GIF: Agentic Generation of Interactive and Functional Object Compositions for Robot Learning

Long Xu[*,1,3] **Zhiqi Zhang**[*,1,2] **Mi Yan**[*,1,2] **Shengliang Deng**[1,4] **Chong Xia**[1,5]
Mingyu Dong[1,5] **Jiayi Chen**[1,2] **Jiangran Lyu**[1,2] **Fei Gao**[1,3] **Zhizheng Zhang**[†,1,6] **He Wang**[†,1,6]

Figure 1: Overview of GIF. Cyan labels index generated task assets and link each standalone asset to its placement in the composed scene. Gray labels mark intra-asset components, emphasizing part-level functional contact rather than coarse whole-object arrangement. Right: generated tasks seed policy-training trajectories under randomized simulation (a) and the trained policy is evaluated on real hardware (b).

Abstract: Embodied foundation models for manipulation require scalable evaluation and data generation over diverse scenarios, with simulation serving as an important pathway. Automated scene generation offers a promising path, yet prior work has largely emphasized coarse-grained scene layouts rather than fine-grained functional object compositions. Motivated by this gap, we present GIF, an agentic **G**eneration framework for **I**nteractive and **F**unctional object compositions. In this framework, we recast this problem as disentangled reconstruction followed by relative pose recovery. CoGen produces instance-disentangled meshes with coarse initial poses leveraging complementary strengths of 2D and 3D generative models, GPRM refines the relative pose under joint geometric and physical guidance, and a VLM verifier selects the candidate that best matches the structured specification. We further construct a benchmark spanning eight representative contact-geometry classes and compare with state-of-the-art generators; GIF improves both asset quality and relation matching, while reducing collision rate to below 1%. Finally, we synthesize data for policy learning, revealing diversity scaling in both simulation and real-world deployment. Our code is available at https://github.com/GaoLon/gif.

1 Introduction

Recently, embodied foundation models [1] have shown increasingly general manipulation capabilities, necessitating scalable pipelines for robot evaluation and data generation. Simulation offers a

[*]Equal contribution. [†] denotes corresponding authors.
Correspondence to zhangzz@galbot.com, hewang@pku.edu.cn.
[1]Galbot, [2]Peking University, [3]Zhejiang University, [4]The University of Hong Kong,
[5]Tsinghua University, [6]Beijing Academy of Artificial Intelligence

low-cost, massively parallelizable alternative, but constructing simulation scenes has traditionally required substantial manual design by domain experts, motivating recent work on automated scene generation [2, 3, 4, 5, 6, 7, 8] that has made promising progress on diverse simulated environments.

Existing systems have advanced diverse and semantically plausible scene generation, particularly for navigation and scene-level reasoning, but they mainly focus on scene-scale layout, specifying approximate placements on tables, furniture, or within rooms. Manipulation instead requires object-level functional relations governed by local geometric compatibility, collision avoidance, and physical stability. Tasks such as hanging, insertion, capping, and containment demand precise relative 6-DoF poses aligned with human intention and physical constraints. Motivated by this gap, we study interactive and functional object composition generation: given a language or image prompt, the goal is to generate independently interactive objects together with relative poses that realize precise functional contact, avoid interpenetration, and remain stable in simulation.

While existing systems already yield diverse and high-fidelity 3D assets [9, 10], their strategies still face substantial challenges when recovering precise relative poses. Language-prompted methods [7, 4, 3] ask VLMs to emit positions or rotation angles yet lack precise local 3D geometry understanding, while visual-prompted methods [11, 8] rely on neural pose estimators that suffer out-of-distribution issues [12] or numerical optimization hindered by ill-posed objectives and unreliable initializations. To bridge this pose gap, motivated by the fine-grained controllability of modern generative models over appearance, geometry, and pose, we propose GIF, an agentic framework that casts scene generation as reconstruction and pose recovery. We focus on two-object interactions as the minimal setting where the pose problem already manifests, while the formulation extends naturally to multi-object scenes (Sec. 4.5). Since recovering correct relative poses presupposes geometrically complete and instance-disentangled assets, GIF first obtains such assets through CoGen, which couples 2D amodal completion with image-conditioned 3D reconstruction so that mutually occluded objects emerge as clean, separately interactive meshes. GPRM then refines the relative pose under joint geometric and physical guidance, using a VLM verifier to inspect simulated rollouts for selecting the candidate that best matches the structured specification. Each verified composition can serves as a compact task specification for downstream policy learning.

In summary, our contributions are threefold. 1) We formulate open-ended generation of functional two-object compositions as a contact-precise, simulation-ready scene-generation problem. 2) We introduce GIF, which combines CoGen for disentangled asset reconstruction with GPRM for geometry- and physics-guided pose refinement. 3) We build a benchmark spanning eight contact-geometry classes and show across 120 generated scenes that GIF outperforms strong baselines on relation matching, object quality, collision rate, and human preference.

2 Related Work

Simulation-Ready Scene Generation for Robot Learning. Robot learning has long relied on simulators and benchmarks that provide physics, assets, tasks, and data-generation interfaces, including AI2-THOR [13], Habitat [14, 15], iGibson [16, 17], TDW [18], BEHAVIOR-1K [19], SAPIEN [20], robosuite [21], ManiSkill2 [22], RLBench [23], and Isaac Gym [24]. Indoor and embodied scene generation extends this infrastructure from priors, procedures, language, or agents, spanning layout datasets and models [25, 26, 27, 28, 29, 30] and recent automated generators [31, 32, 33, 2, 34, 35, 36, 37, 4, 38, 39]. Robotics-oriented systems further add task resources, asset acquisition, domain randomization, and simulator export [40, 41, 6, 7, 42, 43, 44, 45, 46, 3, 5, 8]. These methods mainly reason over rooms, tables, support surfaces, or coarse spatial relations; GIF instead focuses on the contact-precise two-object composition needed for functional relations such as capping, hooking, slotting, spanning, and pegging.

Image-Conditioned Asset Construction and Interactive Object Compositions. Contact-precise composition requires assets that are visually plausible, separately manipulable, and equipped with usable collision geometry. Open-vocabulary perception and editing models support object isolation and reference-image control [47, 48, 49, 50, 51, 52], while text- and image-conditioned 3D generation has advanced through score-distillation and reconstruction models [53, 54, 55, 56, 57, 58, 59,

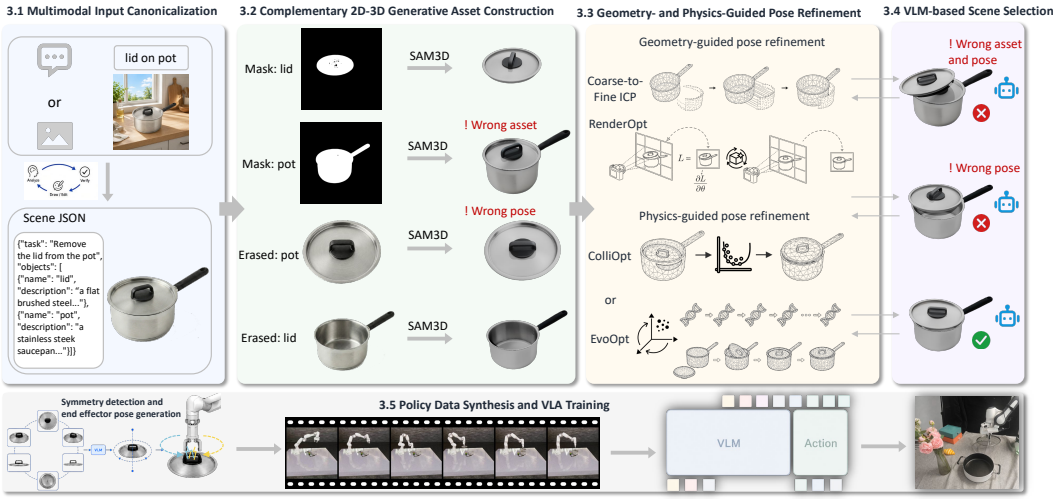


Figure 2: Overview of GIF. The multimodal input agent canonicalizes a language or image prompt into a unified specification $(I_{\text{ref}}, J_{\text{scene}})$. CoGen lifts each object into two complementary mesh and pose candidates via direct image-to-3D reconstruction and an erase-and-regenerate branch that mitigates occlusion and instance entanglement. GPRM iteratively refines the relative pose under joint geometric and physical guidance, while a VLM verifier accepts or returns each refined scene. The verified scene then seeds trajectory synthesis and VLA training for real-world deployment.

60, 61, 62, 63, 64, 65, 66, 67, 68]. Image-conditioned reconstruction and asset-acquisition pipelines recover geometry, texture, and pose from real or generated images [11, 69, 70, 71, 72, 73, 9, 74, 10], and systems such as TabletopGen [8], Interact3D [75], and WorldAct [76] assemble or activate multiple interactive objects. For functional object pairs, however, mutual occlusion occurs exactly at the contact interface, causing direct reconstruction to fuse instances, leak partner geometry, or hallucinate hidden surfaces. CoGen addresses this asset-level bottleneck by producing instance-disentangled meshes and coarse initial poses for downstream refinement.

Object-to-Object Affordance and Composition Pose Recovery. Once assets are available, existing placement and affordance methods still leave contact-precise relative pose recovery for functional object compositions largely unresolved. Language- or agent-prompted scene methods delegate placement to coarse LLM/VLM specifications or layout optimization [7, 8, 4, 3, 35], while visual-prompted systems rely on pose estimators that can fail out of distribution [11, 12, 77, 78] or on classical registration with fragile initialization [79]. Object-to-object affordance, placement, rearrangement, and motion-planning work highlights the importance of relational geometry [80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95], but typically assumes existing assets, fixed placement families, support-surface plausibility, or collision avoidance rather than tight semantic contact. GPRM instead refines CoGen's initial poses through registration, render-based alignment, collision-aware optimization, simulation-aware search, and VLM selection, yielding compositions whose success depends on contact rather than mere non-overlap.

3 Method

Given an input $\mathcal{I}$ specifying a functional object pair (o_1, o_2) together with their attributes and spatial relation, either as a language instruction $\mathcal{I} = T$ or a reference image $\mathcal{I} = I_{\text{input}}$, the goal is to generate a simulation-ready scene $\mathcal{S} = \{(m_i, p_i)\}_{i\in\{1,2\}}$, where m_i is a mesh for the rigid object o_i and $p_i = (r_i, t_i) \in \text{SO}(3) \times \mathbb{R}^3$ is its camera-frame pose.

As shown in Fig. 2, a multimodal agent (Sec. 3.1) canonicalizes $\mathcal{I}$ into a unified specification, CoGen (Sec. 3.2) supplies instance-disentangled meshes together with coarse initial poses, GPRM (Sec. 3.3) refines the relative pose under joint geometric and physical guidance, and a VLM selector (Sec. 3.4) verifies each candidate scene and triggers further refinement when needed. The verified scene is then used as the terminal configuration for trajectory synthesis and VLA policy learning (Sec. 3.5).

3.1 Multimodal Input Canonicalization

A multimodal agent canonicalizes any input form of $\mathcal{I}$ into a unified specification $(I_{\text{ref}}, J_{\text{scene}})$, where I_{ref} is a foreground-only reference image of the desired object pair and J_{scene} is a structured JSON record of object identities, asset attributes, and the target relation. The agent runs an analyze–act–verify loop that first identifies o_1, o_2, and the relation, then invokes either an image-generation tool (when only T is given) or an image-editing tool (when I_{input} is given) to produce a candidate I_{ref}, after which a VLM verifier accepts or calls editing or redrawing until acceptance or a iteration budget is exhausted. How we prepare simulation ready assets with convex decomposition are provided in the supplementary.

3.2 CoGen: Complementary 2D-3D Generative Asset Construction

GPRM requires instance-disentangled meshes together with reasonably aligned initial poses. A natural approach lifts each object directly with an off-the-shelf image-to-3D model [10], conditioned on I_{ref} and a binary instance mask b_i^{ref} derived from J_{scene}, yielding $(m_i^{\text{ref}}, p_i^{\text{ref}})$. But this single-pass reconstruction is insufficient for contact-rich pairs. Existing segmentation and 3D models lack amodal reasoning, so heavy mutual occlusion (e.g., a flower stem inside a vase) leaves meshes that explain only the visible region, and limited instance disentanglement lets partner geometry leak through a single-object mask, e.g., reconstructing a pot together with its lid.

We therefore add a complementary erase-and-regenerate branch that exploits the larger training corpora and stronger object priors of 2D generative models. The branch erases o_j from I_{ref} and completes o_i, obtaining an occlusion-free image I_i^{erase} that SAM3D [10] lifts to $(m_i^{\text{erase}}, p_i^{\text{erase}})$. Since 2D models still provide imperfect pose control and may slightly perturb the kept object, both candidates $(m_i^{\text{ref}}, p_i^{\text{ref}})$ and $(m_i^{\text{erase}}, p_i^{\text{erase}})$ are retained as initial mesh-pose pairs for GPRM. More details can be found in the supplementary materials.

3.3 GPRM: Geometry- and Physics-Guided Pose Refinement Modules

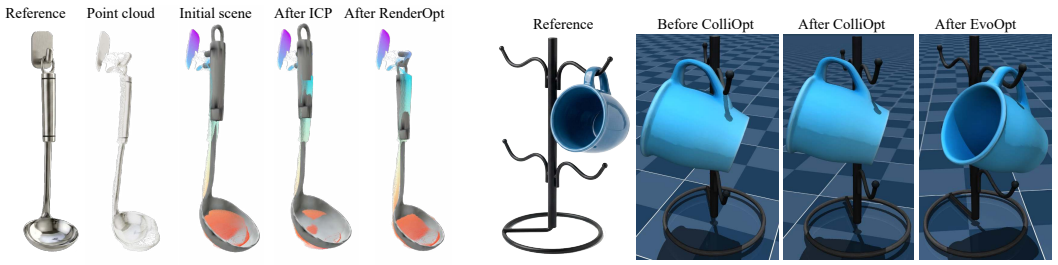
To obtain accurate relative poses, we refine CoGen's coarse mesh–pose candidates with geometry- and physics-guided modules and verify each refined scene with the VLM selector in Sec. 3.4. For each o_i, let $a_i \in \mathcal{A} = \{\text{ref}, \text{erase}\}$ index the two candidates $(m_i^{a_i}, p_i^{a_i})$. We enumerate the four pairings $\mathcal{C} = \{(a_1, a_2)\}$, initialize $\mathcal{S}_c^0 = \{(m_i^{a_i}, p_i^{a_i})\}_{i=1}^2$ for each c, and run a GPRM–VLM loop $\mathcal{S}_c^0 \to \cdots \to \mathcal{S}_c^{K_c}$ until acceptance or budget exhaustion, yielding $\widehat{\mathcal{S}}_c = \mathcal{S}_c^{K_c}$. Fig. 3 illustrates two such cases. Further GPRM details are in the supplementary.

Geometry-Guided Refinement I. Coarse-to-Fine ICP. ICP [79] aligns a 3D model to observed geometry by iterating nearest-neighbour correspondences and minimizing point-to-point distance, requiring no task-specific priors. We sample the source cloud from $m_i^{a_i}$ at the candidate pose and back-project the reference depth from a monocular geometry estimator [96] through the eroded instance mask to form the per-object target cloud, and progressively tighter ICP rounds.

Geometry-Guided Refinement II. Render-Based Optimization (RenderOpt). ICP is sensitive to incomplete point clouds and to errors from the monocular estimator, both of which bias correspondence selection. We therefore add a 2D refinement stage that rasterises each candidate mesh with nvdiffrast [97] and updates the pose and scale so that the rendered silhouette and depth match the reference image, with per-object recall and precision terms preventing leakage between instances and a depth term resolving occlusion order. Regularization terms anchor the solution to the ICP initialization. A refined pose $p_i^{a_i,\text{rend}}$ is accepted only if every per-object loss component improves.

Physics-Guided Refinement I. Collision-Avoidance Optimization (ColliOpt). RenderOpt minimizes image-space discrepancies but does not guarantee physical validity, so the resulting meshes may still interpenetrate and yield undefined behaviour in MuJoCo. We add a constrained pass that finds the nearest collision-free pose to $p_i^{a_i,\text{rend}}$ under a signed convex-hull separation constraint and a ground-clearance constraint, built with Drake [98] and solved with SNOPT [99].

Physics-Guided Refinement II. Simulation-Aware Evolutionary Optimization (EvoOpt). Resolving collisions in isolation does not guarantee that the relation survives gravity. As Fig. 3(b) shows, the cup remains collision-free yet still slips off the mug tree in simulation. We therefore



(a) An example of spoon-hook scene

(b) An example of cup-mug tree scene

Figure 3: Two cases of geometry- and physics-guided refinement. (a) Spoon hooked on a wall hook: ICP drags the spoon inward, and RenderOpt tightens its scale so the ring catches the hook. (b) Cup hung on a mug tree: ColliOpt removes interpenetration but the cup still slips under gravity, while EvoOpt searches under simulation feedback to land on a pose that remains hooked.

close the loop by treating MuJoCo [100] as a black-box objective and search the pose neighbourhood with CMA-ES [101] from EvoTorch [102]. Each candidate is encoded as a translation delta in $\mathbb{R}^3$ and a rotation delta in the 6D representation [103], and is scored under T_{sim} steps by a fitness that penalizes drift, residual penetration, residual velocity, and deviation from the input pose. EvoOpt is launched whenever the VLM selector rejects all earlier refinements.

3.4 VLM-based Scene Selection

For each initial or refined scene, we load it into MuJoCo, simulate until convergence or a step budget is reached, and tile multi-view, multi-timestep renders into a single image whose sub-panels carry view labels and step indices. A VLM verifier inspects this image to check whether the target relation holds in the initial frame and is preserved in the final frame. Among scenes that pass verification, a second VLM call ranks the surviving candidates by horizontally concatenating their per-candidate panels and returning the index that best matches J_{scene}. An example of the cross-candidate montage and full prompts are provided in the supplementary.

3.5 Policy Data Synthesis and VLA Training

Given a VLM-verified target scene $\widehat{\mathcal{S}}$, we treat its arrangement as the task's terminal configuration and synthesize demonstrations that realize it. For each rollout we randomize the table-top poses to obtain $\mathcal{S}^{\text{rand}} = \{(m_i, p_i^{\text{rand}})\}_{i=1}^{2}$ and define a valid terminal scene $\mathcal{S}^\star$ via $(p_2^\star)^{-1} p_1^\star = p_2^{-1} p_1$. With m_1 as manipuland and m_2 fixed, this yields $p_2^\star = p_2^{\text{rand}}$ and $p_1^\star = p_2^{\text{rand}} p_2^{-1} p_1$, after which a motion planner produces a collision-free trajectory from p_1^{rand} to $p_1^\star$. To avoid redundant supervision, a VLM annotates per-object symmetry classes and trajectory targets are defined over the resulting pose-equivalence classes. We pair high-fidelity ray-traced rendering with domain randomization over lighting, textures, camera parameters, and object poses to narrow the sim-to-real gap, following recent VLA pipelines [7, 6, 104]. We train $\pi_{0.5}$ [1] purely on the synthesized trajectories with wrist-view and third-person view images as input and action chunks as output for direct real-robot deployment. Implementation details are provided in the supplementary.

