 0.8], \qquad c_d \in [-0.6, 0.6], \tag{31}$$

with five size bins and four depth bins. The dataset includes pure-size, pure-depth, joint, compensation, identity/no-op, and translation-only command families. The design allocates approximately 15% to exact identity examples; 364 survive certification. To stress instance binding, a same-category bystander is included with probability 0.15. We also enforce coverage of the practically important $c_s > 0, c_d > 0$ regime, where physical enlargement is partially compensated by moving the object farther from the camera.

The certified family counts are 1,664 joint, 364 identity, 171 compensation, 150 size-only, 112 depth-only, and seven translation-only rows. Occlusion counts are 1,862 unoccluded, 331 target-in-front, and 275 target-behind rows. The same-category bystander flag occurs in 30.3% of retained rows; the 0.15 probability above is a per-pair construction draw, not the realized row proportion. Physical enlargement with motion away from the camera ($c_s > 0, c_d > 0$) occurs in 26.4% of rows.

C.3 Constraint-Driven Row Construction

Naively sampling 3D placements and rejecting invalid scenes is inefficient because token support, image-space displacement, frame validity, and physical placement are strongly coupled. We instead solve backward from the desired training cell.

For a fixed asset, camera, and command, projected support is monotonic in camera-relative depth. We therefore use a one-dimensional bisection search to find a depth satisfying the requested target-token band. Given this depth, the projected horizontal centroid varies affinely with translation along the camera-right axis, allowing the required screen-space displacement to be solved directly. This produces a feasible 3D proposal before expensive rendering.

We initially construct 2,500 candidate rows. After rendering and certification, 2,468 are retained. This procedure covers 332 of the 360 joint combinations of command, token-support, and displacement strata considered during data construction.

C.4 Token-Support Constraint

Because the actuator command is written only on source target tokens, we explicitly control this quantity during data construction. A projected bounding hull alone is insufficient for flat, hollow, or highly non-convex assets, for which the hull can substantially overestimate actual visible support. We therefore use three stages.

First, the proposal solver requires at least 35 projected hull tokens. Second, we refine this estimate using per-asset surface area to approximate projected contour support. Finally, after rendering, the actual source segmentation mask is rasterized onto the latent grid and the row is retained only if

$$N_T \geq 20. \tag{32}$$

For the final V7 dataset, source targets cover a median of 89.8 token areas ($|M_T|/16^2$, Eq. (25)); the minimum is 20.2 and the 90th percentile is approximately 315.

C.5 Rendered Geometry and Materialization

The renderer provides explicit source and destination object masks and world-space geometry. Training-time destination centers and apparent scaling are obtained directly from the rendered masks rather than from an auxiliary empirical correction law. For partially occluded destinations, the corresponding amodal render is used. Analytic projections are retained separately as consistency checks.

For each retained row we materialize:

- the source RGB image and target RGB image;
- target and reference segmentation masks;
- source and destination target centers on the latent grid;
- measured source and destination target extents;
- (c_s, c_d) and $a = \exp(c_s - c_d)$;
- camera and scene metadata; and
- identifiers required for exact source–target pairing.

Writing the measured mask extents as $\ell_{\mathrm{src}}^{\mathrm{render}}$ and $\ell_{\mathrm{dst}}^{\mathrm{render}}$, the training warp uses $a^{\mathrm{render}} = \ell_{\mathrm{dst}}^{\mathrm{render}} / \ell_{\mathrm{src}}^{\mathrm{render}}$ and centers $\mathbf{m}_{\mathrm{src}}, \mathbf{m}_{\mathrm{dst}}$ in Eq. (27). This measured ratio is kept distinct from the analytic $a = \exp(c_s - c_d)$.

C.6 Physical Certification

Every rendered row is subjected to deterministic geometric and physical checks before entering training. The certification protocol includes:

1. **Command consistency.** Ground-truth source and destination geometry must reproduce the prescribed $\Delta \log S = c_s$, $\Delta \log d = c_d$, and $\log a = c_s - c_d$ command relations to numerical tolerance. Here a is the analytic command factor; the observed rendered extent ratio is measured separately.
2. **Source integrity.** Source identity and scene metadata must correspond exactly to the intended source render.
3. **Token support.** The measured source target must satisfy Eq. (32).
4. **Instance identity.** Target and reference identities must be unambiguous, with exactly one intended destination target instance.
5. **Projection consistency.** The analytic world-to-image projector must agree with rendered object geometry.
6. **Frame validity.** Training targets may not be truncated by the image boundary.
7. **Physical validity.** Target transformations may not introduce mesh penetration or invalid support relationships with other objects.
8. **Occlusion validity.** Occlusion strata are evaluated using amodal object support rather than intersection of already-occluded visible masks.
9. **Interaction locality.** The radius of image changes caused by the target transformation must converge; examples whose lighting or indirect-interaction changes continue to expand across the frame are rejected.

Of 2,500 rendered candidates, 32 fail at least one final certification criterion, leaving 2,468 training pairs. Thirteen are rejected because the measured source target falls below the 20-token floor, 13 because the influence radius does not converge, and seven because the physical transformation produces scene-wide changes outside the certified influence regions; categories can overlap for multiply failing rows.

As an independent geometry check, the renderer and analytic projector agree on world-space object AABBs to a maximum error of 2.73×10^{-7} m in the final certified set. Planned versus rendered

lateral displacement has a median error of approximately 0.8 pixels and a 90th-percentile error of 6.4 pixels. These quantities are used only for data certification; actuator training itself uses the rendered source–target pairs and the standard flow-matching loss described above.

D Observer and Planner Details

D.1 Physical Observer

D.1.1 Open-vocabulary instance binding

The observer uses Grounding DINO (Liu et al., 2024) (`IDEA-Research/grounding-dino-base`) with one joint prompt containing the queried nouns. A noun's score is the maximum sigmoid score over its token span; the box threshold is 0.15 and at most 24 proposals are retained. SAM (Kirillov et al., 2023) (`facebook/sam-vit-base`) converts each box into a mask, keeping the mask with the highest predicted IoU and discarding masks smaller than 50 pixels.

Same-noun masks with IoU at least 0.70 are merged by retaining the larger mask; nested instances are not removed solely for containment. A Hungarian one-to-one assignment binds nouns to distinct primary instances. Primary scores below 0.15 remain unmatched, and extra instances attach only to nouns with a primary match. These detection-stage rules differ from the overlap rules used later for restoring missing objects. Measurement uses a 512-coordinate grid; the native image grid is 1024 and actuator patches have stride 16.

D.1.2 Ground contact and camera geometry

The contact point uses the mask's lowest row and horizontal box center. Its ray intersects the ground plane to give g_i. Camera intrinsics are fixed at $f/W = 2.4740$ (89.06 mm on a 36 mm sensor), with camera height 0.3237 scene units; camera tilt aligns the projected horizon with the estimated horizon. Observation and planning share these camera and support conventions rather than using pose-predicted translation.

Depth and normals come from `Ruicheng/moge-2-vitl-normal`; ground segmentation uses `nvidia/segformer-b4-finetuned-ade-512-512`, selecting ADE classes from the recorded ground-word list. For the GenScale horizon, candidate MoGe ground points lie below row $0.45H$ and outside the target and reference. When more than 500 candidates have normal component $n_y < -0.80$, only those normal-filtered points are used. Otherwise the lower-image fallback is retained. The fit regresses inverse depth on row, removes residuals above the 90th percentile, and refits; if $1/d = A_{\text{inv}}y + B_{\text{inv}}$, the horizon is $y_h = -B_{\text{inv}}/A_{\text{inv}}$. Acceptance requires at least 300 ground pixels and a finite horizon, not a threshold on fit quality. In the recorded 1,768-source horizon population, 1,740 use normal-filtered points and 28 use the fallback.

The stress-test variant uses the offset-robust model

$$d(y) = \frac{A_h}{y - y_h} + B_h \tag{33}$$

with soft-L_1 nonlinear least squares. This additive-depth-offset model is distinct from the GenScale inverse-depth fit. The original and corrected-horizon evaluations are reported separately in Appendix E.5.

D.1.3 Knowledge-base dimensions and projection fitting

A shape table supplies aspect ratios for 109 categories; absolute typical lengths come from a separate 163-row length table. The latter contains 98 entries shared with the authoritative GenScale table, two pinned entries, and 63 additional entries. All 100 GenScale categories are represented, with one differing typical length. Towel and watch are excluded from the observer. Canonical dimensions $\mathbf{D}_i = (L_i, W_i, H_i)$ combine the shape and length records.

All categories use the same projection measurement as Sec. 3.2. Let Q_i contain the eight corners of the cuboid with knowledge-base dimensions $\mathbf{D}_i = (L_i, W_i, H_i)$, R_i its orientation, g_i its ground

position, and Π the shared camera projector. Writing $b_i(\alpha)$ for the projected 2D box gives

$$b_i(\alpha) = \mathrm{BBox}(\Pi(g_i + \alpha R_i Q_i)), \tag{34}$$

$$\ell_i^{\mathrm{proj}}(\alpha) = \max(w(b_i(\alpha)), h(b_i(\alpha))), \tag{35}$$

$$\ell_i = \max(w(\mathrm{BBox}(M_i)), h(\mathrm{BBox}(M_i))). \tag{36}$$

Thus $\mathrm{extent}(\cdot)$ in the main text is the longest side of this box. The scalar solve finds α_i such that $\ell_i^{\mathrm{proj}}(\alpha_i) = \ell_i$, then returns $r_i = \log \alpha_i$. Bisection starts on $\log \alpha \in [-3, 3]$ and uses 50 iterations; an invalid upper projection bound is repeatedly reduced. Failure to obtain a valid interval with endpoint projections on opposite sides of the observed extent causes abstention. The camera, orientation, and ground position remain fixed during fitting; they are not assumed to be ground truth.

For the representative physical dimension used in Sec. 3.1, let S_i^{kb} be its knowledge-base value and $\widehat{S}_i = \alpha_i S_i^{\mathrm{kb}}$. Equation (7) then implies

$$r_i - r_j = \log \frac{\widehat{S}_i}{\widehat{S}_j} - \log \frac{S_i^{\mathrm{kb}}}{S_j^{\mathrm{kb}}}. \tag{37}$$

Residual differences therefore measure deviations from knowledge-base size ratios. Equal residuals recover those ratios without requiring every object to have its category's typical absolute size.

D.1.4 Pose routing and canonical-axis assignment

For the canonical extents sorted as $L_{(1)} \geq L_{(2)} \geq L_{(3)}$, the routing rule is

$$\mathrm{family}(L_{(1)}, L_{(2)}, L_{(3)}) = \begin{cases} \text{flat-planar}, & L_{(3)}/L_{(1)} < 0.30, \\ \text{elongated}, & L_{(3)}/L_{(1)} \geq 0.30 \text{ and } L_{(2)}/L_{(1)} < 0.55, \\ \text{compact}, & \text{otherwise.} \end{cases} \tag{38}$$

Flat and elongated objects use UniPose9D's public `last.ckpt`; compact objects use an upright identity rotation. The UniPose9D commit was not pinned. The family changes only the orientation supplied to the common projection solver, not its size measurement.

Predicted metric size and translation are not consumed as measurements. For predicted axis extents $\mathbf{s} = (s_x, s_y, s_z)$ and canonical dimensions $\mathbf{D} = (L, W, H)$, normalize each by its geometric mean and choose the axis permutation

$$\widetilde{s}_k = \frac{s_k}{(s_x s_y s_z)^{1/3}}, \quad \widetilde{D}_k = \frac{D_k}{(LWH)^{1/3}}, \quad \pi^\star = \operatorname*{arg\,min}_{\pi \in \mathcal{S}_3} \sum_{k=1}^{3} \left(\log \widetilde{s}_k - \log \widetilde{D}_{\pi(k)} \right)^2. \tag{39}$$

Only the axis assignment and rotation reach the scale solver. This normalization removes any global multiplicative scale in the pose model's predicted dimensions. Rotations are converted from the pose model's object-to-camera convention to the projector's world basis; handedness, axis-permutation, and scale-invariance checks accompany the implementation.

D.1.5 Consistency and abstention

The unfitted box dimension provides a diagnostic independent of the fitted longest side. If $\ell_i^{\perp}$ and $\hat{\ell}_i^{\perp}$ are its observed and predicted values, the recorded score is

$$e_{\mathrm{pose}} = \left| \log \frac{\hat{\ell}_i^{\perp}}{\ell_i^{\perp}} \right|. \tag{40}$$

The deployed observer records this score but does not reject instances on it. It also does not implement a separate severe-truncation gate. An object abstains when its canonical entry is missing or excluded, its foot is at most eight pixels below the horizon, its ray misses the ground, a required flat/elongated pose is missing, or no valid scale bracket exists. Unresolved instance binding prevents a corresponding observation. An unobservable target or reference voids the source; an unobservable bystander is omitted. The observer remains frozen and receives no actuator-training gradient.

D.2 Size and Placement Planner Details

D.2.1 Nominal candidate geometry

A candidate scene offset assigns nominal size commands

$$c_{s,i} = \mu - r_i. \tag{41}$$

The remaining realization variables are $\mathbf{y}_i = (c_{d,i}, \Delta x_i)$. The analytic projector predicts the destination footprint

$$M_i^{\mathrm{dst}} = F_i(M_i, c_{s,i}, c_{d,i}, \Delta x_i), \qquad a_i = \exp(c_{s,i} - c_{d,i}). \tag{42}$$

The deployed command dead zone is 0.26 in log-size units. The exact common-offset relation defines the nominal allocation; the later pair-safe and minimum-edit stages described below can relax equality while retaining the recorded pairwise tolerance conditions.

D.2.2 Prioritized costs

The deployed planner compares a finite set of candidates in priority order, rather than solving a continuous constrained optimization to global optimality. Support and collision precede image-space feasibility, which precedes the editor's apparent-scaling range. The image-change and motion costs are considered only after these constraint tiers tie. The same ordering is used for individual placements and the completed scene.

Candidates are ranked by non-negative violations. The physical tier measures invalid support or penetration. The image-space tier (superscript del, for deliverability) combines overlap/depth ordering with frame truncation and insufficient target-token support. The range tier penalizes excessive apparent warp. Each tier is quantized by

$$\bar{V}_i^k = \left\lfloor V_i^k / 0.02 \right\rfloor, \qquad k \in \{\mathrm{phys}, \mathrm{del}, \mathrm{range}\}. \tag{43}$$

The deployed overlap dead zone is 0.15. In front of another object the overlap term scales its covered fraction by 0.6; behind another object it measures the target's covered fraction. Covering the reference doubles the term. Framing takes the maximum of the out-of-frame fraction and starvation, where starvation is $(16 - N)/16$ when fewer than 16 destination-crop tokens are available. The range penalty is zero for $|c_{s,i} - c_{d,i}| \le 1.2$ and otherwise increases linearly, capped at one.

