NeoWorld-Pro: Programming Interactive Scenes from Monocular Images for Embodied Simulation

Yumeng He[1,2*] **Eachen Soong**[1,2*] **Xiaotian Yang**[3] **Weijia Zhang**[1]
Zanwei Zhou[1] **Junru Gong**[1,2] **Xiaokang Yang**[1] **Yunbo Wang**[1,2†]
[1]Shanghai Jiao Tong University [2]Doxi AI [3]Huazhong University of Science and Technology
https://neoworldproject.github.io/neoworld-pro-website/

Abstract

The advancement of Embodied AI necessitates high-quality simulation assets that faithfully mirror the real world. However, transforming raw visual observations into simulation-ready scenes remains challenging due to the lack of physical grounding and scene-level interactivity in current image-to-URDF methods. We propose **NeoWorld-Pro**, a framework that reformulates monocular scene reconstruction as procedural programming for interactive 3D environments. Leveraging the zero-shot reasoning and code synthesis capabilities of MLLMs, NeoWorld-Pro converts a single RGB image into executable programs specifying object geometry, articulation, and physical properties. A **physics-in-the-loop** mechanism then iteratively refines the generated programs by validating their execution in a physics engine, enforcing physically plausible articulations, valid object compositions and interactions, and accurate spatial relationships. Experiments show that NeoWorld-Pro outperforms open-loop and prior monocular reconstruction methods, while enabling complex downstream tasks such as stable stacking and fine-grained manipulation.

1 Introduction

The rapid advancement of embodied AI and robotics has increased the demand for high-quality, interactive simulation environments for robot learning. In particular, building faithful digital twins requires not only accurate 3D geometry, but also correct kinematic structures, physical properties, and simulation-ready inter-object relationships that support *Real-to-Sim-to-Real transfer*. However, manually constructing such assets is labor-intensive and does not scale to the diversity and complexity of the real world. As a result, automating the Real-to-Sim transition, from raw visual observations to physically functional, interactive simulation scenes, has become a critical research frontier.

Despite its importance, this transition remains constrained by three primary bottlenecks. First, most existing work focuses on *single-asset image-to-URDF reconstruction* [4, 19, 21, 11, 2, 34], lacking scene-level modeling that jointly captures embodied interaction and inter-object spatial and physical dependencies. Second, current approaches typically rely on large curated asset libraries or rigid category-level priors [4, 19, 8, 20, 12, 2, 34], which imposes high data requirements and hinders generalization to *out-of-distribution (OOD)* scenarios. Third, and perhaps most critically, existing pipelines are inherently *open-loop* [4, 34, 15, 8, 2]. By treating reconstruction as a one-way inference task, they lack a mechanism for corrective feedback to resolve physical inconsistencies in geometry, articulation, or scene composition. As a result, previous methods often struggle to produce environments that are truly "simulation-ready" for high-precision downstream tasks. Recent work, NeoWorld [44], constructs explorable virtual worlds from a single image via progressive 3D unfolding, combining object-centric 3D foregrounds with 2D background synthesis for efficient exploration. However, its hybrid neural representation offers limited explicit control over simulation-critical

*Equal contribution.
†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

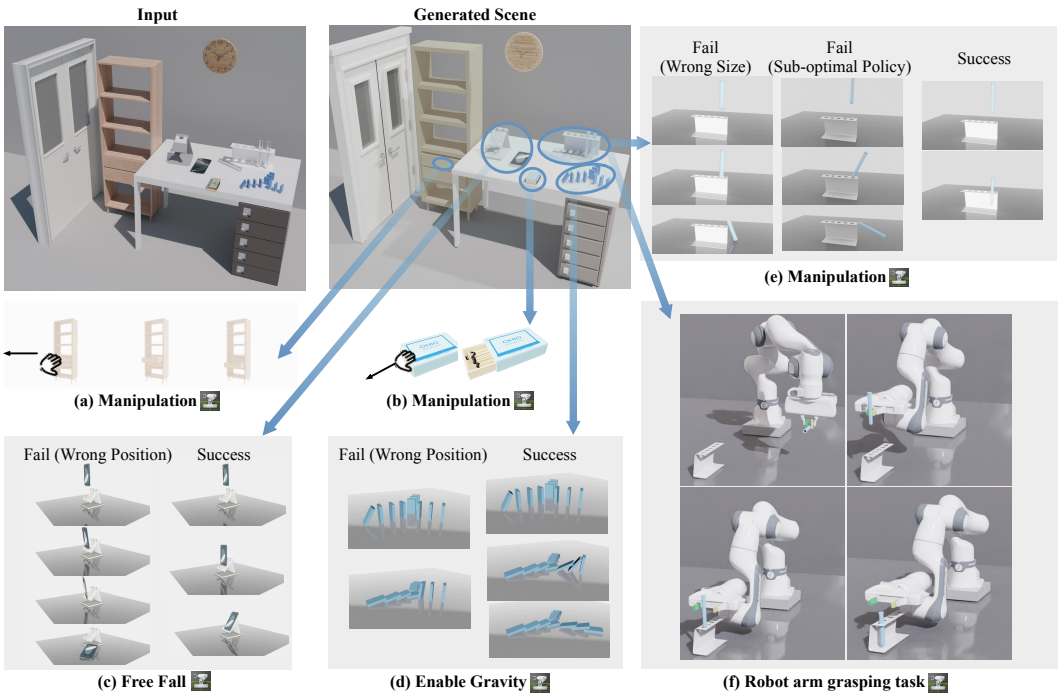

Figure 1: **Demonstration of NeoWorld-Pro.** Our framework ensures simulation-readiness by optimizing: (a-b) physically plausible intra-object articulations under external forces, (c-d) stable inter-object spatial configurations, and (e-f) scene viability for assembly and manipulation tasks.

geometry, articulation, and physical properties. Taken together, these limitations call for an explicit, executable world representation that supports structured editing and physics-based refinement.

We propose NeoWorld-Pro—where "Pro" denotes *programming*—a framework that reformulates monocular interactive scene construction through two core innovations: *(1) asset reconstruction via procedural code synthesis* and *(2) physics-in-the-loop scene refinement*. In contrast to traditional dense reconstruction methods, which represent scenes as unstructured geometric entities like point clouds or meshes and prioritize per-pixel alignment, NeoWorld-Pro introduces a *program-centric paradigm* that prioritizes functional and physical plausibility. By leveraging the zero-shot spatial reasoning and code synthesis capabilities of Multimodal Large Language Models (MLLMs), our framework translates a monocular image into executable programs. These programs explicitly define object geometry via compositional primitives, articulation parameters, physical properties, and scene layout. Unlike static geometric representations, this program-centric approach is executable and editable, exposing the generated scene in a form that can be directly inspected, simulated, and modified. By relying on the rich commonsense priors of MLLMs rather than dataset-specific training, NeoWorld-Pro generalizes effectively to novel categories and logically infers hidden structures, significantly improving robustness to the severe occlusions prevalent in cluttered scenes.

Furthermore, to upgrade superficial plausibility in the static appearance of generated scenes into true physical physical consistency, we introduce a *physics-in-the-loop* validation and refinement mechanism. Instead of passively outputting static assets, NeoWorld-Pro executes the generated programs within a physics engine (*e.g.*, Isaac Sim) to collect dynamic feedback, such as collision events, instability, and articulation failures. This feedback is then fed back to the MLLM to iteratively refine the code. As shown in Figure 1, this closed-loop process validates joint executability at the object level while correcting inter-object penetration, unstable arrangements, and inaccurate relative scales at the scene level, ensuring the final environment is truly simulation-ready.