4 Experimental Results

4.1 Experimental Setup

Taxonomy and Benchmark Suite. We construct a taxonomy of pairwise contact-geometry relations grounded in real human manipulation by deploying an LLM-based agent to classify and aggregate activity descriptions from Ego4D [105], yielding 8 representative types: C.1 Containment, C.2 Capping, C.3 Resting, C.4 Leaning, C.5 Hooking, C.6 Slotting, C.7 Spanning, and C.8 Pegging. The same agent then selects three manipulation tasks per category to form a $8 \times 3 = 24$-task benchmark, with the full task list provided in the supplementary. Each task is instantiated with five semantic variants for a total of $24 \times 5 = 120$ scenes that methods are evaluated on.

Table 1: Main results across the eight contact-geometry classes from C.1 Containment to C.8 Pegging. The upper block reports per-class Object Matching (O.M. ↑) and Relationship Matching (R.M. ↑), each averaged over five seeds per task and three tasks per class. The lower block summarizes Collision Rate (C.R. ↓) averaged across all eight classes. Bold marks the best per-column score within each input setting.

Method	Object Matching ↑ / Relationship Matching ↑								Physical
	C.1	C.2	C.3	C.4	C.5	C.6	C.7	C.8	C.R. ↓
User Input: Image Prompt									
RoLA [11]	8.00 / 3.80	8.60 / 5.07	8.33 / 4.33	8.47 / 3.07	8.00 / 2.80	7.40 / 3.33	7.87 / 4.47	8.20 / 6.20	0.558
TabletopGen [8]	6.60 / 1.33	9.13 / 8.67	9.13 / 3.73	5.33 / 2.07	7.07 / 1.73	7.40 / 3.60	8.60 / 5.00	7.60 / 5.27	**0.000**
Ours-Img	**9.07 / 7.80**	**9.60 / 9.20**	**9.53 / 9.27**	**9.67 / 9.47**	**9.00 / 4.47**	**9.40 / 7.33**	**9.60 / 7.53**	**8.73 / 7.53**	**0.000**
User Input: Language Prompt									
GenieSim [7]	4.87 / 3.47	4.40 / 1.07	4.07 / 2.13	2.73 / 0.80	2.53 / 0.87	6.07 / 2.13	3.07 / 1.00	2.87 / 0.33	0.217
SceneSmith [3]	**8.93** / 6.60	**9.40** / 7.60	8.67 / 6.93	8.27 / 6.33	7.20 / 2.27	8.93 / 6.80	9.47 / 6.67	7.93 / 7.47	0.083
Ours	**8.93 / 7.60**	9.27 / **9.00**	**9.33 / 8.20**	**9.13 / 8.27**	**8.80 / 5.20**	**9.00 / 7.47**	**9.53 / 7.80**	**8.80 / 9.00**	**0.008**

Metrics. We evaluate asset, relation, and physical correctness with three metrics. Object Matching (O.M., $[0, 10]$) compares the asset's appearance and geometry against the reference image, Relationship Matching (R.M., $[0, 10]$) measures whether the final configuration realizes the specified functional contact (e.g., a lid capping a pot rather than merely placed nearby), and Collision Rate (C.R., $[0, 1]$) reports the fraction of seeds whose MuJoCo configuration exhibits inter-object penetration. To make the ratings objective and comprehensive, we score O.M. and R.M. with two complementary protocols: (i) VLM-based scoring against the reference and task specification, and (ii) a user study where humans apply the same criteria to identical outputs. Details are in the supplementary.

Baselines and Comparison Protocol. We compare against four representative systems and field each one under the protocol that best matches its assumptions. RoLA [11] reconstructs simulation-ready assets from a real image, and TabletopGen [8] composes tabletop scenes by generating assets and placing them on a supporting surface. Since both require a table, we evaluate them and our method on the with-table variant of every scene for these two pairings. GenieSim [7] is run without its human-in-the-loop edit step, with a forced asset-acquisition signal appended to the prompt so that it always commits to a runnable scene rather than aborting on missing assets. SceneSmith [3] is a hierarchical room-level system, so we evaluate only its manipulands agents and keep the input scope identical to ours. For fairness, all image-to-3D modules are uniformly replaced with SAM3D [10].

4.2 Main Results

As shown in Tab. 1, under both language- and image-prompt settings, our method attains the highest R.M. in every one of the eight relation classes, averaging 7.82 against 6.33 for SceneSmith (the strongest baseline) and 1.48 for GenieSim [7], with C.5 (Hooking) the hardest class yet still at least 1.6 points ahead of every baseline. On O.M., we exceed SceneSmith by 0.50 on average and never trail by more than 0.13 in any class, confirming that CoGen preserves asset fidelity while improving pose accuracy. Owing to physics-guided refinement, our C.R. is near-zero. The only residual collisions appear in the spoon-hook task, where convex decomposition produces a small ring radius that EvoOpt must trade against the collision term to stay hooked.

Table 2: User study pairwise win-rates. O.M. counts ties as 0.5 wins for each side; R.M. excludes ties. Wilson 95% confidence intervals and Cohen's h are reported.

Ours vs.	O.M. (↑)			R.M. (↑)		
	Win%	95% CI	h	Win%	95% CI	h
SceneSmith [3]	80.7	[75.0, 85.3]	+0.66	95.9	[91.7, 98.0]	+1.16
RoLA [11]	78.5	[73.0, 83.1]	+0.61	97.2	[94.0, 98.7]	+1.23
TabletopGen [8]	80.8	[75.1, 85.4]	+0.66	95.9	[91.7, 98.0]	+1.16
GenieSim [7]	98.4	[95.8, 99.4]	+1.32	100.0	[98.2, 100.0]	+1.57

Human raters corroborate the VLM findings with even larger margins. Against all four baselines, scenes from our method are preferred with large Cohen's h effect sizes and tight Wilson confidence intervals (Tab. 2). The advantage is most pronounced on R.M., where ours wins 95.9% and 100% against SceneSmith [3] and GenieSim [7], respectively. Visual comparisons are shown in Fig. 4.

4.3 Ablation Studies

Effectiveness of CoGen. We ablate complementary reconstruction (c.r.) within CoGen in Tab. 3: disabling c.r. causes a consistent drop in both O.M. and R.M. across all eight classes, as occlusion

Figure 4: Comparisons between our method and baselines after MuJoCo settling. GenieSim [7] is library-bound and frequently lacks a suitable asset. RoLA [11] skips collision optimization, so its placements interpenetrate or fall apart under gravity. SceneSmith [3] emits poses via direct VLM output or hand-crafted heuristics and therefore misses local contact geometry. TabletopGen's [8] per-instance segmentation either drops objects or fuses composites into a single instance. Our method produces contact-precise compositions that remain stable across every prompt.

Table 3: Ablation on complementary reconstruction (c.r.) across 8 classes.

Method	C.1		C.2		C.3		C.4		C.5		C.6		C.7		C.8	
	O.M.	R.M.	O.M.	R.M.	O.M.	R.M.	O.M.	R.M.	O.M.	R.M.	O.M.	R.M.	O.M.	R.M.	O.M.	R.M.
w/o c.r.	7.00	5.80	9.20	2.80	9.07	7.27	8.53	6.53	8.13	2.60	8.93	6.73	8.87	7.20	7.60	4.00
w c.r.	**8.93**	**7.60**	**9.27**	**9.00**	**9.33**	**8.20**	**9.13**	**8.27**	**8.80**	**5.20**	**9.00**	**7.47**	**9.53**	**7.80**	**8.80**	**9.00**

and interference from the partner object yield unreasonable assets that degrade relation matching. Visual comparisons are shown in Fig. 5.

Effectiveness of GPRM. Tab. 4 presents a cumulative ablation of GPRM stages measured by R.M. Each successive module yields consistent gains over the SAM3D initial pose, and EvoOpt provides the largest gain on C.5 (Hooking, +0.33) where simulation stability is most sensitive to small errors, confirming that physical simulation feedback is essential for generating such scenes.

Table 4: Ablation on pose refinement modules across 8 classes, measured by R.M. . Methods are cumulative, with each row adding one stage to the configuration in the row above.

Method	C.1	C.2	C.3	C.4	C.5	C.6	C.7	C.8
SAM3D Only	5.47	7.07	7.47	7.60	4.60	6.13	6.40	7.47
+ ICP	6.80	8.40	7.93	8.13	4.67	6.93	7.13	7.47
+ RenderOpt	7.33	8.93	8.00	**8.27**	4.87	7.33	7.67	8.93
+ EvoOpt	**7.60**	**9.00**	**8.20**	**8.27**	**5.20**	**7.47**	**7.80**	**9.00**

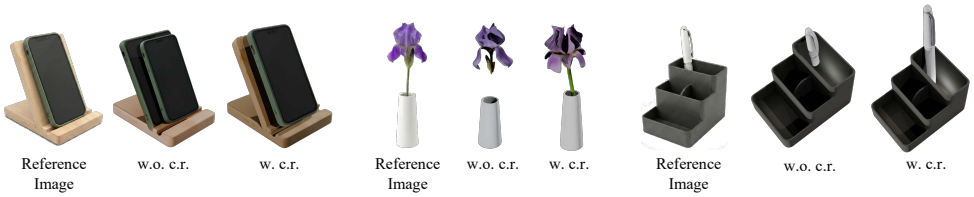


Figure 5: Three examples that show the effectiveness of CoGen. Beyond the over-segmentation case in Fig. 2 (pot lid mask absorbs the pot), three further failure modes drive disentangled reconstruction. Phone–phone-stand reconstruction lets the metallic phone texture leak onto the stand. Flower–vase reconstruction drops the slender stem under incomplete foreground masks, recovered by the erase-and-regenerate branch from a partner-removed view. Pen–pen-holder reconstruction truncates the pen along its hidden length under occlusion.

4.4 GIF-Generated Tasks for Policy Learning

We evaluate GIF-generated compositions as policy-learning tasks. For each task, CoGen and GPRM instantiate task-asset variants, each consisting of simulation-ready assets and a VLM-verified terminal relative pose. We synthesize demonstrations by randomizing initial object poses, scene illumination, and camera poses, train $\pi_{0.5}$ only on the simulated trajectories, and deploy directly on the real robot for two contact-rich tasks: "serve sausage with a paper box" and "cover pot with a lid".

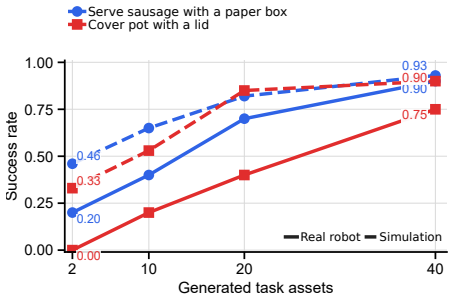


Figure 6: Policy learning on two real-robot tasks. Solid and dashed lines report real-robot and simulation success rates as the number of GIF-generated task-asset variants grows.

Fig. 6 shows that real-robot success improves with the asset budget: serving rises from 0.20 with two assets to 0.90 with forty, while covering rises from 0.00 to 0.75. Simulation success is higher in absolute value but follows the same monotonic trend, suggesting that GIF assets and relative poses define executable policy-learning goals and that asset diversity improves real-world transfer. Training hyperparameters, data-generation details, and more demonstrations are in the supplementary.

4.5 Preliminary Extension to Multi-Object Scenes

Our pipeline extends to scenes with more objects through an iterative loop. At each step, a multimodal agent proposes the next object and its spatial relation, and the image-editor inserts it into the rendered view of the current scene while preserving viewpoint, lighting, and existing layout. CoGen and GPRM are then rerun on the newcomer, with the merged geometry of all previously placed objects held fixed during geometry- and physics-guided refinement. An example is shown in Fig. 7.

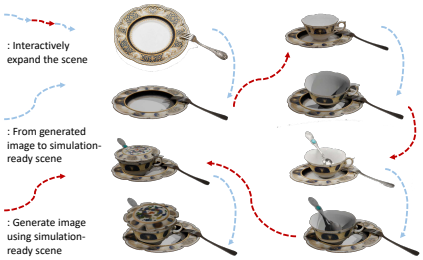


Figure 7: Preliminary extension to a multi-object scene. Further results spanning all eight relation classes are in the supplementary.

5 Limitations and Conclusion

GIF targets rigid object pairs, since convex decomposition [106, 107] and MuJoCo settling assume rigid geometry, leaving articulated and deformable assets out of scope. Room-scale layouts also remain future work; a natural direction is hierarchical decomposition, as in SceneSmith [3], with GIF serving as the contact-precise leaf solver. In summary, GIF casts contact-precise scene generation as reconstruction plus relative-pose recovery, coupling CoGen for instance-disentangled meshes, GPRM for pose refinement, and a VLM verifier. Across eight contact-geometry classes, GIF outperforms strong baselines on relationship matching, object fidelity, collision rate, and human preference. Its simulation data generation, policy training, and real-world deployment further demonstrate that verified functional compositions can support downstream robot learning.

References

[1] K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. R. Equi, C. Finn, N. Fusai, M. Y. Galliker, D. Ghosh, L. Groom, K. Hausman, b. ichter, S. Jakubczak, T. Jones, L. Ke, D. LeBlanc, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, A. Z. Ren, L. X. Shi, L. Smith, J. T. Springenberg, K. Stachowicz, J. Tanner, Q. Vuong, H. Walke, A. Walling, H. Wang, L. Yu, and U. Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization. In *Proceedings of the 9th Conference on Robot Learning (CoRL)*, pages 17–40, 2025.

[2] M. Deitke et al. Holodeck: Language guided generation of 3D embodied AI environments. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[3] N. Pfaff, T. Cohn, S. Zakharov, R. Cory, and R. Tedrake. SceneSmith: Agentic generation of simulation-ready indoor scenes. *arXiv preprint arXiv:2602.09153*, 2026.

[4] Y. Yang, B. Jia, S. Zhang, and S. Huang. SceneWeaver: All-in-one 3D scene synthesis with an extensible and self-reflective agent. *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[5] J. Hao, N. Liang, Z. Luo, X. Xu, W. Zhong, R. Yi, Y. Jin, Z. Lyu, F. Zheng, L. Ma, and J. Pang. MesaTask: Towards task-driven tabletop scene generation via 3D spatial reasoning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[6] T. Chen, Z. Chen, B. Chen, Z. Cai, Y. Liu, Z. Li, Q. Liang, X. Lin, Y. Ge, Z. Gu, et al. RoboTwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.

[7] C. Yin, D. Huang, D. Yang, J. Wang, N. Zhao, C. Xu, W. Sun, L. Hou, Z. Li, J. Wu, Z. Liu, Z. Xiao, S. Zhang, L. Bao, R. Feng, Z. Pang, J. Li, Q. Wang, and M. Yao. Genie Sim 3.0: A high-fidelity comprehensive simulation platform for humanoid robot. *arXiv preprint arXiv:2601.02078*, 2026.

[8] Z. Wang, Y. He, L. Yang, W. Zou, H. Ma, L. Liu, W. Sui, Y. Guo, and H. Su. Tabletop-Gen: Instance-level interactive 3D tabletop scene generation from text or single image. *arXiv preprint arXiv:2512.01204*, 2025.

[9] T. H. Team. Hunyuan3d 2.5: Towards high-fidelity 3d assets generation with ultimate details, 2025. URL https://arxiv.org/abs/2506.16504.

[10] S. D. Team, X. Chen, F.-J. Chu, P. Gleize, K. J. Liang, A. Sax, H. Tang, W. Wang, M. Guo, T. Hardin, X. Li, A. Lin, J. Liu, Z. Ma, A. Sagar, B. Song, X. Wang, J. Yang, B. Zhang, P. Dollár, G. Gkioxari, M. Feiszli, and J. Malik. Sam 3d: 3dfy anything in images, 2026. URL https://arxiv.org/abs/2511.16624.

[11] S. Zhao, J. Mao, W. Chow, Z. Shangguan, T. Shi, R. Xue, Y. Zheng, Y. Weng, Y. You, D. Seita, L. Guibas, S. Zakharov, V. Guizilini, and Y. Wang. Robot learning from any images. In *Proceedings of the 9th Conference on Robot Learning (CoRL)*, 2025.

[12] B. Wen, W. Yang, J. Kautz, and S. Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17868–17879, 2024.

[13] E. Kolve, R. Mottaghi, D. Gordon, Y. Zhu, A. Gupta, and A. Farhadi. AI2-THOR: An interactive 3D environment for visual AI. *arXiv preprint arXiv:1712.05474*, 2017.

[14] M. Savva, A. Malik, D. Parikh, D. Batra, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, et al. Habitat: A platform for embodied AI research. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[15] A. Szot, A. Clegg, E. Undersander, E. Wijmans, Y. Zhao, J. Turner, N. Maestre, M. Mukadam, D. S. Chaplot, O. Maksymets, et al. Habitat 2.0: Training home assistants to rearrange their habitat. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[16] B. Shen, F. Xia, C. Li, R. Martin-Martin, L. Fan, G. Wang, C. Perez-D'Arpino, S. Buch, S. Srivastava, L. Tchapmi, et al. iGibson 1.0: A simulation environment for interactive tasks in large realistic scenes. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

[17] C. Li, F. Xia, R. Martin-Martin, M. Lingelbach, S. Srivastava, B. Shen, K. E. Vainio, C. Gokmen, G. Dharan, T. Jain, et al. iGibson 2.0: Object-centric simulation for robot learning of everyday household tasks. In *Conference on Robot Learning (CoRL)*, 2021.

[18] C. Gan, J. Schwartz, S. Alter, D. Mrowca, M. Schrimpf, J. Traer, J. De Freitas, J. Kubilius, A. Bhandwaldar, N. Haber, et al. ThreeDWorld: A platform for interactive multi-modal physical simulation. *arXiv preprint arXiv:2007.04954*, 2020.

[19] C. Li, R. Zhang, J. Wong, C. Gokmen, S. Srivastava, R. Martin-Martin, C. Wang, G. Levine, M. Lingelbach, J. Sun, et al. BEHAVIOR-1K: A benchmark for embodied AI with 1,000 everyday activities and realistic simulation. In *Conference on Robot Learning (CoRL)*, 2023.

[20] F. Xiang, Y. Qin, K. Mo, Y. Xia, H. Zhu, F. Liu, M. Liu, H. Jiang, Y. Yuan, H. Wang, et al. SAPIEN: A simulated part-based interactive environment. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[21] Y. Zhu, J. Wong, A. Mandlekar, R. Martin-Martin, A. Joshi, S. Nasiriany, and Y. Zhu. robosuite: A modular simulation framework and benchmark for robot learning. *arXiv preprint arXiv:2009.12293*, 2020.