Once higher-priority violations are indistinguishable, the planner favors a smaller image change. On the 512 grid it uses

$$C_i^{\mathrm{change}} = \frac{|M_i \triangle M_i^{\mathrm{dst}}| + A_i^{\mathrm{contact}}}{512^2}, \qquad A_i^{\mathrm{contact}} = 2n_i^{\mathrm{contact}} \cdot 10. \tag{44}$$

Effort averages clipped depth and lateral deviations, normalized by 0.6 and 256 pixels, respectively. For fixed size commands, individual placements are compared by the ordered tuple $(\bar{V}_i^{\mathrm{phys}}, \bar{V}_i^{\mathrm{del}}, \bar{V}_i^{\mathrm{range}}, C_i^{\mathrm{change}}/0.10, C_i^{\mathrm{motion}})$. The scene key aggregates worst-object violations and total costs:

$$\begin{aligned}\mathbf{J}_{\mathrm{scene}} = \big(&\max_i \bar{V}_i^{\mathrm{phys}},\ \max_i \bar{V}_i^{\mathrm{del}},\ \max_i \bar{V}_i^{\mathrm{range}},\\ &\textstyle\sum_i C_i^{\mathrm{change}}/0.10,\ \sum_i C_i^{\mathrm{motion}},\ N_{\mathrm{edit}}\big).\end{aligned} \tag{45}$$

We compare these keys lexicographically: at the first component that differs, the smaller value wins. Thus lower image-change or motion cost cannot compensate for a worse higher-priority violation. This ranking identifies the preferred searched candidate; it does not certify a globally optimal or violation-free plan. The later tolerance refinements need not preserve exact common-offset equality.

D.2.3 Multi-object search and deployed refinements

For each offset hypothesis, objects are processed greedily from largest source area to smallest. Earlier placements become obstacles for later ones. The coarse pass uses three depth values and nine lateral positions, with a 64-pixel lateral step. The fine base grid uses 25 depth values and 65 lateral positions, with an eight-pixel lateral step over ± 256 pixels:

$$K_{\mathrm{coarse}} = 3 \times 9 = 27, \qquad K_{\mathrm{fine}} = 25 \times 65 = 1625. \tag{46}$$

Depth commands lie in $[-0.6, 0.6]$, restricted to the nominal direction; the nominal depth and parent lateral position are also included. The common-offset stage considers 21 grid points over $[\max_i r_i - \Lambda, \min_i r_i + \Lambda]$, with $\Lambda = 1.2$, plus each object's own residual. The chosen offset is replanned on the fine grid, then propagated to additional instances. Greedy placement and coarse offset selection are approximations rather than global joint optimization.

The deployed C1 chain includes two further safeguards. Pair-safe re-selection accepts a different shared offset only if no pair's excess outside $b = \log 1.2$ increases, the mean excess strictly decreases, and the first three scene-key tiers are no worse than the base plan. Pair errors are computed with execution dose $\eta = 1.08$, the factor by which size commands are scaled when passed to the actuator (Algorithm 1). The accepted size commands are then held fixed for fine-grid placement and propagated to additional instances of the same object category. The base plan is retained when no alternative passes these conditions.

A subsequent minimum-edit linear program reduces the sum of absolute size commands while preserving their signs, not exceeding their parent magnitudes, and not worsening any parent's pairwise band excess. Active magnitudes remain between 0.26 and 1.2; unchanged objects remain unchanged. These refinements, the command dead zone, and the execution dose distinguish the delivered commands from the exact common-offset allocation in Eq. (8). In particular, final residuals need not all be equal.

D.2.4 Complexity and exact lazy token checks

For N objects and n_μ offset hypotheses, the base search evaluates

$$O(n_\mu N K_{\text{coarse}} + N K_{\text{fine}}) \tag{47}$$

candidates, ignoring fixed-resolution raster work. Accounting for interaction with all other objects gives the conservative bound $O(n_\mu N^2 K_{\text{coarse}} + N^2 K_{\text{fine}})$. Bounding-box rejection avoids many pairwise tests, and lateral candidates at a fixed depth can reuse their projected footprint.

Crop-token evaluation is more expensive than the remaining key terms and can only worsen a candidate's key. The unchecked key is therefore an optimistic lower bound. The accelerated implementation sorts candidates by this key and a deterministic index, checks them in order, and stops once the next optimistic key is strictly worse than the best fully checked key. Ties are still inspected under the same index rule. This preserves exhaustive-search selection rather than approximating token support. The efficiency comparison applies this acceleration to C1; it does not time the stress-test variant end to end.

D.2.5 Stress-test variant and execution checks

The stress-test variant expands the depth-command domain to $[-1.9, 1.9]$, keeps lateral motion within ± 256 pixels, and effectively disables the actuator-range penalty. It prioritizes side overhang, physical validity, overlap/framing, and rear overhang before the remaining costs. Its overlap dead zone is 0.03; additional preferences use a 16-pixel image margin and a 140–212-pixel longest-side band. Support estimation interpolates table regions hidden by the target, and horizon estimation uses Eq. (33).

The fallback ladder returns strict, relaxed, or fallback candidates, but all are scored against the original evaluation gates. Neither ranking an imperfect candidate nor returning a fallback certifies a valid final output. Token-constrained cropping can be infeasible, and an unexecuted plan is scored as a failure in the photorealistic stress test. Appendix E.5 separates planner variants, final rendering measurements, and the original versus corrected-horizon results.

D.2.6 Planner Pseudocode

Algorithm 1 summarizes shared-scale search and refinement in the frozen C1 planner. Algorithm 2 places objects for fixed size commands using the geometry and keys in Eqs. (42)–(45). Pair-safe re-selection can change the nominal allocation, and minimum-edit refinement can shorten its commands; neither requires the final residuals to share one exact value. Grid construction and lazy token checking are specified in Sections D.2.3 and D.2.4. The stress-test variant is described separately in Section D.2.5.

Algorithm 1 Shared-scale planning and command refinement (C1)

Require: Observed scene $\mathcal{S}$ with primary instances, masks M_i, residuals r_i, camera and support geometry
Ensure: Size–depth commands and lateral placements for the edited instances

Set $\Lambda = 1.2$, command dead zone $\tau = 0.26$, pair band $b = \log 1.2$, and dose $\eta = 1.08$.
$\mathcal{U} \leftarrow \mathrm{linspace}(\max_i r_i - \Lambda, \min_i r_i + \Lambda, 21) \cup \{r_i\}_i$.
Initialize the candidate-plan set $\mathcal{C} \leftarrow \varnothing$.
for $\mu \in \mathcal{U}$ **do**
    Assign nominal size commands $c_{s,i} \leftarrow \mu - r_i$ (Eq. (8)).
    Set commands inside the dead zone τ to zero.
    $(P_\mu, \mathbf{J}_\mu) \leftarrow$ PLACEOBJECTS$(\mathcal{S}, \{c_{s,i}\}, \mathcal{G}_{\mathrm{coarse}})$; add $(P_\mu, \mathbf{J}_\mu)$ to $\mathcal{C}$.
Choose base plan P_0 with lexicographically smallest $\mathbf{J}_{\mathrm{scene}}$ in $\mathcal{C}$.
Retain alternatives whose constraint-tier key is no worse than P_0 and whose e_{jk} never increases.
Pair-safe re-select only if mean e_{jk} strictly decreases; otherwise retain P_0.
Hold the accepted size commands fixed and replan with $\mathcal{G}_{\mathrm{fine}}$.
Propagate the selected correction to additional instances of the same category.
Apply minimum-edit LP refinement under the sign, magnitude, and pair-band restrictions in Section D.2.3.
return the refined commands and placements.

For a plan P, the predicted excess of pair (j, k) at execution dose 1.08 is

$$e_{jk}(P) = \max\{0, |(r_j + 1.08 c_{s,j}(P)) - (r_k + 1.08 c_{s,k}(P))| - \log 1.2\}\,. \tag{48}$$

Algorithm 2 PLACEOBJECTS: greedy placement with fixed size commands

Require: Scene $\mathcal{S}$, fixed size commands $\{c_{s,i}\}$, and a depth–lateral grid $\mathcal{G}$
Ensure: Candidate plan P and its scene key $\mathbf{J}_{\mathrm{scene}}$

Sort edited primary instances by decreasing source-mask area $|M_i|$.
Initialize P with unchanged instances and the shared support geometry.
for each edited instance i in this order **do**
    Initialize its selected placement as unset.
    for each allowed $(c_{d,i}, \Delta x_i) \in \mathcal{G}_i$ **do**
        Project its destination mask and apparent scale using Eq. (42).
        Measure support and collision violations against the current scene.
        Measure overlap/depth-order, frame, and target-token violations.
        Measure the apparent-scaling-range violation.
        Quantize the three tiers using Eq. (43).
        Compute image-change and motion costs as in Section D.2.2.
        Retain the candidate with the lexicographically smallest placement key.
    Commit its placement to P; it becomes an obstacle for subsequent objects.
Aggregate the worst-object tiers and total costs into $\mathbf{J}_{\mathrm{scene}}$ (Eq. (45)).
return $(P, \mathbf{J}_{\mathrm{scene}})$.

The base grids contain 3×9 coarse and 25×65 fine depth–lateral samples before the direction restrictions and extra candidates in Section D.2.3. Exact lazy token checking (Section D.2.4) avoids unnecessary crop evaluations without changing candidate selection.

E EXPERIMENTAL PROTOCOLS AND ADDITIONAL RESULTS

E.1 IMPLEMENTATION AND STATISTICAL SETUP

The observer and planner configurations are specified in Appendix D, actuator architecture and optimization in Appendix B, and 3D training pairs in Appendix C. Only the actuator receives task-specific training. Gemini-3.1-Pro-preview serves as the comparison observer or planner and as an offline judge, not as a correction-decision component of PhyScale. The following sections preserve experiment-specific populations and configurations rather than treating the deployment, stress-test, and diagnostic variants as a single run.

E.1.1 STATISTICS

Intervals are 95% bootstrap intervals using each experiment's analysis unit: source images for source-level comparisons, scene groups for ObjectMover, and product families for planner stress tests. Most experiments use 10,000 draws; the main GenScale comparison uses 4,000 draws stratified by source generator. Observer size-identification comparisons use paired scene resampling unless stated otherwise. Holm adjustments are applied only to the contrast families specified in their original records. Intervals containing zero do not establish equivalence or noninferiority. Human ratings, offline model ratings, and geometric gate outcomes are reported separately. Main-text tables show point estimates only; bold and underline mark the best and second-best reported values, not statistical significance.

E.2 GENSCALE SCORING AND SENSITIVITY

GenScale Task 1 contains 400 sources from each of seven generators and 8,001 designed object relations. The evaluated generators are Nano Banana 2 (Google DeepMind, 2026b), GPT-Image-2 (OpenAI, 2026), Z-Image-Turbo (Z-Image Team et al., 2025), Grok Imagine (xAI, 2026), Qwen-Image 2512 (Qwen Team, 2025b), FLUX.2 (Black Forest Labs, 2025), and SD3.5-Large (Stability AI, 2024). The offline judge is Gemini-3.1-Pro-preview, with temperature 0.35 for size scoring and 0.1 for scene prefiltering. Five votes are aggregated by a unique mode or, when needed, a rounded median. The human-calibrated rubric assigns score three to the valid size band, with relation error $|s - 3|$. The main view (P2 in Table 9) retains all designed relations and assigns error two to protocol-defined unjudgeability or skips; technical missingness is tracked separately rather than silently treated as a semantic failure. Additional visual comparisons with ReScale appear in Fig. 15.

The main record combines historical batches rather than a synchronized three-arm rerun. In the final incremental update, 35 changed images were scored in a new batch while other scores were reused. We therefore supplement it with the same-batch re-evaluation below; source bootstrap alone does not account for systematic batch effects. Table 8 gives generator-specific paired intervals; Table 9 shows that the point-estimate ordering depends on the scoring population and weighting. A post-hoc exclusion of 39 human-screened sources leaves 2,761 sources and 7,865 relations, with a ReScale-minus-PhyScale difference of 0.008773 $[-0.024558, 0.040850]$; this sensitivity does not replace the unfiltered main result.

Table 8: **Generator-level paired differences.** ReScale error minus PhyScale error under the main P2 protocol. Per-generator intervals use 4,000 source-bootstrap resamples within each generator; the aggregate uses source bootstrap stratified by generator. No reported 95% interval excludes zero.

Generator	Difference	95% CI
Nano Banana 2	-0.013123	$[-0.091384, 0.065094]$
GPT-Image-2	$+0.066492$	$[-0.012142, 0.142364]$
Z-Image-Turbo	$+0.009624$	$[-0.072899, 0.090592]$
Grok Imagine	-0.021872	$[-0.103791, 0.058618]$
Qwen-Image 2512	-0.083115	$[-0.166515, 0.000850]$
FLUX.2	$+0.007874$	$[-0.078179, 0.096507]$
SD3.5-Large	$+0.096238$	$[-0.008475, 0.196850]$
All seven	$+0.008874$	$[-0.023515, 0.040720]$

Table 9: **GenScale evaluation sensitivities.** Lower error is better; a positive ReScale-minus-PhyScale difference favors PhyScale. Denominators are relations except for the source-equal row, which uses 2,800 sources. These views do not replace the main unfiltered relation-equal result.

View	Denominator	ReScale	PhyScale	Difference
P0: jointly scorable relations	6,764	0.479598	0.507836	-0.028238
P1: source-scorable relations	7,452	0.566425	0.576758	-0.010333
P2: all designed relations	8,001	0.649044	0.640170	$+0.008874$
P2: equal source weighting	2,800	0.587857	0.612679	-0.024821
P2: prefilter-completed sensitivity	8,001	0.649044	0.646794	$+0.002250$

E.2.1 Same-Batch Re-Evaluation

We re-evaluate both correction methods in one judging batch on a stratified subset of 700 sources, selected by a fixed hash order with 100 sources per generator. The subset contains 2,029 relations. Sampling and protocol were frozen before scoring. We use the frozen PhyScale delivery and the corresponding ReScale outputs; pixel-identical outputs share one judgment. The judge, prefilter, temperature, and five-vote aggregation match the main comparison. The first size-scoring attempt allows 2,048 output tokens instead of 512, avoiding the truncation and 2,048-token retry observed at the smaller initial budget. Intervals use 10,000 source-bootstrap draws stratified by generator.