We evaluate NeoWorld-Pro on asset reconstruction, articulation prediction, scene assembly, and manipulation task execution. Our results show that, by explicitly optimizing for simulation readiness through physics-in-the-loop refinement, NeoWorld-Pro significantly outperforms existing open-loop methods in generating environments that are immediately usable for robot learning.

In summary, the main contributions of this work are as follows:

- We introduce NeoWorld-Pro, formulating monocular scene construction as MLLM-driven program synthesis, reducing reliance on specific asset databases and supervised category-level priors.
- We propose a physics-in-the-loop scene refinement pipeline that leverages simulation feedback to iteratively improve both intra-object articulations and inter-object relationships.
- We construct a simulation benchmark comprising 90 articulated object categories and 30 scenes in Universal Scene Description (USD) format, and validate the effectiveness of NeoWorld-Pro on it.

2 Related Work

2.1 Articulated Asset Reconstruction

The capability to model articulated assets is fundamental for enabling intelligent agents to perform complex manipulations in physical environments. Yet, manually authoring such assets is labor-intensive, requiring explicit part decomposition, geometry modeling, and kinematic specification. Recent learning-based methods thus aim to automate articulated asset construction from visual or geometric observations [14, 11, 17, 6]. Existing research predominantly relies on structured 3D inputs, such as static/dynamic point clouds [7, 28, 10, 38, 32, 18, 39] or meshes [42, 37, 22, 9], which are often unavailable in the wild. Alternative paradigms utilize multi-view-based 3D reconstruction [36, 21] or multi-state images [33, 24, 13, 41, 16, 3] to resolve kinematic ambiguity, yet incur significant data-acquisition overhead and limited scalability.

One line of work mitigates this ambiguity via retrieval- or assembly-based formulations, grounding symbolic structures in mesh libraries or templates [4, 19]. However, their generalization are constrained by the coverage and canonical biases of the underlying assets or training corpora. Learning-based methods instead exploit category-level priors over part layouts, shapes, and motions [8, 20, 12], reducing explicit retrieval dependence but remaining limited by the distributional coverage of training corpora and thus generalizing poorly to out-of-distribution structures or articulations. More recent systems further target simulation-oriented asset construction by converting reconstructed geometry and inferred kinematics into executable representations for physics engines [27, 34, 2].

Although some of them evaluate generated assets in simulation [4, 34, 15] or downstream robotic tasks [11, 27, 1, 8], such evaluation is typically post-hoc rather than used as feedback to refine the generation process. As a result, articulation and contact failures often emerge only after asset generation. In contrast, our framework reformulates single-image asset construction as MLLM-driven procedural modeling and introduces physics-in-the-loop validation to refine both intra-object articulations and inter-object spatial consistency for simulation-ready scene generation.

2.2 LLMs and MLLMs as 3D Generation Priors

Recent studies have explored LLMs and MLLMs as priors for articulated 3D asset generation. Some methods use foundation models mainly to construct supervision signals, textual conditions, physical annotations, or data augmentations [25, 2, 34]. Beyond such auxiliary usage, LLMs/MLLMs have been used as interpretive priors that expose the latent part structure and motion semantics of visual or geometric observations, while leaving the final asset construction to downstream retrieval, abstraction, or template-based modules [19, 8, 22, 27, 20]. SINGAPO [19] uses GPT-4o to infer an object-level part connectivity graph before grounding it through part abstraction and retrieval, while SPARK [8] leverages GPT-4o to parse parts and infer joints for constructing coarse URDF templates. Articulate-Anything [11] further introduces an actor-critic formulation, where MLLM agents iteratively propose and critique link placement and joint configurations to improve articulation plausibility.

More recently, language models have been adapted from interpreters into direct generators of articulated asset specifications, producing structured outputs that can be converted into executable or simulator-compatible assets [21, 15, 39, 2]. Among them, PhysX-Anything [2] targets single-image physical asset generation by fine-tuning a MLLM to predict a unified structured representation of part geometry, articulation, and physical attributes. Nevertheless, these methods are largely constrained by curated articulated-object or task-specific 3D and physical corpora, making their outputs inherit the structural and kinematic biases of the underlying datasetss. In contrast, our framework reformulates single-view articulated asset generation as zero-shot code-driven procedural modeling, enabling more flexible generation beyond category-specific asset distributions.

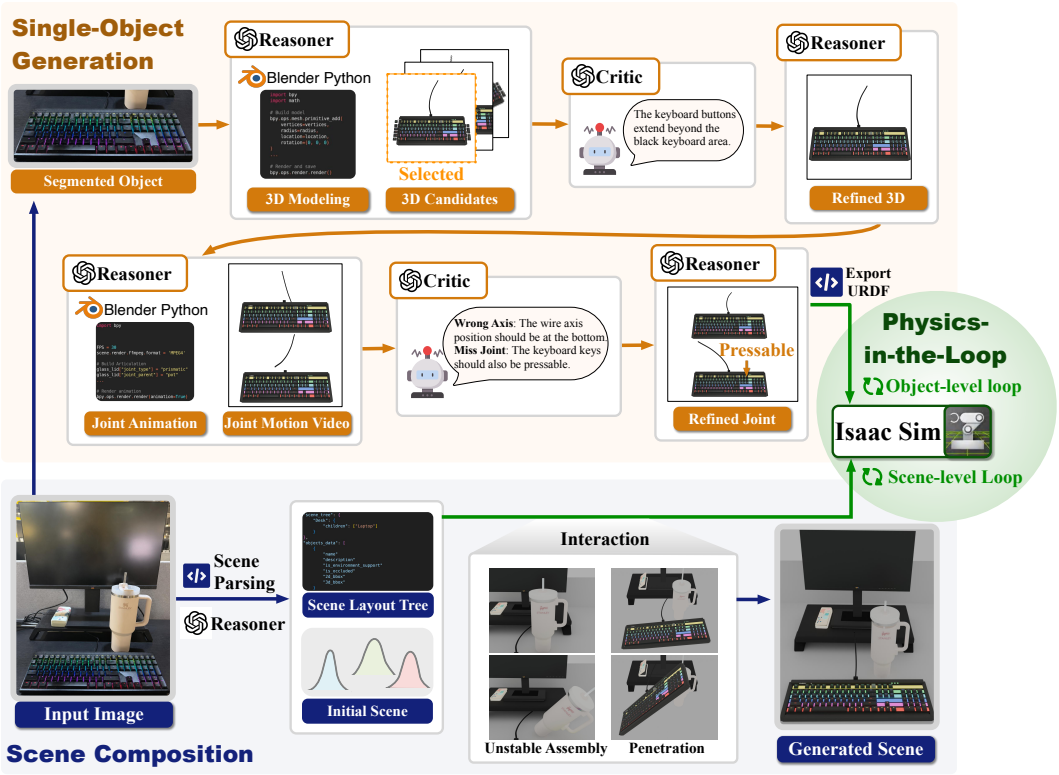

Figure 2: **Overview of the pipeline.** Given a monocular RGB image, NeoWorld-Pro first performs scene parsing and reasoning to establish a scene tree and layout. Each foreground object is then programmed into an executable Blender script to generate geometry and articulation. A two-level *physics-in-the-loop* mechanism ensures simulation-readiness: (1) an *object-level loop* for resting stability and joint executability, and (2) a CEM-based *scene-level loop* that optimizes scene plausibility, such as relative object scale and placement, collision-free spatial consistency, and stability.

