3D-Layout-R1: Structured Reasoning for Language-Instructed Spatial Editing

Haoyu Zhen[1,2,*], Xiaolong Li[1,*], Yilin Zhao[1,*], Han Zhang[1], Sifei Liu[1], Kaichun Mo[1], Chuang Gan[2], and Subhashree Radhakrishnan[1,†]

[1] NVIDIA, Santa Clara CA 95051, USA
[2] University of Massachusetts Amherst, Amherst MA 01003, USA

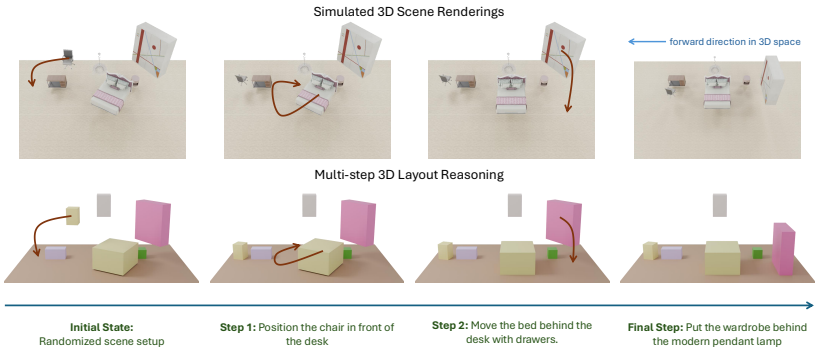

Fig. 1: We introduce **3D-Layout-R1**, which performs multi-step language-guided 3D layout editing, iteratively updating an initially randomized scene into a sequence of spatially consistent intermediate layouts.

Abstract. Large Language Models (LLMs) and Vision Language Models (VLMs) have shown impressive reasoning abilities, yet they struggle with spatial understanding and layout consistency when performing fine-grained visual editing. We introduce a Structured Reasoning framework that performs text-conditioned spatial layout editing via scene-graph reasoning. Given an input scene graph and a natural-language instruction, the model reasons over the graph to generate an updated scene graph that satisfies the text condition while maintaining spatial coherence. By explicitly guiding the reasoning process through structured relational representations, our approach improves both interpretability and control over spatial relationships. We evaluate our method on a new text-guided layout editing benchmark encompassing sorting, spatial alignment, and room-editing tasks. Our training paradigm yields an average 15% improvement in IoU and 25% reduction in center-distance error compared

* Equal contribution.
† Corresponding author: Subhashree Radhakrishnan (subhashreer@nvidia.com).

to Chain of thought Fine-tuning (CoT-SFT) and vanilla GRPO baselines. Compared to SOTA zero-shot LLMs, our best models achieve up to 20% higher mIoU, demonstrating markedly improved spatial precision.

Keywords: Spatial Reasoning · Vision-Language Models · 3D Editing

1 Introduction

Understanding and manipulating 3D scenes through natural language is a fundamental capability for agents and content creation systems. Beyond passive perception, agents must be able to rearrange their surroundings, e.g., "move the chair from the desk to align with the sofa", which requires a sophisticated blend of capabilities: comprehending compositional spatial relationships, understanding semantic intent, and adhering to strict physical constraints to produce a plausible layout. While recent progress in 3D perception and multimodal foundation models has advanced the ability of VLMs to answer spatial questions, far fewer methods can execute structured and multi-step 3D layout edits in response to natural language instructions.

Existing VLM-based spatial reasoning frameworks have primarily focused on passive 3D understanding. Models like SpatialRGPT [8], SpatialLLM [28], SpatialReasoner [27], and 3D-R1 [20] enhance spatial VQA through depth cues, implicit 3D representations, or coordinate-based reasoning. Despite these gains, such systems do not modify the underlying 3D scene and lack the structured mechanisms needed for long-horizon editing. This gap motivates a shift from answering spatial queries to acting upon 3D layouts in an interpretable and physically consistent manner.

Language-driven 3D layout editing has recently emerged as a promising direction. A recent line of works use LLMs as high-level planners to edit existing layouts. In this dominant paradigm, the VLM generates a high-level plan or a set of spatial constraints, which is then passed to a separate, external module, such as a constraint solver [38] or a differentiable optimizer [11, 34], to compute the final 3D poses. While effective for ensuring physical feasibility, these pipelines are limited by manually specified rules or objectives, reduced flexibility for diverse instructions, difficulty handling long-horizon compositional edits, and a largely opaque reasoning process that offers little interpretability or opportunity for correction. Other methods perform one-shot layout prediction or single-object placement [1], but these do not generalize to multi-object rearrangement or sequential editing of existing scenes. To the best of our knowledge, no existing system supports fully integrated, multi-step 3D layout editing that directly reasons over structured spatial representations.

In this work, we introduce **3D-Layout-R1**, a framework that performs structured, interpretable, and language-directed 3D layout editing by reasoning directly over a 3D bounding-box based scene graph as an iterative canvas. Instead of generating a vague, free-form chain-of-thought (CoT), our model, 3D-Layout-R1, produces a structured trace of scene-graph transformations (Fig 1). Each