Table 10: **Same-batch robustness check.** Lower error is better. $\Delta = E_{\mathrm{ReScale}} - E_{\mathrm{ours}}$, so positive values favor PhyScale. The historical row uses exactly the same 700 sources, not the full main-table population.

Evaluation view	Ours	ReScale	Δ	95% CI
Same batch: all 2,029 relations	0.681	0.640	-0.041	$[-0.106, 0.025]$
Jointly scorable: 1,728 relations	0.527	0.490	-0.037	$[-0.094, 0.020]$
Equal source weighting: 700 sources	—	—	-0.042	$[-0.101, 0.016]$
Exclude 41 sources prefiltered for only one method	—	—	-0.041	$[-0.098, 0.013]$
Historical scores: same 700 sources	0.667	0.630	-0.037	$[-0.101, 0.029]$

The same-batch and historical comparisons on this subset give similar differences and neither detects a separation (Table 10). This supports the stability of that conclusion under the check, rather than proving zero batch effect. The subset's point estimates favor ReScale; they do not show that PhyScale outperforms it. One relation per method lacks valid judge JSON; excluding these technical-missing entries leaves the reported difference and interval unchanged at the shown precision. This robustness check supplements, rather than replaces, the 2,800-source main table.

E.3 Front-End–Actuator Swap

The factorial comparison uses 196 randomly sampled GenScale development sources (28 prompts $\times$ seven generators), with 679 relations. All four combinations use the same subset; restricting this comparison limits the cost of repeated VLM-based diagnosis and editing. Front ends are the PhyScale observer/planner and ReScale's Gemini front end; back ends are the PhyScale actuator and ReScale's actuator. The offline judge provides one vote per relation, with missing objects assigned error two. Source images have mean error 0.865 before correction. Table 11 reports the five specified contrasts with Holm adjustment. With the PhyScale front end fixed, switching to our actuator improves correction; neither front-end contrast nor the interaction survives the same adjustment. The reused development subset and single-vote judge distinguish this study from the full five-vote GenScale comparison.

Quality–cost trade-off. To place these configurations on a common estimated-cost scale, we use stage unit costs from the 30-source timing study (Appendix E.9) and each front end's actual edit count on the 196-source subset. We separate Gemini planning calls (scoring, global review, and layout grounding) from execution calls (background cleanup and over-removal checks). The recorded front-end estimates are \$0.454 per source for Gemini and \$0.006 for PhyScale; per-edit estimates are \$0.217 for ReScale and \$0.0356 for our actuator, including restoration. The front ends issue 1.28 and 1.42 edits per source, respectively. We estimate total cost as front-end cost plus edits per source times actuator cost per edit. Table 12 retains the recorded totals; independently rounded entries need not sum exactly.

Table 11: **Paired component-swap contrasts.** Positive reductions favor the first named component. Contrasts use relations scored in both compared arms (GR: 673/679), so they can differ slightly from differences of Table 12 means; interaction $= (\mathrm{PR} - \mathrm{PP}) - (\mathrm{GR} - \mathrm{GP})$. Intervals use the experiment's paired resampling; p values are Holm-adjusted within these five contrasts.

Contrast	Error reduction	95% CI	Holm p
Our actuator vs. ReScale; our front end	$+0.171$	$[0.074, 0.270]$	0.0035
Our actuator vs. ReScale; Gemini front end	$+0.137$	$[0.035, 0.256]$	0.058
Gemini vs. our front end; our actuator	$+0.084$	$[0.008, 0.155]$	0.078
Gemini vs. our front end; ReScale actuator	$+0.116$	$[0.014, 0.226]$	0.078
Interaction	$+0.031$	$[-0.084, 0.147]$	0.600

Table 12: **Quality and estimated cost of the four compositions.** P: PhyScale; G: Gemini front end; R: ReScale actuator. Costs are projected to the 196-source swap population from measured stage unit costs, not invoices or a new end-to-end timing run. VLM decisions refers to the correction front end, not background-cleanup calls.

Front end	Actuator	Error ↓	Front ($)	Edits ($)	Total ($)	VLM decisions
P	P	0.654	0.006	0.051	0.057	No
G	P	0.570	0.454	0.046	0.499	Yes
P	R	0.826	0.006	0.308	0.314	No
G	R	0.701	0.454	0.277	0.731	Yes

Using Gemini with our actuator gives the lowest error point estimate, at approximately nine times the estimated cost of the complete PhyScale system. The adjusted front-end test does not establish equal accuracy or noninferiority. The 30-source pipeline means in Table 2 use that study's own action counts and therefore are not interchangeable with these projected totals. A separate one-rater swap study is descriptive only: actuator quality preferences were 20:4 for our actuator, while 89/100 placement judgments were ties.

E.4 Observer Evaluation

Each scene contains three anonymously labeled objects, exactly one at an incorrect physical size. Methods receive the source image, object boxes, correct mesh dimensions, and the specified image-estimated horizon. The observer identifies the most mutually consistent object pair and infers the remaining object's correction; pose-sensitive objects use predicted rather than true rotations. Success requires the correct object and a linear correction-factor error no greater than 20%. Gemini-3.1-Pro-preview uses temperature 0.1 and structured output. The final comparison uses the 116 Blender scenes and 109 accepted photorealistic counterparts described in Section E.5.

E.4.1 Reconstruction-Based Alternative

The SAM 3D + MoGe-3 baseline replaces the projection-fitting observer with an object/scene reconstruction pipeline. SAM 3D Objects (Chen et al., 2026) is an image-and-mask-conditioned object-reconstruction model, while MoGe-3 (Kong et al., 2026) estimates monocular scene geometry. This comparison tests a replaceable observation stage, not a swap of UniPose9D alone. Table 13 compares this alternative with our observer and Gemini under the same evaluation protocol. The reconstruction pipeline has mixed accuracy relative to our observer and substantially higher recorded runtime; these point estimates do not establish equivalence.

Table 13: **Alternative observers on the final evaluation sets.** Accuracy (%) uses the same wrong-object and correction-factor criterion as Table 3. Time and estimated cost are the recorded per-image observer measurements, not pipeline totals.

Observer	Blender	Photorealistic	Time (s)	Cost ($)
Gemini 3.1 Pro	44.8	38.5	58.5	0.094
SAM 3D + MoGe-3	45.7	**47.7**	176	0.045
PhyScale	**56.9**	44.0	**0.34**	**0.0002**

E.4.2 Matched Horizon and Orientation Diagnostic

To localize observation errors, we replace the horizon and object orientation independently with ground truth on the same 116 Blender scenes, keeping all other inputs and processing fixed and without running the editor. The predicted-horizon/predicted-orientation outputs match the frozen observation results. The protocol was frozen before scoring, including two pre-scoring corrections. Confidence intervals resample object families 10,000 times; Holm correction is applied to the three primary contrasts.

Table 14: **Matched observer input replacements.** Accuracy (%) on 116 Blender scenes. Ground-truth inputs diagnose errors; they are not available at deployment.

Horizon	Predicted orientation	Ground-truth orientation
Predicted	56.9	57.8
Ground truth	73.3	65.5

With predicted orientation, replacing the horizon improves accuracy by 16.4 percentage points [4.5, 27.9] (Holm-adjusted $p = 0.019$). With the predicted horizon, replacing orientation changes accuracy by 0.9 points $[-10.5, 12.0]$, with no detected improvement; the recorded median per-axis UniPose orientation error is about 5°. The joint replacement is not the sum of these individual effects: ground-truth orientation lowers the point estimate under a ground-truth horizon. On the 109 photorealistic counterparts, the recorded horizon replacement improves accuracy by 26.6 points [14.2, 38.5] and removes the 17% abstention rate associated with the predicted horizon.

With both horizon and orientation set to ground truth, replacing the cuboid approximation with the true object shape adds 28.4 points [19.3, 37.6]. Replacing placement with ground truth as well yields 116/116 correct diagnoses. These input-replacement tests identify horizon estimation and shape approximation as important error sources on this set; they do not show that pose estimation is irrelevant or guarantee error-free observation outside these controlled conditions.

E.4.3 Earlier Observer Diagnostics

Table 15 retains earlier horizon/pose snapshots. Their populations and configurations differ from the matched experiment above, so their intervals are not assigned to the final main-table point estimates.

Table 15: **Earlier observer diagnostic snapshots.** Accuracy in percent; differences are ours minus Gemini in percentage points. Ground-truth horizon is a nondeployment diagnostic. Its Gemini number comes from the recorded reference comparison, not a newly established matched true-horizon arm.

Setting	Ours	Gemini	Difference [95% CI]
Blender, estimated corrected horizon	53.1	47.4	$+5.6\ [-2.6, 13.8]$
Blender, upright control	53.1	—	—
Blender, old frozen horizon	33.2	47.4	$-14.3\ [-23.0, -6.1]$
Photorealistic, frozen horizon	33.0	39.4	$-6.4\ [-14.7, 1.8]$
Photorealistic, corrected horizon	45.0	50.5	$-5.5\ [-15.6, 4.6]$
Photorealistic, true horizon diagnostic	72.5	39.4	—

These earlier diagnostic snapshots contain 196 Blender scenes and 109 photorealistic scenes. Object identification alone is nearly saturated in the Blender comparison, whereas correction magnitude remains harder. The deployed-pose and upright point estimates coincide; this experiment does not establish a benefit of predicted pose. The true-horizon diagnostic implicates horizon estimation as a bottleneck but is not a deployable accuracy result.

E.5 Planner Stress Tests

E.5.1 Construction and Automatic Evaluation

Each scene contains three GSO (Downs et al., 2022)/YCB (Calli et al., 2015) objects on a tabletop and a locked correction factor for one target. The planner may change only its depth and lateral position. Three rendered counterexamples define the tasks: enlarging by $4\times$ truncates at least 25% of the target with at least 48 pixels beyond the frame; enlarging by $3\times$ reduces a neighbor's visibility by

at least 0.35; shrinking to $1/6$ leaves a longest side of at most 28 pixels and area at most 0.3% of the frame. Tight-table variants add only three centimeters around the union of the relevant object footprints. Scenes are deduplicated by geometry and capped by asset family within task type. The final set contains 116 scenes from 61 target families and 82 distinct assets: 59 normal-table and 57 tight-table scenes. The permitted camera field of view is 45–60 degrees; the retained horizontal fields of view span $45.4°$–$55.8°$ (median $46.9°$). Table 16 gives the six task–table combinations and accepted photorealistic counts.

Table 16: **Planner stress-test composition.** Counts of Blender scenes by failure mode of in-place correction and table variant, and of accepted photorealistic counterparts. The prescribed factor is the linear physical-size correction of the target.

Failure of in-place fix	Factor	Normal	Tight	Blender	Photorealistic
Truncation	$\times 4$	20	20	40	35
Occlusion	$\times 3$	20	20	40	39
Small target	$\times 1/6$	19	17	36	35
Total		59	57	116	109

Both planners receive the same image, annotations, image-estimated horizon, 64×64 support grid, forward formulas, numeric requirements, and fallback rules. Gemini-3.1-Pro-preview uses temperature 0.1, JSON output, and a 16,384-token output limit. Both plans are executed by Blender Cycles on CPU with 64 samples, no denoising, six maximum bounces, four transparent bounces, and 512×512 output. The renderer preserves target orientation and non-target geometry. Amodal masks come from alpha, visible masks from object-ID renders, and full-frame membership from mesh projection through an overscan camera.

Table 17: **Frozen Blender layout gates.** The main pass rate requires all gates except the last row. Full tabletop support is reported separately and as an additional-gate sensitivity.

Gate	Target	Each neighbor
Longest side	128–224 pixels	96–192 pixels
Short side / area	At least 48 pixels / 1.5% of frame	Same
Image boundary	Fully in frame; margin at least 12 pixels	Margin at least 12 pixels
Visibility	At least 0.90	At least 0.90; decrease no more than 0.05
Execution	Exact prescribed physical scale; orientation unchanged	Unmoved
Additional support	Entire footprint on the table	—

Every scene also stores a feasible reference correction combining the prescribed physical-size change with a different depth and lateral position that satisfies the layout checks in Table 17. This reference demonstrates a valid solution to the constructed task but is never supplied to either planner. Figure 9 shows nine examples, three per failure mode. The displayed task examples are recorded in a saved selection manifest and illustrate construction rather than comparative method performance.

E.5.2 Development Sequence and Subgroups

The first frozen planner was evaluated on 107 scenes and underperformed Gemini. Those scenes subsequently became development data. Revisions addressed saturated small-target tie-breaking, occluded-table interpolation, overlap tolerances, image margins, horizon offsets, and conservative table-edge observations. A separate 89-scene evaluation followed, then a 30-scene development pilot motivated tight tables. The final comparison used 116 new scenes after freezing its sample list, protocol, and analysis plan. This sequence is reported rather than presenting all evaluations as tests of an unchanged method.

In the final set, the advantage is concentrated in occlusion scenes: 35/40 pass for PhyScale and 26/40 for Gemini, a difference of 22.5 $[8.5, 38.9]$ percentage points. Both methods pass 34/40 truncation scenes and 33/36 small-target scenes. Results are 53/59 versus 44/59 on normal tables and 49/57 for both methods on tight tables. Adding full tabletop support yields 88/116 versus 80/116, a difference of 6.9 $[0.0, 14.8]$ percentage points. Its interval reaches zero and is reported as inconclusive. The

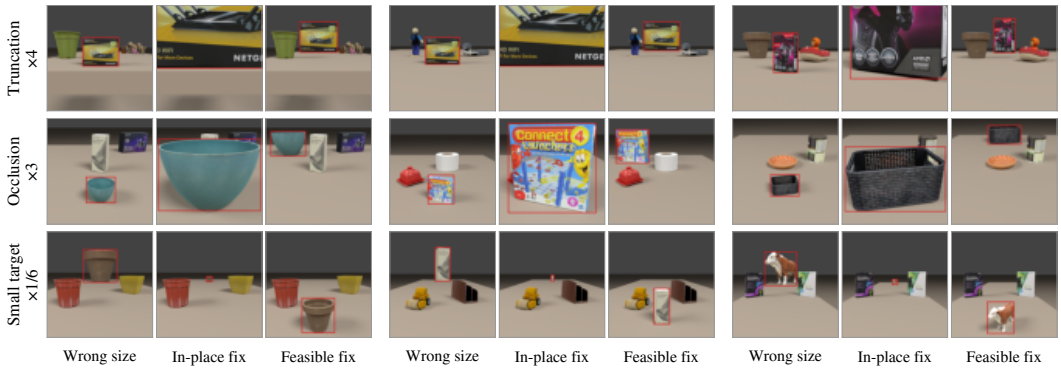

Figure 9: **How the stress tasks are constructed.** Three examples per failure mode; red boxes mark targets. Each triplet shows the wrong-size input, the prescribed correction applied in place, and the stored feasible reference. In-place correction truncates an enlarged target ($\times 4$), hides a neighbor ($\times 3$), or leaves a shrunken target too small ($\times 1/6$). The reference uses the same physical-size change with adjusted placement and is not a planner input.