[22] J. Gu, F. Xiang, X. Li, Z. Ling, X. Liu, T. Mu, Y. Tang, S. Tao, X. Wei, Y. Yao, et al. ManiSkill2: A unified benchmark for generalizable manipulation skills. In *International Conference on Learning Representations (ICLR)*, 2023.

[23] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison. RLBench: The robot learning benchmark and learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.

[24] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State. Isaac Gym: High performance GPU-based physics simulation for robot learning. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2021.

[25] H. Fu, B. Cai, L. Gao, L. Zhang, C. Li, Q. Zeng, C. Sun, R. Jia, B. Zhao, and H. Zhang. 3D-FRONT: 3D furnished rooms with layouts and semantics. *arXiv preprint arXiv:2011.09127*, 2021.

[26] D. Ritchie, K. Wang, and Y.-A. Lin. Fast and flexible indoor scene synthesis via deep convolutional generative models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[27] D. Paschalidou, A. Kar, M. Shugrina, K. Kreis, A. Geiger, and S. Fidler. ATISS: Autoregressive transformers for indoor scene synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[28] X. Wang et al. SceneFormer: Indoor scene generation with transformers. *arXiv preprint arXiv:2012.09793*, 2021.

[29] L. Yang, Z. Zhang, Y. Song, S. Hong, R. Xu, Y. Zhao, Y. Shao, W. Zhang, B. Cui, and M.-H. Yang. CommonScenes: Generating commonsense 3D indoor scenes with scene graph diffusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[30] L. Hollein, A. Cao, A. Owens, J. Johnson, and M. Nießner. Text2Room: Extracting textured 3D meshes from 2D text-to-image models. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[31] M. Deitke et al. ProcTHOR: Large-scale embodied AI using procedural generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[32] A. Raistrick, L. Mei, K. Kayan, D. Yan, Y. Zuo, B. Han, H. Wen, M. Parakh, S. Alexandropoulos, L. Lipson, Z. Ma, and J. Deng. Infinigen Indoors: Photorealistic indoor scenes using procedural generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[33] J. Tang, Y. Nie, L. Markhasin, A. Dai, J. Thies, and M. Nießner. DiffuScene: Denoising diffusion models for generative indoor scene synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[34] Z. Bian, R. Ren, Y. Yang, and C. Callison-Burch. Holodeck 2.0: Vision-language-guided 3D world generation with editing. *arXiv preprint arXiv:2508.05899*, 2025.

[35] F.-Y. Sun, W. Liu, S. Gu, D. Lim, G. Bhat, F. Tombari, M. Li, N. Haber, and J. Wu. LayoutVLM: Differentiable optimization of 3D layout via vision-language models. *arXiv preprint arXiv:2412.02193*, 2024.

[36] H. Xia, X. Li, Z. Li, Q. Ma, J. Xu, M.-Y. Liu, Y. Cui, T.-Y. Lin, W.-C. Ma, S. Wang, S. Song, and F. Wei. SAGE: Scalable agentic 3D scene generation for embodied AI. *arXiv preprint arXiv:2602.10116*, 2026.

[37] L. Ling, C.-H. Lin, T.-Y. Lin, Y. Ding, Y. Zeng, Y. Sheng, Y. Ge, M.-Y. Liu, A. Bera, and Z. Li. Scenethesis: A language and vision agentic framework for 3D scene generation. *arXiv preprint arXiv:2505.02836*, 2025.

[38] G. Lin, K. Huang, M. Liu, R. Gao, H. Chen, L. Chen, B. Lu, T. Komura, Y. Liu, J.-Y. Zhu, et al. Pat3d: Physics-augmented text-to-3d scene generation. *arXiv preprint arXiv:2511.21978*, 2025.

[39] L. Che, S. Wen, S. Huang, C. Wang, Y. Yang, G. Dudek, X. Wang, and J. Su. Mansion: Multi-floor language-to-3d scene generation for long-horizon tasks. *arXiv preprint arXiv:2603.11554*, 2026.

[40] S. Nasiriany et al. RoboCasa: Large-scale simulation of everyday tasks for generalist robots. In *Robotics: Science and Systems (RSS)*, 2024.

[41] Y. Wang, Z. Xian, F. Chen, T.-H. Wang, Y. Wang, K. Fragkiadaki, Z. Erickson, D. Held, and C. Gan. RoboGen: Towards unleashing infinite data for automated robot learning via generative simulation. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, pages 51936–51983, 2024.

[42] L. Wang, Y. Ling, Z. Yuan, M. Shridhar, C. Bao, Y. Qin, B. Wang, H. Xu, and X. Wang. GenSim: Generating robotic simulation tasks via large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

[43] P. Katara, Z. Xian, and K. Fragkiadaki. Gen2Sim: Scaling up robot learning in simulation with generative models. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[44] A. Zook, F.-Y. Sun, J. Spjut, V. Blukis, S. Birchfield, and J. Tremblay. GRS: Generating robotic simulation tasks from real-world images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2025.

[45] P. Nguyen, T.-H. Wang, Z.-W. Hong, E. Aasi, A. Silva, G. Rosman, S. Karaman, and D. Rus. ReGen: Generative robot simulation via inverse design. In *International Conference on Learning Representations (ICLR)*, 2025.

[46] S. Lee, S. Park, and H. Kim. DynScene: Scalable generation of dynamic robotic manipulation scenes for embodied AI. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12166–12175, 2025.

[47] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su, et al. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. In *European Conference on Computer Vision (ECCV)*, 2024.

[48] L. H. Li, P. Zhang, H. Zhang, J. Yang, C. Li, Y. Zhong, L. Wang, L. Yuan, L. Zhang, J.-N. Hwang, et al. GLIP: Grounded language-image pre-training. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[49] M. Minderer, A. Gritsenko, A. Stone, M. Neumann, D. Weissenborn, A. Dosovitskiy, A. Mahendran, A. Arnab, M. Dehghani, Z. Shen, et al. Simple open-vocabulary object detection with vision transformers. In *European Conference on Computer Vision (ECCV)*, 2022.

[50] R. Suvorov, E. Logacheva, A. Mashikhin, A. Remizova, A. Ashukha, A. Silvestrov, N. Kong, H. Goka, K. Park, and V. Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2022.

[51] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, and L. Van Gool. RePaint: Inpainting using denoising diffusion probabilistic models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[52] L. Zhang and M. Agrawala. Adding conditional control to text-to-image diffusion models. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[53] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall. DreamFusion: Text-to-3D using 2D diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

[54] C.-H. Lin, J. Gao, L. Tang, T. Takikawa, X. Zeng, X. Huang, K. Kreis, S. Fidler, M.-Y. Liu, and T.-Y. Lin. Magic3D: High-resolution text-to-3D content creation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[55] R. Chen, Y. Chen, N. Jiao, and K. Jia. Fantasia3D: Disentangling geometry and appearance for high-quality text-to-3D content creation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[56] Z. Wang, C. Lu, Y. Wang, F. Bao, C. Li, H. Su, and J. Zhu. ProlificDreamer: High-fidelity and diverse text-to-3D generation with variational score distillation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[57] J. Gao, T. Shen, Z. Wang, W. Chen, K. Yin, D. Li, O. Litany, Z. Gojcic, and S. Fidler. GET3D: A generative model of high quality 3D textured shapes learned from images. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[58] A. Nichol, H. Jun, P. Dhariwal, P. Mishkin, and M. Chen. Point-E: A system for generating 3D point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022.

[59] H. Jun and A. Nichol. Shap-E: Generating conditional 3D implicit functions. *arXiv preprint arXiv:2305.02463*, 2023.

[60] R. Liu, R. Wu, B. Van Hoorick, P. Tokmakov, S. Zakharov, and C. Vondrick. Zero-1-to-3: Zero-shot one image to 3D object. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[61] Y. Liu, C. Lin, Z. Zeng, X. Long, L. Liu, T. Komura, and W. Wang. Syncdreamer: Generating multiview-consistent images from a single-view image. In *International conference on learning representations*, volume 2024, pages 27676–27697, 2024.

[62] X. Long, Y.-C. Guo, C. Lin, Y. Liu, Z. Dou, L. Liu, Y. Ma, S.-H. Zhang, M. Habermann, C. Theobalt, et al. Wonder3d: Single image to 3d using cross-domain diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9970–9980, 2024.

[63] M. Liu, C. Xu, H. Jin, L. Chen, M. Varma T, Z. Xu, and H. Su. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. *Advances in Neural Information Processing Systems*, 36:22226–22246, 2023.

[64] Y. Hong, K. Zhang, J. Gu, S. Bi, Y. Zhou, D. Liu, F. Liu, K. Sunkavalli, T. Bui, and H. Tan. Lrm: Large reconstruction model for single image to 3d. In *International Conference on Learning Representations*, volume 2024, pages 50678–50702, 2024.

[65] D. Tochilkin, D. Pankratz, Z. Liu, Z. Huang, A. Letts, Y. Li, D. Liang, C. Laforte, V. Jampani, and Y.-P. Cao. TripoSR: Fast 3D object reconstruction from a single image. *arXiv preprint arXiv:2403.02151*, 2024.

[66] J. Xu, W. Cheng, Y. Gao, X. Wang, S. Gao, and Y. Shan. InstantMesh: Efficient 3D mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191*, 2024.

[67] J. Tang, Z. Chen, X. Chen, T. Wang, G. Zeng, and Z. Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In *European Conference on Computer Vision*, pages 1–18. Springer, 2024.

[68] L. Wang, H.-X. Guo, X. Wang, F. Sun, K. Sun, P. Liu, H. Xiao, Z. Wang, G. Fu, E. Li, Y. Liu, and Y. Wang. SceneTransporter: Optimal transport-guided compositional latent diffusion for single-image structured 3D scene generation. In *International Conference on Learning Representations (ICLR)*, 2026.

[69] T. Dai, J. Wong, Y. Jiang, C. Wang, C. Gokmen, R. Zhang, J. Wu, and L. Fei-Fei. Automated creation of digital cousins for robust policy learning. *arXiv preprint arXiv:2410.07408*, 2024.

[70] H. Yu, B. Jia, Y. Chen, Y. Yang, P. Li, R. Su, J. Li, Q. Li, W. Liang, S.-C. Zhu, T. Liu, and S. Huang. MetaScenes: Towards automated replica creation for real-world 3D scans. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[71] Y. Chen, S. Guo, R. Jin, T. Yang, X. Cai, Y. Luo, M. Yang, M. Yu, L. Xu, and T. Xue. AnyRecon: Arbitrary-view 3D reconstruction with video diffusion model. *arXiv preprint arXiv:2604.19747*, 2026.

[72] S. Yin, J. Ge, Z. Z. Wang, C. Wang, X. Li, M. J. Black, T. Darrell, A. Kanazawa, and H. Feng. Vision-as-inverse-graphics agent via interleaved multimodal reasoning. *arXiv preprint arXiv:2601.11109*, 2026.

[73] T. H. Team. Hunyuan3D 2.0: Scaling diffusion models for high-resolution textured 3D asset generation. *arXiv preprint arXiv:2501.12202*, 2025.

[74] Y. Chen, T. He, D. Huang, W. Ye, S. Chen, J. Tang, Z. Cai, L. Yang, G. Yu, G. Lin, et al. Meshanything: Artist-created mesh generation with autoregressive transformers. In *International Conference on Learning Representations*, volume 2025, pages 51369–51389, 2025.

[75] H. Shan, K. Luo, M. Li, S. Zheng, Y. Fu, Z. Chen, and X. Huang. Interact3D: Compositional 3D generation of interactive objects. *arXiv preprint arXiv:2603.16085*, 2026.

[76] J. Hu, J. Guo, J. Cen, C. Yang, S. Li, and W. Shen. Worldact: Activating monolithic 3d worlds into interactive-ready object-centric scenes. *arXiv preprint arXiv:2605.15843*, 2026.

[77] C. Wang, D. Xu, Y. Zhu, R. Martin-Martin, C. Lu, L. Fei-Fei, and S. Savarese. DenseFusion: 6D object pose estimation by iterative dense fusion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[78] T. Hodan, F. Michel, E. Brachmann, W. Kehl, A. GlentBuch, D. Kraft, B. Drost, J. Vidal, S. Ihrke, X. Zabulis, et al. Bop: Benchmark for 6d object pose estimation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 19–34, 2018.

[79] P. J. Besl and N. D. McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. Spie, 1992.

[80] A. Simeonov, Y. Du, A. Tagliasacchi, J. B. Tenenbaum, A. Rodriguez, P. Agrawal, and V. Sitzmann. Neural descriptor fields: SE(3)-equivariant object representations for manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

[81] C. Pan, B. Okorn, H. Zhang, B. Eisner, and D. Held. TAX-Pose: Task-specific cross-pose estimation for robot manipulation. In *Conference on Robot Learning (CoRL)*, 2023.

[82] T. Tian, X. Kang, and Y.-L. Kuo. O^3Afford: One-shot 3D object-to-object affordance grounding for generalizable robotic manipulation. *arXiv preprint arXiv:2509.06233*, 2025.

[83] B. Eisner et al. AnyPlace: Learning generalizable object placement for robot manipulation. In *Conference on Robot Learning (CoRL)*, 2024.

[84] W. Liu, C. Paxton, T. Hermans, and D. Fox. StructFormer: Learning spatial structure for language-guided semantic rearrangement of novel objects. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

[85] W. Liu, T. Hermans, S. Chernova, and C. Paxton. Structdiffusion: Object-centric diffusion for semantic rearrangement of novel objects. In *Workshop on Language and Robotics at CoRL 2022*, 2022.

[86] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, A. Wahid, et al. Transporter networks: Rearranging the visual world for robotic manipulation. In *Conference on Robot Learning (CoRL)*, 2021.

[87] M. Shridhar, L. Manuelli, and D. Fox. CLIPort: What and where pathways for robotic manipulation. In *Conference on Robot Learning (CoRL)*, 2022.

[88] M. Shridhar, L. Manuelli, and D. Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning (CoRL)*, 2023.

[89] K. Mo, Y. Qin, F. Xiang, H. Su, and L. J. Guibas. Where2Act: From pixels to actions for articulated 3D objects. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[90] Y. Yang et al. ClutterGen: A cluttered scene generator for robot learning. In *Conference on Robot Learning (CoRL)*, 2024.

[91] Y. Yang et al. PhyScene: Physically interactable 3D scene synthesis for embodied AI. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[92] Y. Wang, X. Qiu, J. Liu, Z. Chen, J. Cai, Y. Wang, T.-H. Wang, Z. Xian, and C. Gan. ARCHITECT: Generating vivid and interactive 3D scenes with hierarchical 2D inpainting. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[93] J. Luo, J. Tang, R. Lu, and G. Zeng. SceneAssistant: A visual feedback agent for open-vocabulary 3D scene generation. *arXiv preprint arXiv:2603.12238*, 2026.

[94] X. Liu, Y.-W. Tai, and C.-K. Tang. Agentic 3D scene generation with spatially contextualized VLMs. *arXiv preprint arXiv:2505.20129*, 2025.

[95] M. Dalal, J. Yang, R. Mendonca, Y. Khaky, R. Salakhutdinov, and D. Pathak. Neural MP: A generalist neural motion planner. In *arXiv preprint arXiv:2409.05864*, 2024.

[96] R. Wang, S. Xu, Y. Dong, Y. Deng, J. Xiang, Z. Lv, G. Sun, X. Tong, and J. Yang. Moge-2: Accurate monocular geometry with metric scale and sharp details. *Advances in Neural Information Processing Systems*, 38:35928–35959, 2026.

[97] S. Laine, J. Hellsten, T. Karras, Y. Seol, J. Lehtinen, and T. Aila. Modular primitives for high-performance differentiable rendering. *ACM Transactions on Graphics (ToG)*, 39(6): 1–14, 2020.

[98] R. Tedrake and the Drake Development Team. Drake: Model-based design and verification for robotics, 2019. URL https://drake.mit.edu.

[99] P. E. Gill, W. Murray, and M. A. Saunders. Snopt: An sqp algorithm for large-scale constrained optimization. *SIAM review*, 47(1):99–131, 2005.

[100] E. Todorov, T. Erez, and Y. Tassa. MuJoCo: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.

[101] N. Hansen, S. D. Müller, and P. Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). *Evolutionary computation*, 11(1):1–18, 2003.

[102] N. E. Toklu, T. Atkinson, V. Micka, P. Liskowski, and R. K. Srivastava. EvoTorch: Scalable evolutionary computation in Python. *arXiv preprint*, 2023. https://arxiv.org/abs/2302.12600.

[103] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5745–5753, 2019.

[104] S. Deng, M. Yan, S. Wei, H. Ma, Y. Yang, J. Chen, Z. Zhang, T. Yang, X. Zhang, H. Cui, Z. Zhang, and H. Wang. GraspVLA: a grasping foundation model pre-trained on billion-scale synthetic action data. In *Proceedings of the 9th Conference on Robot Learning (CoRL)*, pages 1004–1029, 2025.

[105] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu, M. Martin, T. Nagarajan, S. K. Ramakrishnan, F. Ryan, J. Sharma, M. Wray, M. Xu, E. Z. Xu, C. Zhao, et al. Ego4D: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18995–19012, 2022.

[106] X. Wei, M. Liu, Z. Ling, and H. Su. Approximate convex decomposition for 3D meshes with collision-aware concavity and tree search. *ACM Transactions on Graphics (TOG)*, 41 (4):1–18, 2022.

[107] K. Mamou and F. Ghorbel. A simple and efficient approach for 3d mesh approximate convex decomposition. In *2009 16th IEEE international conference on image processing (ICIP)*, pages 3501–3504. IEEE, 2009.

[108] S. Bai, Y. Cai, R. Chen, K. Chen, X. Chen, Z. Cheng, L. Deng, W. Ding, C. Gao, C. Ge, W. Ge, Z. Guo, Q. Huang, J. Huang, F. Huang, B. Hui, S. Jiang, Z. Li, M. Li, M. Li, K. Li, Z. Lin, J. Lin, X. Liu, J. Liu, C. Liu, Y. Liu, D. Liu, S. Liu, D. Lu, R. Luo, C. Lv, R. Men, L. Meng, X. Ren, X. Ren, S. Song, Y. Sun, J. Tang, J. Tu, J. Wan, P. Wang, P. Wang, Q. Wang, Y. Wang, T. Xie, Y. Xu, H. Xu, J. Xu, Z. Yang, M. Yang, J. Yang, A. Yang, B. Yu, F. Zhang, H. Zhang, X. Zhang, B. Zheng, H. Zhong, J. Zhou, F. Zhou, J. Zhou, Y. Zhu, and K. Zhu. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025. URL https://arxiv.org/abs/2511.21631.

[109] Google DeepMind. Gemini 3.1 Flash Image model card. https://deepmind.google/models/model-cards/gemini-3-1-flash-image/, 2026.