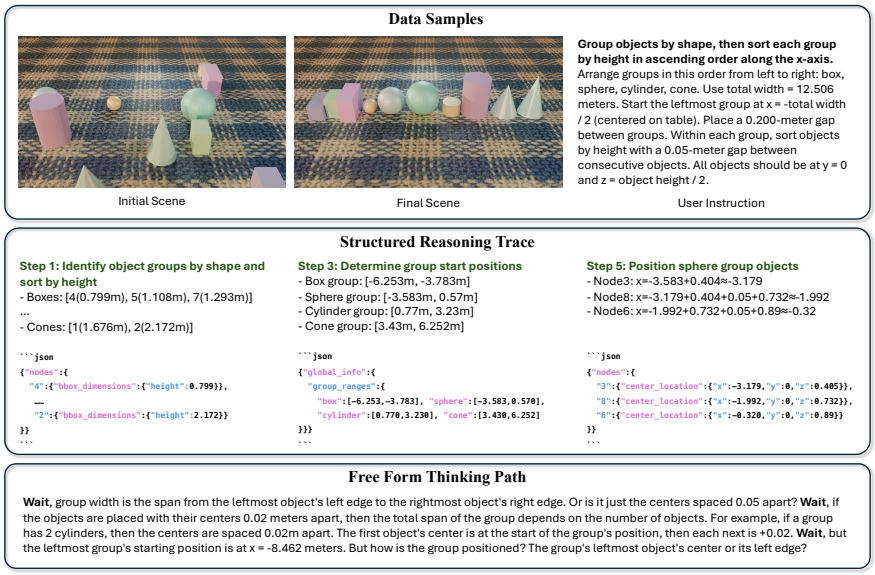

Fig. 2: Example from the synthetic 3D sorting benchmark. Given a instruction to group and sort objects by shape and height, our model generates a concise structured reasoning trace with JSON scene-graph updates that transforms the initial scene into the final layout, in contrast to a long, ambiguous free-form thinking path.

reasoning step is an explicit, verifiable graph edit that directly updates the scene's state. This approach embeds the 3D spatial logic directly within the model's generation process. This allows 3D-Layout-R1 to plan and execute complex, multi-step rearrangements (e.g., "first move the box, then place the lamp next to the book") while ensuring each intermediate step is interpretable and geometrically coherent.

To achieve this, we integrate a GRPO-based reinforcement learning stage that optimizes layout accuracy using a dense 3D IoU reward and collision-aware penalties. By jointly leveraging structured scene-graph reasoning and RL-driven refinement, the model learns to generate precise, physically consistent layout edits that reliably satisfy complex textual instructions.

Our contributions are summarized as follows:

1. We introduce **3D-Layout-R1**, a framework that directly performs multi-step 3D scene editing over a 3D-Layout using language-guided, interpretable chain-of-graph-edits reasoning across three tasks: Sorting, Spatial Alignment, and Room Editing.
2. We release a dataset of 15k 3D scenes with natural language instructions, intermediate chain-of-thought graph edits, and target layouts, providing the first benchmark dedicated to multi-step editing of existing 3D scenes.

3. We design geometric reward functions based on 3D IoU for policy optimization and evaluation, offering continuous and geometrically faithful supervision than distance-based or binary success metrics.
4. 3D-Layout-R1 improves mean IoU by **15%** and reduces center-distance errors by **25–30%** compared to CoT-SFT and GRPO baselines. It achieves up to **20%** higher IoU compared to leading zero-shot LLMs and prior spatial reasoning or layout editing systems.

2 Related Works

Spatial Reasoning. Recent efforts to endow VLMs with spatial intelligence have explored diverse modalities and representations. Several approaches enhance 2D VLMs with 3D positional cues from RGB-D inputs or multi-view images [5,7,8,10,45], while others learn implicit "cognitive maps" from video [36,41] or ground reasoning in 2D coordinates and large-scale embodied datasets [26,31, 33,43]. The most proximate works leverage explicit 3D representations for spatial reasoning. SpatialVLM [6] taught metric relationships by extracting 3D properties from 2D images, while SpatialRGPT [8] generated region-aware Visual Question Answering (VQA) using implicit 3D scene graphs. Furthermore, [28] and [27] incorporated estimated depth, distances, and explicit 3D coordinates to enable multi-step reasoning. In contrast to these works, which primarily focus on spatial VQA using implicit or coordinate-based representations, we investigate whether LLMs/VLMs can reason and update over the structure of a 3D scene graphs.

Language-Driven 3D Layout Editing. Traditional 3D scene editing requires expert knowledge and manual operation. Recent language-driven approaches aim to automate spatial manipulation using LLMs/VLMs. Some methods operate via intermediate representations, such as [23,44], which either edit in 2D before lifting to 3D or construct semantic graph priors for scene decoding. Others model 3D structure directly, including diffusion- or placement-based techniques [1,42], though these are often restricted to simple object-level edits. A complementary line of work uses LLMs as *high-level planners* that produce spatial constraints or initial poses for external optimizers [11, 13, 34, 38], or leverage symbolic and agentic pipelines for procedural scene manipulation [14, 18]. More recent efforts explore direct 3D object generation with supervised or DPO-tuned LLMs [37]. While these approaches demonstrate the promise of natural-language 3D editing, most rely on external solvers and struggle with complex compositional instructions. Concurrent work such as [30] explores end-to-end layout reasoning in 2D, whereas we focus on enabling a LLM to directly reason over and iteratively update a structured 3D scene graph for long-horizon compositional editing without external optimization.

Structured Chain-of-Thought. Recent advances in reasoning [15, 35] highlight that while Chain-of-thought (CoT) reasoning is powerful, unconstrained generation from standard RL methods [32, 39, 40] can produce incoherent or hallucinated reasoning traces [19]. To mitigate this, researchers have introduced

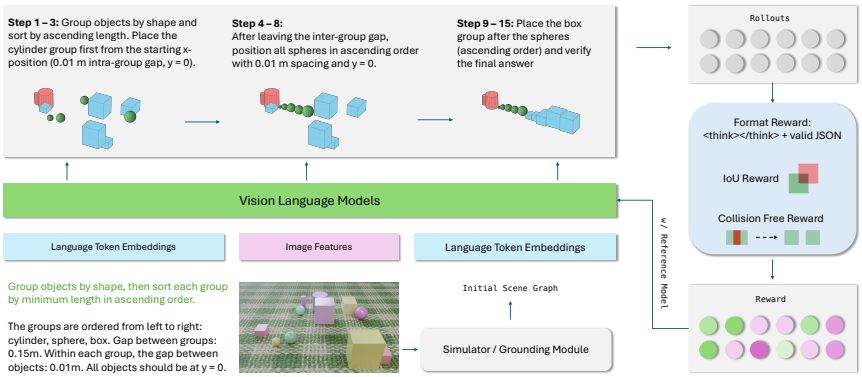

Fig. 3: Overview of our training pipeline. The vision-language model predicts step-by-step layout edits from the instruction and initial scene graph, and rollouts are optimized using a combination of format, IoU, and collision-free rewards.