Table 18: **Planner development and held-out comparisons.** Pass rates (%); differences are ours minus Gemini in percentage points. The 30-scene pilot is development data.

Stage	n	Ours	Gemini	Difference [95% CI]
Initial frozen planner	107	63.6	74.8	-11.2 $[-23.2, -1.5]$
Intermediate revision (held out)	89	92.1	80.9	$+11.2$ $[5.0, 17.8]$
Hard-scene development pilot	30	93.3	90.0	$+3.3$ $[-12.9, 16.7]$
Final frozen planner	116	87.9	80.2	$+7.8$ $[1.6, 14.5]$

main constraints can also miss visually crowded arrangements, a known failure mode of the explicit planner.

E.5.3 Photorealistic Transfer

Gemini-3.1-Flash-Lite-Image (Google DeepMind, 2026a) generates a photorealistic counterpart from each wrong-size Blender reference. We consider up to ten candidates and retain the first that passes the consistency gates in Table 19. The accepted set contains 109/116 scenes; the seven missing counterparts comprise five truncation scenes, one occlusion scene, and one small-target scene. Prompts use the recorded normal- and tight-table templates; generation is at 1024 pixels with a deterministic scene/environment/attempt seed. Figure 10 shows four accepted pairs per task, selected by sorting the SHA-256 hashes of the string `pair` followed by each sample identifier.

Table 19: **Acceptance checks for photorealistic counterparts.** These checks filter dataset inputs before planning; they are distinct from the output-level success gates. Pixel thresholds use the 512-coordinate grid. Size error is $\max(|w' - w|/w,\ |h' - h|/h)$ between the SAM mask bounding boxes of an object in the generated image (w', h') and in the Blender reference (w, h).

Check against the Blender reference	Requirement
Target identity (DINOv2 cosine similarity)	≥ 0.60
Object center/contact drift	≤ 12 pixels
Neighbor size error	≤ 0.075
Target size error	≤ 0.25
Median tight-table edge offset	≤ 12 pixels

The accepted images span eight recorded environment templates: oak worktops (47), small oak tables (39), small white tables (10), marble counters (6), small birch tables (2), small walnut tables (2), wooden workbenches (2), and steel worktops (1). Small freestanding tables are used for tight-table scenes, while normal-table scenes mostly use wide worktops. These are generated counterparts, not independently captured photographs, and use of the Gemini model family for both generation and comparison is a possible unmeasured source of bias.

Let γ denote the prescribed linear correction factor and m the target's measured generation-induced size factor. We adjust the photorealistic request to $\gamma_{\text{photo}} = \gamma/m$ rather than assuming image

generation preserves target size exactly. SAM resegments the generated images using rendered boxes as prompts.

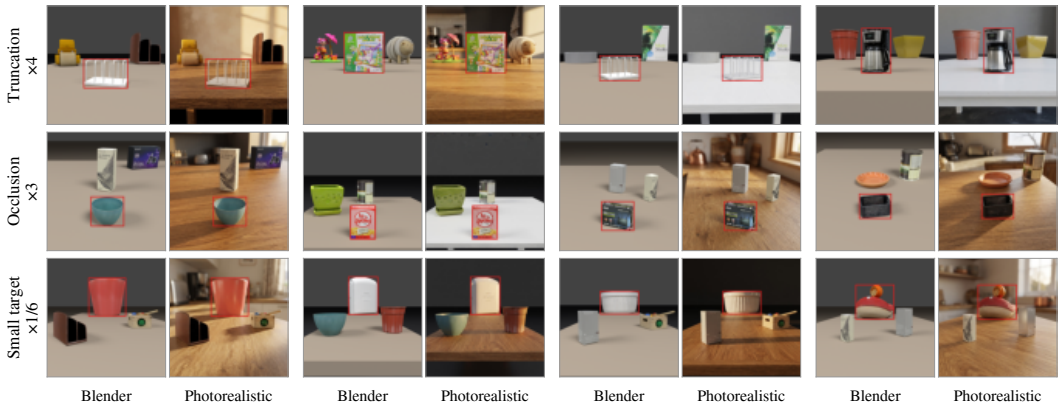

Figure 10: **Blender inputs and accepted photorealistic counterparts.** Four pairs per failure mode. The counterpart changes surface appearance, lighting, and background while passing the identity and geometry checks in Table 19. Red boxes identify targets. Both images retain the size error; the photorealistic image is not a corrected output.

Both plans are executed on each photorealistic source by the same frozen V7 actuator. The primary output-level criterion applies the layout gates to detected/segmented outputs and additionally requires target identity cosine at least 0.60, scale error no greater than 20%, and stationary neighbors. Unexecuted plans and undetected objects count as failures. As a secondary diagnostic, the plans are executed in their Blender parent scenes. These two evaluations differ because the photorealistic criterion includes editing and measurement failures. The original frozen-horizon variant and the corrected-horizon variant are reported separately; the corrected-horizon variant was run after the frozen-horizon evaluation and is not a first-pass result. Figure 17 shows additional photorealistic placement examples under the shared actuator.

Table 20: **Photorealistic planner evaluation.** Automatic pass rates (%); paired differences in percentage points. The placement-only variant is post hoc. All rows use 109 scenes.

Measurement and variant	Ours	Gemini	Difference [95% CI]
V7 output, frozen horizon	39.4	46.8	-7.3 $[-13.9, -0.9]$
V7 output, corrected horizon	46.8	56.0	-9.2 $[-18.7, 0.9]$
Blender parent execution, corrected horizon	70.6	66.1	$+4.6$ $[-6.4, 15.7]$
V7 output, placement only	51.4	57.8	-6.4 $[-15.3, 2.9]$

E.5.4 Blind Placement Preferences

Five raters each evaluate every scene in a domain. Each trial displays the source, the boxed target and prescribed correction, and two outputs with randomized left/right order and hidden method identities. Raters select the placement that keeps the target visible, moderately sized, away from image boundaries, and without hiding neighboring objects, or choose a tie. A separate optional flag marks both outputs as clearly unacceptable. Catch trials comprise two identical pairs and two swapped repeats per rater. The scene score averages $+1$ for PhyScale, -1 for Gemini, and zero for ties across raters; confidence intervals resample product families.

For Blender, majority outcomes are 30 PhyScale wins, 15 Gemini wins, and 71 ties; 437 of 580 individual votes are ties. The mean preference is 0.122 $[0.049, 0.202]$, with Fleiss' $\kappa = 0.64$. Two initial raters redid their sheets after clarification of the boundary criterion. Substituting their original sheets gives 0.078 $[0.019, 0.143]$. For photorealistic scenes, majority outcomes are 40 wins, eight losses, and 61 ties; the mean preference is 0.172 $[0.104, 0.245]$, with $\kappa = 0.58$. Each domain has 10/10 correct identical-pair catches and 9/10 consistent swapped repeats. The human analysis was not separately preregistered. These placement judgments do not measure identity preservation or general image naturalness.

E.5.5 Complete Stress-Test Galleries

Figures 11 and 12 show all 116 Blender inputs and all 109 accepted photorealistic inputs. Both galleries group scenes by failure mode and place normal-table scenes before tight-table scenes within each group. The photorealistic gallery omits only the seven rejected counterparts. Unlike the planner-output comparisons in Section F.3, these galleries document the evaluation inputs, not selected successful edits.



Figure 11: **All 116 Blender stress-test inputs.** Scenes are grouped by the failure caused by in-place correction; within each group, normal-table scenes precede tight-table scenes. Red boxes mark the wrong-size targets.

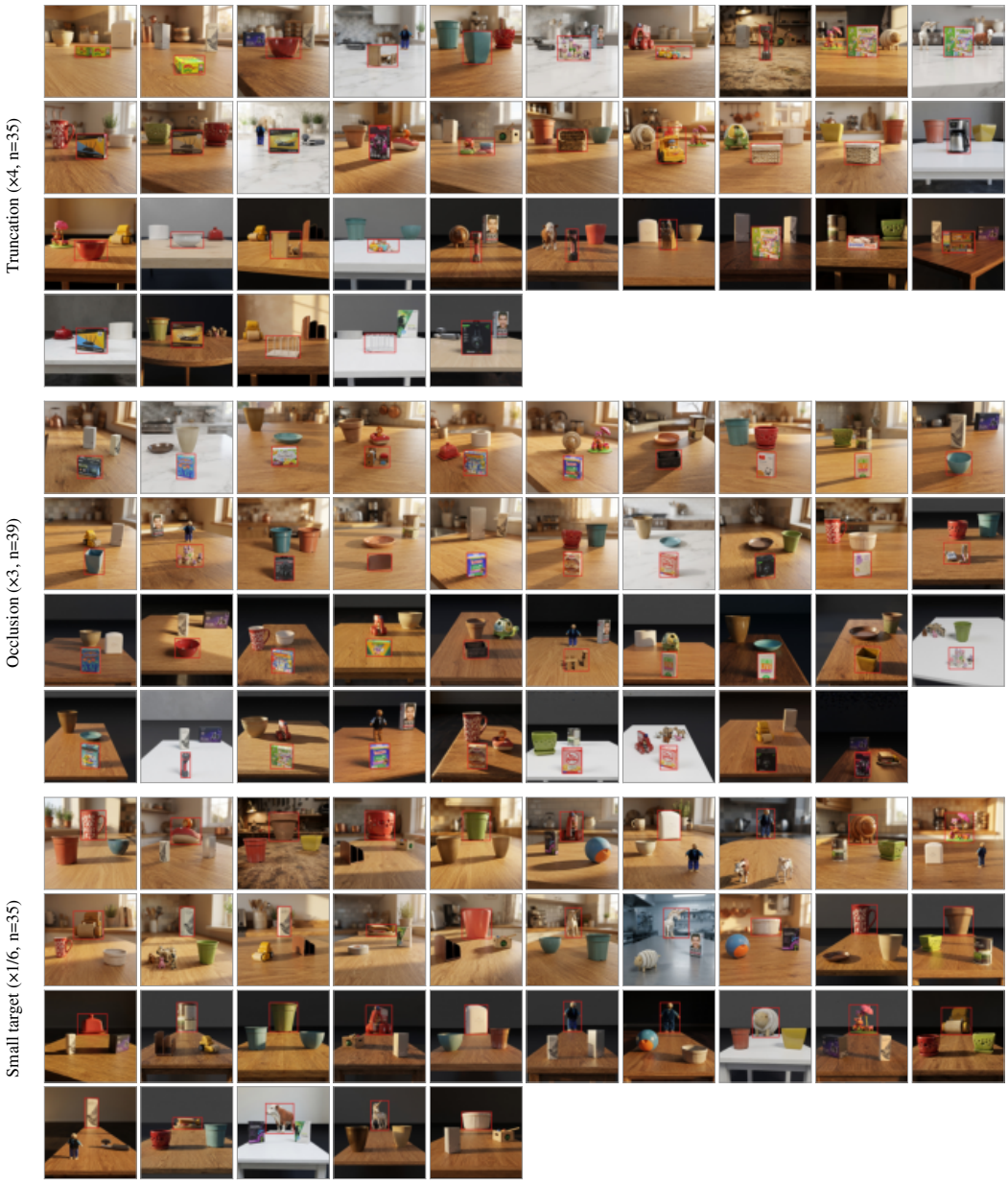

Figure 12: **All 109 accepted photorealistic stress-test inputs.** Grouping and relative order follow Figure 11, with seven unaccepted counterparts omitted. Red boxes mark the targets. The images are generated counterparts, not captured photographs.

E.6 Controlled Editing and Training Ablations

E.6.1 ObjectMover Protocol and Metrics

Of 200 public requests, 12 are used for request-encoding development and 188 are reported, grouped into 175 scenes. On September 14, the encoding was developed on the 12 requests at 04:56–05:30 UTC; the main protocol was frozen at 05:35 UTC, before the remaining destination photographs or archived outputs were downloaded. The already-trained V7 actuator was evaluated once on the 188 main requests at 06:55–06:58 UTC, after freezing weights, inference settings, and encoding. Subsequent baselines and ablations reused this split without changing V7; this reuse does not turn each later analysis into a new independent confirmation. A 186-request sensitivity subset excludes scene overlap with development. Repeated V7 score calculations agree within 5×10^{-5}. Each box request specifies apparent scale $a = \sqrt{|B_{\mathrm{dst}}|/|B_{\mathrm{src}}|}$ and a destination mask centroid obtained by applying the same box-centered similarity transform. Coordinates are measured on a 512 grid and mapped to the 48×48 token grid. PhyScale uses the full 768 frame, seed 42, 40 steps, guidance one, and RoPE control for 20 steps, without the observer, planner, local crop, or paste-back.

Target DINO is the DINO-v1 ViT-S/16 CLS cosine between tight target-box crops of output and destination photograph. Identity DINO instead compares the output target with the source-object crop. DreamSim uses the recorded 0.2.1 default ensemble; PSNR is measured on the 512 grid. These are similarity metrics, not placement-success rates or naturalness ratings. ObjectMover uses its public output archive, whose sampling conditions are unknown. PhyEdit uses its Qwen-Image-Edit-2511 LoRA with 28 steps and Depth-Anything-3 input. The plain Qwen baseline receives a gray-background reference crop and textual coordinates, with 40 steps and guidance four; its performance applies to that interface. The released ReScale actuator receives the same ground-truth move/resize transform with its VLM planner removed; Gemini removes the source object from the conditioning background before editing. Figure 7 shows a three-row same-request comparison; its panels are illustrative rather than an additional evaluation sample. Figure 16 extends this comparison with seven further displayed requests.