[110] N. Carion, L. Gustafson, Y.-T. Hu, S. Debnath, R. Hu, D. Suris, C. Ryali, K. V. Alwala, H. Khedr, A. Huang, J. Lei, T. Ma, B. Guo, A. Kalla, M. Marks, J. Greer, M. Wang, P. Sun, R. Rädle, T. Afouras, E. Mavroudi, K. Xu, T.-H. Wu, Y. Zhou, L. Momeni, R. Hazra, S. Ding, S. Vaze, F. Porcher, F. Li, S. Li, A. Kamath, H. K. Cheng, P. Dollár, N. Ravi, K. Saenko,

P. Zhang, and C. Feichtenhofer. SAM 3: Segment anything with concepts, 2025. URL https://arxiv.org/abs/2511.16719.

[111] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[112] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, et al. SAM 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[113] Microsoft. Playwright: A framework for web testing and automation. https://playwright.dev/, 2020.

Supplementary Materials

GIF: Agentic Generation of Interactive and Functional Object Compositions for Robot Learning

A Methodology Details

This section provides implementation details for the modules introduced in Sec. 3, while keeping the presentation aligned with the pipeline abstraction used in the main paper. We follow the same order as the main text: multimodal input canonicalization, complementary 2D–3D asset construction (CoGen), geometry- and physics-guided pose refinement (GPRM), VLM-based scene selection, and policy data synthesis for VLA training.

A.1 Implementation Backends

Tab. S1 summarizes the model and software backends used by the agentic pipeline. The prompt templates in Sec. D define the task-level behavior of the LLM/VLM calls; the table below specifies the underlying modules.

Table S1: Implementation backends used in GIF.

Component	Backend
Pure-text LLM calls	gpt-4o-mini
VLM verifier, candidate voter, and VLM scene scorer	qwen3-vl-plus [108]
Reference-image generation and image editing	gemini-3.1-flash-image-preview [109] for input canonicalization, reference-image synthesis, and the CoGen erase branch
Instance-mask preparation	SAM3 [110] binary object masks derived from the canonical scene specification and reference image, then used as SAM3D conditioning masks
Image-to-3D lifting	SAM3D [10]
Monocular geometry for ICP targets	MoGeV2 [96]
Convex decomposition	CoACD [106]
Differentiable rendering	nvdiffrast [97]
Collision optimization and solver	Drake [98] with SNOPT [99]
Physics and simulation-aware search	MuJoCo [100] with CMA-ES from EvoTorch [101, 102]
Policy-data trajectory engine	GraspSimulator with CuRobo planning and Isaac Sim ray-traced rendering

A.2 Multimodal Input Canonicalization

The multimodal input agent in Sec. 3.1 is implemented as a unified goal-conditioned loop, rather than as a set of modality-specific branches. It accepts a task instruction, a reference scene image, optional object-level asset images, or any combination of these inputs, and returns the canonical pair $(I_{\text{ref}}, J_{\text{scene}})$. Here I_{ref} is the foreground-only reference image used by CoGen, and J_{scene} records the objects, their attributes, and the desired relation.

Tools and state. The agent uses two image operations: prompt-conditioned generation for text-only or asset-guided inputs, and instruction-conditioned editing when a usable scene image is already available. Both operations call the same image-generation and image-editing backends used later in CoGen, which keeps the visual distribution of I_{ref} consistent across the pipeline. The agent also maintains the current candidate image across iterations, enabling corrective edits to build on the previous result instead of reinitializing the scene.

Analyze, act, verify. The prompt in Sec. D.1 specifies the admissible relation vocabulary, scene-quality constraints, and output schema. Under this prompt, the agent repeatedly performs three operations: it first analyzes the available modalities to identify the objects, and relation; it then

generates or edits a candidate reference image, using any supplied images as appearance references; finally, it verifies the candidate against constraints on background simplicity, object count, visibility, contact plausibility, camera viewpoint, physical plausibility, and rigid-body separability. Failed candidates are repaired through targeted edits whenever possible, which helps preserve the user-specified objects while correcting localized violations.

Asset GLB pre-rendering. If the input includes a 3D asset rather than an RGB image, we first render it from a perspective viewpoint on a plain background and pass the rendered view to the same analyzer. This preprocessing step makes mesh and image inputs share the same visual interface, so the canonicalization prompt and verification rules do not depend on the original asset format.

Convex decomposition for simulation-ready assets. Once a reference image is accepted, the object entries in J_{scene} are used to prepare SAM3 [110] binary instance masks b_i^{ref} for the target objects. Each cleaned object image is lifted by SAM3D [10] into a textured visual mesh and an initial camera-frame transform. For simulation, we separately convert the visual mesh into a set of convex collision parts with CoACD [106]; the textured mesh is retained for rendering, while the convex decomposition is used for stable contact simulation in MuJoCo [100]. This separation between visual and collision geometry is used throughout the downstream pose pipeline.

A.3 CoGen: Complementary 2D-3D Generative Asset Construction

CoGen converts the canonical pair $(I_{\text{ref}}, J_{\text{scene}})$ into two mesh–pose candidates per object: a direct reconstruction $(m_i^{\text{ref}}, p_i^{\text{ref}})$ from the original pair image and a partner-removed reconstruction $(m_i^{\text{erase}}, p_i^{\text{erase}})$. GPRM later refines all cross-branch pairings, and the VLM selector chooses among the physically settled candidates. We highlight two details that make this complementary construction robust on long-tail contact pairs.

Synonym-driven retry for mask preparation. Mask preparation is sensitive to the surface form of an object name: a phrase such as "cup-rack tree" may fail even when visually equivalent prompts such as "mug rack" or "cup tree" succeed. We therefore add a synonym-driven retry stage. When the first mask-preparation attempt is empty or low confidence, an LLM proposes five alternative noun phrases using the prompt in Sec. D.2. The alternatives are tried in rank order until a mask passes basic area and shape checks. If all phrasings fail, the branch falls back to a coarse spatial prompt on I_{ref}. This mechanism improves coverage of long-tail household objects without introducing a hand-written vocabulary.

Erase-and-regenerate as a closed-loop agent. The erase branch constructs I_i^{erase} by removing the partner object o_j while preserving the geometry and pose of o_i. A single inpainting call is often insufficient for contact-rich pairs: it can shift the preserved object, leave traces of the erased object, or introduce background artefacts near the contact region. We instead use a closed-loop erase agent that alternates between image editing and visual verification. The verifier compares the edited image with the original and assigns a score under a pose-first rubric: pose drift of the preserved object receives the largest penalty, followed by incomplete removal, appearance changes, and background artefacts. The loop returns the first image above the acceptance threshold, or the best-scored image if the iteration budget is exhausted. The operative erase prompt is listed in Sec. D.3, and the scoring weights are summarized in Tab. S2.

Image-to-3D lifting and pose readout. Both branches use the same image-to-3D model to predict a textured mesh and a coarse camera-frame transform. The transform initializes $p_i^{a_i}$ for GPRM, while the mesh is converted into the visual and collision geometries described above. The reference branch conditions reconstruction on I_{ref} and the instance mask, whereas the erase branch reconstructs from the partner-removed image I_i^{erase}. Because the two branches fail in different ways, the four cross-branch pairings $\mathcal{C} = \{(a_1, a_2)\}$ provide meaningful alternatives for downstream geometric and physical selection.

Table S2: Score deductions used by the erase agent's verification rubric. Pose drift dominates because subsequent ICP and render-based optimization can absorb mild appearance change but not a shifted partner object.

Failure mode	Deduction
Preserved object pose changed (position, scale, or orientation differs)	-50
Target object still partially visible after erase	-25
Preserved object appearance changed while pose intact	-15
Background artefacts or unnatural fill	-10

A.4 GPRM Optimization Details

This section gives the optimization objectives underlying the GPRM stages summarized in Sec. 3.3.

ICP target cloud. For each object o_i we lift the reference image I_{ref} into a metric depth map $d \in \mathbb{R}_+^{H\times W}$ with a monocular geometry estimator [96], and back-project it through the camera intrinsics K to a scene cloud $\mathcal{P} = \{K^{-1}[u\,d_{uv},\, v\,d_{uv},\, d_{uv}]^\top \mid d_{uv} > 0\}$. The per-object target cloud $\mathcal{P}_i^{\text{tgt}} = \{p \in \mathcal{P} \mid (u,v) \in b_i^{ref,\text{erode}}\}$ is obtained by sampling $\mathcal{P}$ inside an eroded instance mask. The source cloud is sampled from the candidate mesh surface $m_i^{a_i}$ at pose $p_i^{a_i}$. We then run progressively tighter ICP rounds with shrinking distance thresholds, yielding the refined transform $p_i^{a_i,\text{icp}}$.

RenderOpt objective. Given $(m_i^{a_i}, p_i^{a_i})$, we rasterise the mesh with nvdiffrast [97] to obtain the per-object soft mask $\hat{b}_i \in [0,1]^{H\times W}$ and depth $\hat{d}_i \in \mathbb{R}_+^{H\times W}$, together with their composited counterparts $\hat{b}$ and $\hat{d}$ across all objects. The objective is

$$\begin{aligned}\mathcal{L}_{\text{rend}} = &\underbrace{\mathcal{L}_{\text{BCE}}(\hat{b}, b^{ref}) + \mathcal{L}_{\text{L1}}(\hat{d}, d)}_{\text{global}} + \underbrace{\lambda_p \mathcal{L}_{\text{pose-reg}} + \lambda_s \mathcal{L}_{\text{scale-reg}}}_{\text{regularization}} \\ &+ \sum_i \Big[\mathcal{L}_{\text{recall}}(\hat{b}_i, b_i^{ref}) + \mathcal{L}_{\text{prec}}(\hat{b}_i, b^{ref}) \\ &\qquad + \mathcal{L}_{\text{d-self}}(\hat{d}_i, d, b_i^{ref}) + \mathcal{L}_{\text{d-occ}}(\hat{d}_i, d, b_i^{ref})\Big]_{\text{per-object}},\end{aligned} \tag{S1}$$

where $b^{ref} = \bigcup_i b_i^{ref}$ and $\mathcal{L}_{\text{BCE}}$ is the focal variant [111]. The per-object terms enforce that each rendered mask covers its target ($\mathcal{L}_{\text{recall}}$), does not spill beyond valid image regions ($\mathcal{L}_{\text{prec}}$), matches observed depth within the object region ($\mathcal{L}_{\text{d-self}}$), and does not incorrectly occlude other objects ($\mathcal{L}_{\text{d-occ}}$). The regularizers anchor the pose and scale to the ICP initialization. The optimized pose $p_i^{a_i,\text{rend}}$ is accepted only if all four per-object loss components improve, which guards against degenerate optima.

ColliOpt formulation. Let $\text{cvx}(m_i, p_i)$ denote the convex decomposition of m_i at pose p_i. Starting from p_i^{rend}, ColliOpt seeks the nearest collision-free pose by solving

$$\begin{aligned}\min_{\{p_i\}} \; &\sum_i \Big[\big(r_i \ominus r_i^{\text{rend}}\big)^\top (r_i \ominus r_i^{\text{rend}}) + \|t_i - t_i^{\text{rend}}\|^2\Big] \\ \text{s.t.} \quad &d_{\min}(\text{cvx}(m_i, p_i),\, \text{cvx}(m_j, p_j)) \geq -\varepsilon, \quad \forall i \neq j,\end{aligned} \tag{S2}$$

where $r_i \ominus r_i^{\text{rend}}$ is the geodesic distance in $SO(3)$, $d_{\min}$ is the signed minimum convex-hull separation, and ε is the allowed interpenetration tolerance. We further add a ground-clearance constraint $t_{i,z} \geq z_{\text{floor}} + \varepsilon$ to prevent objects from sinking below the floor. The problem is built with Drake [98] and solved with SNOPT [99].

EvoOpt fitness. EvoOpt searches in the neighbourhood of $p_i^{a_i,\text{rend}}$ rather than $p_i^{a_i,\text{coll}}$, since the latter may already be at a bad local minimum. Each CMA-ES [101] candidate is a perturbation $x_i \in \mathbb{R}^9$ per body, encoding a translation delta in $\mathbb{R}^3$ and a rotation delta in the 6D representation [103].

Denote the perturbed pose $\tilde{p}_i^{a_i}(x_i)$. We score it by loading the perturbed MuJoCo [100] scene and running T_{sim} physics steps,

$$f(x_i) = \sum_{t=1}^{T_{\text{sim}}} \Big[\underbrace{w_q \|q_t - q_{t-1}\|}_{\text{drift}} + \underbrace{w_{\text{col}} \sum_{c_t} \max(0,\, -d_{c_t})}_{\text{collision}} + \underbrace{w_v \|v_t\|}_{\text{velocity}} \Big] + \underbrace{w_e \|\tilde{p}_i^{a_i}(x_i) - p_i^{a_i,\text{rend}}\|}_{\text{effort}}, \quad \text{(S3)}$$

where q_t is the generalized position at step t, d_{c_t} is the signed penetration depth of contact pair c_t, and v_t is the generalized velocity. The four terms penalize dynamic drift, residual interpenetration, residual kinetic energy, and deviation from the input pose. EvoOpt is launched only when the VLM selector rejects all candidates produced by the earlier three refinements.

A.5 Cross-Candidate Voting Example

Sec. 3.4 describes a two-stage VLM selector that first verifies each refined scene in isolation and then ranks the surviving candidates jointly. Fig. S1 visualizes a representative input to the cross-candidate stage, where each panel is itself a single-scene verification image. Detailed prompts for the verifier and the voter are provided in Sec. D.5 and Sec. D.6.

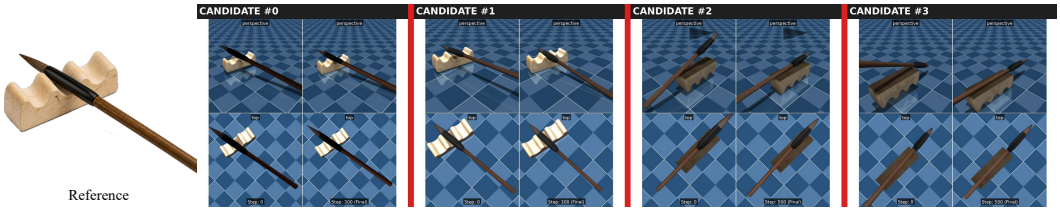


Figure S1: Cross-candidate voting. Per-candidate timeline images are tiled into a single montage with bold labels and red separators. The VLM selects the best candidate based on J_{scene}. In each panel, left or right columns show the initial or the gravity-settled final states of the simulation. Top and bottom rows show perspective and top-down views, respectively.

A.6 Policy Data Synthesis and VLA Training

Trajectory synthesis. For each VLM-verified terminal composition $\widehat{\mathcal{S}} = \{(m_i, p_i)\}_{i=1}^{2}$, we generate demonstrations by randomizing an initial tabletop scene and planning a trajectory that realizes the verified relative pose. Let m_1 be the manipulated object and m_2 be the reference object. The scene-generation pipeline defines the target relative transform $\Delta p = p_2^{-1} p_1$. For each rollout, we sample both the initial manipulated-object pose p_1^{rand} and the reference-object pose p_2^{rand}. The reference object is placed at this sampled pose and is not manipulated by the planner; the desired terminal pose of m_1 is obtained by transporting Δp into the randomized reference-object frame:

$$p_2^\star = p_2^{\text{rand}}, \qquad p_1^\star = p_2^{\text{rand}} \Delta p = p_2^{\text{rand}} p_2^{-1} p_1. \quad \text{(S4)}$$

The planner then produces a collision-free pick-and-place trajectory from p_1^{rand} to $p_1^\star$ with approach, grasp, lift, transfer, place, and retreat phases. Failed plans, unstable terminal states, and executions that break the target relation after MuJoCo settling are discarded before policy training.

Symmetry-aware target sampling. Many functional relations admit equivalent terminal poses; for example, a cylindrical object inserted into a container can often rotate about its long axis without changing the task outcome. We therefore run the symmetry analyzer of Sec. D.4 on each task object and obtain a task-conditioned symmetry group G_i over the admissible axes. During trajectory synthesis, the target pose is treated as an equivalence class $\{p_i^\star g \mid g \in G_i\}$ rather than a single rigid transform. For finite cyclic symmetries (C_2, C_3, C_4), we sample one of the admissible rotations; for continuous symmetry (C_∞), we sample a rotation angle uniformly around the accepted axis. This avoids penalizing functionally identical terminal states and reduces redundant supervision in the generated trajectories.

Policy architecture. We use the $\pi_{0.5}$ VLA architecture [1] as the policy template, but replace its original VLM backbone with Qwen3-VL-2B-Instruct; references to $\pi_{0.5}$ in this section therefore denote the architecture template rather than the original released checkpoint. The policy uses a flow-matching action expert for continuous action prediction. Training starts from the pretrained VLM weights, while the action expert is initialized from scratch. The vision encoder and visual merger are frozen, while the language backbone and action expert are trainable. The model observes one third-person main camera and one wrist camera, both resized to 224×224, together with robot proprioception. Actions and proprioception are represented in 7 dimensions, the proprioceptive history length is 2, the action chunk length is 4, and the control interval is $\Delta t = 0.3$s.

Table S3: VLA policy training configuration used for the real-robot policy-learning experiments.

Item	Setting
Policy template	$\pi_{0.5}$ architecture with Qwen3-VL-2B-Instruct VLM backbone
Prediction head	Flow-matching action expert, 256 action tokens
Trainable modules	Language backbone and action expert
Frozen modules	Vision encoder and visual merger
Training data	50,000 simulated trajectories per policy-learning task
Training steps	50,000 optimizer steps from pretrained VLM weights and a randomly initialized action expert
Camera inputs	Third-person main view and wrist view
Image resolution	224×224
Action / proprioception dimension	7 / 7
Proprioception history / action chunk	2 steps / 4 steps
Control interval	0.3s
Optimizer	AdamW, learning rate 1.6×10^{-4}, constant schedule
Regularization	Weight decay 0, gradient norm clipped to 1.0
Precision and parallelism	bfloat16, FSDP full-shard
Batch size	24 per GPU, 384 global
Hardware	2 nodes, 8 H100 GPUs per node

Policy-scaling protocol. For the asset-scaling experiment in Fig. 6, the number of training trajectories is held fixed across asset budgets: every policy is trained with 50,000 simulated trajectories for the corresponding task. The asset budget controls how many GIF-generated task-asset variants are used to synthesize those trajectories. Simulation success is evaluated on newly generated held-out assets that are not used in policy training, with 80 rollouts per task and asset budget. The simulator uses the same programmatic success predicates as data filtering: the sausage task requires the sausage's projected body to lie inside the paper box after release, and the pot-lid task requires the lid to be released, approximately horizontal, centered over the pot opening, overlapping the container footprint, and supported near the pot rim without sinking into the pot.

Simulation randomization. During data generation we randomize the initial object poses, distractor objects, table geometry, scene illumination, camera poses, and visual materials so that the policy sees a broad distribution around each verified terminal relation. The final rendered observations use the same two-view convention as the real robot: one third-person front camera and one wrist-mounted camera. Tab. S4 summarizes the settings that materially affect the generated demonstrations; low-level engineering parameters such as worker count, queue size, and batching are omitted because they affect throughput rather than the data distribution.

Table S4: Trajectory-generation and rendering settings for policy-learning data.