3 Method

The NeoWorld-Pro framework reformulates monocular scene reconstruction by shifting from unstructured geometric generation to a structured, program-centric paradigm. By representing scenes as compositions of executable programs, NeoWorld-Pro transforms visual scenes into simulation-ready functional entities. In this section, we present the technical details of the NeoWorld-Pro framework.

3.1 Overview: Scene Reconstruction as Physics-in-the-Loop Program Synthesis

Given a single RGB image I, our goal is to generate a simulation-ready interactive scene, represented as a collection of articulated object URDFs assembled within a Universal Scene Description (USD) stage S. This requires the simultaneous inference of part-level geometry, kinematics, and spatial relationships. To ensure these components are mutually consistent and physically stable within a simulator, NeoWorld-Pro moves beyond traditional dense reconstruction toward a structured, program-centric representation that bridges the gap between perception and physical execution.

As shown in Figure 2, NeoWorld-Pro first leverages MLLMs to synthesize executable Blender Python programs as a unified intermediate representation, which are subsequently compiled into standard URDF and USD assets. This program-centric paradigm offers several key advantages: it is *executable* for direct physical simulation, *interpretable* for MLLM-based inspection and verification, and *editable* for iterative scene refinement. Moreover, by exploiting the rich commonsense priors of foundation models, NeoWorld-Pro can logically infer occluded internal structures and missing physical attributes without relying on dataset-specific category priors.

Algorithm 1 CEM for Scene Composition Refinement

Inputs: Initial layout $\mu^{(0)}$, covariance $\Sigma^{(0)}$, iterations T, samples N, elite ratio ρ, input image I
for $t = 0 \dots T-1$ **do**
  Sample N candidates $\{z^{(n)}\}_{n=1}^{N} \sim \mathcal{N}(\mu^{(t)}, \Sigma^{(t)})$
  for $n = 1 \dots N$ **do**
    Compose USD stage using base layout perturbed by $z^{(n)}$
    Run an Isaac Sim simulation rollout
    Compute physical penalties $D_{\text{pen}}(z^{(n)})$ and $D_{\text{drift}}(z^{(n)})$
    Render stage from canonical view and query MLLM for semantic score $S_{\text{sem}}(z^{(n)}; I)$
    Compute total reward $R(z^{(n)})$
  end for
  Select top-K candidates where $K = \rho N$
  Update $\mu^{(t+1)}, \Sigma^{(t+1)}$ using the sample mean and variance of the top-K elites
  if convergence criteria met **then**
    break
  end if
end for
Return: Best candidate layout

Once the executable representation is established, the remaining challenge is to ensure its physical validity and simulation readiness. We address this through a *two-level physics-in-the-loop* framework that decomposes the pipeline into object-level and scene-level refinement stages, each designed to handle different types of reconstruction errors:

- Object-level loop (Sec. 3.2): Refines discrete structural and physical attributes, including kinematic topology, joint limits, and inertial properties, through iterative MLLM-guided code editing.
- Scene-level loop (Sec. 3.3): Optimizes continuous scene assembly variables, such as relative scale, orientation, and spatial offsets, using the Cross-Entropy Method (CEM) for derivative-free optimization under simulation feedback.

3.2 Object-Level Procedural Programming and Articulation Refinement

This section describes how NeoWorld-Pro reconstructs physically consistent articulated objects and the corresponding scene layout from a single RGB image. The pipeline consists of three stages: scene parsing, procedural synthesis, and object-level physics refinement. Each stage follows a lightweight generate-critic-refine paradigm to improve robustness and physical consistency.

Step 1: Scene parsing. An `MLLM-Reasoner` (*e.g.*, GPT-5.6-sol) first predicts (i) the scene hierarchy tree, (ii) 2D and 3D bounding boxes, and (iii) semantic attributes indicating whether an object serves as environmental support or is partially occluded. This joint prediction allows relational information (*e.g., a cup resting on a table*) to directly constrain object localization and layout estimation, avoiding heuristic downstream assembly. To improve parsing quality, an `MLLM-Critic` evaluates the predictions based on bounding-box tightness, completeness, and hierarchical consistency. The resulting object crops are then forwarded to the procedural synthesis stage.

Step 2: Procedural asset programming. For each segmented object crop, an `MLLM-Programmer` sequentially generates (i) a Blender geometry program, (ii) an articulation program, and (iii) a URDF export script. Each stage is wrapped within a runtime debugging loop to resolve execution failures, followed by a critic-refine loop in which rendered outputs are compared against the input crop. We deliberately choose executable code over direct mesh generation as our intermediate representation, as it is highly reusable across stages and naturally leverages the MLLM's commonsense reasoning to hallucinate occluded internal structures (*e.g., inferring a complete drawer cavity from a visible handle*) without relying on dataset-specific priors. The resulting URDFs are therefore structurally valid and articulation-aware, but their physical behavior remains unverified.

Step 3: Object-level physics-in-the-loop refinement. To ensure physical validity, the generated URDFs are evaluated in Isaac Sim under two standardized physics scenarios: *resting stability* and *joint executability*. We use CoACD [31] to generate collision geometries for the reconstructed objects. The stability test assesses whether each object maintains its intended resting configuration under

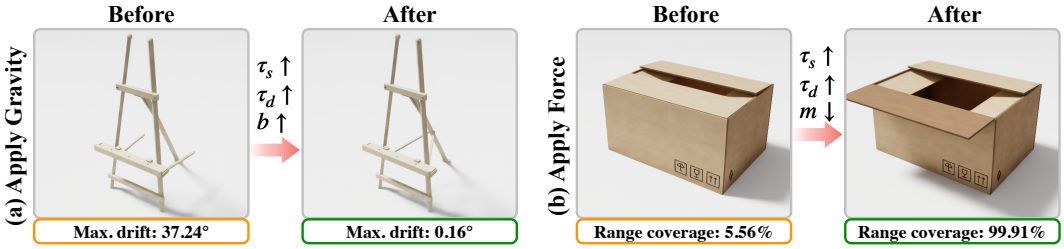

Figure 3: **Physics-in-the-loop refinement improves resting stability and joint executability.** (a) Increasing static friction torque τ_s, dynamic friction torque τ_d, and passive viscous damping b reduces the maximum angular drift of the easel braces under gravity from $37.24°$ to $0.16°$. (b) Reducing τ_s, τ_d, and flap mass m increases joint range coverage from 5.56% to 99.91% under force limit.