structured constraints into the reasoning process, enhancing performance on tasks requiring complex, structured outputs [21, 22, 24]. Extending this to the multimodal domain, recent works improve model performance by leveraging explicit visual meta-information [2, 16, 17] or adopting specialized modules for complex spatio-temporal graphs [12]. However, directly integrating structured 3D information into the reasoning process of LLMs/VLMs is less explored. We posit that embedding our 3D bounding-box scene graph within the reasoning trace guides more faithful spatial logic, mitigates multimodal hallucination, and promotes more accurate, generalizable reasoning.

3 Method

3.1 Preliminaries

Scene graph representation. We represent each 3D layout as a directed scene graph with nodes corresponding to objects and supporting regions, and edges encoding contact or containment relations. In practice, each graph is serialized as a JSON dictionary keyed by integer node identifiers, where each node stores attributes such as `node_type`, `center_location` (a 3D position), `dimension` (axis-aligned length, width, height), `rotation` (roll, pitch, yaw), and an optional natural-language `caption`.

CoT-SFT and GRPO Training. Modern approaches to enhancing reasoning capabilities in large language models [15] typically follow a two-stage pipeline: CoT-SFT followed by RL optimization. During the RL stage, recent methods [32] leverage a policy optimization framework that enables the model to sample multiple candidate outputs and update its parameters via a clipped surrogate ob-

jective:

$$\mathcal{J}_{\mathrm{GRPO}}(\theta) = \mathbb{E}_{q\sim P(Q),\,\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\mathrm{old}}}(O|q)}\Bigg[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|} \sum_{t=1}^{|o_i|}\Bigg\{\min\Big(r_{i,t}\hat{A}_{i,t}, \mathrm{clip}\big(r_{i,t}, 1-\epsilon, 1+\epsilon\big)\hat{A}_{i,t}\Big) - \beta\,\mathbb{D}_{\mathrm{KL}}\big(\pi_\theta \,\|\, \pi_{\mathrm{ref}}\big)\Bigg\}\Bigg], \tag{1}$$

where $r_{i,t} = \pi_\theta(o_{i,t}|q, o_{i,<t})/\pi_{\theta_{old}}(o_{i,t}|q, o_{i,<t})$ denotes the likelihood ratio between the current and previous policies at step t. The hyperparameter ϵ denoting the controls the clipping range to stabilize updates, while β balances the main optimization objective against the KL divergence term $\mathbb{D}_{\mathrm{KL}}[\pi_\theta \,||\, \pi_{\mathrm{ref}}]$ which regularizes the learned policy toward a reference policy. The actual learning signal is derived from the advantage estimate, a normalized reward computed across multiple samples within the same query group:

$$\hat{A}_{i,t} = \frac{r_{i,t} - \mathrm{mean}\{r_{1,t}, \ldots, r_{N,t}\}}{\mathrm{std}\{r_{i,t}, \ldots, r_{N,t}\}}. \tag{2}$$

Where $r_{i,t}$ denotes the scalar reward assigned to the i-th generated sample

While CoT-SFT brings about the basic capability of generating long reasoning responses to the model, it does not directly guaranty superior final accuracy nor generalizability. The following RL stage tends to fail when the starting point model does not potentially master the underlying task since learning from low quality reasoning traces hardly helps further improvement. When the task is complicated, an exact match reward function will also be sparse for the model to receive meaningful training signal and hinder the converging. Unfortunately, current state of vision language models on real world 3D tasks suffers simultaneously from these complications.

In the following sections, we introduce the components used throughout our approach. First, we construct a data pipeline that produces controllable and interpretable reasoning trajectories, where text-based scene graphs serve as structured representations of spatial relations. Based on these trajectories, we apply CoT-SFT to expose the model to step-wise structured reasoning. We then describe a GRPO-based reinforcement learning stage that further refines generation quality.

3.2 Dataset Creation

Our training data consist of paired 3D layouts and language, organized as tuples $(I, G_0, G^\star, x, y)$ of rendered images I, an initial scene graph G_0 (see Paragraph 3.1), a target scene graph $G^\star$, an instruction x, and a reasoning trace y. The initial scene graph is obtained using a rule-based python auto-labeling framework leveraging the blender annotations. The reasoning trace provides a step-by-step explanation that connects the instruction and input layout to the desired final layout, and is later used for Structured CoT-SFT. We generate reasoning traces y using the DeepSeek-R1 model [15]. Given an instruction x,

an initial graph G_0, and a target graph $G^\star$, we prompt the model to produce a structured chain-of-thought "thinking path" that explicitly describes how to transform G_0 into $G^\star$. The generated trace maps textual constraints in the instruction—such as grouping rules, sorting criteria, relational statements, or distance specifications—to concrete object-level operations inside the scene graph.

Sorting task. The first and largest task is a synthetic 3D sorting benchmark of 10k scenes. Each instance contains a scene and a rule-heavy instruction that specifies how to group objects and arrange them along a chosen axis. The instructions combine multiple constraints, such as (1) grouping objects by semantic attributes (e.g., shape), (2) sorting objects within each group by a geometric attribute (e.g., height), (3) placing groups in a prescribed left-to-right order, and (4) enforcing global layout constraints like a fixed total span and fixed gaps between groups and between consecutive objects. All objects are finally snapped to a common support surface, with vertical positions determined by their height and the table plane.