Item	Setting
Trajectory engine	GraspSimulator with MuJoCo physics, CuRobo motion planning, and Isaac Sim ray-traced rendering
Robot embodiment	Franka arm with a parallel gripper
Task specification	Object identities, the manipulated object, the reference object, and the GIF-verified terminal relative transform used to compute $p_1^\star$ for each sampled reference-object pose
Scene content	Target object pair plus sampled distractors, with 2–8 objects total per scene
Execution phases	Approach, grasp, lift, transfer, place or lower, release, and retreat; unreachable grasps, failed plans, unstable rollouts, and relation failures are discarded
Camera views	One randomized third-person front view and one randomized wrist-mounted view. The third-person camera perturbs azimuth, elevation, distance, look-at point, and roll around the workspace, while the wrist camera perturbs its mounting offset and orientation around the end effector. Samples that fail to keep the task objects and gripper visible are rejected
Rendered resolution	256×256 during data generation; images are resized to 224×224 for VLA training
Lighting	Randomized area-light color temperature, intensity, size, position, and orientation
Visual randomization	Randomized table, ground, background, robot, and object material parameters, with additional procedural texture variation for rendered backgrounds
Data-generation hardware	RTX 4090 GPU workers

Real-robot evaluation protocol. The real-robot setup uses the same observation interface as the simulation data: one fixed third-person main camera, one wrist camera, and robot proprioception. The sim-trained policy is deployed directly on the robot without real-world fine-tuning. Tab. S5 summarizes the evaluation setup and the initial-state protocol.

Table S5: Real-robot evaluation setup.

Item	Setting
Robot platform	Franka arm with a parallel gripper
Observation inputs	Intel RealSense fixed third-person main camera, ZED wrist-mounted camera, and robot proprioception
Main-camera placement	RealSense camera placed opposite the robot, within a 0–40° horizontal azimuth range around the robot-forward direction, matching the simulation camera-randomization envelope
Policy interface	Same as simulation training: two 224×224 RGB views, 7-D proprioception, 7-D actions, action chunk length 4, and control interval $\Delta t = 0.3$s
Deployment	Simulation-trained policy evaluated directly on the real robot, without real-robot fine-tuning
Tasks	Serving a sausage with a paper box; covering a pot with a lid
Initial object layouts	Target objects are randomly placed on the tabletop within the Franka reachable workspace. For each task and asset-budget condition, success is computed from 10 fixed initial layouts with three repeats per layout
Distractors	1–3 distractor objects per trial
Trials	30 real-robot trials per task and asset-budget condition, corresponding to 10 layouts $\times$ 3 repeats

For the sausage task, a trial is successful if the robot places the sausage into the paper box and the sausage remains inside without falling out; at most two repeated grasp attempts on the sausage are allowed within a trial. For the pot-lid task, a trial is successful if the robot grasps the lid within at most two attempts, places it accurately on the pot rim so that it covers the opening, and leaves

it stably supported by the pot without dropping it, falling outside the pot, or sinking into the pot interior.

B Experimental Design Details

This section gives the experimental details behind Sec. 4: how the benchmark is derived from egocentric manipulation data, how the automatic and human metrics are computed, how each baseline is adapted to the benchmark, and how the preliminary multi-object extension is evaluated.

B.1 Benchmark Scene Setup

Source corpus and extraction. We derive the contact-geometry taxonomy from Ego4D [105], using both free-form camera-wearer narrations and the structured FHO action annotations. From the narration stream, we retain camera-wearer utterances marked as action descriptions and keep their video identifiers, timestamps, and pass indices. From the FHO stream, we retain valid narrated actions together with their normalized verb, free-form verb, state transition, and temporal extent. The two sources are deduplicated at the annotation level and stored in a unified corpus for downstream filtering. This preprocessing yields millions of narration-level manipulation descriptions and roughly 0.6M valid FHO actions across thousands of videos.

Filtering for two-object contact events. The raw corpus includes many activities outside the scope of pairwise object interaction, such as navigation, conversation, and whole-body motion. We filter it in two stages. First, we keep only hand-object interaction verbs derived from the FHO verb taxonomy. Second, we remove descriptions that do not name a concrete manipulated object or that involve more than one explicit contact partner. The remaining examples describe one target object being manipulated with respect to one contact object, matching the pairwise scene specification used by GIF. We draw a random subset of $50{,}000$ filtered examples for taxonomy construction.

LLM-based clustering into eight contact-geometry classes. An LLM-based agent clusters the filtered examples by contact geometry. To reduce sensitivity to wording, the agent first summarizes each shard into candidate relation phrases, then merges equivalent phrases across shards, and finally clusters the merged phrases according to geometric contact pattern rather than linguistic surface form. The clustering prompt requires a mutually exclusive and exhaustive partition with six to ten classes. The stable solution contains eight classes: C.1 Containment, C.2 Capping, C.3 Resting, C.4 Leaning, C.5 Hooking, C.6 Slotting, C.7 Spanning, and C.8 Pegging. The prompts for shard summarization, phrase merging, and final clustering are listed in Sec. D.9.

Per-class task selection and variant generation. For each class, the agent selects three representative tasks subject to three constraints: the task must instantiate the class with two distinct everyday objects, the two objects must be separable rigid bodies, and the selected tasks should cover a mixture of household and workplace scenarios. This produces the $8 \times 3 = 24$ task suite. We then instantiate each task with five semantic variants that change material, color, or substyle while preserving the same relation, yielding $24 \times 5 = 120$ benchmark scenes. Sec. C lists the full task descriptions and shows the corresponding reference-image galleries.

B.2 Metrics

B.2.1 VLM-based scoring

For each generated scene, we render the gravity-settled MuJoCo state from a fixed perspective view and provide this render, the reference image, and the task description to the VLM scorer in Sec. D.8, using the backend listed in Tab. S1. The scorer returns integer Object Matching (O.M.) and Relationship Matching (R.M.) scores in $[0, 10]$. O.M. evaluates whether the generated assets preserve the appearance and geometry of the reference objects. R.M. evaluates whether the target relation is realized after settling, using a checklist covering contact geometry, common-sense placement, in-

terpenetration, and physical plausibility. Scores are averaged over the five variants of each task and then aggregated by relation class.

B.2.2 User study

This subsection describes the human evaluation behind Tab. 2. We follow the pairwise comparison format used by SceneSmith [3], adapted to our contact-geometry tasks.

Participants. Raters were recruited from a university population and had normal vision and basic familiarity with everyday object interactions. They were blind to the methods under comparison and to the study hypothesis. Before rating, each participant read a tutorial page explaining the task, the two metrics, and the *Equal* option (Fig. S2). Across all sessions, we collected more than 1,000 pairwise judgments distributed across the four baseline pairings and both prompt settings.

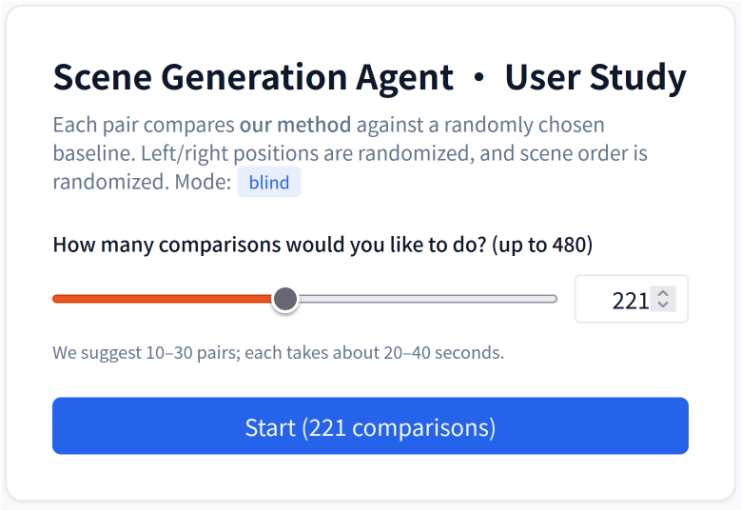


Figure S2: Tutorial landing page that introduces the rating task before any judgments are collected.

Rating interface and protocol. Each trial shows the input prompt and two anonymized outputs, one from our method and one from a baseline. Left–right placement is randomized and method names are hidden. The rater chooses which output is better, or selects *Equal* when the two are indistinguishable; an additional confirmation dialog is shown for ties (Fig. S3). O.M. is judged from the generated assets and their rendered appearance, while R.M. is judged from the gravity-settled MuJoCo state and its render. The four interface variants, crossing prompt modality with two view types, are shown in Fig. S4.

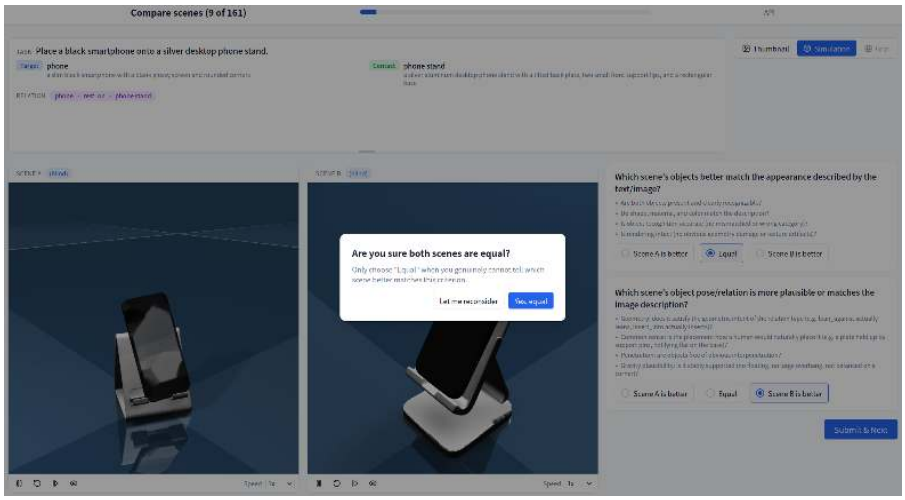
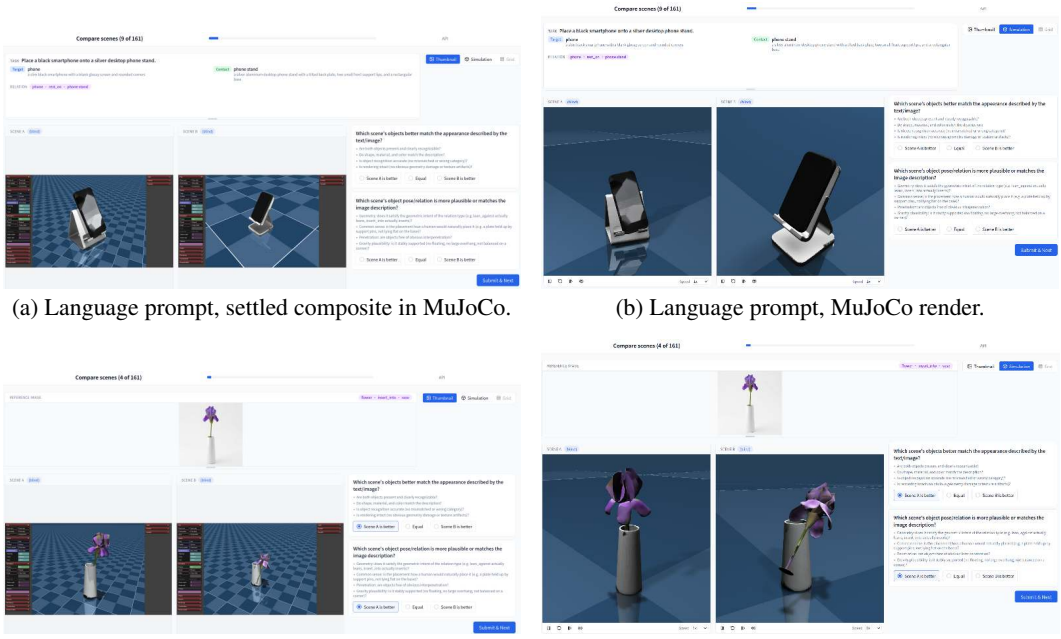


Figure S3: Confirmation dialog shown when a rater selects *Equal*.



(a) Language prompt, settled composite in MuJoCo.

(b) Language prompt, MuJoCo render.

(c) Image prompt, settled composite in MuJoCo.

(d) Image prompt, MuJoCo render.

Figure S4: Four rating panels obtained by crossing the prompt modality, language and image, with the view modality, settled composite in MuJoCo and MuJoCo render. The MuJoCo render supports interactive inspection, including running, pausing, and resetting the simulation.

Aggregation and statistical analysis. For O.M., tied votes contribute half a win to each method. For R.M., ties are excluded before computing the win rate, so the reported value reflects decisive relation-quality preferences. For every comparison we report the win rate of our method, the 95% Wilson confidence interval, and Cohen's h effect size against the 50% null.

Quality control. Each session begins with a short calibration set containing clearly successful and clearly unsuccessful scene pairs; raters below the calibration threshold are excluded from the final analysis. Trial order, side assignment, and method-baseline pairing are independently randomized, and method names remain hidden throughout the study.

B.3 Baselines and Comparison Protocol

This subsection specifies how each baseline is adapted to the $24 \times 5 = 120$ benchmark scenes. Whenever a baseline uses an image-to-3D backend, we replace that backend with SAM3D [10] in our evaluation, so that comparisons focus on scene composition, relation reasoning, and physical placement rather than on differences in asset-generation quality.

Table S6: Summary of baseline adaptations for the pairwise contact-geometry benchmark. All exported scenes are settled in MuJoCo and evaluated with the same renderer and scoring protocols described in Sec. B.2.

Baseline	Original assumption	Benchmark mismatch	Adaptation	Fairness control / residual caveat
RoLA [11]	Reconstructs simulation-ready assets from a single RGB image and instance masks.	Requires image inputs and masks, whereas the benchmark includes canonical language and image specifications.	Provide masks through the SAM2 annotation interface distributed with RoLA; pass the resulting crops through RoLA's reconstruction and pose-composition stages with SAM3D as the shared 3D backend.	Uses RoLA's expected input format; mask preparation is applied consistently across RoLA runs.
TabletopGen [8]	Generates tabletop assets independently and places them on a fixed support surface.	Assumes a tabletop support surface, while some benchmark relations are not naturally table-only.	Use table-conditioned benchmark images and replace the image-to-3D backend with SAM3D, leaving the remaining segmentation, asset-generation, and layout stages unchanged.	Preserves the released pipeline except for the shared 3D backend; the protocol satisfies TabletopGen's table assumption.
RoLA / TabletopGen image protocol	Both systems expect an image of a tabletop scene.	Foreground-only references omit the support surface required by these baselines.	Edit each foreground reference into a canonical white-table scene while preserving object viewpoint and adding no unrelated items; for wall-hanging cases, add a minimal wall support.	The same composites are used for RoLA, TabletopGen, and Ours-Img.
GenieSim [7]	Interactive generation with optional human-in-the-loop editing and asset acquisition.	Human intervention cannot be used in a fully automated 120-scene comparison.	Automate the released web interface with Playwright, fix the model choice, disable human editing, and append an instruction requiring asset acquisition rather than early termination.	Gives every run an automatic export path; does not evaluate manual interactive correction.
SceneSmith [3]	Hierarchical room-level scene synthesis with room context and routing strategies.	The benchmark evaluates pairwise rigid-object contact rather than room-scale layout or articulated/thin-covering routing.	Evaluate the manipuland-placement stage in a fixed empty room with a single work desk; supply object names as required manipulands and express the task as a tabletop placement constraint.	Matches the pairwise object scope; room-level context and routing strategies outside the benchmark scope are not evaluated.

RoLA. RoLA [11] reconstructs simulation-ready assets from a single RGB image and composes them into a physical scene. The released implementation supports SAM3D [10] as an alternative to its original Hunyuan3D [73] backend, but it expects instance masks as input. We therefore provide masks using the SAM2 [112] annotation interface distributed with their source code, applying the same mask preparation procedure across all RoLA inputs. The resulting object crops are passed to RoLA's mesh reconstruction and pose-composition stages.

TabletopGen. TabletopGen [8] generates each tabletop asset independently and places the assets on a fixed support surface. We replace its original Hunyuan3D [73] backend with SAM3D [10] and otherwise leave its segmentation, asset generation, and layout stages unchanged.

Image prompt protocol for table-required baselines. RoLA and TabletopGen expect an image of a tabletop scene. To match their input assumptions, we synthesize a table-conditioned image prompt for each benchmark instance. The image-editing backend in Tab. S1 receives the foreground reference and a canonical white-table image, and is instructed to place the same objects on the table while preserving their viewpoint and adding no unrelated items. For wall-hanging relations, the

editor is additionally instructed to add a minimal wall support on the table and hang the object from it. These composites serve as the image inputs to RoLA, TabletopGen and "Ours-Img".

GenieSim. GenieSim [7] is evaluated without its human-in-the-loop editing stage. Since the released system is primarily exposed through an interactive web interface, we automate the same interaction with Playwright [113]: the task description is submitted to the generator panel, the model choice is fixed across all runs, and the exported MuJoCo scene is collected for evaluation. We append a short instruction that requires the system to acquire or synthesize the necessary assets rather than terminating when a library asset is unavailable. Each generated scene is then settled under gravity and rendered with the same evaluation renderer used for all methods.

SceneSmith. SceneSmith [3] targets hierarchical room-level synthesis. To align its scope with our pairwise benchmark, we evaluate only the manipuland-placement stage and provide a fixed empty room containing a single rectangular work desk. Objects names are supplied as required manipulands, and the task description is expressed as a tabletop placement constraint on that desk. SceneSmith then runs its standard asset routing, image-editing, and pose-composition stages before export to MuJoCo. We disable routing strategies for articulated objects and thin coverings, since our benchmark focuses on separable rigid-body pairs.

B.4 Preliminary Extension to Multi-Object Scenes

Fig. S5 shows the full nine-step expansion summarized in Sec. 4.5.