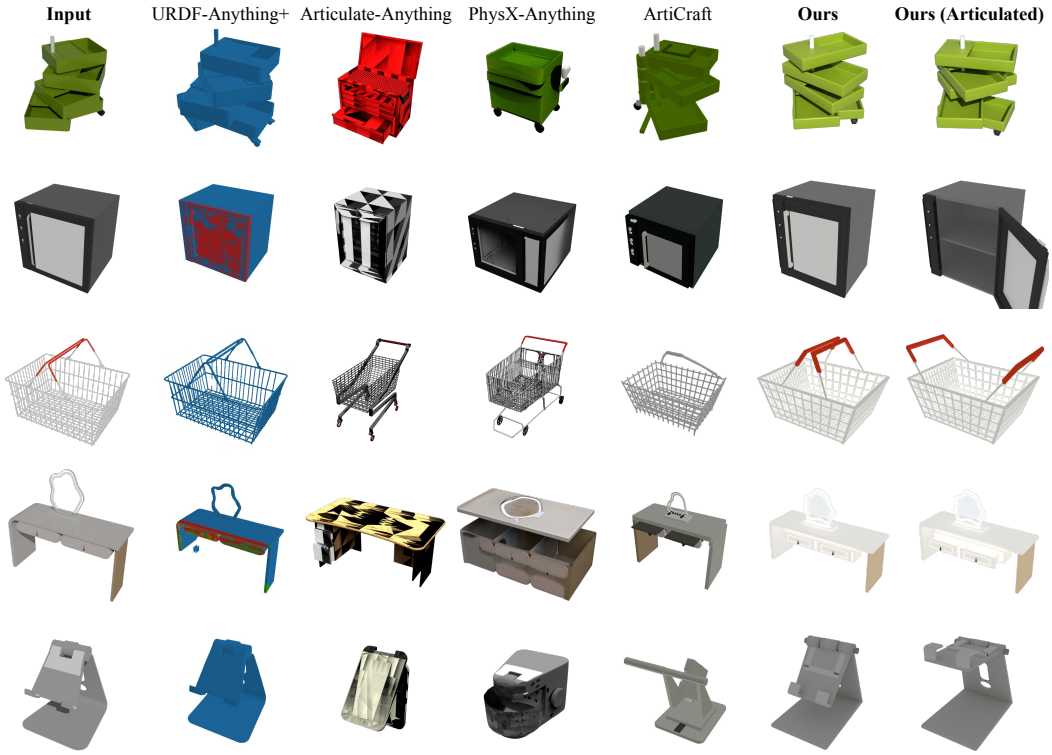

Figure 4: **Object-level reconstruction results of articulated assets.** Compared with existing methods, NeoWorld-Pro recovers more faithful geometry, appearance, and articulation structure.

gravity, without toppling, penetration, or unintended disassembly. For articulated objects, actuation tests determine whether the intended joint motion can be completed under physically plausible force or torque limits. Acceptance is determined by numerical criteria computed from simulator-recorded trajectories. For failed tests, the `MLLM-Reasoner` integrates trajectory measurements with simulation frames to diagnose failure mechanisms and propose parameter updates, as illustrated in Fig. 3. The validated resting pose of each object subsequently serves as its initial configuration for scene-level composition. At this stage, each reconstructed object is individually physically consistent. However, ensuring correct relative scale, placement, and interaction among objects requires scene-level optimization, which we introduce next.

3.3 Scene-Level Composition and CEM Refinement

This section assembles the generated URDF assets into a complete and refined USD, which consists of two stages: deterministic scene composition from the inferred layout, followed by stochastic optimization via a reward-driven CEM that integrates physical simulation and MLLM-based evaluation.



Figure 5: **Qualitative scene reconstruction on our synthetic scene dataset.** Each row compares reconstructions from the same monocular input. NeoWorld-Pro more faithfully preserves object identity, relative scale and orientation, and inter-object layout across both tabletop and larger workspace scenes. "N/A" denotes examples outside TabletopGen's tabletop setting.

Step 4: Scene composition. We construct the initial scene via a top-down traversal of the scene tree, using each object's validated resting pose as its local configuration. The horizontal coordinates (x, y) and yaw rotation are initialized from the parsed spatial layout, while the vertical coordinate is determined by aligning each child's transformed bottom surface with the top surface of its parent. This strategy provides a geometrically consistent initialization while reducing floating and inter-object penetration. Non-environment objects are instantiated via USD references to preserve their URDF articulation properties, whereas environment supports are instantiated as static primitives and excluded from subsequent optimization. Although this deterministic construction provides a plausible initialization, it often introduces physical artifacts such as inter-object penetration, unstable stacking, scale inconsistency, and incorrect relative orientation. These errors are difficult for MLLMs to correct in continuous pose space but can be directly quantified using a physics engine.

Step 5: CEM-based scene refinement. We maintain a diagonal Gaussian distribution over a four-dimensional perturbation vector for each non-environment object o_i. We treat the initial layout as the zero-mean configuration, with $z_i = (\Delta x_i, \Delta y_i, \Delta \psi_i, \log s_i) \in \mathbb{R}^4$ representing axial translation, yaw rotation, and isotropic scale in parent coordinates. We use blockwise CEM, selecting an active object



Figure 6: **Reconstruction from real-world images.** The results illustrate generalization to real-world appearance and clutter while retaining functional articulations.

group based on the current failure relationships. Each iteration samples $N = 16$ candidates from the bounded Gaussian distributions using rejection sampling. The top $K = 4$ candidates determine the updated means and variances of the active group. Standard deviations are bounded below by 20% of their initial values. If all candidates have equal rewards, the distributions remain unchanged. At each iteration, we sample N candidate layouts, instantiate them in USD, and simulate them in Isaac Sim with gravity enabled to evaluate the stability of inter-object support relationships. The top-K candidates are updated under the Gaussian distribution via a hybrid reward, where $w_* \geq 0$ are scalar weights, I is the input image, and z is the candidate perturbation vector:

$$R(z) = w_{\text{sem}} S_{\text{sem}}(z; I) - w_{\text{pen}} D_{\text{pen}}(z) - w_{\text{drift}} D_{\text{drift}}(z). \tag{1}$$

Here, physical terms are measured directly in simulation: D_{pen} aggregates threshold exceedances of maximum PhysX contact penetration across object pairs, while D_{drift} combines orientation deviation, terminal pose and velocity variation, and support and attachment errors. The semantic term S_{sem} uses an MLLM to assess rendered candidates against the input image, emphasizing relational cues such as depth ordering, relative scale, and spatial consistency. This process repeats until convergence or a fixed iteration budget (Algorithm 1). Implementation details are provided in Appendix C. By resolving discrete structural attributes in Sec. 3.2 and continuous scene optimization in this section, NeoWorld-Pro forms a unified physics-in-the-loop framework that jointly enforces articulation correctness and global spatial consistency, distinguishing it from prior open-loop approaches.

4 Experiments

4.1 Experimental Setups

We evaluate NeoWorld-Pro on both a public articulated-object benchmark and a newly constructed multi-object simulation benchmark. We introduce a synthetic scene dataset for monocular reconstruction in physically realistic multi-object environments. The dataset contains 90 articulated object categories, composing 30 USD-format object assemblies in total. Each scene contains multiple articulated or rigid objects with both intra-object and inter-object occlusions, and is paired with several downstream manipulation tasks for evaluation beyond reconstruction and articulation prediction.

Table 1: **Object-level evaluation of joint articulation and kinematics.** *#Total*, *#Pred*, and *#Hit* are the number of ground-truth joints, predicted joints, and matched hits. We report the joint miss rate (*Miss*), joint type identification error (*Type*), orientation error (*Axis*), and positional error (*Pivot*).

Method	Total	#Pred	#Hit	Miss ↓	Type↓	Axis↓	Pivot↓
Articulate-Anything [11]	245	170	95	61.22	69.39	63.25	0.32
PhysX-Anything [2]	245	192	126	48.57	58.78	45.80	0.37
URDF-Anything+ [34]	245	96	74	69.80	82.86	50.41	0.40
Articraft [46]	245	304	190	22.45	29.39	30.14	0.23
Ours	**245**	**268**	**216**	**11.83**	**18.78**	**28.52**	**0.19**

Table 2: **Scene-level results on the full benchmark and the 11-scene tabletop subset.** S-Pen, P-Pen, S-Stab, and O-Stab denote scene-level (inter-object), part-level (inner-object) penetration rate, and scene-/object-level stability, respectively. These metrics are percentages. MLLM scores (the last three columns) use a 1–7 scale.