Spatial alignment task. The second task is a more challenging rearrangement benchmark with 1k Blender-rendered scenes used for training. Here, the underlying target layout is a clean $N \times M$ grid derived from an initially well-organized scene graph, where each grid cell encodes the intended position and orientation of an object, grouped by category. We then randomly perturb a subset of objects by translating and rotating them away from their grid cells, yielding a corrupted initial graph G_0. The instruction asks the model to restore order by returning displaced objects to their appropriate grid locations and orientations, while leaving correctly placed objects unchanged.

Room editing task. We further construct a room-editing benchmark based on the instruction corpus from the InstructScene dataset [23]. Although InstructScene provides rich relational descriptions (e.g., "place the lamp next to the sofa"), such instructions generally do not specify a unique target configuration, as many scene layouts can simultaneously satisfy the same set of relations. Moreover, common scene-generation metrics are inherently ambiguous and fail to sufficiently constrain the solution space. To reduce this indeterminacy, we augment each object-insertion instruction with additional geometric constraints: for every object to be added, we provide its distances to two or three existing reference objects. These distance constraints significantly narrow the set of feasible placements and lead to an almost uniquely determined target graph $G^\star$, enabling more reliable supervision for both SFT and RL.

3.3 3D Layout Reinforcement Learning

We adopt a two-stage training pipeline. First, we perform CoT-SFT to provide the model with a strong cold start, ensuring it can already produce structured reasoning traces and roughly correct 3D layouts. Building upon this foundation, we then apply GRPO reinforcement learning to further refine layout accuracy, physical plausibility, and output formatting. This RL stage leverages

Table 1: Results on caption-free and noisy-input scene-graph grounding and editing.

Model	Task 1: Caption-Free		Task 2: Noisy-Input	
	IoU	Collision Free	IoU	Collision Free
Qwen3-VL-235B-Instruct	0.408	0.631	0.428	0.679
Qwen3-VL-235B-Thinking	0.472	0.744	0.378	0.701
Qwen3-8B (Ours, for reference)	0.209	**0.984**	0.664	0.939
Qwen2.5-VL-7B (Ans-SFT)	0.521	0.700	0.619	0.667
Qwen2.5-VL-7B (CoT-SFT)	0.641	0.829	0.773	0.844
Qwen2.5-VL-7B (Ours)	**0.787**	0.979	**0.822**	**0.980**

task-specific rewards and benefits substantially from the structured behaviors acquired during CoT-SFT.

We optimize a policy to synthesize a physically plausible 3D scene graph G_T while remaining faithful to a ground-truth graph $G^\star$. The objective is a weighted sum of three terms that jointly measure semantic alignment, internal physical feasibility, and output-format compliance:

$$r \;=\; \mathrm{IoU}(G_{\mathrm{pred}}, G^\star) \;+\; \lambda_1\, \mathrm{Coll}(G_{\mathrm{pred}}) \;+\; \lambda_2\, \mathrm{Fmt}(G_{\mathrm{pred}})$$

Here, the weights λ_{Coll} and λ_{Fmt} control the relative contribution of collision avoidance and formatting fidelity, respectively. The IoU term anchors the layout to ground-truth object geometry, the collision term suppresses physically implausible interpenetrations inside the prediction, and the format term preserves a standardized interface for downstream parsing and analysis. Jointly optimizing these signals encourages G_T to be both accurate and physically coherent while remaining reproducible and inspectable.

IoU reward. To evaluate cross-graph alignment we first establish correspondences between nodes in G_{pred} and $G^\star$. For every predicted node we attempt a semantic match using textual descriptions when available; if no unique semantic match is found, we select the ground-truth node that maximizes 3D intersection over union (IoU). Let b_i and b_j be the axis-aligned 3D boxes of a matched pair. The IoU is $\mathrm{IoU}(b_i, b_j) \;=\; V(b_i \cap b_j)/\left[V(b_i) + V(b_j) - V(b_i \cap b_j)\right]$, The intersection volume is computed by overlapping the one-dimensional extents on each axis. The IoU reward is the average over matched pairs,

$$r_{\mathrm{IoU}} \;=\; \frac{1}{|G_{\mathrm{pred}}|} \sum_{(i,j)\in\mathcal{M}} \mathrm{IoU}(b_i, b_j),$$

with $\mathcal{M}$ the set of description- or IoU-based matches.

Collision reward. To promote internal physical plausibility we penalize interpenetrations among non-container objects within G_T. For every unordered pair

of nodes (p, q) we compute the axis-aligned intersection volume $V_{pq} = V(b_p \cap b_q)$ using the same overlap operator. A pair is deemed colliding if $V_{pq} > \varepsilon$ for a small tolerance $\varepsilon > 0$. Let $\mathcal{C} = \{(p, q) : V_{pq} > \varepsilon\}$ and let N be the number of non-container objects. We define a normalized collision-free score $\text{Coll}(G) = 1 - |\mathcal{C}|/N$, which rises as collisions vanish and reaches one when all pairs are disjoint. This term mirrors the computation used in evaluation, including the exclusion of container nodes and the use of a nonzero tolerance to avoid numerical artifacts.

Format reward. To ensure outputs remain verifiable, we require a strict reasoning-and-answer pattern: `<think> structured reasoning that includes a JSON block </think> Final Scene Graph‘‘‘json...‘‘‘` The format reward assigns $\text{Fmt}(G_T) = 1$ when the response contains the exact tag pair `<think></think>`, at least one syntactically valid JSON block inside the `<think>` section, and a second well-formed JSON block in the final answer segment. Minor deviations such as unmatched braces or misplaced tags reduce the score continuously toward zero. This constraint stabilizes training by coupling layout predictions with auditable, structured reasoning rather than free-form prose.

4 Experiments

4.1 Experiment Setup

Baselines. We compare our method against two categories of state-of-the-art models. (1) **Open-Sourced Models:** We evaluate top-performing LLMs and VLMs, including the Qwen3-235B and Qwen3-VL-235B with Instruct and Thinking versions [35], and the DeepSeek series (Deepseek-V3 [25], DeepSeek-R1, and DeepSeek-R1-0528 [15]). (2) **Proprietary Models:** We evaluate the performance of Gemini 2.5 Pro and Flash [9] as strong, closed-source baselines.