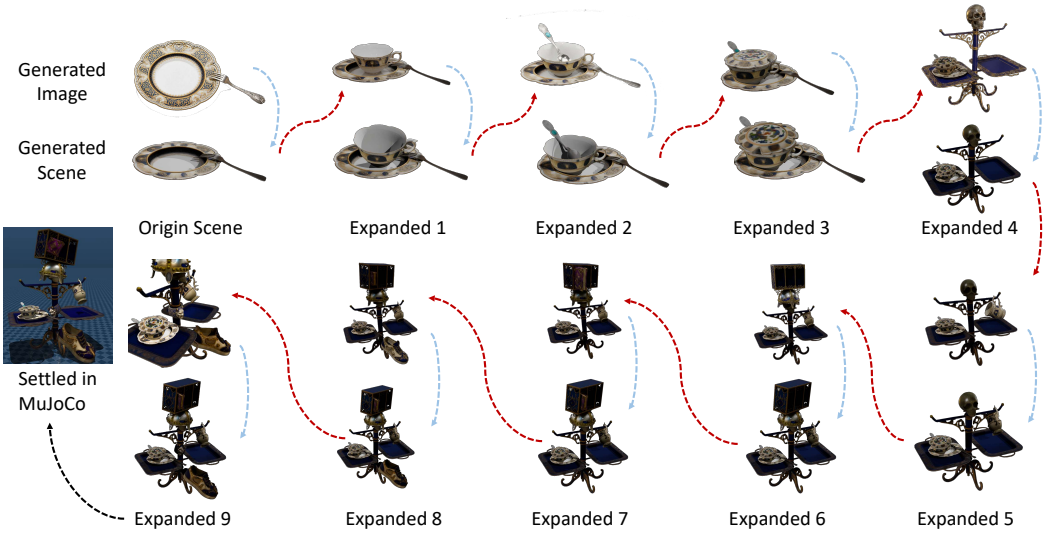


Figure S5: Preliminary extension to multi-object scenes. Starting from a verified fork-on-plate pair, the multimodal agent inserts one new object at a time, and CoGen together with GPRM is rerun on the newcomer while the merged geometry of all previously placed objects is held fixed. The nine insertions are: an empty cup placed on the plate, a spoon set into the cup, a notched lid capping the cup, an ornate skull-shaped rack standing beside the plate, a mug hooked onto a side branch, a helmet-shaped bookshelf resting on the skull, a magic book slotted into the shelf, a skateboard shoe leaning against the central pole, and a ring pegged around it. Together, these additions span all eight taxonomy classes from **C.1 Containment** to **C.8 Pegging**.

C Task Suite Descriptions and Scene Galleries

This section lists the $8 \times 3 = 24$ benchmark tasks and the five semantic variants used for each task, giving the natural-language prompt for every one of the 120 reference scenes. The galleries are organized by contact-geometry class: each 3×5 grid uses one row per task and five columns for its variants.

C.1 C.1 Containment

Task 1: Place a flower into a vase.

- The leafless flower stem is inserted vertically through the mouth of the vase, with the blossom fully visible above the rim. The flower is a single real red rose with one long green stem, a full red blossom, no leaves anywhere on the stem, and natural green sepals allowed beneath the blossom. The vase is a narrow transparent glass bud vase with a round base, a slim neck, and an open circular mouth.
- The leafless flower stem is inside the vase opening and the white tulip bloom stands above the vase. The flower is a single real white tulip with a smooth pale green stem, a closed cup-shaped white bloom, no leaves attached to the stem, and natural sepals allowed near the bloom. The vase is a small matte blue ceramic vase with a rounded body, narrow neck, and open top.
- The leafless flower stem is inserted into the center of the vase opening, with the yellow flower head clear of the rim. The flower is a single real yellow daisy with a dark center, a thin green stem, no leaves attached to the stem, and small natural sepals allowed under the flower head. The vase is a short brushed stainless steel vase with vertical ribbing and a flared open lip.
- The leafless flower stem goes down into the vase mouth and the purple iris blossom is upright above the vase. The flower is a single real purple iris with a slender green stem, layered purple petals, no leaves on the stem, and natural sepals allowed at the base of the flower. The vase is a tall glossy white porcelain vase with a tapered cylindrical body and a simple open mouth.
- The bare leafless flower stem is inserted into the square opening of the empty vase while the pink gerbera flower head remains outside and above it. The flower is a single real pink gerbera daisy with a round flower head, a long bare green stem, no leaves, no side shoots, no bracts along the stem, and only a small natural calyx directly under the flower head. The vase is a clear square glass vase with thick walls, a flat base, and an open square top.

Task 2: Place a pen into a multi-compartment desktop organizer.

- The pen is standing inside one narrow rectangular compartment of the white modern desktop organizer. The pen is a single slim black ballpoint pen with a smooth cylindrical barrel and a small clip. The desktop organizer is a multi-compartment desktop organizer with an open top structure, white matte plastic finish, clean rectangular sections of different sizes, crisp edges, and a compact minimalist geometric form.
- The pen stands inside the tallest rear compartment of the dark gray complex organizer box. The pen is a glossy white gel pen with a transparent grip section, a silver clip, and a fine conical tip. The complex organizer box is a dark gray stepped plastic organizer box with three different-height rectangular compartments, rounded internal dividers, and a low front storage tray, all compartments empty.
- The pen rests inside one narrow vertical compartment of the wooden desktop organizer. The pen is a single red marker style pen with a rounded cap and simple cylindrical body. The desktop organizer is a multi-compartment desktop organizer made of light wood, with open top rectangular sections of different sizes, clean sharp edges, a compact block-like structure, and a modern desk accessory appearance.
- The pen is inserted diagonally into the tall rectangular compartment of the acrylic complex organizer box. The pen is a navy blue rollerball pen with a straight cylindrical barrel, a rounded cap end, and a subtle metal clip. The complex organizer box is a clear smoky acrylic organizer box with visible thick internal dividers, one tall rectangular compartment, two small square wells, and a low front tray, all compartments empty.
- The pen is inserted into the deep oval opening of the ceramic complex organizer box. The pen is a yellow fine liner pen with a slim plastic barrel, a small cap, and a simple pointed

tip. The complex organizer box is a matte cream ceramic organizer box with curved wave-like internal partitions, one deep oval opening, two smaller rounded pockets, and a shallow empty tray area.

Task 3: Place a flowering plant pot into a hanging flower pot stand.

- The flowering plant pot is sitting inside the circular basket ring of the black hanging flower pot stand. The flowering plant pot is a beige ceramic flower pot containing a compact green leafy plant with several small yellow flowers. The hanging flower pot stand is a black metal hanging flower pot stand with a circular basket ring, vertical wire side bars, a flat round support base, and two curved hanging hooks, empty except for the pot.
- The flowering plant pot sits in the open cage of the black wire hanging flower pot stand. The flowering plant pot is a glossy cream ceramic flower pot with a leafy green plant and small white flowers rising above the rim. The hanging flower pot stand is a simple black wire hanging flower pot stand with a circular top ring, open cage-like sides, a round base support, and two slim upward hanging hooks.
- The flowering plant pot is seated inside the brass circular basket of the hanging flower pot stand. The flowering plant pot is a small matte charcoal flower pot containing broad green leaves and several red flowers, the pot body fully inside the frame. The hanging flower pot stand is a decorative brass-colored hanging flower pot stand with a circular basket rim, curved vertical ribs, a raised round bottom ring, and two symmetrical hanging hooks.
- The flowering plant pot rests on the shallow circular tray inside the black hanging flower pot stand. The flowering plant pot is a white cylindrical flower pot with compact green foliage and tiny orange flowers. The hanging flower pot stand is a compact black metal hanging flower pot stand with a shallow circular tray, short vertical guard bars, a top retaining ring, and rear hooks shaped for a balcony rail.
- The flowering plant pot is nested inside the wide circular basket of the green hanging flower pot stand. The flowering plant pot is a tan clay flower pot containing lush green leaves and a few small blue flowers. The hanging flower pot stand is a wide dark green metal hanging flower pot stand with an open circular basket, a low bottom ring, spaced vertical rods, and two rounded hook arms.

Figure S6: Reference scenes for **C.1 Containment**. Top to bottom rows show flower-vase, pen-organizer, and pot-hanging-stand variants.

C.2 C.2 Capping

Task 1: Fit a pot lid onto a pot.

- The lid is centered on the pot rim and covers the open circular top of the pot. The lid is a round transparent glass lid with a shiny metal rim and a black knob handle centered on top. The pot is a medium stainless steel pot with two side handles, straight walls, and an open circular rim.
- The lid sits flush on the upper rim of the pot and fully covers its opening. The lid is a heavy black cast iron lid with a raised circular knob and a slightly domed profile. The pot is a black cast iron Dutch oven with short side handles and a wide open mouth.
- The lid is aligned over the pot opening and rests on the rim as a cover. The lid is a glossy red enamel lid with a silver knob and a shallow dome. The pot is a matching glossy red enamel pot with a cream interior, loop side handles, and an open rim.
- The lid covers the saucepan opening with its edge seated on the pot rim. The lid is a flat brushed steel lid with a low black grip and a thin circular edge. The pot is a small stainless steel saucepan with one long handle and a clean open circular top.
- The lid is resting on the top rim of the stockpot and covers the opening. The lid is a round silver metal lid with a small steam vent hole and a dark round knob. The pot is a tall deep stockpot made of brushed stainless steel with two loop handles and an open top.

Task 2: Fit a teacup lid onto its teacup.

- The teacup lid is seated on the circular rim of the matte black teacup. The teacup lid is a complete matching teacup lid with a small central knob, smooth rim, and clean unbranded surface, matte black accent details. The teacup is one complete matte black ceramic teacup with a visible handle, round open rim, stable base, and no tea bag or extra props.
- The teacup lid is seated on the circular rim of the natural wood teacup. The teacup lid is a complete matching teacup lid with a small central knob, smooth rim, and clean unbranded surface, natural wood accent details. The teacup is one complete natural wood ceramic teacup with a visible handle, round open rim, stable base, and no tea bag or extra props.
- The teacup lid is seated on the circular rim of the brushed metal teacup. The teacup lid is a complete matching teacup lid with a small central knob, smooth rim, and clean unbranded surface, brushed metal accent details. The teacup is one complete brushed metal ceramic teacup with a visible handle, round open rim, stable base, and no tea bag or extra props.
- The teacup lid is seated on the circular rim of the deep green teacup. The teacup lid is a complete matching teacup lid with a small central knob, smooth rim, and clean unbranded surface, deep green accent details. The teacup is one complete deep green ceramic teacup with a visible handle, round open rim, stable base, and no tea bag or extra props.
- The teacup lid is seated on the circular rim of the cream colored teacup. The teacup lid is a complete matching teacup lid with a small central knob, smooth rim, and clean unbranded surface, cream colored accent details. The teacup is one complete cream colored ceramic teacup with a visible handle, round open rim, stable base, and no tea bag or extra props.

Task 3: Insert a wine cork into a red wine bottle.

- The wine cork is inserted into the mouth of the red wine bottle. The wine cork is one natural cylindrical wine cork with visible cork texture, flat top, and no label or text. The red wine bottle is one complete dark glass red wine bottle with a clean neck, round bottle mouth, simple unbranded body, and no label text.
- The natural wine cork is inserted into the mouth of the red wine bottle. The wine cork is one natural cylindrical wine cork with visible cork fibers and pores, a flat cut top and slightly irregular sides, no branding or text, clean and unbroken. The red wine bottle is one

complete realistic dark glass red wine bottle with a round mouth opening and a neck sized for the cork; includes a practical applied paper label with generic winery-style graphics and subtle surface wear; visible glass reflections and fine texture on the bottle body.

- The wine cork is inserted into the mouth of the red wine bottle. The wine cork is one natural cylindrical wine cork with visible cork fibers and small irregular pores, flat cut top, intact rounded bottom, no text or markings, matte texture. The red wine bottle is one complete realistic dark glass red wine bottle with a clean round mouth (open neck), visible embossed glass imperfections, and a realistic paper wine label with decorative colors and subtle texture but no readable text.
- The wine cork is inserted into the mouth of the red wine bottle. The wine cork is one natural cylindrical wine cork with visible cork pores and slight unevenness, a flat top surface, and no markings, held as a single complete object. The red wine bottle is one complete red wine bottle made of dark glass with a clean realistic bottle neck and a round bottle mouth; the bottle body has a visible textured paper label with subtle wrinkles and printed-style artwork but no readable text, and the label is fully attached without any extra objects.
- The wine cork is inserted into the mouth of the red wine bottle. The wine cork is one natural cylindrical wine cork with visible cork texture, flat top, and no label or text. The red wine bottle is one complete dark glass red wine bottle with a clean neck, round bottle mouth, simple unbranded body, and no label text.

Figure S7: Reference scenes for **C.2 Capping**. Top to bottom rows show pot-lid, teacup-lid, and wine-cork variants.

C.3 C.3 Resting

Task 1: Rest a hat on a mannequin head.

- The hat is fitted over the crown of the mannequin head with the brim encircling the forehead. The hat is a tan woven straw hat with a medium circular brim and a shallow rounded crown. The mannequin head is a smooth white mannequin head with simplified facial features and a visible neck base.
- The hat sits on top of the mannequin head with the bill projecting forward over the face. The hat is a black baseball cap with a curved front bill and a soft rounded crown. The mannequin head is a matte gray mannequin head with minimal facial contours and an upright neck.
- The hat covers the upper part of the mannequin head and wraps around the forehead. The hat is a red ribbed knit beanie with a folded cuff and a rounded closed top. The mannequin head is a cream foam mannequin head with a simple nose, chin, and stable flat base.

- The hat is seated over the top of the mannequin head with the brim drooping around the sides. The hat is a navy fabric bucket hat with a soft downward brim and stitched crown panels. The mannequin head is a beige mannequin head with smooth facial features and a short cylindrical neck.
- The hat rests over the crown of the mannequin head and covers the top surface. The hat is a gray felt fedora with a pinched crown, narrow black band, and structured brim. The mannequin head is a black mannequin head with subtle facial contours, ears, and a flat neck base.

Task 2: Rest a cup of coffee on a coffee machine.

- The cup of coffee sits on the drip tray directly under the spout of the warm white coffee machine. The cup of coffee is one complete small cup filled with dark coffee, visible coffee surface, simple handle, and no spoon or saucer. The coffee machine is one complete warm white compact coffee machine with a visible central spout, drip tray, simple body, and no brand text or extra cup.
- The cup of coffee sits on the drip tray directly under the spout of the natural wood coffee machine. The cup of coffee is one complete small cup filled with dark coffee, visible coffee surface, simple handle, and no spoon or saucer. The coffee machine is one complete natural wood compact coffee machine with a visible central spout, drip tray, simple body, and no brand text or extra cup.
- The cup of coffee sits on the drip tray directly under the spout of the brushed metal coffee machine. The cup of coffee is one complete small cup filled with dark coffee, visible coffee surface, simple handle, and no spoon or saucer. The coffee machine is one complete brushed metal compact coffee machine with a visible central spout, drip tray, simple body, and no brand text or extra cup.
- The cup of coffee sits on the drip tray directly under the spout of the translucent smoky acrylic coffee machine. The cup of coffee is one complete small cup filled with dark coffee, visible coffee surface, simple handle, and no spoon or saucer. The coffee machine is one complete translucent smoky acrylic compact coffee machine with a visible central spout, drip tray, simple body, and no brand text or extra cup.
- The cup of coffee sits on the drip tray directly under the spout of the deep green coffee machine. The cup of coffee is one complete small cup filled with dark coffee, visible coffee surface, simple handle, and no spoon or saucer. The coffee machine is one complete deep green compact coffee machine with a visible central spout, drip tray, simple body, and no brand text or extra cup.

Task 3: Rest an egg in an egg carton.

- The single egg is sitting inside one molded cup of the matte black egg carton. The egg is one complete smooth chicken egg, oval shape, pale shell. The egg carton is one empty matte black egg carton section with multiple molded cups, raised dividers, and no other eggs.
- The single egg is sitting inside one molded cup of the warm white egg carton. The egg is one complete smooth chicken egg, oval shape, pale shell. The egg carton is one empty warm white egg carton section with multiple molded cups, raised dividers, and no other eggs.
- The single egg is sitting inside one molded cup of the brushed metal egg carton. The egg is one complete smooth chicken egg, oval shape, pale shell. The egg carton is one empty brushed metal egg carton section with multiple molded cups, raised dividers, and no other eggs.
- The single egg is sitting inside one molded cup of the translucent smoky acrylic egg carton. The egg is one complete smooth chicken egg, oval shape, pale shell. The egg carton is

one empty translucent smoky acrylic egg carton section with multiple molded cups, raised dividers, and no other eggs.

- The single egg is sitting inside one molded cup of the deep green egg carton. The egg is one complete smooth chicken egg, oval shape, pale shell. The egg carton is one empty deep green egg carton section with multiple molded cups, raised dividers, and no other eggs.

Figure S8: Reference scenes for **C.3 Resting**. Top to bottom rows show hat-mannequin, coffee-machine, and egg-carton variants.

C.4 C.4 Leaning

Task 1: Lean a phone against a phone stand.

- The phone is upright against the tilted back plate of the phone stand with its bottom edge resting on the support lips. The phone is a slim black smartphone with a blank glossy screen and rounded corners. The phone stand is a silver aluminum desktop phone stand with a tilted back plate, two small front support lips, and a rectangular base.
- The phone stands upright on the phone stand with its bottom edge seated on the horizontal ledge behind the front retaining lips and its back leaning against the angled support panel. The phone is a white smartphone with a dark inactive glass screen, a thin white bezel, rounded rectangular corners. The phone stand is a realistic compact black folding plastic desktop phone stand with two triangular side supports, a hinged angled back panel, a horizontal lower ledge, small front retaining lips, and a stable base.
- The phone is seated in the slot of the phone stand and leans back against the wooden support. The phone is a smartphone in a muted green protective case with a flat dark glass screen. The phone stand is a light wood phone stand with a slotted base and a vertical angled backing board.
- The phone is supported by the front prongs of the phone stand and rests against the angled support. The phone is a large dark gray smartphone with a blank reflective screen and rounded edges. The phone stand is a white adjustable phone stand with a hinged vertical support, two front prongs, and a broad base.
- The phone rests on the curved lower shelf of the transparent phone stand, held by the molded front lip, while the back of the phone leans against the clear slanted panel. The phone is a blue smartphone with a black inactive glass screen, slim side buttons, rounded corners. The phone stand is a realistic transparent acrylic desktop phone stand made from a single bent clear sheet, with a slanted back panel, broad stable base, curved lower shelf, and molded front lip that prevents slipping.

Task 2: Lean a shoe against a sloped shoe display stand.

- The cream sneaker rests along the dark charcoal sloped ramp, with its sole flat against the incline and its front outsole blocked by the vertical front stop wall. The shoe is one complete cream athletic sneaker with beige suede panels, white laces, a padded collar, and a gum-colored outsole. The sloped shoe display stand is one dark charcoal wedge-shaped shoe display stand with a broad inclined top ramp and a vertical rectangular stop wall at the lower front edge, the stop wall touching the front outsole to prevent sliding.
- The black sneaker leans back on the white curved sloped surface while the vertical front stop wall catches the lower sole and prevents the shoe from sliding down. The shoe is one complete black low-top sneaker with black laces, smooth leather-like upper, rounded toe, and white midsole. The sloped shoe display stand is one glossy white sloped shoe display stand with a gently curved concave ramp surface and a short vertical front stop wall rising at the lower edge, stable rectangular base.
- The red running shoe rests on the brushed aluminum incline, and the vertical metal front stop plate supports the toe area so the shoe cannot slide forward. The shoe is one complete red running shoe with breathable mesh texture, black laces, sculpted white midsole, and black outsole. The sloped shoe display stand is one brushed aluminum sloped display stand with a triangular side profile, flat inclined ramp, and a vertical metal stop plate at the low front edge that is tall enough to block the shoe toe.
- The pastel pink sneaker rests on the clear acrylic sloped ramp, and the transparent vertical front stop panel blocks the lower sole from sliding forward. The shoe is one complete pastel pink sneaker with matching laces, soft fabric panels, rounded toe, and pale cream sole. The sloped shoe display stand is one clear translucent acrylic shoe display stand with a smooth sloped ramp, visible transparent side edges, and a vertical clear front stop panel at the low edge.
- The burgundy sneaker rests on the cream triangular sloped plane, and the raised vertical stop wall at the lower front edge prevents the shoe from sliding down. The shoe is one complete burgundy sneaker with dark red laces, soft suede-like upper, rounded toe, and white chunky sole. The sloped shoe display stand is one cream colored triangular wedge shoe display stand with a sloped top plane and a raised vertical stop wall at the lower front edge.