Method	S-Pen↓	P-Pen↓	S-Stab↑	O-Stab↑	CLIP↑	DINOv2↑	Complete↑	Layout↑	Function↑
Full benchmark									
VIGA [40]	56.67	63.37	23.33	27.33	0.8831	0.6087	4.70	3.95	3.45
SAGE [35]	6.67	2.41	70.00	91.60	0.8729	0.5379	3.80	2.60	2.90
Ours	**0.00**	**0.00**	**100.00**	**100.00**	**0.9128**	**0.7855**	**6.20**	**5.35**	**5.58**
Tabletop subset									
VIGA [40]	54.55	67.26	18.18	25.66	0.8764	0.5950	4.55	3.91	3.00
TabletopGen [30]	45.45	26.53	36.36	67.35	0.8917	0.6266	4.36	3.73	3.64
SAGE [35]	9.09	4.17	**100.00**	**100.00**	0.8741	0.4742	3.45	2.64	2.73
Ours	**0.00**	**0.00**	**100.00**	**100.00**	**0.9205**	**0.7166**	**6.10**	**5.70**	**5.58**

For single-object image-to-URDF reconstruction, we compare NeoWorld-Pro with four representative baselines: PhysX-Anything [2] and URDF-Anything+ [34], which generate simulation-compatible object assets from visual inputs; Articulate-Anything [11], which follows a retrieval-and-critic paradigm for articulated structure estimation; and Articraft [46], a state-of-the-art generation method that synthesizes articulated assets through agentic program synthesis, which mainly focuses on text input. For scene-level reconstruction, we compare against TabletopGen [30], VIGA [40], and SAGE [35], spanning simulation-ready generation from tabletop to room-scale scenes.

4.2 Object-Level Reconstruction Results of Articulated Assets

We present qualitative results of single-object image-to-URDF reconstruction in Figure 4. As shown, our method achieves the most accurate recovery of articulated joints and produces highly accurate geometric reconstructions. Table 1 reports quantitative articulation results. NeoWorld-Pro achieves the lowest joint miss rate, the largest number of matched joints, and the lowest joint type, axis, and pivot errors among the evaluated methods, indicating more complete and accurate joint recovery. Detailed evaluation protocols and additional results on appearance, geometry, and cross-modal similarity are provided in Appendix A.1.

4.3 Scene-Level Reconstruction and Assembly Results

We provide a qualitative comparison on our synthetic scene dataset in Figure 5. Across tabletop, appliance, and workspace layouts, NeoWorld-Pro preserves the complete set of foreground objects while more faithfully recovering their relative scales, orientations, and spatial relationships. In contrast, competing methods often omit or replace objects and distort layouts, especially as the number and diversity of objects increase. Beyond static reconstruction quality, we further demonstrate that our reconstructed scenes are free of object penetration, remain stable under dynamic simulation, and support physically meaningful downstream interactions, including stable phone–stand support, robotic grasping of test tubes from racks, and controllable domino-pushing behaviors. These scenarios require accurate recovery of inter-object geometry, articulation, contact, and spatial dependencies, which existing open-loop methods struggle to achieve reliably.

For quantitative comparisons, Table 2 summarizes scene-level inter-object penetration rate, part-level self-penetration rate, and scene- and object-level stability under gravity. All physical metrics are evaluated in a physics simulator (*e.g.*, Isaac Sim). We additionally evaluate CLIP and DINOv2 similarity, together with MLLM-based judgments of scene completeness, layout fidelity, and functional feasibility. MLLM-based scores assess whether the reconstructed scene preserves all relevant objects (*completeness*), faithfully reproduces their spatial arrangement (*layout fidelity*), and supports the intended physical interactions (*functional feasibility*). Across both the full benchmark and the tabletop subset, NeoWorld-Pro eliminates inter-object collisions, maintains strong stability, and achieves the highest semantic scores. Figure 6 further demonstrates generalization to two real-world monocular images. Despite background clutter, illumination variation, and partial occlusion, NeoWorld-Pro reconstructs the primary objects and their spatial relationships. The actuated renderings additionally show that the inferred movable components remain functional after scene assembly.

4.4 Image-to-Policy Demonstration for the Manipulation Task

As illustrated in Figure 1, we further demonstrate image-to-policy execution on a test-tube placement task. Given a monocular scene image containing a test tube and a rack, NeoWorld-Pro reconstructs both objects and their relative spatial configuration. Given the test-tube placement task, which requires the tube to pass through an opening and be supported by the rack, an MLLM specifies the initial end-effector position and displacement for executing the manipulation in Isaac Sim. Errors in the relative pose or scale of the reconstructed objects may render the predicted motion infeasible and cause task failure. We use this execution outcome as a feedback signal to iteratively refine the scene layout until the placement succeeds, demonstrating that the reconstructed scene supports physically grounded downstream manipulation rather than merely visual resemblance.

4.5 Ablation Studies

We validate the refinement components of NeoWorld-Pro at both object and scene levels, evaluating their effects on articulation recovery and scene-level task execution. At the object level, we remove either the geometry or articulation critic to assess their impact on joint recovery. At the scene level, we removes closed-loop simulation feedback and directly uses the initially composed scene.

Table 3: **Analyses of model component.**

Method	Success (%)	Miss (%)
w/o geom. refine critic	76.47	14.84
w/o artic. refine critic	83.33	17.18
w/o simul. (open-loop)	44.05	11.72
NeoWorld-Pro	**92.85**	**7.03**

As shown in Table 3, without physical validation, errors from parsing and generation remain uncorrected, leading to failures such as penetration, unstable stacking, incorrect scale, and invalid articulation, as illustrated in Figure 1. Ablations of the CEM reward terms are reported in Appendix A.2. Removing the semantic score, inter-object penetration penalty, or stability penalty reduces task success, indicating that all three terms contribute to reliable scene-level execution.

5 Conclusion and Limitation

We presented NeoWorld-Pro, a physics-in-the-loop framework for monocular real-to-sim scene reconstruction. By reformulating reconstruction as MLLM-driven program synthesis and coupling it with object-level and scene-level simulation feedback, NeoWorld-Pro produces simulation-ready assets with accurate geometry, articulation, and spatial relationships. Extensive experiments demonstrate strong improvements over open-loop baselines in appearance fidelity, geometric accuracy, articulation prediction, and downstream manipulation tasks, highlighting the effectiveness of integrating executable representations with physical validation.

NeoWorld-Pro is currently limited to rigid and articulated objects under quasi-static physics. It does not yet model highly deformable or continuous media such as cloth, fluids, or granular materials, where dynamics are significantly more complex and less amenable to URDF-style representations. Our evaluation is also limited to scenes with fewer than 10 objects. Extending the framework to support deformable assets, richer material properties, and denser scenes remains an important direction for future work. Improving computational efficiency to accelerate scene reconstruction is another important direction for future work.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant 62250062), the Shanghai Municipal Science and Technology Major Project (Grant 2021SHZDZX0102), and the Fundamental Research Funds for the Central Universities.

Competing-interests Disclosure

The authors declare that they have no competing interests.

References

[1] Ziang Cao, Zhaoxi Chen, Liang Pan, and Ziwei Liu. Physx-3d: Physical-grounded 3d asset generation. *arXiv preprint arXiv:2507.12465*, 2025.

[2] Ziang Cao, Fangzhou Hong, Zhaoxi Chen, Liang Pan, and Ziwei Liu. Physx-anything: Simulation-ready physical 3d assets from single image. *arXiv preprint arXiv:2511.13648*, 2025.