Table 21: **Additional object-editing metrics on the same 188 requests.** Target and full-image PSNR compare outputs with the destination photograph. Identity DINO is a proxy; these columns do not define task success or naturalness. Qwen and PhyEdit use the adapted interfaces described in the text.

Method	Target DINO ↑	DreamSim ↓	ID DINO ↑	Target PSNR ↑	Full PSNR ↑
PhyScale actuator (V7)	0.8987	0.1050	0.8564	16.2659	27.0417
ObjectMover (archive)	0.8672	0.1402	0.8133	15.8262	24.8347
PhyEdit (adapted)	0.7912	0.2156	0.7383	14.3457	21.8593
Qwen (crop + text)	0.4101	0.6223	0.3790	10.5613	14.5137
Unchanged source	0.4309	0.6450	0.3193	11.1302	20.4608

E.6.2 Same-Backbone Controls

The untrained RoPE control keeps the position channel and removes task adaptation. In the checkpoint scoring run (Table 24), its target DINO and DreamSim are 0.6338 and 0.3994; the main-table identity DINO is 0.561. The matched training increment improves target DINO by $+0.265$ $[0.229, 0.302]$ (Table 23). These controls do not form a complete matched 2×2 training-by-RoPE design: the plain crop/text interface has different input and guidance settings. Section E.6.4 reports a same-data text-box/RoPE retraining control with corrected per-sample visual inputs and matched training hardware.

The structured Qwen baseline adds blockwise depth ControlNet conditioning, a target-only hole dilated by eight pixels, a Gemini-cleaned background, and ReScale-style zoom crops. Its LoRA variant trains on 3,462 locally reconstructed pairs from the eight source datasets listed in ReScale, using rank 16, 3,000 optimizer steps, and effective batch eight. Its 24,000 sample exposures are close to V7's 24,680 exposures, not an equal number of optimizer updates. Crucially, the reconstructed pairs never change object size and are not a verified reproduction of ReScale's original data. Results test these specific reconstructed training and input configurations, not the intrinsic superiority of one recipe. Structured inference uses 30 steps and seed 42.

Table 22: **Same-backbone structured controls.** The three paired DINO contrasts use the 175-scene grouping and their own Holm-adjusted family. The V7 DreamSim value is retained at the precision recorded in this experiment.

Qwen-Image-Edit-2511 configuration	Target DINO ↑	DreamSim ↓	Identity DINO ↑
Structured, untrained	0.6957	0.3256	0.6530
Structured, reconstructed-data LoRA	0.6393	0.3908	0.5857
PhyScale V7	0.8987	0.1051	0.8564

V7 minus structured-untrained target DINO is 0.203 $[0.169, 0.238]$; V7 minus structured-LoRA is 0.259 $[0.223, 0.297]$. The reconstructed-data LoRA changes DINO by -0.056 $[-0.090, -0.023]$ relative to its untrained counterpart. These contrasts have Holm-adjusted p between 0.0003 and 0.0006 in their specified family.

Table 23: **Target-DINO contrasts from the individual actuator experiments.** Higher target DINO is better. The mask contrast averages two-pixel erosion and dilation variants. Intervals are copied from their respective experiments and are not jointly multiplicity-adjusted.

Contrast	Difference	Reported CI	Requests
V7 − ObjectMover archive	$+0.031484$	$[0.022327, 0.040418]$	188
V7 − PhyEdit	$+0.107437$	$[0.083411, 0.133518]$	188
V7 − Qwen (crop + text)	$+0.488566$	$[0.456006, 0.521376]$	188
V7 increment on − off	$+0.264674$	$[0.228537, 0.301586]$	188
Mean of seeds 43, 44 − archive	$+0.027493$	$[0.018120, 0.036328]$	188
Mean of mask perturbations − reference	-0.005470	$[-0.011716, -0.001038]$	188

E.6.3 Checkpoints and Supervision

Table 24: **Checkpoint comparisons.** Exposure counts are samples, not optimizer updates. Values come from this experiment's own scoring run and can differ from other tables in the fourth decimal.

Variant	Training rows	Exposures	Target DINO ↑	DreamSim ↓
V7 full	2,468	24,680	0.8985	0.1050
V7 half	2,468	12,340	0.8955	0.1095
Small-data, 3D targets	783	7,830	0.8713	0.1375
Small-data, 2.5D targets	783	7,830	0.8684	0.1294
RoPE only, untrained	—	—	0.6338	0.3994

Full versus half V7 training gives a target-DINO difference of 0.0030 $[-0.0007, 0.0070]$ with Holm-adjusted $p = 0.232$. V7 exceeds the older small-data model by 0.0272 $[0.0134, 0.0434]$, but this is not a pure data-volume ablation: the asset pool and command composition also change. On the older matched 783-row data, 3D versus 2.5D supervision gives 0.0030 $[-0.0140, 0.0189]$, with adjusted $p = 0.718$. Neither null contrast establishes equivalence.

E.6.4 Textual versus Positional Commands

Matched retraining. To compare command interfaces independently of backbone and data choice, we retrain a RoPE model and a text-box model from the same Qwen-Image-Edit-2511 checkpoint. Both use the same 2,468 rendered pairs, row order, 24,680 sample exposures, rank-16 LoRA, optimizer, and seed, on five B300 GPUs. RoPE uses the source-target positional transformation in Eq. (9). The text-box model instead receives source and destination coordinates in its prompt: "Move the {target} from box [x0,y0,x1,y1] to box [...] (0–1000 coordinates), resizing it to fit the new box. Keep ... unchanged." These boxes come from the same prescribed similarity transform. Each model's auxiliary visual encoder receives the source image of its current training pair.

Input-alignment audit. The original trainer cached prompt embeddings by asset pair, so the auxiliary visual encoder could reuse a source image from another camera or lighting configuration of that pair. This affected the frozen V7 model, both small-data supervision models, and the original text-box control. Replaying all 24,680 exposures in training order gives 17,976 mismatches (72.8%) under the old cache rule and zero after correction; the new training logs also report zero. This

repair does not retroactively change the original model's training history. The frozen V7 remains the main-results model, and its repaired RoPE retraining is a separate control.

ObjectMover evaluation. We use the same 188 ObjectMover-A requests, frozen ObjectMover scorer, inference stack, and common geometric-association/failure-counting rules. The repaired RoPE and text-box models, frozen V7, and original text-box model are evaluated in one container. The protocol was frozen before complete outputs were available; paired intervals use 10,000 scene-group bootstrap samples. Table 25 reports the repaired comparison and the main model as reference.

Table 25: **Command interfaces with correctly aligned training inputs.** Same-backbone, same-data retraining, evaluated on 188 ObjectMover requests. The frozen main model is a reference, not one of the two repaired training arms. Geometric values are per-model medians, not paired-effect estimates.

Metric	RoPE (aligned)	Text boxes (aligned)	Frozen V7
Target DINO ↑	0.8953	0.4388	0.8987
DreamSim ↓	0.108	0.620	0.105
Identity DINO ↑	0.856	0.365	0.856
Median destination error (pixels) ↓	2.4	216.5	2.4
Median absolute log-scale error ↓	0.026	0.118	0.025

The aligned RoPE-minus-text-box target-DINO difference is $+0.4565$ [0.4248, 0.4889]. DreamSim, identity, destination, and scale contrasts also favor RoPE after Holm adjustment. The aligned RoPE-minus-frozen-V7 difference is -0.003 [-0.011, 0.002], with no detected difference. This check supports the main model's measured performance under the tested repair; it does not establish equivalence or show that input alignment never matters.

Held-out rendered requests. To distinguish command following from transfer to photographs, we also evaluate 150 rendered requests with ground-truth destinations (Table 26). RoPE-minus-text-box differences are $+0.400$ [0.363, 0.437] for target DINO and $+0.712$ [0.666, 0.753] for silhouette IoU. The text-box model often fails to place the object at the destination even in this rendered domain, so real-image transfer cannot be the sole explanation for its ObjectMover gap. Excluding eight requests previously used for smoke testing preserves this conclusion.

Table 26: **Rendered command-following control.** Results on 150 requests, including destination-placement failures in the reported counts. The eight smoke-test requests are excluded in a separate sensitivity check.

Metric	RoPE (aligned)	Text boxes (aligned)	Frozen V7
Target-box DINO ↑	0.921	0.522	0.921
Silhouette IoU ↑	0.807	0.095	0.811
Object placed at destination	149/150	67/150	149/150

Historical control and scope. The original text-box run had target DINO 0.4644, DreamSim 0.5975, and identity DINO 0.4025. Its comparison with frozen V7 gave target-DINO difference $+0.4342$ [0.4027, 0.4663], but shared the old cache behavior and used a different training environment. The repaired comparison above replaces that run as the evidence for the main-text retrained controls. Each repaired arm still has one training run and one prompt template. Text boxes and mask-conditioned RoPE do not supply identical target-selection information, so this is a command-interface comparison, not a pure causal estimate of the RoPE mathematical operation or evidence that text-based control cannot learn.

E.6.5 Held-Out 3D versus 2.5D Supervision Tests

The original 783-row matched comparison does not separate 3D and 2.5D supervision on ObjectMover. To test scene effects more directly, we evaluate the same two small-data (783-row) training arms on 150 newly rendered requests with true destination images. Both arms use 783 training rows and 7,830 exposures and differ in whether the training destination is a full 3D re-render or an analytic 2.5D composite. The 2.5D target warps and composites the source object rather than re-rendering the full scene. Both small-data models retain the original auxiliary-visual-input cache described in Section E.6.4; the matched comparison controls this shared behavior, but does not test its interaction

with supervision type. V7 is a reference arm with a larger and differently composed training set, not a matched supervision ablation.

The new scenes use the frozen V7 rendering engine, with camera and command ranges restricted to the small-data training distribution. Their assets are disjoint from all training sets used in this work and its development runs after removing 15 overlapping assets found in the earlier candidate holdout pool. Of 150 requests, 108 change depth and 42 change only size. Rendered overlap defines 40 target-in-front, 46 target-behind, and 64 unoccluded requests. The three arms receive identical source pixels, commands, prompts, and seeds, and each returns 150/150 outputs without technical failures. The protocol was frozen before generation; intervals use 10,000 asset-pair cluster-bootstrap draws.

Table 27: **Prespecified supervision tests on held-out renders.** Differences are 3D minus 2.5D; occlusion differences are percentage points. The primary metric and three secondary metrics use the reported Holm-adjusted family.

Measure	3D	2.5D	Difference	95% CI	Holm p
Target DINO, primary ($n = 150$) ↑	0.895	0.886	$+0.009$	$[-0.007, 0.023]$	0.72
Occlusion order ($n = 86$) ↑	86.0%	81.4%	$+4.7$	$[-5.4, 14.9]$	0.79
Depth-change DINO ($n = 108$) ↑	0.891	0.882	$+0.009$	$[-0.013, 0.028]$	0.79
Silhouette IoU ($n = 150$) ↑	0.725	0.764	-0.039	$[-0.072, -0.010]$	0.026

Occlusion order compares the overlap region with the true destination and a counterfactual rendering with reversed front/back order. Neither the primary appearance metric nor the prespecified order metric detects a difference; silhouette IoU favors 2.5D (Table 27). The 3D arm misses the destination on 6/150 requests, versus 2/150 for 2.5D and 1/150 for V7. On the 144 requests where all arms place the target correctly, the post-hoc IoU difference is -0.022 without a reported interval; this conditional diagnostic does not replace the prespecified all-request result.

Exploratory occlusion and contact diagnostics. The target-behind subset favors 3D in correct order (93.5% versus 80.4%; $+13.0$ $[2.2, 25.0]$ percentage points), target DINO ($+0.031$ $[0.012, 0.048]$), and overlap-region PSNR ($+3.8$ $[2.3, 5.5]$ dB). The target-in-front subset does not separate. These unadjusted, post-hoc subgroup contrasts suggest a benefit for rendering objects behind occluders but do not establish a prespecified subgroup result. Across all requests, a contact-shadow-ring PSNR diagnostic favors 3D by 1.16 $[0.52, 1.79]$ dB. V7 exceeds the small 3D and 2.5D arms in target DINO by 0.026 $[0.013, 0.042]$ and 0.035 $[0.027, 0.044]$, respectively, and reaches 93.0% correct order; its data and composition differences preclude attributing these gains to supervision alone.

Exploratory shadow fidelity. Because target-box appearance metrics largely exclude ground shadows, we add pixel measurements against the true 3D render and a companion render with the moved object absent. Darkening relative to that companion defines shadow regions. We evaluate both the new target location and residual shadow at the old location. This add-on protocol was frozen before its computations but after inspecting the primary results; it is exploratory. Table 28 reports the recorded errors and intervals. Shadow depth is relative to the true-render darkening (one denotes matching depth), not the fraction of correct shadows. The same qualitative conclusion holds on the 144 requests where all arms place the target correctly.

Table 28: **Exploratory shadow measurements.** Lower errors are better; shadow depth should approach one. Differences are 3D minus 2.5D. V7 is a nonmatched reference. Shadow-error p values use the add-on's reported Holm adjustment, not the primary-test family.

Measure	3D	2.5D	V7	Difference [95% CI]	Holm p
Shadow luminance error ($n = 147$)	0.261	0.373	0.214	-0.112 $[-0.141, -0.086]$	0.0003
New-location shadow error ($n = 137$)	0.384	0.564	0.316	-0.180 $[-0.229, -0.136]$	0.0003
Old-location shadow residue ($n = 132$)	0.078	0.100	0.064	-0.022 $[-0.034, -0.008]$	0.0042
Shadow depth (true render = 1)	0.61	0.06	0.88	$+0.54$ $[0.49, 0.60]$	—

Exploratory model preference. Gemini-3.1-Pro-preview provides a separate offline two-alternative judgment with method names hidden and both left/right orders queried. The 150 requests yield 300 calls, not 300 independent scenes. There are 259 judgments for 3D, six ties, and 35 for 2.5D. Scoring a tie as one half gives an overall 3D preference of 87.3% with reported interval $[83.0, 91.1]$; the

strict win proportion is 86.3%. The shadow preference is 86.7%, whereas occlusion and lighting preferences are 54.9% and 57.7%, with many ties. A further 300-call true-render-versus-output check prefers the true render at 80.3% $[75.8, 84.7]$ on the same tie-aware scale. This is a limited judge sanity check, not human validation or evidence of unbiased physical reasoning. These model preferences remain exploratory and are reported separately from the human studies.