Evaluation Metrics. We evaluate the final generated scene graph using a suite of metrics. For geometric accuracy, we compute: **Mean IoU**, the average 3D IoU between predicted and ground-truth boxes with semantic matching; **IoU@**x, the percentage of nodes achieving at least x IoU (e.g., 0.50); and **Center Distance (Ctr. Dist.)**, the average Euclidean distance between the centroids of predicted bounding boxes and ground-truths. For physical plausibility, we report the **Collision-Free Score (Col. Free)**, which is the normalized score $\text{Coll}(G)$ defined in Section 3.3. Finally, for the sorting task, we also evaluate the **Edit Distance (Edit Dist.)**, calculated as the average Levenshtein edit distance required to transform the predicted object orders into ground-truths.

Post-training models. For training experiments, we instantiate our framework on several open-source 7B/8B backbones, including Qwen2.5-7B [29], Qwen2.5-VL-7B [4], Qwen3-8B [35], and Cosmos-Reason1-7B [3]. For each backbone, we report Ans-SFT, CoT-SFT, Vanilla GRPO, and our structured reasoning variant.

Table 2: Quantitative results on the 3D Sorting benchmark. Models equipped with our structured scene-graph reasoning ("Ours") consistently outperform both zero-shot LLM/VLM baselines and standard CoT-SFT fine-tuning.

Model	Mean IoU ↑	Ctr. Dist. ↓	Col. Free ↑	Edit Dist. ↓
Large Language Models (Zero-shot)				
Qwen3-235B-Instruct	0.122	2.702	0.548	6.285
Qwen3-235B-Thinking	0.708	0.856	0.957	2.208
Qwen3-VL-235B-Instruct	0.145	2.323	0.421	8.120
Qwen3-VL-235B-Thinking	0.706	0.568	0.918	1.900
Deepseek-V3	0.236	1.242	0.620	2.010
Deepseek-R1	0.776	0.566	0.903	0.783
Deepseek-R1-0528	0.601	1.163	0.799	2.021
Gemini 2.5 Flash	0.746	**0.458**	0.968	1.840
Gemini 2.5 Pro	0.724	0.593	0.976	2.180
LayoutGPT (Qwen3-235B)	0.763	0.518	0.971	0.910
LayoutGPT (Gemini-2.5)	**0.781**	0.471	**0.982**	**0.709**
Instruct Models				
Qwen2.5-VL-7B (Ans-SFT)	0.570	0.423	0.698	0.900
Qwen2.5-VL-7B (CoT-SFT)	0.713	0.330	0.847	0.853
Qwen2.5-VL-7B (Ours)	0.798	0.228	0.961	**0.613**
Qwen2.5-7B (Ans-SFT)	0.622	0.348	0.745	1.175
Qwen2.5-7B (CoT-SFT)	0.712	0.345	0.898	0.992
Qwen2.5-7B (Ours)	**0.879**	**0.179**	**1.000**	0.627
Reasoning Models				
Cosmos-7B (Ans-SFT)	0.501	0.577	0.683	1.794
Cosmos-7B (CoT-SFT)	0.641	0.410	0.757	1.383
Cosmos-7B (Vanilla GRPO)	0.729	0.297	0.937	0.811
Cosmos-7B (Ours)	0.875	0.234	1.000	0.592
Qwen3-8B (Ans-SFT)	0.552	0.493	0.700	1.130
Qwen3-8B (CoT-SFT)	0.635	0.513	0.854	1.610
Qwen3-8B (Vanilla GRPO)	0.850	0.274	0.954	0.684
Qwen3-8B (Ours)	**0.924**	**0.145**	**1.000**	**0.510**

4.2 Sorting

In real-world or simulators, object captions may be missing, and detected bounding boxes can be inaccurate, making grounding from the scene graph ambiguous. We therefore evaluate two settings in Table 1: (1) Caption-Free, where node captions are removed and the model must infer object semantics and relations from geometry (and visual cues when available) to recover descriptions and output the updated graph; and (2) Noisy-Input, where we perturb the input by adding 5% jitter to object bounding boxes and enforcing partial visibility, testing robustness to imperfect localization and occlusion. As shown in Table 1, our method is effective with a VLM and yields strong gains under both settings.

To further reveal the capability of our structured reasoning, we additionally evaluate the sorting task under a perfect-input setting, as shown in Table 2.

Table 3: Quantitative results on the spatial alignment benchmark, evaluating grid-structure recovery and object repositioning accuracy.

Model	Mean IoU ↑	IoU@0.5 ↑	Ctr. Dist. ↓
Qwen3-235B-Instruct	0.592	0.587	0.243
Qwen3-235B-Thinking	0.445	0.423	0.320
Qwen3-VL-235B-Instruct	**0.786**	**0.787**	0.177
Qwen3-VL-235B-Thinking	0.623	0.624	0.189
Deepseek-V3	0.553	0.559	0.278
Deepseek-R1	0.602	0.607	0.184
Deepseek-R1-0528	0.599	0.606	0.252
Gemini 2.5 Flash	0.613	0.611	0.231
Gemini 2.5 Pro	0.759	0.772	**0.156**
Qwen2.5-VL-7B (Ans-SFT)	0.728	0.725	0.201
Qwen2.5-VL-7B (CoT-SFT)	0.785	0.779	0.193
Qwen2.5-VL-7B (Ours)	0.798	0.792	0.181
Qwen2.5-7B (Ans-SFT)	0.733	0.729	0.201
Qwen2.5-7B (CoT-SFT)	0.764	0.758	0.189
Qwen2.5-7B (Ours)	**0.809**	**0.803**	**0.176**
Qwen3-8B (Ans-SFT)	0.649	0.636	0.248
Qwen3-8B (CoT-SFT)	0.707	0.700	0.227
Qwen3-8B (Ours)	0.790	0.784	0.190

Direct SFT to generate the final scene graph consistently underperforms, as the task requires multi-step decomposition, intermediate-state perception, and recursive update—capabilities that the base model cannot acquire through SFT alone. In contrast, incorporating structured CoT-SFT enables the model to reliably construct intermediate states and reach the correct final configuration. Adding RL on top of CoT-SFT further improves performance, indicating that synthetic traces alone do not cover the harder regions of the task distribution and that RL provides additional exploration and reward-driven refinement for further improvements. To isolate the contribution of structured content, we additionally compare Vanilla GRPO training from a Qwen3-8B base model. Skipping the structured CoT-SFT stage results in a 7.4% drop in mIoU gain, confirming that scene-graph-based trajectories offer crucial guidance.