[3] Chuhao Chen, Isabella Liu, Xinyue Wei, Hao Su, and Minghua Liu. Freeart3d: Training-free articulated object generation using 3d diffusion. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pages 1–13, 2025.

[4] Zoey Chen, Aaron Walsman, Marius Memmel, Kaichun Mo, Alex Fang, Karthikeya Vemuri, Alan Wu, Dieter Fox, and Abhishek Gupta. Urdformer: A pipeline for constructing articulated simulation environments from real-world images. *arXiv preprint arXiv:2405.11656*, 2024.

[5] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017.

[6] Haoran Geng, Helin Xu, Chengyang Zhao, Chao Xu, Li Yi, Siyuan Huang, and He Wang. Gapartnet: Cross-category domain-generalizable object perception and manipulation via generalizable and actionable parts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7081–7091, 2023.

[7] Anna-Maria Halacheva, Yang Miao, Jan-Nico Zaech, Xi Wang, Luc Van Gool, and Danda Pani Paudel. Articulate3d: Holistic understanding of 3d scenes as universal scene description. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5633–5644, 2025.

[8] Yumeng He, Ying Jiang, Jiayin Lu, Yin Yang, and Chenfanfu Jiang. Spark: Sim-ready part-level articulated reconstruction with vlm knowledge. *arXiv preprint arXiv:2512.01629*, 2025.

[9] Zhao Huang, Boyang Sun, Alexandros Delitzas, Jiaqi Chen, and Marc Pollefeys. React3d: Recovering articulations for interactive physical 3d scenes. *IEEE Robotics and Automation Letters*, 2026.

[10] Jens U Kreber and Joerg Stueckler. Guiding diffusion-based articulated object generation by partial point cloud alignment and physical plausibility constraints. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3206–3214, 2025.

[11] Long Le, Jason Xie, William Liang, Hung-Ju Wang, Yue Yang, Yecheng Jason Ma, Kyle Vedder, Arjun Krishna, Dinesh Jayaraman, and Eric Eaton. Articulate-anything: Automatic modeling of articulated objects via a vision-language foundation model. *arXiv preprint arXiv:2410.13882*, 2024.

[12] Haitian Li, Haozhe Xie, Junxiang Xu, Beichen Wen, Fangzhou Hong, and Ziwei Liu. Monoart: Progressive structural reasoning for monocular articulated 3d reconstruction. *arXiv preprint arXiv:2603.19231*, 2026.

[13] Jialin Li, Bin Fu, Ruiping Wang, and Xilin Chen. Gear: Geometry-motion alternating refinement for articulated object modeling with gaussian splatting. *arXiv preprint arXiv:2604.07728*, 2026.

[14] Xiaolong Li, He Wang, Li Yi, Leonidas J Guibas, A Lynn Abbott, and Shuran Song. Category-level articulated object pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3706–3715, 2020.

[15] Zhe Li, Xiang Bai, Jieyu Zhang, Zhuangzhe Wu, Che Xu, Ying Li, Chengkai Hou, and Shanghang Zhang. Urdf-anything: Constructing articulated objects with 3d multimodal language model. *arXiv preprint arXiv:2511.00940*, 2025.

[16] Zizhang Li, Cheng Zhang, Zhengqin Li, Henry Howard-Jenkins, Zhaoyang Lv, Chen Geng, Jiajun Wu, Richard Newcombe, Jakob Engel, and Zhao Dong. Art: Articulated reconstruction transformer. *arXiv preprint arXiv:2512.14671*, 2025.

[17] Xinyu Lian, Zichao Yu, Ruiming Liang, Yitong Wang, Li Ray Luo, Kaixu Chen, Yuanzhen Zhou, Qihong Tang, Xudong Xu, Zhaoyang Lyu, et al. Infinite mobility: Scalable high-fidelity synthesis of articulated objects via procedural generation. *arXiv preprint arXiv:2503.13424*, 2025.

[18] Jiong Lin, Lechen Zhang, Kwansoo Lee, Jialong Ning, Judah Goldfeder, and Hod Lipson. Autourdf: Unsupervised robot modeling from point cloud frames using cluster registration. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 27628–27637, 2025.

[19] Jiayi Liu, Denys Iliash, Angel X Chang, Manolis Savva, and Ali Mahdavi-Amiri. Singapo: Single image controlled generation of articulated parts in objects. *arXiv preprint arXiv:2410.16499*, 2024.

[20] Ruijie Lu, Yu Liu, Jiaxiang Tang, Junfeng Ni, Yuxiang Wang, Diwen Wan, Gang Zeng, Yixin Chen, and Siyuan Huang. Dreamart: Generating interactable articulated objects from a single image. *arXiv preprint arXiv:2507.05763*, 2025.

[21] Zhao Mandi, Yijia Weng, Dominik Bauer, and Shuran Song. Real2code: Reconstruct articulated objects via code generation. *arXiv preprint arXiv:2406.08474*, 2024.

[22] Xiaowen Qiu, Jincheng Yang, Yian Wang, Zhehuan Chen, Yufei Wang, Tsun-Hsuan Wang, Zhou Xian, and Chuang Gan. Articulate anymesh: Open-vocabulary 3d articulated objects modeling. *arXiv preprint arXiv:2502.02590*, 2025.

[23] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[24] Licheng Shen, Saining Zhang, Honghan Li, Peilin Yang, Zihao Huang, Zongzheng Zhang, and Hao Zhao. Gaussianart: Unified modeling of geometry and motion for articulated objects. *arXiv preprint arXiv:2508.14891*, 2025.

[25] Jiayi Su, Youhe Feng, Zheng Li, Jinhua Song, Yangfan He, Botao Ren, and Botian Xu. Artformer: Controllable generation of diverse 3d articulated objects. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1894–1904, 2025.

[26] Maxim Tatarchenko, Stephan R Richter, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3d reconstruction networks learn? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3405–3414, 2019.

[27] Jiawei Wang, Dingyou Wang, Jiaming Hu, Qixuan Zhang, Jingyi Yu, and Lan Xu. Kinematify: Open-vocabulary synthesis of high-dof articulated objects. *arXiv preprint arXiv:2511.01294*, 2025.

[28] Penghao Wang, Siyuan Xie, Hongyu Yan, Xianghui Yang, Jingwei Huang, Chunchao Guo, and Jiayuan Gu. Artllm: Generating articulated assets via 3d llm. *arXiv preprint arXiv:2603.01142*, 2026.

[29] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[30] Ziqian Wang, Yonghao He, Licheng Yang, Wei Zou, Hongxuan Ma, Liu Liu, Wei Sui, Yuxin Guo, and Hu Su. Tabletopgen: Instance-level interactive 3d tabletop scene generation from text or single image. *arXiv preprint arXiv:2512.01204*, 2025.

[31] Xinyue Wei, Minghua Liu, Zhan Ling, and Hao Su. Approximate convex decomposition for 3d meshes with collision-aware concavity and tree search. *ACM Transactions on Graphics (TOG)*, 41(4):1–18, 2022.

[32] Yijia Weng, He Wang, Qiang Zhou, Yuzhe Qin, Yueqi Duan, Qingnan Fan, Baoquan Chen, Hao Su, and Leonidas J Guibas. Captra: Category-level pose tracking for rigid and articulated objects from point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13209–13218, 2021.

[33] Ruiqi Wu, Xinjie Wang, Liu Liu, Chunle Guo, Jiaxiong Qiu, Chongyi Li, Lichao Huang, Zhizhong Su, and Ming-Ming Cheng. Dipo: Dual-state images controlled articulated object generation powered by diverse data. *arXiv preprint arXiv:2505.20460*, 2025.