The real-image examples in Figs. 8 and 18 illustrate residual shadows, source-object ghosts, and surrounding scene changes, whereas Tables 27–28 quantify the separate rendered test. Each training arm has one run and the controlled scenes contain two objects, limiting conclusions about training variability and more complex interactions.

User study on physical plausibility. We add an independent measurement of the same two small-data models by collecting blinded judgments on 90 of the existing 150 rendered requests. A fixed hash rule samples proportionally by occlusion class without consulting outputs or scores: 24 target-in-front, 28 target-behind, and 38 unoccluded requests. All three raters evaluate every request. The interface shows the source image, solid source and dashed destination boxes, and the two outputs in randomized left/right order, without method names or ground-truth destination renders. Two identical-image checks and two left/right-swapped repeats accompany the trials. All three raters pass the identical-image checks; none is excluded.

Raters answer three questions: whether each output completes the request (Q1); which output is more physically plausible, considering new contact and shadows, old-location remnants, and occlusion (Q2); and which has more plausible shadows, with a not-applicable option (Q3). Q2 is the sole primary metric. The protocol was frozen before the interface was built; intervals resample asset pairs 10,000 times. Preferences assign one to 3D, zero to 2.5D, and one half to a tie. Table 29 reports the study; Fleiss' κ for Q2 is 0.64.

Table 29: **User study comparing 3D and 2.5D supervision.** Three raters assess 90 rendered requests. Preference values favor 3D and count ties as one half; 50% is neutral. The completion contrast is 3D minus 2.5D in percentage points.

Measure	Estimate	95% CI
Physical-plausibility preference, Q2 (primary)	83.5%	[76.6, 89.8]
Shadow preference, Q3	85.5%	[78.8, 91.5]
Q2 on the 84 requests both methods completed	88.9%	[84.2, 93.3]
Strict completion: 3D / 2.5D	93.3% / 98.9%	—
Strict-completion difference	-5.6	$[-10.5, -1.3]$

The 270 primary judgments comprise 206 preferences for 3D, 39 ties, and 25 for 2.5D. Q3 has 14 not-applicable judgments out of 270; its reported preference excludes them. Exploratory Q2 preferences are 84.7% for target-in-front, 80.4% for target-behind, and 85.1% for unoccluded requests; each reported interval excludes 50%. The 84-request subset requires both outputs to be judged complete by a majority of raters and is a conditional sensitivity analysis, not a replacement primary population. The results support a scene-effect benefit with lower request completion, not a uniform improvement in editing. This study adds a new measurement method on reused rendered outputs, not an independent test set or a direct human evaluation of transfer to photographs.

E.6.6 Training Dynamics

Figure 13 evaluates checkpoints by sample exposures, the common unit across the two training scripts. V7 learns from true 3D move/resize pairs; the ReScale-style Qwen adaptation learns from the reconstructed insertion data described above, whose objects do not change size. Thus these are recipe-level trajectories, not a same-data comparison of training efficiency or a trajectory of the released ReScale actuator. Both curves use the reused 188-request ObjectMover split. Bootstrap bands quantify evaluation-sample uncertainty, not variability over independent training runs.

V7 reaches target DINO 0.8788 after 2,000 exposures and then remains near 0.89–0.90 over the measured range. Its retrained endpoint differs from the original frozen V7 by -0.0025 $[-0.00733, 0.00085]$, a no-detected-difference result rather than an equivalence test. The endpoint difference against the reconstructed ReScale-style recipe is $+0.216$ $[0.18084, 0.25282]$. Against that recipe's highest tested checkpoint, selected post hoc in its favor, the difference is

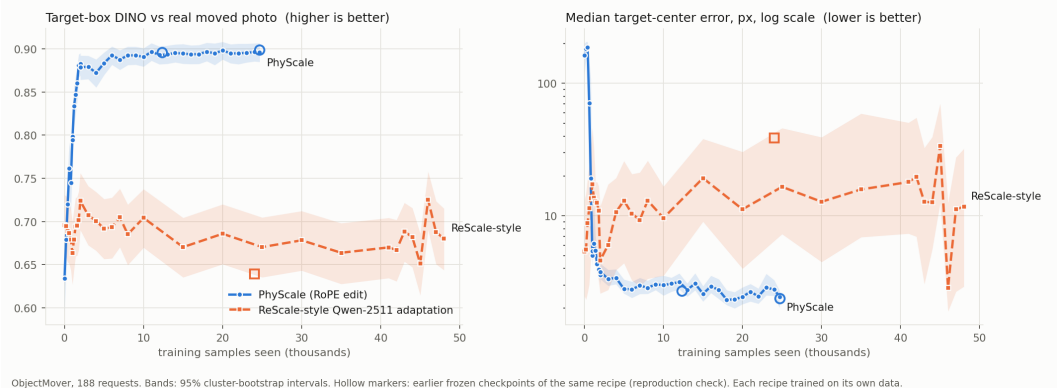

Figure 13: **Training dynamics on ObjectMover.** Left: target DINO against real destination photographs (↑). Right: median target-center error in pixels on a logarithmic axis (↓). The horizontal axis is training sample exposures in thousands. Shaded bands are 95% cluster-bootstrap intervals over evaluation samples. Hollow markers are retained checkpoint references, not additional independent seeds. The two recipes use different training data; the orange curve is the reconstructed ReScale-style adaptation, not the released ReScale actuator.

Table 30: **Selected training-curve checkpoints.** Sample exposures are not unique training pairs or optimizer updates. The unchanged frozen V7 remains the main-table model; the retrained endpoint is a separate run.

Recipe / checkpoint	Exposures	Target DINO	95% CI	Center error (px)
V7, untrained RoPE	0	0.6340	[0.5959, 0.6706]	163.005
V7, early	1,000	0.7946	[0.7643, 0.8230]	5.410
V7, early	2,000	0.8788	[0.8642, 0.8922]	3.749
V7, intermediate	12,000	0.8928	[0.8821, 0.9028]	3.145
V7, retrained endpoint	24,680	0.8962	[0.8849, 0.9066]	2.439
V7, original frozen model	24,680	0.8987	[0.8882, 0.9084]	2.357
ReScale-style, untrained	0	0.6957	[0.6605, 0.7306]	5.363
ReScale-style, early	2,000	0.7237	[0.6900, 0.7560]	4.589
ReScale-style, highest tested DINO	46,000	0.7250	[0.6904, 0.7576]	2.831
ReScale-style, endpoint	48,000	0.6801	[0.6437, 0.7150]	11.645

$+0.171$ $[0.13887, 0.20526]$. These curves show no sustained improvement from simply extending that reconstructed recipe's exposure budget; they do not establish its best achievable performance. Intermediate ReScale-style checkpoints were sampled sparsely after the early stage, with 26 intermediate checkpoints not evaluated.

E.6.7 Command and Geometric Diagnostics

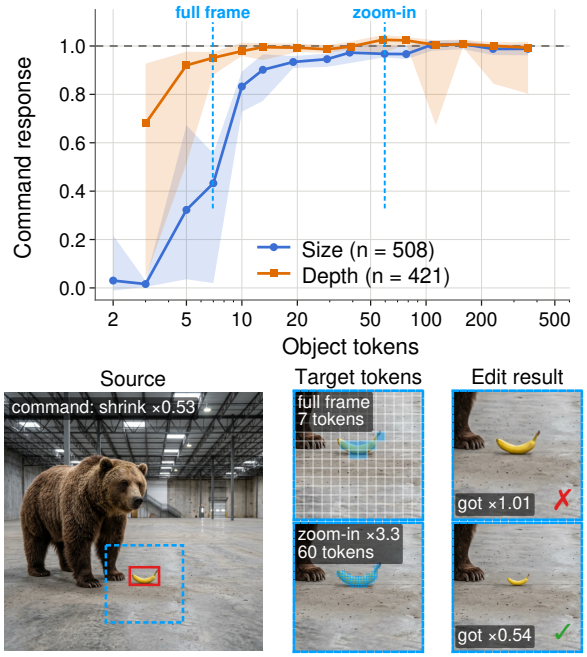

Figure 14: **Target-token support and local crop enlargement.** Top: size and depth curves report the fraction of the requested command executed, not a binary success rate; unit response follows the request. Bottom: a GenScale target requested at $0.53\times$ has seven command-carrying tokens in the full frame and remains at $1.01\times$. A $3.3\times$ enlarged crop provides 60 tokens and reaches $0.54\times$. Dashed lines on the curves mark these support levels. Points are medians within logarithmic token bins containing at least 12 rows; bands are 95% row-bootstrap intervals of the median; n counts rows.

The controlled size–depth response test contains 24 groups with nine command combinations. Estimated own-command responses are 0.910 $[0.798, 0.991]$ for size and 1.014 $[1.005, 1.030]$ for depth; cross-responses include zero. Because a two-dimensional mask baseline can score well on this readout, the test establishes projected command following rather than independent recovery of physical 3D state. Figure 14 likewise reports command response, not binary success. Low token support attenuates size response; enlarged local crops increase the number of command-carrying tokens and recover the requested change in the illustrated example.

Geometric diagnostics use automatic association and a separate post-hoc identity layer in which an AI assistant resolved ambiguous instance identities from mask overlays. Neither is human geometric ground truth; admissions differ by method. Under the resolved association, the paired position difference against the ObjectMover archive is -2.5783 $[-4.5468, 0.0998]$ pixels, whereas the absolute log-scale difference is -0.0206 $[-0.0322, -0.0086]$. Per-method medians must not be subtracted to infer these paired effects. Seed and mask-perturbation contrasts are retained in Table 23.

Table 31: **Automatic and AI-resolved geometric diagnostics.** Position is measured on a 512-coordinate grid. Each entry is a per-method median conditional on its admitted rows; denominators differ, and these medians must not be subtracted to infer paired mean differences. Neither association layer is human geometric ground truth.

Method	Admitted / designed	Position (px) ↓	$\lvert \log \text{scale} \rvert$ ↓	Mask IoU ↑
Automatic association				
PhyScale actuator	188 / 188	2.3572	0.0251	0.8919
ObjectMover (archive)	188 / 188	5.4315	0.0466	0.7739
PhyEdit (adapted)	188 / 188	11.6666	0.0513	0.6682
Qwen (crop + text)	188 / 188	141.0741	0.2629	0.0000
Unchanged source	188 / 188	229.4228	0.0613	0.0000
Post-hoc AI-resolved identity association				
PhyScale actuator	184 / 188	2.3942	0.0237	0.8919
ObjectMover (archive)	183 / 188	5.3158	0.0467	0.7765
PhyEdit (adapted)	174 / 188	10.4558	0.0479	0.6934
Qwen (crop + text)	185 / 188	139.8362	0.2218	0.0000
Unchanged source	187 / 188	226.6463	0.0600	0.0000

E.7 Additional Component Contrasts

Table 32 reports matched component contrasts. Each row belongs to its own experiment, metric, denominator, and interval construction; the table is not a pooled analysis. Full placement versus in-place scaling and predicted-pose versus upright observation do not show a detected downstream-size difference on the development sources. In the placement study, 107/197 source outputs are bit-identical between full planning and in-place scaling because the planner chooses no displacement. This limits the subset's ability to distinguish placement policies and motivates the stress tests; it does not establish equivalence of the policies. The naturalness row is an offline model rating, not a human study.

Table 32: **Matched component contrasts.** Differences are first variant minus second variant. Size gain is the reduction in source-weighted size error. The training comparison uses 188 editing requests; the two size comparisons each use 197 development sources in separate experiments. Naturalness is an offline model rating over 174 scorable sources, not human evaluation. The reported confidence intervals are experiment-specific and are not jointly adjusted across rows.

Comparison	Measure	N	Difference	CI
Training increment: on − off	Target DINO ↑	188	$+0.2647$	$[0.2285, 0.3016]$
Full placement − in-place	Size gain ↑	197	$+0.0381$	$[-0.0212, 0.1035]$
Upright − predicted pose	Size gain ↑	197	-0.0372	$[-0.1122, 0.0402]$
Full placement − in-place	Naturalness ↑	174	-0.0465	$[-0.1346, 0.0382]$

Table 33: **Placement-strategy size and quality results.** Size error and gain use 197 development sources and 614 relations, with source weighting and a single offline judge vote. Identity, naturalness, and MUSIQ come from the separate 176-action-source, 316-target quality evaluation. N_{nat} counts sources contributing scorable naturalness ratings; absent or replaced targets are excluded. The columns have different estimands and denominators and must not be combined into a success score. MUSIQ measures whole-image quality, not physical integration.

Strategy	Error ↓	Gain ↑	Identity ↑	Naturalness ↑	N_{nat}	MUSIQ ↑
Unchanged source	0.9653	0.0000	3.8627	3.9025	176	73.3206
Full placement	0.4738	0.4915	3.3799	3.7112	174	71.7621
Depth only	0.4679	0.4975	3.4154	3.7297	175	71.6814
Lateral only	0.4924	0.4729	3.3676	3.6768	173	71.6814
In-place scaling	0.5118	0.4535	3.4408	3.7577	174	71.7214
No token penalty	0.4746	0.4907	3.3913	3.7107	174	71.7631

The quality evaluation covers 176 action sources and 316 targets, but naturalness denominators depend on scorable targets. Absent or replaced targets are excluded from those means. Mean identity, naturalness, and whole-image MUSIQ are lower after editing than for unchanged sources; the full-versus-in-place naturalness contrast remains inconclusive. These results do not establish cost-free quality preservation.

Table 34: **Observer comparison with downstream modules fixed.** Size results use 197 development sources and 614 relations. The interval is for improvement relative to the unchanged source in this experiment, not a pairwise observer contrast. Its scoring run differs from that of Table 33; the baseline errors must not be merged. The U-minus-C contrast is reported separately in Table 32.

Observer	Error ↓	Improvement ↑	Improvement CI
Unchanged source	0.915398	0.000000	—
Predicted-pose observer (C)	0.486464	0.428934	$[0.309773, 0.550002]$
Upright observer (U)	0.523689	0.391709	$[0.271569, 0.515077]$

Table 35: **Auxiliary planning and observer contrasts.** Higher plan-mask IoU and lower response error are better. N follows the analysis unit of each experiment rather than denoting a common pool. C is the deployed observer, U its upright simplification, and SAM denotes the SAM 3D + MoGe-3 observer of Appendix E.4.1. The plan-IoU variants use different self-selected plans.