Task 3: Lean a bamboo steamer lid against a bamboo steamer basket.

- The bamboo steamer lid is slightly open at front and leaning against the rim of the empty bamboo steamer basket. The bamboo steamer lid is one complete circular bamboo steamer lid with woven top pattern, raised rim, small central handle or cross rib, clean and empty. The bamboo steamer basket is one complete empty round pale bamboo bamboo steamer basket with cylindrical side wall, visible open interior, woven base.
- The lid is slanted and leaning against the rim of the empty basket, with the lid lower edge resting on the basket rim and side wall so it is physically supported rather than floating. The bamboo steamer lid is one complete circular bamboo steamer lid with a deep cinnabar red stained bamboo rim, pale woven bamboo center, thin gold-toned band around the raised rim, small central cross rib. The bamboo steamer basket is one complete empty round bamboo steamer basket with a deep cinnabar red stained cylindrical side wall, thin gold-toned bands near top and bottom, visible open interior with pale woven bamboo base.
- The lid leans off-center at a diagonal angle against the basket rim, with a clear contact point between the lid edge and the basket's upper rim while the basket interior remains visible. The bamboo steamer lid is one complete circular bamboo steamer lid with a muted jade green painted bamboo rim, a woven top panel using alternating pale straw and light green strips, raised outer rim, simple cross rib handle. The bamboo steamer basket is one complete empty round bamboo steamer basket with muted jade green painted cylindrical side wall, pale bamboo interior, visible woven base, slightly darker green binding bands.

- The bamboo steamer lid leans at about 45° against the rim of the empty basket, with the lower rim of the lid touching the basket rim and the lid partially over but not covering the whole opening. The bamboo steamer lid is one complete circular bamboo steamer lid with a deep indigo-blue dyed bamboo rim, pale woven center panel, and small white geometric triangle motifs around the raised rim, small central cross rib. The bamboo steamer basket is one complete empty round bamboo steamer basket with deep indigo-blue dyed cylindrical side wall decorated with small white geometric triangle motifs, visible open interior and pale woven base.
- The lid is slanted over the open steamer and leaning against the upper rim, with the lid edge clearly supported by the basket rim and side wall and the empty interior still visible. The bamboo steamer lid is one complete circular bamboo steamer lid with a soft blue-gray dyed bamboo outer rim, pale woven center panel, raised rim, small central cross rib. The bamboo steamer basket is one complete empty round bamboo steamer basket with soft blue-gray dyed cylindrical side wall, pale bamboo interior and woven base, subtle darker blue-gray edge bands.

Figure S9: Reference scenes for **C.4 Leaning**. Top to bottom rows show phone-stand, shoe-stand, and bamboo-steamer-lid variants.

C.5 C.5 Hooking

Task 1: Hook a cup on a cup-rack tree.

- The cup handle is looped over one angled peg of the cup rack tree so the cup hangs freely beside the pole. The cup is a white ceramic mug with a round open top, smooth cylindrical body, and a C-shaped side handle. The cup rack tree is a light wooden cup rack tree with a central vertical pole, a round base, and several short upward angled pegs.
- The cup is suspended from one hook of the cup rack tree by its handle. The cup is a glossy blue ceramic cup with a thick side handle and a slightly flared rim. The cup rack tree is a black metal cup rack tree with a circular base, a straight center post, and four curved hook-like arms.
- The cup handle hangs over a horizontal peg of the cup rack tree while the cup body remains below the peg. The cup is a yellow ceramic mug with a solid handle, cylindrical body, and thick bottom. The cup rack tree is a shiny chrome cup rack tree with a weighted round base and multiple short horizontal pegs.
- The cup is hanging from a short peg of the cup rack tree through its small handle. The cup is a small cream espresso cup with a tiny loop handle, round body, and open top. The

cup rack tree is a bamboo cup rack tree with a square base, one vertical stem, and short staggered pegs.

- The cup handle is hooked onto one peg of the cup rack tree and the cup hangs downward from it. The cup is a red ceramic mug with a smooth rounded body, open circular mouth, and large side handle. The cup rack tree is a white metal cup rack tree with a stable round base, central upright pole, and six rounded pegs.

Task 2: Hook a hat on a coat rack.

- The hat is hanging from the outer part of exactly one long straight upper hook on the matte black coat rack; at least half of that hook shaft and its tip remain visible outside the hat, and all other hooks are empty and do not touch, support, pierce, or overlap the hat. The hat is one complete hat with a rounded crown and visible brim, soft fabric texture. The coat rack is one matte black coat rack with a stable vertical post, one prominent long straight upper hook or peg projecting from the post to hold the hat.
- The medium-small hat hangs from only the outer end of one curved hook on the warm white coat rack; more than half of that hook, including its root and rounded tip, is visible, and all other hooks are visible, empty, and not touching the hat. The hat is one complete medium-small hat with a rounded crown and visible brim, soft fabric texture. The coat rack is one warm white coat rack with a stable vertical post, several visible empty curved hooks near the top, and a simple base.
- The hat is hanging from the outer part of exactly one long straight upper hook on the natural wood coat rack. The hat is one complete hat with a rounded crown and visible brim, soft fabric texture. The coat rack is one natural wood coat rack with a stable vertical post, one prominent long straight upper hook projecting from the post to hold the hat.
- The medium-small hat hangs from the outer part of one translucent curved hook, leaving most of the hook visible and all other hooks empty, visible, and not in contact with the hat. The hat is one complete medium-small hat with a rounded crown and visible brim, soft fabric texture. The coat rack is one translucent smoky acrylic coat rack with a stable vertical post, several visible curved hooks, and a simple base.
- The hat is hanging from the outer part of exactly one long straight upper hook on the light gray coat rack. The hat is one complete hat with a rounded crown and visible brim, soft fabric texture. The coat rack is one light gray coat rack with a stable vertical post, one prominent long straight upper hook projecting from the post to hold the hat.

Task 3: Hook a soup spoon on a hanging spoon hook.

- The soup spoon hangs from the spoon hook by its handle hole or hanging loop, with the hook supporting the spoon vertically. The soup spoon is one complete stainless steel soup ladle with a round bowl and long handle, hanging vertically by the handle end. The spoon hook is one small wall-mounted hook holding the soup ladle by its handle end.
- The soup spoon hangs from the spoon hook by its handle hole or hanging loop, with the hook supporting the spoon vertically. The soup spoon is one complete stainless steel soup ladle with a round bowl and long handle, hanging vertically from a rail hook. The spoon hook is one wall-mounted horizontal hook rail with several hooks, exactly one hook holding the soup ladle and the other hooks empty.
- The soup spoon hangs from the spoon hook by its handle hole or hanging loop. The soup spoon is one complete stainless steel soup ladle with a round bowl and long handle, hanging vertically by the handle end. The spoon hook is one small wall-mounted hook holding the soup ladle by its handle end.
- The soup spoon hangs from the spoon hook by its handle hole or hanging loop. The soup spoon is one complete black soup spoon with a deep round bowl and long handle, hanging vertically by the handle end. The spoon hook is one small wall-mounted hook holding the black soup spoon by its handle end.

- The soup spoon hangs from the spoon hook by its handle hole or hanging loop. The soup spoon is one complete matte black soup spoon with a deep bowl and long handle, hanging vertically by the handle end. The spoon hook is one square wall-mounted hook holding the black soup spoon by its handle end.

Figure S10: Reference scenes for **C.5 Hooking**. Top to bottom rows show cup-rack-tree, hat-coat-rack, and soup-spoon-hook variants.

C.6 C.6 Slotting

Task 1: Slot a plate into a dish rack.

- The plate stands nearly vertical in one slot between two upright bamboo pegs of the dish rack, with its lower curved edge resting between the two open parallel bamboo base rails. The plate is a round white ceramic dinner plate with a shallow raised rim, glossy surface, complete circular outline. The dish rack is a realistic narrow single-row bamboo tabletop dish rack like a simple plate stand, made from two straight parallel bamboo base rails connected by short cross rods, with one line of evenly spaced upright round bamboo pegs.
- The plate is inserted vertically into one slot of the dish rack and rests against the wooden dividers. The plate is a round matte blue ceramic plate with a raised rim and flat center. The dish rack is a bamboo dish rack with evenly spaced vertical wooden slots and a simple rectangular frame.
- The square plate is held upright in one slot between two round bamboo pegs of the dish rack, with its bottom edge resting between the low parallel wooden base rails and its face leaning lightly against an adjacent peg. The plate is a square white porcelain plate with softly rounded corners, a shallow raised edge, a glossy face. The dish rack is a realistic single-row bamboo tabletop dish rack with two long parallel wooden base rails, evenly spaced upright round bamboo pegs, a low open frame, small wooden feet.
- The plate is slotted upright between two round bamboo pegs of the dish rack, with its lower rim resting between the two open parallel wooden base rails. The plate is a pale green shallow ceramic plate with a slightly concave center, smooth rounded rim, glossy finish. The dish rack is a realistic single-row bamboo tabletop dish rack with two long parallel wooden base rails, evenly spaced upright round bamboo pegs, a simple low rectangular frame.
- The plate is inserted into the dish rack. The plate is a round ivory ceramic plate with a thin dark rim pattern, glossy face, shallow raised lip. The dish rack is a single-row bamboo

tabletop dish rack with two long parallel wooden base rails and evenly spaced round vertical pegs, an open low frame.

Task 2: Slot a red wine bottle into a wine rack.

- The red wine bottle is seated in the wine rack, supported by the rack structure so it cannot roll away. The red wine bottle is one complete dark glass red wine bottle with a cream label and burgundy neck capsule, seated through one diamond opening of the rack. The wine rack is one black metal diamond-shaped wire wine rack with crossed open cells, one cell supporting the bottle.
- The red wine bottle is seated in the wine rack, lying horizontally in one upper groove of the rack. The red wine bottle is one complete dark glass red wine bottle with a cream label and burgundy neck capsule. The wine rack is one light wooden rectangular wine rack with horizontal rails and U-shaped cradle grooves, one upper groove supporting the bottle.
- The red wine bottle is seated in the wine rack, supported by the rack structure so it cannot roll away. The red wine bottle is one complete dark glass red wine bottle with a cream label and burgundy neck capsule, seated diagonally in one crossed wooden cell of the rack. The wine rack is one light wooden crossed-board wine rack with diagonal boards forming open cells, one cell supporting the bottle.
- The red wine bottle is seated in the wine rack, supported in the curved wire holder. The red wine bottle is one complete dark glass red wine bottle with a cream label and burgundy neck capsule. The wine rack is one black decorative metal wire wine rack with rounded loop supports, one support holding the bottle.
- The red wine bottle is seated in the wine rack, lying horizontally in one cell of the wooden grid rack. The red wine bottle is one complete dark glass red wine bottle with a cream label and burgundy neck capsule. The wine rack is one pale wooden grid wine rack with rectangular cells and horizontal supports, one cell supporting the bottle.

Task 3: Slot a book into a multi-compartment book stand.

- The book is placed upright inside exactly one rectangular compartment of the multi-compartment book stand, fully contained within that compartment and not touching the dividers of adjacent compartments. The book is one complete closed book with a plain cover and a visible spine. The multi-compartment book stand is a realistic natural wood multi-compartment book stand with a stable base and several rectangular grid compartments separated by vertical dividers; at least one compartment is clearly sized so a single upright book fits within it.
- The closed book is placed upright inside exactly one rectangular compartment of the dark ceramic multi-compartment book stand, fitting entirely within that compartment without occupying neighboring compartments. The book is one complete closed book with a plain dark cover and visible spine. The multi-compartment book stand is one dark ceramic multi-compartment book stand with a stable base and clearly separated rectangular compartments separated by vertical divider walls.
- The book stands upright inside one rectangular compartment of the translucent smoky acrylic multi-compartment book stand. The book is one complete closed book with plain cover and visible spine. The multi-compartment book stand is one translucent smoky acrylic multi-compartment book stand with several empty rectangular grid slots, vertical dividers, and stable base.
- The closed book is placed upright inside exactly one compartment of the brass multi-compartment book stand, contained fully within that single slot and not spanning across the divider into neighboring compartments. The book is one complete closed book with a plain cover and visible spine. The multi-compartment book stand is one brass colored multi-compartment book stand with several clearly separated rectangular compartments, vertical dividers, and a stable base.

- The book stands upright inside one rectangular compartment of the light gray multi-compartment book stand. The book is one complete closed book with plain cover and visible spine. The multi-compartment book stand is one light gray multi-compartment book stand with several empty rectangular grid slots, vertical dividers, and stable base.

Figure S11: Reference scenes for **C.6 Slotting**. Top to bottom rows show plate-dish-rack, wine-rack, and book-stand variants.

C.7 C.7 Spanning

Task 1: Span a fork across a plate rim.

- The fork bridges the plate rim like a small ramp: the fork head and downward-facing tines are inside the plate, the fork neck crosses the rim, and the handle slopes downward outside the plate until the handle end rests on the tabletop, not on the plate. The fork is a polished stainless steel dinner fork with four tines and a long narrow handle, shown upside down with the back side facing upward. The plate is a round white ceramic plate with a shallow raised rim and glossy surface, fully visible on a clean white tabletop.
- The fork is tilted like a ramp over the plate rim: the fork head and downward-facing tines touch the inside of the plate, the fork neck is supported by the raised rim, and the handle angles downward outside the plate with the rounded handle tip visibly touching the tabletop below the rim. The fork is a matte black metal fork with four slim tines and a straight handle, shown upside down with the back side facing upward. The plate is a round matte gray stoneware plate with a raised rim, fully visible on a clean white tabletop.
- The fork lies diagonally over the rim of the plate, with the tines pointing downward toward the center of the plate and the handle extending outside the rim. The fork is a gold-colored fork with a reflective handle and four rounded tines. The plate is a glossy blue ceramic plate with a shallow bowl-like center and smooth circular rim.
- The fork is tilted like a ramp over one side edge of the plate: the fork head and downward-facing tines touch the inside of the plate, the fork neck is supported by the raised plate edge, and the handle angles downward outside the plate with the rounded handle tip visibly touching the tabletop below the edge. The fork is a small silver dessert fork with short tines and a tapered handle, shown upside down with the back side facing upward. The plate is a square white porcelain plate with rounded corners and a raised edge.
- The fork is tilted like a ramp over the plate rim with the handle angling downward outside the plate and the rounded handle tip visibly touching the tabletop below the rim. The fork has a brushed steel head, four tines, and a light wooden handle, shown upside down with the

back side facing upward. The plate is a round cream ceramic plate with a slightly speckled glaze and a raised rim.

Task 2: Span a calligraphy brush across a brush stand.

- The calligraphy brush rests on the brush stand with visible support under the brush shaft so it is stable and cannot roll away. The calligraphy brush is one complete traditional calligraphy brush with a pale wooden handle, black ferrule, tapered dark bristles, and a small hanging loop at the rear end. The brush stand is one small dark wooden brush stand with carved U-shaped cradle notches supporting the brush shaft from below.
- The calligraphy brush rests on the brush stand with visible support under the brush shaft. The calligraphy brush is one complete calligraphy brush with dark bristles, a dark ferrule, warm brown handle, and rear hanging loop. The brush stand is one dark carved wooden brush stand with multiple U-shaped notches, two notches visibly supporting the brush shaft from below.
- The calligraphy brush rests on the brush stand with visible support under the brush shaft. The calligraphy brush is one complete calligraphy brush with dark pointed bristles, black ferrule, orange-brown handle, and rear hanging loop. The brush stand is one small blue and white ceramic brush stand with a raised cradle groove supporting the brush shaft.
- The calligraphy brush rests on the brush stand with visible support under the brush shaft. The calligraphy brush is one complete calligraphy brush with dark pointed bristles, black ferrule, brown wooden handle, and rear hanging loop. The brush stand is one small cream ceramic brush stand with two rounded cradle humps supporting the brush shaft.
- The calligraphy brush rests on the brush stand with visible support under the brush shaft. The calligraphy brush is one complete calligraphy brush with dark pointed bristles, black ferrule, warm brown handle, and rear hanging loop. The brush stand is one small blue and white ceramic brush stand with a triangular upright cradle supporting the brush shaft.

Task 3: Span a soup ladle across a vertical ladle stand.

- The soup ladle rests upright in the lower circular cradle of the vertical ladle stand, while the handle leans upward through and is guided by the stand's tall wire guide. The soup ladle is one complete soup ladle with a deep rounded bowl and a long slim handle, clean reflective metal with subtle brushed highlights, plus a richer mixed-tone finish, with a hanging hole at the handle tip. The vertical ladle stand is one complete brushed-metal vertical ladle stand with a tall wire frame shaped like a triangular tri-prong guide, a stable weighted base, and a lower circular saucer cradle specifically sized to hold the ladle bowl.
- The soup ladle bowl rests in the lower circular cradle while the long handle leans upright through the tall wire guide of the gold-toned wire ladle stand. The soup ladle is one complete soup ladle with a deep rounded bowl, long slim handle with hanging hole, clean reflective metal surface. The vertical ladle stand is one gold-toned wire vertical soup ladle stand with a decorative upright loop and small round bowl support, stable wire base.
- The soup ladle rests upright with its bowl sitting in the lower circular cradle of the vertical ladle stand while the long handle passes through and is guided by the tall wire channel. The soup ladle is one complete soup ladle with a deep rounded bowl and a long slim handle, clean surface with a richer matte color finish, subtle realistic metal sheen, and a hanging hole at the handle end. The vertical ladle stand is one realistic matte steel vertical ladle stand with a minimal rod-and-wire frame, a front lower circular cradle that supports the ladle bowl, and a tall rear wire channel that aligns the handle upright.
- The multicolored soup ladle rests upright on the lower circular cradle of the white enamel vertical ladle stand, with the long handle guided vertically through the stand's tall wire frame. The soup ladle is one complete soup ladle with a deep rounded bowl and a long slim handle featuring a hanging hole; the bowl has a rich multicolor enamel finish, and the

handle is smooth metallic or enamel coated. The vertical ladle stand is one white enamel wire vertical ladle stand with a stable base and a lower circular cradle that supports the ladle bowl; the upper section forms a tall, open wire guide channel that the ladle handle passes through.

- The soup ladle bowl rests in the lower circular cradle while the long handle leans upright through the tall wire guide of the dark bronze wire ladle stand. The soup ladle is one complete soup ladle with a deep rounded bowl, long slim handle with hanging hole, clean reflective metal surface. The vertical ladle stand is one dark bronze wire vertical soup ladle stand, upright loop frame with stable circular foot.