[34] Zhuangzhe Wu, Yue Xin, Chengkai Hou, Minghao Chen, Yaoxu Lyu, Jieyu Zhang, and Shanghang Zhang. Urdf-anything+: Autoregressive articulated 3d models generation for physical simulation. *arXiv preprint arXiv:2603.14010*, 2026.

[35] Hongchi Xia, Xuan Li, Zhaoshuo Li, Qianli Ma, Jiashu Xu, Ming-Yu Liu, Yin Cui, Tsung-Yi Lin, Wei-Chiu Ma, Shenlong Wang, Shuran Song, and Fangyin Wei. Sage: Scalable agentic 3d scene generation for embodied ai. *arXiv preprint arXiv:2602.10116*, 2026.

[36] Hongchi Xia, Entong Su, Marius Memmel, Arhan Jain, Raymond Yu, Numfor Mbiziwo-Tiapo, Ali Farhadi, Abhishek Gupta, Shenlong Wang, and Wei-Chiu Ma. Drawer: Digital reconstruction and articulation with environment realism. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21771–21782, 2025.

[37] WenBo Xu, Liu Liu, Li Zhang, Dan Guo, and RuoNan Liu. Motionanymesh: Physics-grounded articulation for simulation-ready digital twins. *arXiv preprint arXiv:2603.12936*, 2026.

[38] Zihao Yan, Ruizhen Hu, Xingguang Yan, Luanmin Chen, Oliver Van Kaick, Hao Zhang, and Hui Huang. Rpm-net: recurrent prediction of motion and parts from point cloud. *arXiv preprint arXiv:2006.14865*, 2020.

[39] Yixuan Yang, Luyang Xie, Zhen Luo, Zixiang Zhao, Tongsheng Ding, Mingqi Gao, and Feng Zheng. Artiworld: Llm-driven articulation of 3d objects in scenes. *arXiv preprint arXiv:2511.12977*, 2025.

[40] Shaofeng Yin, Jiaxin Ge, Zora Zhiruo Wang, Chenyang Wang, Xiuyu Li, Michael J Black, Trevor Darrell, Angjoo Kanazawa, and Haiwen Feng. Vision-as-inverse-graphics agent via interleaved multimodal reasoning. *arXiv preprint arXiv:2601.11109*, 2026.

[41] Sylvia Yuan, Ruoxi Shi, Xinyue Wei, Xiaoshuai Zhang, Hao Su, and Minghua Liu. Larm: A large articulated object reconstruction model. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pages 1–12, 2025.

[42] Chuanrui Zhang, Minghan Qin, Yuang Wang, Baifeng Xie, Hang Li, and Ziwei Wang. Simart: Decomposing monolithic meshes into sim-ready articulated assets via mllm. *arXiv preprint arXiv:2603.23386*, 2026.

[43] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[44] Yanpeng Zhao, Shanyan Guan, Yunbo Wang, Yanhao Ge, Wei Li, and Xiaokang Yang. Neoworld: Neural simulation of explorable virtual worlds via progressive 3d unfolding. *arXiv preprint arXiv:2509.24441*, 2025.

[45] Junsheng Zhou, Jinsheng Wang, Baorui Ma, Yu-Shen Liu, Tiejun Huang, and Xinlong Wang. Uni3d: Exploring unified 3d representation at scale. *arXiv preprint arXiv:2310.06773*, 2023.

[46] Matt Zhou, Ruining Li, Xiaoyang Lyu, Zhaomou Song, Zhening Huang, Chuanxia Zheng, Christian Rupprecht, Andrea Vedaldi, and Shangzhe Wu. Articraft: An agentic system for scalable articulated 3d asset generation. *arXiv preprint arXiv:2605.15187*, 2026.

Appendix

A Additional Results

A.1 Object-Level Quantitative Evaluation

We further quantify the comparisons from three perspectives. From the **appearance** perspective, (i) we measure *photometric rendering fidelity* using SSIM [29] and LPIPS [43]. (ii) We assess *semantic similarity* using CLIP similarity [23]. From the **geometric** view, (iii) we quantify mesh reconstruction quality using Chamfer distance [5] and F-score [26]. (iv) We report Uni3D similarity [45] as a cross-modal alignment metric between input images and reconstructed 3D meshes. Finally, for the **articulation** results reported in Table 1 of the main text, (v) given a predicted URDF and its ground truth, we render videos by sequentially actuating corresponding joints in both models. An MLLM compares the paired videos to determine whether each ground-truth joint is correctly recovered. We report the total number of joints and hits, as well as the miss rate. Table 4 summarizes the appearance, geometry, and cross-modal similarity results. NeoWorld-Pro demonstrates strong overall performance across these metrics. It achieves the best LPIPS and CLIP scores while remaining competitive in SSIM, obtains the highest F-scores at both thresholds and the second-lowest Chamfer distance, and provides the strongest cross-modal 3D alignment according to Uni3D similarity.

Table 4: **Object-level reconstruction results in appearance, cross-modal similarity, and geometry evaluation. Bold** indicates the best result and underline indicates the second best. '/' denotes methods that do not produce appearance outputs.

	Appearance Evaluation			**Similarity**	**Geometry Evaluation**		
Method	SSIM↑	LPIPS↓	CLIP↑	Uni3D↑	CD×10 ↓	F@0.01↑	F@0.05↑
Articulate-Anything [11]	0.7646	0.2930	0.7056	0.2125	0.3989	10.76	41.07
PhysX-Anything [2]	0.7657	0.2991	0.7937	0.2709	0.7150	4.61	24.72
URDF-Anything+ [34]	/	/	/	0.1638	**0.1696**	24.30	65.18
Articraft [46]	**0.8926**	0.2149	0.8999	0.2566	0.2155	22.14	62.25
Ours	0.8894	**0.1989**	**0.9042**	**0.2812**	0.1809	**25.77**	**67.06**

A.2 Ablation of CEM Reward Terms

Table 5 reports ablations of the scene-level CEM reward terms. We report task success rate because these terms primarily affect scene execution rather than object-level joint discovery. The variant w/o S_{sem} removes the MLLM-based semantic score and relies solely on simulator-derived physical signals, reducing task success from 92.85% to 48.81%. This result highlights the importance of semantic feedback for preserving task-relevant object arrangements and consistency with the input image. The variants w/o D_{pen} and w/o D_{drift} remove the inter-object penetration penalty and the stability penalty, respectively, reducing task success to 76.47% and 88.24%. These results indicate that both semantic and physical reward terms contribute to reliable task execution.

Table 5: **Ablation of CEM reward terms.**

Method	Success↑ (%)
w/o S_{sem}	48.81
w/o D_{pen}	76.47
w/o D_{drift}	88.24
NeoWorld-Pro	**92.85**

B Scene Task Configurations

Figure 7 shows additional examples from our scene-level task set. These tasks cover diverse object configurations and manipulation goals, including placement, assembly, and interaction with articulated

parts. They are designed to test whether the reconstructed scene is not only visually plausible, but also physically executable in simulation: objects must have reasonable relative scale and pose, avoid severe interpenetration, and expose task-relevant affordances for downstream manipulation.