Experiment and contrast	Metric	N	Difference	Reported CI
Full placement − in-place	Plan-mask IoU	48	$+0.053727$	$[0.027591, 0.079981]$
SAM − C	Response error	84	$+0.000553$	$[-0.021445, 0.018685]$
C − SAM	Response error	24	$+0.009537$	$[-0.022162, 0.041209]$
U − C	Response error	19	$+0.005899$	$[-0.015835, 0.026790]$

E.8 User Study on Editing Quality

The original snapshot contains six completed questionnaires, 180 paired judgments, 360 single-image rating opportunities, and 50 unique sources. Identity and naturalness use 0–4 scales. Methods cover different source sets, with only ten sources shared across the original four methods. Two questionnaires fail both attention checks but remain in the all-six snapshot; the original admission rule retains four. Table 7 therefore retains descriptive pooled means, while the within-question paired analysis below compares each output only with the opponent shown for the same source. These paired tests do not require every opponent to share an identical overall source set. Figure 19 shows the editing-study interface, with the source request, two unnamed outputs, and separate preference, identity, and naturalness questions.

Table 36: **Coverage and denominators for the all-six human snapshot.** Means pool available ratings on the 0–4 scales. There are 180 paired judgments and 360 single-image rating opportunities; one Qwen naturalness rating is unavailable. Counts do not denote independent images or participants. Only ten sources are shared by all four methods.

Method	Identity	Naturalness	N_{id}	N_{nat}	Sources
ObjectMover (archive)	3.766667	3.633333	60	60	30
PhyEdit	2.200000	2.416667	60	60	30
Qwen (crop + text)	3.233333	1.627119	60	59	30
PhyScale	3.861111	3.727778	180	180	50

Within-question paired comparisons. Each questionnaire item presents our output and one opponent's output for the same source. We compute within-item differences in identity and naturalness and preference with wins/ties/losses scored as $1/0.5/0$. Intervals use 10,000 source-bootstrap draws with seed 20260924. Each opponent contributes 60 paired items across 30 sources in the all-six analysis. Table 37 reports these paired comparisons rather than differences between unmatched pooled means. All three metrics favor our outputs over PhyEdit and Qwen, but ObjectMover does not separate. The limited participant pool remains a restriction: source resampling does not measure uncertainty over a new population of raters.

Registered-inclusion sensitivity. The original rule excludes questionnaires that fail both attention checks. Two of six questionnaires meet this exclusion criterion; among the remaining four, one passes both checks and three pass one. Applying the rule leaves 40 paired items per opponent across the same 30 sources. The direction and interval-based conclusions in Table 37 are unchanged. In the all-six analysis, ours/tie/opponent counts are 15/34/11 against ObjectMover, 41/12/7 against PhyEdit, and 57/3/0 against Qwen. For the admitted four they are 12/19/9, 26/8/6, and 38/2/0, respectively. The four-questionnaire pooled identity/naturalness means are 3.875/3.683 for ours, 3.700/3.525 for

Table 37: **Paired human editing results.** Identity and naturalness differences are ours minus opponent, with 95% source-bootstrap intervals. Preference gives ties half credit. The all-six analysis supplies the main descriptive snapshot; the four-questionnaire analysis applies the originally registered exclusion rule. ReScale has no item-level record and is not included.

Opponent	Identity difference	Naturalness difference	Preference (%)
Six completed questionnaires: 60 paired items per opponent			
ObjectMover	$0.00\ [-0.48, 0.29]$	$-0.03\ [-0.40, 0.28]$	$53.3\ [43.6, 62.9]$
PhyEdit	$+1.65\ [0.96, 2.29]$	$+1.22\ [0.63, 1.79]$	$78.3\ [65.2, 90.0]$
Qwen	$+0.73\ [0.41, 1.06]$	$+2.32\ [1.80, 2.82]$	$97.5\ [94.4, 100]$
Four admitted questionnaires: 40 paired items per opponent			
ObjectMover	$+0.10\ [-0.39, 0.39]$	$+0.08\ [-0.28, 0.40]$	$53.8\ [41.7, 65.8]$
PhyEdit	$+1.65\ [0.90, 2.37]$	$+1.10\ [0.45, 1.76]$	$75.0\ [60.5, 88.2]$
Qwen	$+0.58\ [0.28, 0.92]$	$+2.35\ [1.84, 2.86]$	$97.5\ [93.4, 100]$

ObjectMover, 2.175/2.425 for PhyEdit, and 3.425/1.575 for Qwen. Retaining six questionnaires in the main descriptive table is distinct from following the original admission rule; the paired sensitivity is reported explicitly rather than treating that deviation as absent.

Separate ReScale aggregate. The ReScale supplement contains 60 identity and naturalness ratings but no item-level records. Its means are 3.600 and 3.100. Pairwise preferences are 32 for PhyScale, 11 for ReScale, and 17 ties. Completion labels exist for 50 outputs: 39 completed, eight partially completed, and three not completed, giving 78.0% fully completed. For the other methods, completion is the fraction of outputs labeled fully completed among those with a completion label: Qwen 1/50, PhyEdit 15/48, ObjectMover 37/50, and PhyScale 133/148. This aggregate appears as a separately marked column in the main human table; it is not merged into the six-questionnaire sample and cannot support a paired item-level test.

E.9 Pipeline Time and Cost

The cost comparison uses 30 GenScale sources and reports per-source sums of native stage runtimes, not a new simultaneous wall-clock rerun of every accelerated pipeline stage. Exact lazy token checking changes the planner implementation without changing the selected plan. The accelerated plans match the originally executed plans bit-for-bit on all 27 sources with recorded planning actions; the new per-source planner runtimes replace the old planner stage, while other measured stages are reused. Mean planner time decreases from 189.7 to 18.1 seconds, yielding the main table's summed-stage comparison. This is the deployed C1 configuration; the stress-test variant has not been timed end to end.

Table 38: **Recorded pipeline costs.** Means over 30 sources. Prices are estimates at the recorded unit rates, not invoices.

Pipeline	Summed stage time (s)	Estimated cost ($)
PhyScale, exhaustive token checks	224.3	0.039
PhyScale, exact lazy token checks	52.8	0.037
ReScale	98.0	0.696

The paired accelerated-PhyScale-minus-ReScale time difference is $-45.2\ [-68.1, -24.8]$ seconds; the cost difference is $-\$0.659\ [-0.86, -0.47]$. Recorded compute prices are \$0.000542 per L40S second, \$0.001261 per H200 second, \$0.0000131 per CPU core-second, and \$0.00000222 per GiB-second. Gemini-3.1-Pro input/output rates are \$2/\$12 per million tokens in this cost record; these are historical assumptions, not a claim about current prices. ReScale's mean API cost is \$0.554 per source, including \$0.425 for diagnosis and planning.

E.10 Earlier Stress Tests and Scoring Diagnostics

Before the rendered stress tests of Appendix E.5, a 154-scene synthetic comparison used a common V7 executor. Plan-level validity was 0.981 for PhyScale and 0.844 for Gemini; the Gemini-minus-PhyScale difference was $-0.136\ [-0.188, -0.078]$. However, our planner moved every target backward, with median contact-depth displacement 0.415 m, whereas Gemini often retained the

original placement. After execution, fidelity did not separate, and all invalid Gemini plans were executed as requested. Offline VLM judgments favored Gemini for size plausibility and overall preference but favored our outputs for naturalness. This disagreement motivated the later evaluation with actual renders and explicit layout gates rather than treating plan validity alone as a complete outcome.

A policy sweep that prioritized minimal depth motion reduced validity from approximately 0.98 to 0.56. A finite-relaxation setting selected on 42 development scenes still reduced held-out validity by $0.169\ [0.091, 0.248]$. These are development diagnostics, not evidence that smaller motion is always preferable or that the final stress-test planner was unchanged throughout the study.

The GenScale scoring audit also illustrates denominator sensitivity. Under the full 8,001-relation view, differences in unjudgeable or skipped outputs materially affect aggregate error. A separate 21-by-seven-cell reproduction of the earlier ReScale table covered 6,898 relations; restoring 496 post-edit unjudgeable relations with error two reduced the reported improvement from 0.353 to 0.259. This reconstructed view does not replace the unfiltered main comparison or its separately reported weighting sensitivities.

F Additional Visualizations and User-Study Interface

The following panels expand the main-text visual comparisons and document the user study on editing quality. They supplement the aggregate results and protocols in Appendix E; the displayed examples are not a separate quantitative evaluation.

F.1 GenScale Relative-Size Correction

Figure 15 expands Fig. 5 with a wider range of object pairs and size errors. The examples include undersized and oversized objects, as well as scenes with more than one incorrectly sized object. Comparing each result with its source makes both the relative-size change and preservation of the surrounding scene visible.

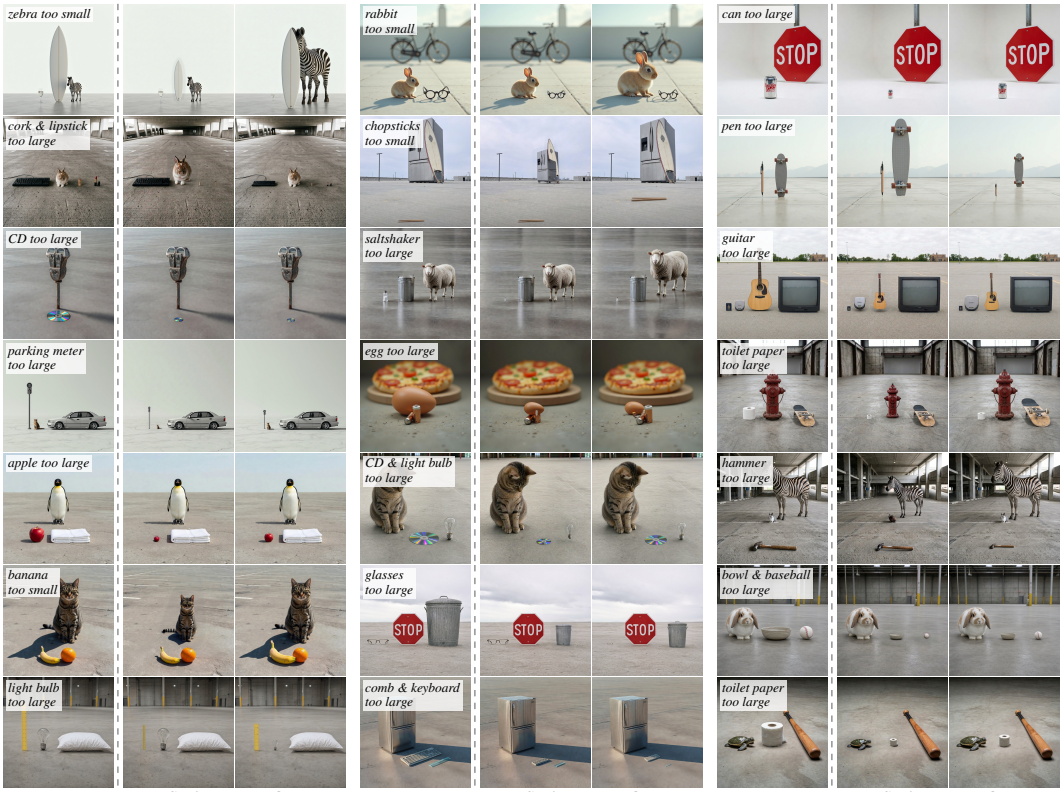

Figure 15: **Additional GenScale correction examples.** Twenty-one examples are arranged in three groups of input, ReScale, and PhyScale columns. Annotations on the inputs identify the original relative-size errors. Each triplet compares corrections of the same source; the aggregate comparison and scoring sensitivities are reported in Table 1 and Appendix E.2.

F.2 Prescribed Object Movement and Resizing

Figure 16 extends Fig. 7 with same-request comparisons across varied targets and scene materials. The shared source and destination boxes allow direct inspection of requested placement and size, target identity, source-location cleanup, and changes to neighboring objects. The observer and planner are not part of this comparison.

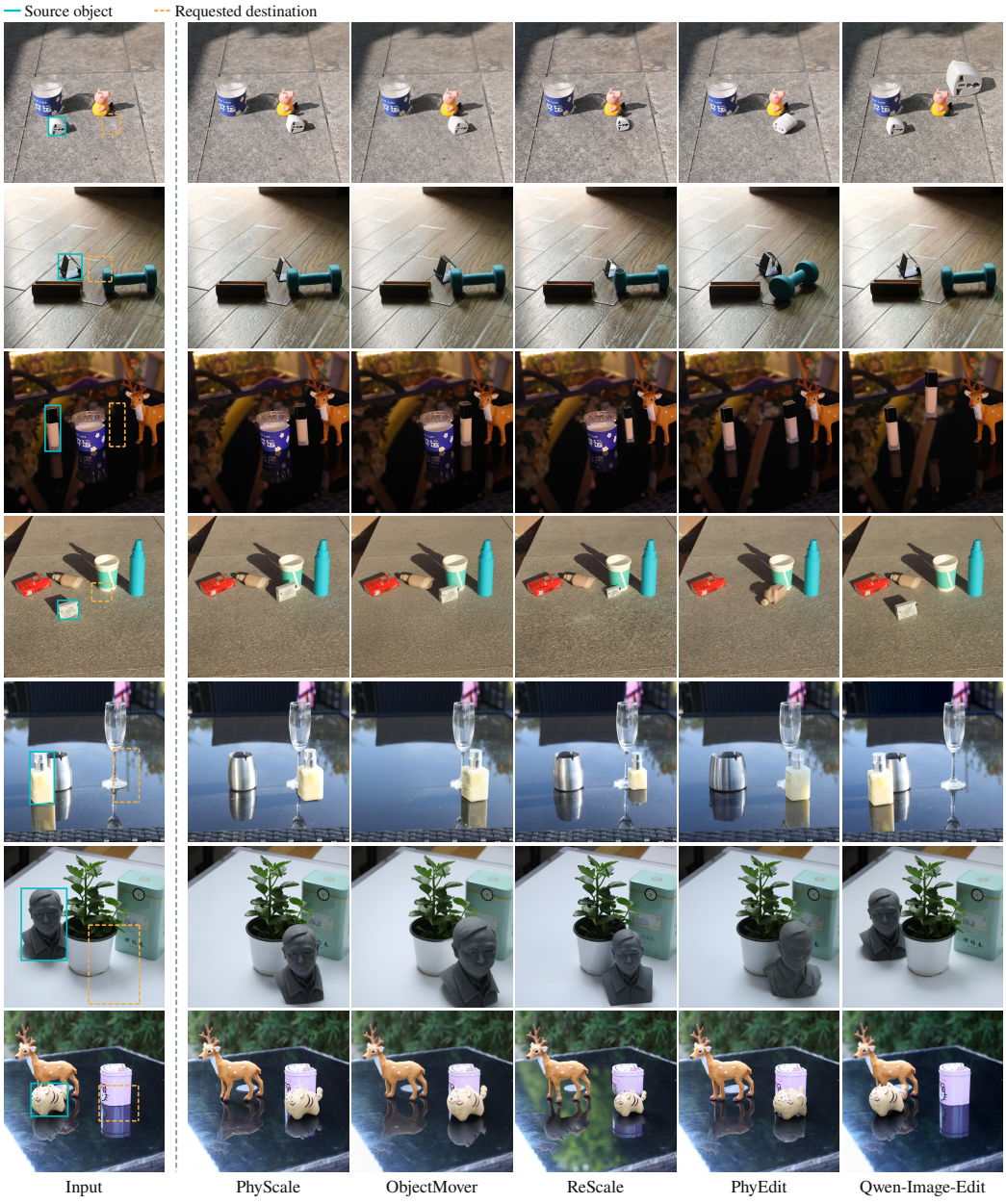

Figure 16: **Additional ObjectMover actuator comparisons.** Seven prescribed edits, with columns showing the input, PhyScale, ObjectMover, the ReScale actuator, PhyEdit, and Qwen-Image-Edit. Solid cyan boxes mark the source objects and dashed orange boxes mark the requested destinations. All methods receive the same geometric request through their respective interfaces; ReScale is evaluated without its planner. Quantitative results and input adaptations are given in Table 6 and Appendix E.6.