Figure S12: Reference scenes for **C.7 Spanning**. Top to bottom rows show fork-plate-rim, calligraphy-brush-stand, and ladle-stand variants.

C.8 C.8 Pegging

Task 1: Peg a toss ring onto a ring toss stand.

- The single red toss ring is looped around one upright peg of the ring toss stand and sits visibly on the peg. The toss ring is one complete red circular toss ring with smooth rounded tube shape. The ring toss stand is one toy-like ring toss stand with a stable base and several upright empty pegs, colorful simple plastic or painted wood construction.
- The single yellow toss ring is looped around one upright peg of the ring toss stand and sits visibly on the peg. The toss ring is one complete yellow circular toss ring with smooth rounded tube shape. The ring toss stand is one toy-like ring toss stand with a stable base and several upright empty pegs.
- The single green ring encircles the vertical tapered cone post near the lower third of the stand; the post passes through the ring's center hole and the ring rests against the cone above the solid base. The toss ring is one complete green translucent circular toss ring with smooth rounded tube shape. The ring toss stand is one toy-like ring toss stand with a single solid round disk base and one vertical tapered cone post, stacked colored bands on the post.
- The orange ring is looped around the tapered wooden cone so the cone passes through the ring center; the ring sits tilted slightly on the lower cone and touches the base, physically captured by the post. The toss ring is one complete matte orange rubber toss ring with a thick rounded tube. The ring toss stand is one single-post ring toss stand with a circular wooden base and a smooth vertical tapered wooden cone, natural light wood grain with a darker walnut top cap.

- The pink ring is looped around the vertical cone post and rests around the lower part of the cone above the solid pedestal base. The toss ring is one complete pink silicone circular toss ring with thick soft rounded tube shape. The ring toss stand is one single upright ring toss cone stand with a solid oval pastel lavender pedestal base and one vertical tapered cone post, satin plastic material.

Task 2: Peg a jewelry ring onto a ring holder.

- The jewelry ring is fitted around the tapered support of the matte black ring holder. The jewelry ring is one complete small jewelry ring with a smooth metallic band, simple gemstone or plain top. The ring holder is one matte black ring holder with a tapered cone or upright finger-shaped support, stable base, and no other jewelry.
- The jewelry ring is fitted around the tapered support of the warm white ring holder. The jewelry ring is one complete small jewelry ring with a smooth metallic band, simple gemstone or plain top. The ring holder is one warm white ring holder with a tapered cone or upright finger-shaped support, stable base.
- The jewelry ring is fitted around the tapered support of the natural wood ring holder. The jewelry ring is one complete small jewelry ring with a smooth metallic band, simple gemstone or plain top. The ring holder is one natural wood ring holder with a tapered cone or upright finger-shaped support, stable base.
- The jewelry ring is fitted around the tapered support of the brass colored ring holder. The jewelry ring is one complete small jewelry ring with a smooth metallic band, simple gemstone or plain top. The ring holder is one brass colored ring holder with a tapered cone or upright finger-shaped support, stable base.
- The jewelry ring is fitted around the tapered support of the light gray ring holder. The jewelry ring is one complete small jewelry ring with a smooth metallic band, simple gemstone or plain top. The ring holder is one light gray ring holder with a tapered cone or upright finger-shaped support, stable base.

Task 3: Peg a mug onto a cup rack.

- The mug hangs upside down on one vertical peg of the wooden cup rack, with the peg entering the open mouth of the mug and supporting it securely. The mug is one complete dark navy blue mug with a handle, open mouth facing downward. The cup rack is one complete light wooden cup rack with a rectangular base, multiple vertical cylindrical pegs, and simple cross rails.
- The mug hangs upside down on one vertical peg of the cup rack, with the peg entering the open mouth of the mug. The mug is one complete clear glass cup with open mouth facing downward. The cup rack is one white tabletop cup rack with a round base and multiple vertical curved pegs, one peg supporting the cup.
- The mug hangs upside down on one vertical peg of the cup rack, with the peg entering the open mouth of the mug. The mug is one complete sage green mug with handle and open mouth facing downward. The cup rack is one gray rectangular cup rack with several vertical cylindrical pegs and a flat base.
- The mug hangs upside down on one vertical peg of the cup rack, with the peg entering the open mouth of the mug. The mug is one complete clear glass cup with blue rim and open mouth facing downward. The cup rack is one chrome rotating cup rack with a circular base and multiple vertical arms.
- The mug hangs upside down on one vertical peg of the cup rack, with the peg entering the open mouth of the mug. The mug is one complete amber glass cup with open mouth facing downward. The cup rack is one silver metal cup rack with forked vertical supports and a rectangular base.

Figure S13: Reference scenes for **C.8 Pegging**. Top to bottom rows show ring-toss-stand, jewelry-ring-holder, and mug-cup-rack variants.

D Prompt Templates

This section lists the prompt templates that instantiate the agentic decisions described in the main paper and earlier sections of the supplement. Each box preserves the operative constraints and machine-readable output schema of the corresponding template, while omitting incidental boilerplate.

D.1 Input scene agent

The input scene agent of Sec. A.2 consumes an arbitrary combination of language, scene image, and asset images, and decides whether to draw a fresh scene or edit the user-supplied one. Its system prompt encodes the scene rules, the closed vocabulary of relations, the image-quality criteria, and the JSON output schema.

Input scene agent (system)

```
[Goal]
Produce a clean reference image suitable for 3D reconstruction together with a RootData JSON that
records the primary target, the contact object, their relation, and a background color.

[Scene rules]
Exactly two objects, primary target plus contact object, with no distractors. Solid background such
as white, light gray, beige, or a hex value. Camera at an elevated angle so both objects are clearly
visible. Target and contact must be physically separable as independent rigid bodies and must not be
connected by thin continuous structures such as stems, vines, chains, or ropes.

[Relation vocabulary]
on, in, inside, above, under, stack_on, cover, hang_on, lean_against, rest_on, attach_to, insert_into,
fit_in.

[Inputs and policy]
You may receive a task description, a scene image that may be messy, one or two asset images each
labeled TARGET or CONTACT, and an optional task hint. Analyze all provided modalities to identify
the target, contact, and relation. Then act: prefer edit_image when a usable scene image is given
and only minor cleanup is needed, otherwise call draw_image with the available asset and scene paths
in reference_image_paths so the generator preserves their appearance. Fall back to text-only
generation only when no images are provided. After every action, verify the result against the
quality criteria below and retry up to {max_iterations} times.

[Quality criteria]
1. Background is solid or near-solid with no textures, walls, or floor patterns. 2. Object count is
exactly one target and one contact; structurally connected parts of the same object are still one
object. 3. Both objects are substantially visible; functional occlusion such as contents inside a
container is acceptable. 4. The depicted contact is the way a human would actually arrange these
```

specific objects for the given relation. 5. Camera is elevated and not directly overhead or directly side-on. 6. No floating, interpenetrating, or geometrically broken parts. 7. Target and contact are independent rigid bodies, with no thin continuous links.

[Output JSON]

{ task_description, primary_target{name, description, role}, contact_object{name, description, role}, target_contact_relation, reference_image }

D.2 Synonym retry for mask preparation

When the mask-preparation step returns an empty or low-confidence mask, the synonym module proposes alternative phrasings that are more likely to ground to the intended object.

Mask-preparation synonym generator

[Role]

You generate alternative text prompts for a mask-preparation step that fails on the original object name. The mask-preparation step prefers short noun phrases, struggles with overly specific or niche terms, and benefits from visual descriptors of color, material, and shape.

[Strategies (apply in order)]

1. Generalization: move up the category hierarchy, e.g. ‘‘sundial’’ → ‘‘statue’’.
2. Visual adjectives: add color, material, or shape, e.g. ‘‘ball’’ → ‘‘red ball’’.
3. Category names: use the superclass, e.g. ‘‘mug’’ → ‘‘cup’’.
4. Creative indirect terms: use related visual concepts, e.g. ‘‘nose’’ → ‘‘black marking’’.
5. Simplification: strip non-visual modifiers, e.g. ‘‘ceramic coffee mug with handle’’ → ‘‘mug’’.

[Output JSON]

{ alternatives [exactly 5 entries, ranked best-first, all distinct from the original name], reasoning }

D.3 Asset erase agent

The erase agent of Sec. A.3 alternates between editing and verifying with a strict pose-first rubric. Its system prompt is reproduced below.

Asset erase agent (system)

[Goal]

Erase the named target object from the input image while preserving the partner object exactly in its current position, orientation, scale, and posture. Minor appearance changes such as color shift or texture smoothing are acceptable; pose changes are not.

[Workflow (repeat up to {max_iterations} times)]

1. Call edit_image with a precise prompt that names the target object and explicitly forbids any motion of the preserved object. 2. Call compare_images to obtain a labeled side-by-side comparison whose left panel is the original and whose right panel is the current edit. 3. Inspect both panels and call score_image with an integer score in [0,100] and a short feedback string. Apply the scoring procedure below.

[Scoring procedure]

Step A: locate the preserved object in both panels and check whether its pixel position, size, and orientation match. Step B: check whether the target object has been fully removed from the right panel. Step C: check the background fill for naturalness and absence of artefacts. Step D: apply the rubric of Tab. S2: -50 for any pose change, -25 for residual visibility, -15 for appearance change of the preserved object with intact pose, -10 for background artefacts.

[Tips for edit prompts]

Name the target precisely with color, shape, and screen location. Say explicitly that the preserved object must not move, rotate, or resize. Specify that the background fill should match the surrounding area seamlessly. If the previous attempt shifted the preserved object, repeat in the next prompt that the preserved object must stay at exactly the same pixel location and orientation.

[Termination]

If the score is at least 90, output the result path and stop. Otherwise, if iterations remain, call edit_image again. After all iterations, output the path of the best-scored result.

D.4 Asset symmetry analyzer

Symmetry tagging guides the pose-equivalence sampling of Sec. 3.5. Rather than asking for purely geometric symmetry, the analyzer is constrained to task-driven functional equivalence: rotations are admitted only when the rotated object still fulfills the contact relation.

Asset symmetry analyzer

[Role]
You are an expert in 3D kinematics, geometry, and robotic grasping. Symmetry types are the cyclic groups C1, C2, C3, C4, Cinf.

[Coordinate system]
Do NOT use default axis assumptions. Read the labeled arrows in the input image: red+‘‘X’’ is X, green+‘‘Y’’ is Y, blue+‘‘Z’’ is Z.

[Task-driven functional equivalence]
Symmetry must be judged with respect to the contact relation. Identify the alignment requirement, idealize away minor bends and textures, and apply the lift-rotate-place test: lift the object, rotate it by a candidate angle, place it back; if the post-placement state still fulfills the task, that rotation is admissible. Path collisions during the rotation are irrelevant.

[Examples]
Mug hang_on hook: C2 about vertical axis, since a 180° flip still lets the hook pass through the handle.
Mug rest_on coaster: Cinf about vertical axis, since the bottom contacts identically at any yaw.
Sausage in rectangular box: Cinf about long axis after idealization, C2 about vertical axis.

[Output JSON]
{ object_name, step1_shape_idealization{...}, step2_task_requirements{...}, x_axis{motion, 180/90 checks, symmetry_type, reasoning}, y_axis{...}, z_axis{...}, overall_confidence }

D.5 Multi-view scene verifier

After the pose pipeline produces a candidate scene, the verifier inspects MuJoCo renderings, optionally arranged as multi-view or time-series collages, and decides whether the contact relation is preserved under gravity.

Multi-view scene verifier

[Context]
Images are MuJoCo renders, not AI-generated. Physics, multi-view consistency, and temporal consistency are guaranteed by the simulator. A time-series, when present, is a passive gravity-only stability rollout, not a robot executing the task. The final frame is therefore the gravity-settled state, not the expected manipulation outcome.

[Task]
Verify three properties: (i) the target-contact relation matches the expected relation; (ii) for time series, the relation is preserved at the final settled frame; (iii) the resulting scene is feasible for grasping. Do not reward candidates whose ‘‘final’’ state corresponds to a removed or separated target, even if it visually resembles a completed manipulation.

[Critical defects]
Target or contact object missing, wrong spatial relation, or final settled state breaking the expected relation. Texture artefacts and minor lighting differences are acceptable.

[Output JSON]
{ passes, relation_ok, reason, issues, relation_issues, confidence }

D.6 Candidate voter

When multiple pose candidates survive verification, the voter compares them on a single horizontal montage where each candidate occupies one labeled panel separated by red dividers, and selects the best one.

Candidate voter

[Image structure]
A single horizontal montage. Candidates are arranged left to right, separated by thick red vertical bars. Each panel carries a CANDIDATE #N header. Inside the panel a per-candidate timeline (perspective view on the left, top view on the right) progresses left to right; this timeline is a passive gravity-only MuJoCo rollout, not a robot execution.

[Critical interpretation rule]
A candidate is an initial scene that should already satisfy the contact relation; the gravity-settled final frame must continue to satisfy that relation. Do not prefer a candidate where the target is removed, separated, or fallen away merely because it resembles task completion.

[Evaluation priorities]

1. Relation preservation under gravity at the final frame.
2. Asset quality: target and contact must be separate clean meshes, not fused or doubled; reject reconstruction artefacts even if alignment looks tight.
3. Physical plausibility: no severe interpenetration, flying, or instability.
4. Overall reasonableness for robotic manipulation. A candidate with perfect alignment but corrupted geometry must score lower than one with mild misplacement and clean meshes.

[Output JSON]

{ best_index, reason, confidence }

D.7 Multi-object scene expander

The expander plans up to {max_new_objects} insertions for an already verified scene. It reads a pullback rendering together with the current object list, where each entry contains the object name, role, and description. If a user request is provided, the planner must include the requested objects subject to the insertion budget; otherwise, it autonomously selects semantically related additions. In both cases, it outputs reconstruction-oriented visual descriptions, spatial relations to existing objects, an image-editing prompt, and a preservation description for the editor.

Scene expander

[System]

You are a scene expansion planner for robotic grasping scenarios. Given an existing scene with objects, plan additional objects to add. When the user specifies objects, follow their instructions. When no user request is given, suggest objects that make the scene more realistic and natural. Output valid JSON only, with no markdown.

[Inputs]

Analyze the rendered scene image and plan up to {max_new_objects} new objects to add. Existing objects are provided as - name (role): description. An optional user request may specify the objects to add.

[Planning policy]

If a user request is provided, include the requested objects in the plan. Parse the request to determine object names, visual descriptions, reasonable positions relative to existing objects, and appropriate spatial relations. Descriptions may be adjusted to improve 3D reconstruction quality, but the core requested objects must not be changed. If the insertion budget is restrictive, add fewer requested objects while prioritizing the user's choices. If no user request is provided, suggest objects that are semantically related to the existing scene, e.g. pen-holder → pens, desk → books or ruler, coat rack → coat or hat.

[Requirements]

1. Each new object has a clear spatial relation to an existing object.
2. Sizes are reasonable relative to existing objects.
3. Generate an image-editing prompt that preserves the existing scene, viewpoint, lighting, and style.

[Output JSON]

{ objects_to_add[{name, description, suggested_position}], spatial_relations[{subject, relation ∈ {in, on, left_of, right_of, behind, stack_on, lean_against, in_front_of, cover, hang_on}, object}], edit_prompt, preserve_description, reasoning }

D.8 VLM scene scorer

The VLM scorer rates a single rendered scene on Object Matching and Relationship Matching, supplying the per-scene scores aggregated in Tab. 1 of the main paper.

VLM scene scorer

[Context]

A 3D scene was reconstructed from images, placed in MuJoCo, and allowed to settle under gravity. Some upstream objects may be held static, so the settled image can still expose floating, balanced, or overhung configurations. Judge plausibility under gravity directly, do not assume settling guarantees plausibility.

[Score 1, Object Matching (0-10)]

0: at least one object missing. 1-3: both present but appearance differs substantially. 4-6: rough match with noticeable discrepancies. 7-9: close match with minor differences only. 10: perfect match.

[Score 2, Relationship Matching (0-10)]

Apply the decision procedure: (a) does the geometry satisfy the named relation; (b) is the arrangement common-sense correct, that is, how a human would actually place the objects for this relation; (c) is there visible interpenetration; (d) is the pose physically plausible under gravity. Any failure caps the score: 0 for absent or reversed; 1-3 for a clear failure on geometry, plausibility, or common sense; 4-6 for partial satisfaction with a clear defect; 7-9 only when geometry, plausibility, and common-sense placement are all unambiguously satisfied; 10 for perfect.

[Output JSON]

{ target_present, contact_present, object_match_score, object_match_reasoning, relation_score, relation_reasoning }

D.9 Ego4D-driven taxonomy clustering

The benchmark of Sec. B.1 is built by an LLM agent that ingests filtered Ego4D narrations in shards. Each shard is summarized into candidate relation phrases, the phrases are merged across shards, and the merged set is clustered by contact geometry. The three prompts that drive these stages are listed below.

Stage 1: per-shard relation summarization

[Role]

You read short egocentric narrations of human manipulation, each describing one camera-wearer manipulating one named target with respect to one named contact partner.

[Task]

For every narration, output a single relation phrase that names the contact geometry between the target and the partner, ignoring linguistic surface form. Use short noun-style phrases such as ‘‘ring around peg’’ or ‘‘lid covering opening’’, and do not invent objects that are not in the narration.

[Output JSON]

{ narrations[{video_uid, narration_text, target, contact, relation_phrase}] }

Stage 2: cross-shard phrase merging

[Role]

You merge relation phrases that describe the same contact geometry but were produced by different shards. Phrases are equivalent only when the rotation, translation, and contact pattern of the target with respect to the partner are interchangeable.

[Task]

Group equivalent phrases under a single canonical phrase, choose the most precise wording for each canonical, and discard phrases that no longer describe a single contact geometry after merging.

[Output JSON]

{ canonical_phrases[{canonical, members[...] }] }

Stage 3: contact-geometry clustering

[Role]

You partition canonical relation phrases into a small number of contact-geometry classes. The partition must be mutually exclusive, exhaustive, and have between six and ten classes.

[Constraints]

Each class must correspond to a distinct contact pattern that admits a closed-form geometric template, such as a containment volume, a planar contact, an axial peg, or a hooking link. Avoid splits that depend only on object identity. Provide a one-line definition for each class together with two to three representative member phrases.

[Output JSON]

{ classes[{class_id, name, definition, members[...] }] }

Per-class task selection

[Role]

You select three representative manipulation tasks per contact-geometry class for benchmark construction.

[Constraints]

For each class produce three tasks under the following rules: (i) each task instantiates the class with two distinct everyday objects; (ii) the two objects must be physically separable as independent

rigid bodies; (iii) the union of the three tasks should cover both household and workplace scenarios. Provide a one-sentence natural-language instruction per task and a short justification per choice.

[Output JSON]

{ class_id, tasks[{task_name, target, contact, instruction, justification}] }