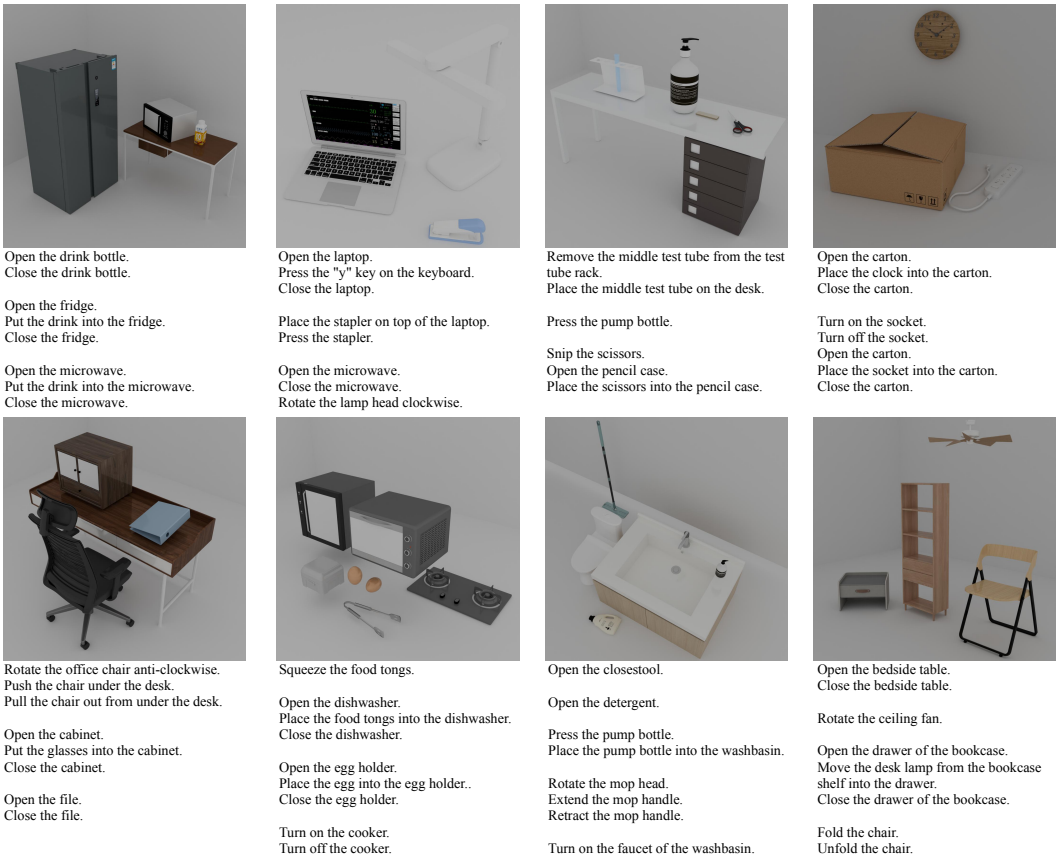

Figure 7: **Examples of scenes and their corresponding manipulation tasks.**

C Scene-Level Refinement Implementation Details

During scene composition, each object URDF produced by the object-level pipeline is deterministically rescaled to match the corresponding 3D bounding box predicted during scene parsing. This step is necessary because the procedural Blender programs are generated at a convenient modeling scale, which is not guaranteed to match the physical scale implied by the input scene. We therefore treat the predicted bounding box size as the absolute scale initialization before CEM refinement. The subsequent CEM scale variable is defined in relative log-scale space, so $\log s = 0$ corresponds to preserving the bounding-box-derived scale. This design separates absolute scale initialization from local scale correction, allowing CEM to focus on small, physically grounded refinements rather than recovering global object size from scratch.

For each non-environment object, the CEM distribution is initialized as a diagonal Gaussian over $(\Delta x, \Delta y, \Delta\psi, \log s)$. The planar standard deviation is set proportional to the object size, with σ_{xy} equal to 5% of the horizontal bounding-box diagonal. This scale-adaptive choice prevents small objects from being over-perturbed while allowing large objects to move sufficiently during optimization. We set $\sigma_\psi = 15^\circ$ for yaw and $\sigma_{\log s} = 0.1$, corresponding to an approximate 10% scale perturbation. In our experiments, we use blockwise CEM, selecting the active object group from the current failure relationships. Each iteration samples $N = 16$ candidate layouts and retains $K = 4$ elites ranked by reward. The active distributions are updated using the elite sample means and variances, with standard deviations bounded below by 20% of their initial values. Inactive objects

retain the reference layout for that iteration. The search runs for at most $T = 10$ iterations and does not terminate solely because reward improvement stagnates.

The reward terms in Section 3.2 are normalized before weighting so that no single term dominates purely because of its numerical scale. The semantic score S_{sem} is computed by rendering the candidate stage from a fixed canonical camera and providing both the original input image and the candidate rendering to the MLLM. The MLLM scores four relative-layout criteria: depth ordering, relative size, relative orientation, and visible inter-object intersections. Each criterion is assigned a discrete score and the aggregated result is normalized to $[0, 1]$ to obtain S_{sem}. The physical terms are similarly normalized via fixed-threshold normalization before being combined with the semantic term. In implementation, reward assignment prioritizes physical feasibility: candidates that fail the physical checks receive only physical penalties, whereas those that pass receive a unit bonus, a semantic reward, and a small layout-deviation penalty.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly summarize the main contributions, which are consistent with the methods and results presented in the paper

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See Section 5.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: : This paper proposes an empirical framework and system pipeline for scene generation, and thus does not introduce formal theoretical results or mathematical proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide a detailed explanation of the PRISM framework in Section 3. Furthermore, experimental settings, baseline details, CEM hyperparameters, and MLLM prompts are thoroughly documented in Section 4 and Appendix 5. We also plan to open-source our code and the constructed synthetic dataset.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We include the code in the supplementary in an anonymous pattern, and we will soon release our latest version of the code publicly before the decision of the paper acceptance.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Our framework uses zero-shot MLLMs and does not involve a traditional neural-network training phase, so training data splits, training optimizers, and training hyperparameters are not applicable. The paper specifies the evaluation datasets, test settings, baselines, and metrics in Section 4, and details the CEM optimization procedure and its hyperparameters in Appendix 5.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We report average evaluation metrics aggregated over the full benchmark, but do not include error bars, confidence intervals, or statistical significance tests across multiple independent runs. This is due to the high computational cost of repeatedly querying MLLMs and running physics-based simulation evaluations over the entire benchmark.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: The paper provides implementation details in Appendix 5, but does not fully specify the compute resources, memory usage,and execution time required for each experiment. We will include detailed resource requirements in the released code documentation to facilitate reproducibility.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper adheres to the NeurIPS Code of Ethics in all aspects.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: PRISM can positively impact embodied AI and robotics by reducing the manual effort required to construct interactive simulation environments and enabling safer pre-deployment testing in simulation. Potential negative impacts include misuse in unsafe robotic workflows and over-reliance on automatically generated simulations whose physical assumptions may be imperfect.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: This paper does not involve the responsible release of data or models that have a high risk for misuse.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All existing datasets (e.g., PartNet-Mobility), simulation platforms (e.g., Isaac Sim), and pre-trained foundation models utilized in our framework are properly cited in the References section. We strictly adhere to their respective licenses and terms of use for academic research.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The paper introduces a new synthetic multi-object simulation benchmark. The benchmark construction, task setup, evaluation protocol, and dataset statistics are documented in the paper, and the dataset documentation will be released together with the code.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: This paper does not involve crowdsourcing experiments and research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: The core methodology of our PRISM framework heavily relies on Multimodal Large Language Models (MLLMs). We explicitly detail the roles of the MLLM-Reasoner, MLLM-Programmer, and MLLM-Critic for procedural scene programming and physics-in-the-loop refinement in Section 3, and provide the exact prompts in Appendix **??**.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.