F.3 PHOTOREALISTIC PLANNER COMPARISONS

Figure 17 complements Fig. 6 with photorealistic stress scenes. Both planners receive the same fixed size correction, and their plans are executed by the same actuator. The displayed cases highlight placement differences after enlargement: the annotated Gemini outputs place targets beyond the support surface, against the image boundary, or partly outside the frame. These examples illustrate the support and framing criteria evaluated in Appendix E.5.

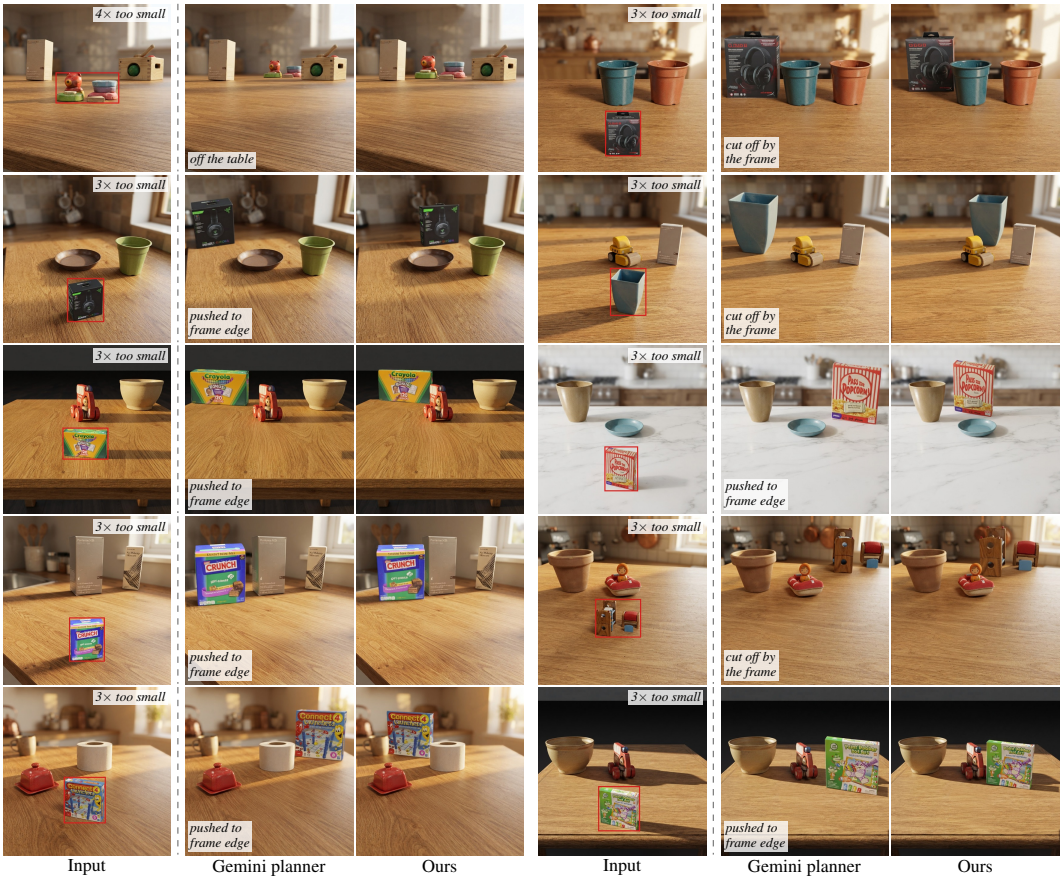

Figure 17: **Additional planner comparisons on photorealistic stress scenes.** Ten examples are arranged in two groups of input, Gemini-planner output, and PhyScale-planner output. Red boxes identify the targets, and input annotations indicate the original size errors. Output annotations highlight support or frame-boundary problems in the displayed cases. The photorealistic images are generated counterparts of the rendered stress scenes, not independently captured photographs; aggregate automatic and human placement results appear in Table 5.

F.4 3D VERSUS 2.5D SUPERVISION

Figure 18 expands Fig. 8 with real-image comparisons of the two supervision types. In addition to the edited object's appearance, the panels expose effects outside its destination box, including remnants at the original location and changes to shadows and reflections. These qualitative examples complement, but are distinct from, the held-out rendered measurements in Appendix E.6.5.

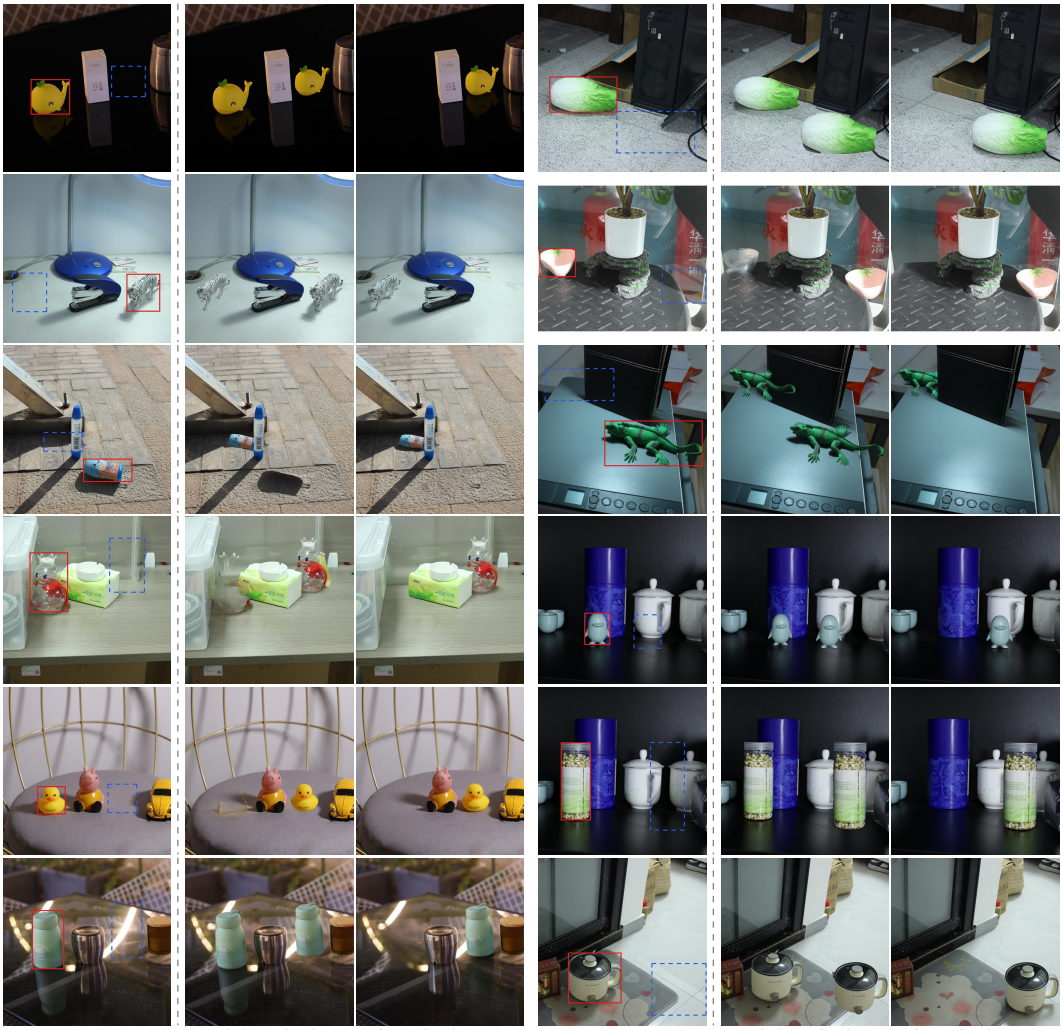

Figure 18: **Additional visual comparisons of 2.5D and 3D supervision.** Twelve displayed requests are arranged in two groups of input, 2.5D-trained output, and 3D-trained output. Red source boxes and dashed blue destination boxes specify the edits. The examples show why inspecting only the object at its new location can miss residual copies, old shadows, or inconsistent surrounding appearance. These photographic examples do not supply the ground-truth renders used for the separate quantitative tests in Tables 27 and 28.

F.5 User-Study Interface for Editing Quality

Figure 19 documents the interface for the user study on editing quality in Appendix E.8, distinct from the planner-only placement study. Raters see the original edit request alongside two outputs labeled only by their left/right position. Zoom and focus controls support inspection of the original and requested regions. The preference question considers request completion, identity, and scene naturalness; separate 0–4 scales then assess identity preservation and naturalness for each output.

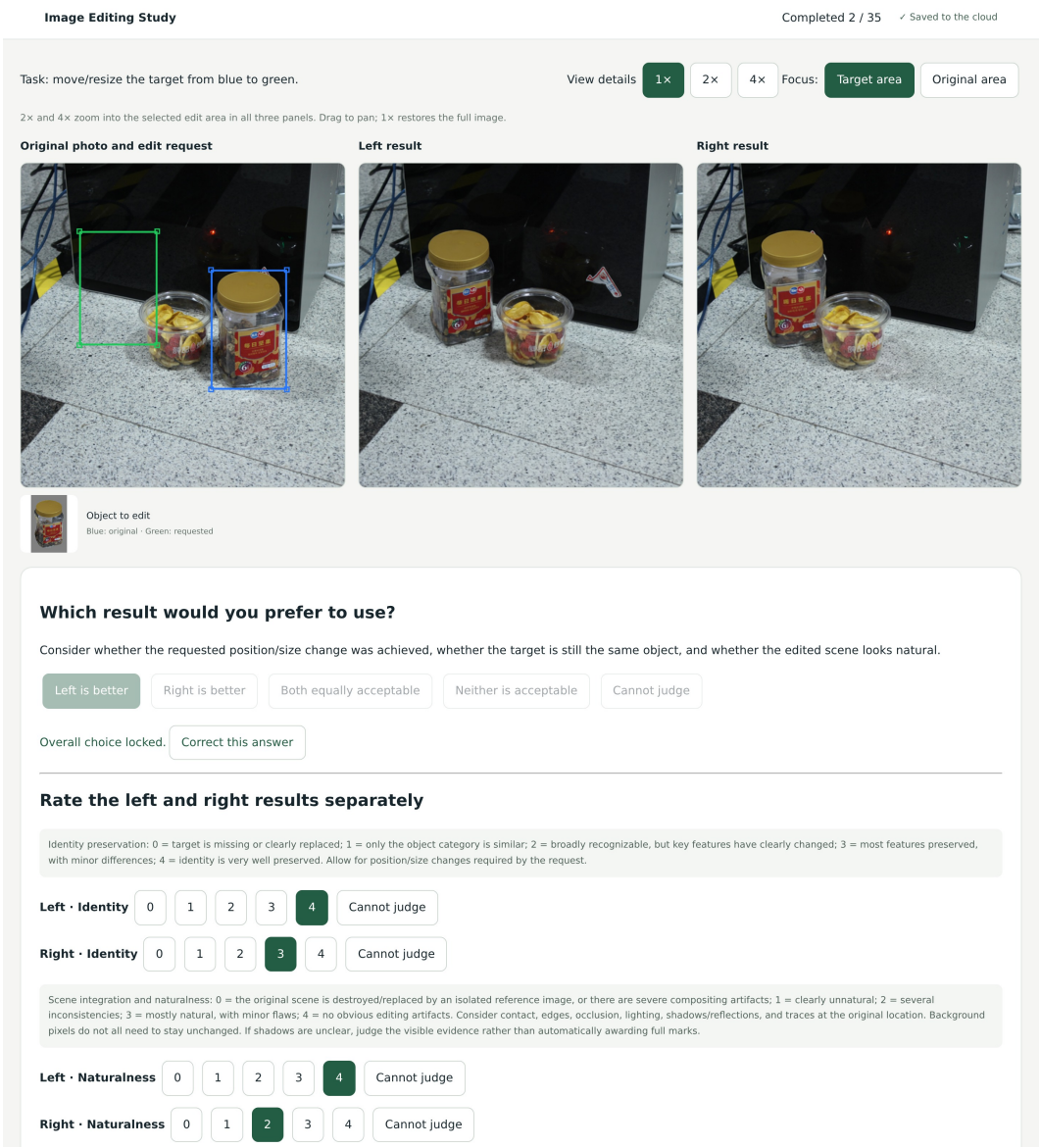

Figure 19: **Example trial from the editing-quality user study.** Blue and green boxes in the source panel mark the original object and requested destination, respectively. The two candidate outputs are shown without method names. Preference options include either result, both equally acceptable, neither acceptable, and inability to judge; identity and naturalness are rated separately for each output. The selected responses illustrate one interface state, not aggregate study outcomes. Coverage, inclusion rules, and paired results are reported in Appendix E.8.