Discussion. In the perfect-input setting, the problem largely collapses to text-based structured reasoning over a clean scene graph, so adding a VLM backbone brings limited extra benefit beyond what the LLM already provides. In contrast, the core advantage of VLMs is stronger visual/geometry grounding under missing captions and imperfect perception; when captions are removed or boxes are noisy/partially visible, LLM-only models degrade much more, while VLM-based grounding remains more robust.

Table 4: Quantitative results on the RoomEditing benchmark, evaluating instruction-guided object placement under relational and distance constraints.

Model	Mean IoU ↑	IoU@0.5 ↑	Ctr. Dist. ↓
Qwen3-VL-235B-Instruct	0.502	0.497	0.719
Qwen3-VL-235B-Thinking	0.496	0.487	0.684
Deepseek-V3	0.473	0.469	0.859
Deepseek-R1	0.454	0.449	0.754
Deepseek-R1-0528	0.471	0.463	0.871
Gemini 2.5 Flash	0.443	0.441	0.924
Gemini 2.5 Pro	**0.591**	**0.587**	**0.621**
Qwen2.5-7B (Ans-SFT)	0.869	0.863	0.138
Qwen2.5-7B (CoT-SFT)	0.950	0.950	0.051
Qwen2.5-7B (Ours)	0.974	0.973	0.033
Qwen3-8B (Ans-SFT)	0.895	0.897	0.117
Qwen3-8B (CoT-SFT)	0.972	0.972	**0.026**
Qwen3-8B (Vanilla GRPO)	0.768	0.774	0.336
Qwen3-8B (Ours)	**0.983**	**0.983**	0.031

4.3 Spatial Alignment

The spatial alignment task poses a slightly higher difficulty than the sorting task, as it requires understanding the underlying $N \times M$ group structure, identifying inconsistencies relative to a group anchor, and locating the misaligned instance based on center-position deviations. This makes the benchmark challenging even for SOTA models, as shown in Table 3.

From our post training experiments, answer-SFT models that directly predict the final coordinates fall noticeably short. In contrast, our structured layout reasoning can further improve the performance of smaller 7B/8B models, enabling them to match or even exceed the accuracy of these large commercial models. Strategically, our 3D-Layout-R1 learns to (1) infer the underlying grid pattern, (2) detect misaligned objects via center-location comparisons, and (3) recover the correct target position by either extending the end node or interpolating along the relevant grid line.

An additional observation is that large VLMs consistently outperform their LLM counterparts by over 10% on this task. However, our own post-training experiments on VLM backbones do not show comparable gains, suggesting that specialized visual prompting or tailored visual representations may be necessary to fully exploit VLM capabilities. We provide further discussion and ablations in the supplementary material.

4.4 Room Layout Editing

For the layout editing task, we conduct a parallel set of experiments and observe consistent trends. As shown in Table 4, applying RL on top of structured CoT-SFT again produces the strongest overall performance. While the gap between

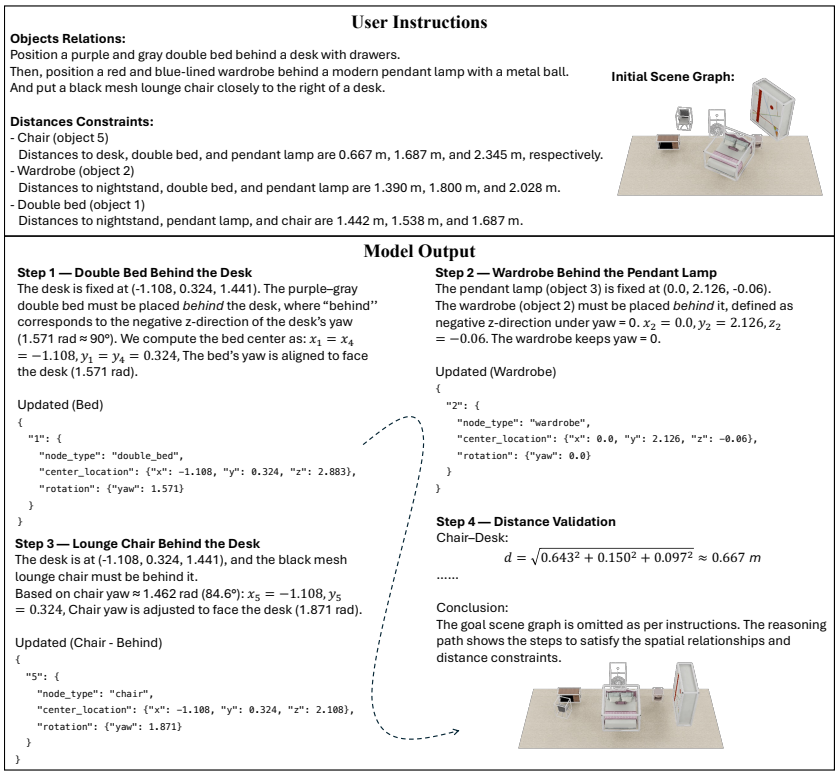

Fig. 4: Example of text-guided 3D room layout reasoning, showing how the model interprets constraints to update object poses step by step and validate distances.

direct answer SFT and methods incorporating structured reasoning persists, a notable difference emerges: for this task, the performance gap between RL and its corresponding structured CoT-SFT baseline is significantly smaller than in the sorting task. This suggests that CoT-SFT already captures most of the structured patterns required for layout editing, leaving less headroom for RL to further optimize. We attribute this to the relative simplicity of the task, each editing example modifies only a small portion of the scene, and the model primarily needs to learn how to move the specified objects to achieve a high IoU.

4.5 Qualitative Results

Structured 3D sorting traces. Fig. 2 shows an results from the synthetic sorting benchmark. Given a cluttered scene and a rule-heavy instruction, our model generates a trace that (1) groups and sorts obejcts by attributes, (2) allocates group spans along the target axis, and (3) places each object with the required gaps. Each step is accompanied by a compact scene-graph update, producing layouts that match the target.

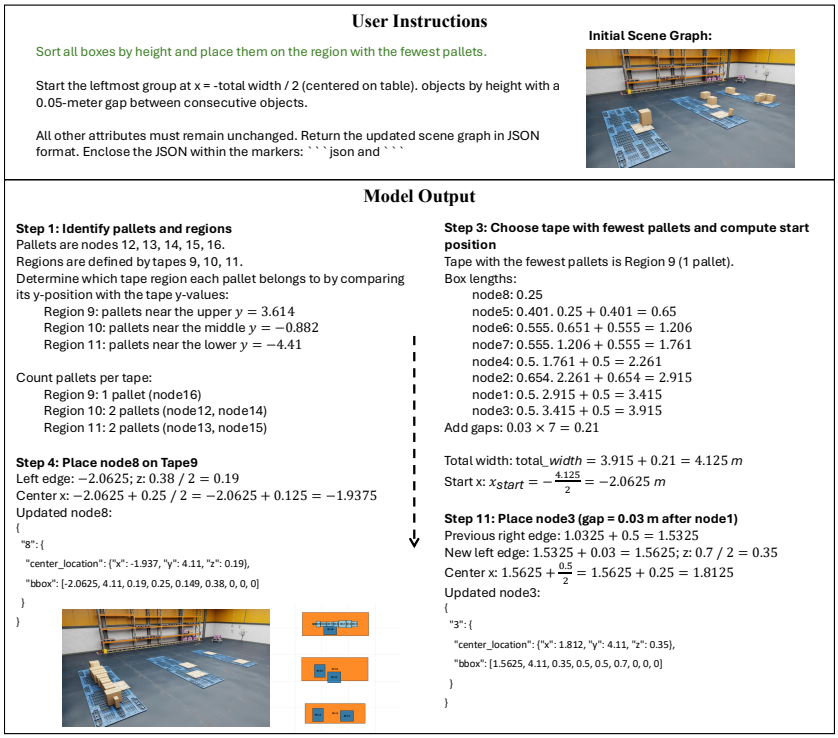

Fig. 5: Out-of-domain warehouse simulation results showing our model correctly following user instructions with 3D boxes.

Room-editing reasoning. Fig. 4 presents a room-editing example from the InstructScene-based task. Starting from an initial scene graph and relational instructions with metric constraints, the model incrementally updates object poses in JSON form. The trace explicitly verifies the Euclidean distances used in the constraints, and the resulting rendered layout is both visually plausible and quantitatively consistent with the ground-truth scene graph.

Warehouse. We additionally show out-of-domain qualitative results in a warehouse simulation. In this setting, the simulator provides perfect 3D bounding boxes, and the user specifies practical rearrangement rules. As shown in Fig. 5, without any retraining on warehouse data, our model follows these unseen instructions and produces correct, physically consistent scene-graph updates. To further evaluate flexibility, we include an additional multi-turn example in Fig. 6. The first turn asks the model to sort boxes in the left region and move pallets to the middle region, producing an intermediate layout. Starting from this generated layout, the second turn issues an under-specified instruction to select the tallest box and place it onto any pallet. Our model selects a feasible placement while preserving the existing layout structure. This demonstrates that the model

Fig. 6: Flexible multi-turn warehouse layout editing.

can handle sequential edits and flexible, multi-solution instructions beyond fixed one-shot rearrangement.

Real-world Robot. We demonstrate our method in Fig. 7 on two real-world table-top tasks: (i) a spatial alignment task, and (ii) a pick-and-place task that places the yellow cup in the yellow bowl. For both tasks, we use Qwen3-VL to extract object bounding boxes from the input image and then apply our VLM-based scene editing to synthesize the goal state. Given known grasping skills for the detected objects, a rule-based and goal-conditioned robot can execute the task. We do not explicitly model continuous dynamics or collisions inside the model; instead, our focus is high-level goal generation, which is complementary to existing motion planning and control.

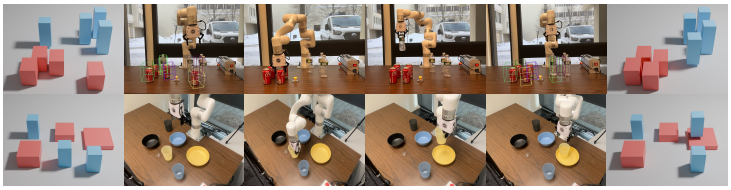

Fig. 7: Real-world results, including spatial rearrangement and placing execution.

5 Conclusion

We presented 3D-Layout-R1, a structured reasoning framework that improves the accuracy and interpretability of language-guided 3D layout editing. By injecting layout structure into multi-step reasoning and refining outputs with GRPO, our approach produces spatially consistent, physically coherent scene edits across a wide range of layout manipulation tasks. This work demonstrates the value of explicit graph-based reasoning for robust and controllable 3D understanding. Across all benchmarks, 3D-Layout-R1 significantly reduces layout error and consistently outperforms zero-shot and CoT-based baselines, particularly in demanding spatial alignment and room-editing settings. These results highlight the framework's strong generalization and its effectiveness in translating natural language instructions into precise 3D transformations.

References

1. Abdelreheem, A., Aleotti, F., Watson, J., Qureshi, Z., Eldesokey, A., Wonka, P., Brostow, G., Vicente, S., Garcia-Hernando, G.: Placeit3d: Language-guided object placement in real 3d scenes. arXiv preprint arXiv:2505.05288 (2025)
2. Arnab, A., Iscen, A., Caron, M., Fathi, A., Schmid, C.: Temporal chain of thought: Long-video understanding by thinking in frames. CoRR **abs/2507.02001** (2025). `https://doi.org/10.48550/ARXIV.2507.02001`, `https://doi.org/10.48550/arXiv.2507.02001`, accessed: 2026-06-29
3. Azzolini, A., Bai, J., Brandon, H., Cao, J., Chattopadhyay, P., Chen, H., Chu, J., Cui, Y., Diamond, J., Ding, Y., et al.: Cosmos-reason1: From physical common sense to embodied reasoning. arXiv preprint arXiv:2503.15558 (2025)
4. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., Zhong, H., Zhu, Y., Yang, M., Li, Z., Wan, J., Wang, P., Ding, W., Fu, Z., Xu, Y., Ye, J., Zhang, X., Xie, T., Cheng, Z., Zhang, H., Yang, Z., Xu, H., Lin, J.: Qwen2.5-vl technical report (2025), `https://arxiv.org/abs/2502.13923`, accessed: 2026-06-29
5. Cai, W., Ponomarenko, I., Yuan, J., Li, X., Yang, W., Dong, H., Zhao, B.: Spatialbot: Precise spatial understanding with vision language models. In: 2025 IEEE International Conference on Robotics and Automation (ICRA). pp. 9490–9498. IEEE (2025)
6. Chen, B., Xu, Z., Kirmani, S., Ichter, B., Sadigh, D., Guibas, L., Xia, F.: Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14455–14465 (2024)
7. Cheng, A.C., Fu, Y., Chen, Y., Liu, Z., Li, X., Radhakrishnan, S., Han, S., Lu, Y., Kautz, J., Molchanov, P., Yin, H., Wang, X., Liu, S.: 3d aware region prompted vision language model. arXiv preprint arXiv:2509.13317 (2025)
8. Cheng, A.C., Yin, H., Fu, Y., Guo, Q., Yang, R., Kautz, J., Wang, X., Liu, S.: Spatialrgpt: Grounded spatial reasoning in vision-language models. Advances in Neural Information Processing Systems **37**, 135062–135093 (2024)
9. Comanici, G., Bieber, E., Schaekermann, M., Pasupat, I., Sachdeva, N., Dhillon, I., Blistein, M., Ram, O., Zhang, D., Rosen, E., et al.: Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. arXiv preprint arXiv:2507.06261 (2025)
10. Daxberger, E., Wenzel, N., Griffiths, D., Gang, H., Lazarow, J., Kohavi, G., Kang, K., Eichner, M., Yang, Y., Dehghan, A., et al.: Mm-spatial: Exploring 3d spatial understanding in multimodal llms. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7395–7408 (2025)
11. El Amine Boudjoghra, M., Laptev, I., Dai, A.: Scanedit: Hierarchically-guided functional 3d scan editing. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 27105–27115 (2025)
12. Fei, H., Wu, S., Ji, W., Zhang, H., Zhang, M., Lee, M.L., Hsu, W.: Video-of-thought: step-by-step video reasoning from perception to cognition. In: Proceedings of the 41st International Conference on Machine Learning. ICML'24, JMLR.org (2024)
13. Feng, W., Zhu, W., Fu, T.j., Jampani, V., Akula, A., He, X., Basu, S., Wang, X.E., Wang, W.Y.: Layoutgpt: Compositional visual planning and generation with large language models. Advances in Neural Information Processing Systems **36**, 18225–18250 (2023)

14. Gu, Y., Huang, I., Je, J., Yang, G., Guibas, L.: Blendergym: Benchmarking foundational model systems for graphics editing. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 18574–18583 (2025)
15. Guo, D., Yang, D., Zhang, H., Song, J., Wang, P., Zhu, Q., Xu, R., Zhang, R., Ma, S., Bi, X., Zhang, X., Yu, X., Wu, Y., Wu, Z.F., Gou, Z., Shao, Z., Li, Z., Gao, Z., Liu, A., Xue, B., Wang, B., Wu, B., Feng, B., Lu, C., Zhao, C., Deng, C., Ruan, C., Dai, D., Chen, D., Ji, D., Li, E., Lin, F., Dai, F., Luo, F., Hao, G., Chen, G., Li, G., Zhang, H., Xu, H., Ding, H., Gao, H., Qu, H., Li, H., Guo, J., Li, J., Chen, J., Yuan, J., Tu, J., Qiu, J., Li, J., Cai, J.L., Ni, J., Liang, J., Chen, J., Dong, K., Hu, K., You, K., Gao, K., Guan, K., Huang, K., Yu, K., Wang, L., Zhang, L., Zhao, L., Wang, L., Zhang, L., Xu, L., Xia, L., Zhang, M., Zhang, M., Tang, M., Zhou, M., Li, M., Wang, M., Li, M., Tian, N., Huang, P., Zhang, P., Wang, Q., Chen, Q., Du, Q., Ge, R., Zhang, R., Pan, R., Wang, R., Chen, R.J., Jin, R.L., Chen, R., Lu, S., Zhou, S., Chen, S., Ye, S., Wang, S., Yu, S., Zhou, S., Pan, S., Li, S.S., Zhou, S., Wu, S., Yun, T., Pei, T., Sun, T., Wang, T., Zeng, W., Liu, W., Liang, W., Gao, W., Yu, W., Zhang, W., Xiao, W.L., An, W., Liu, X., Wang, X., Chen, X., Nie, X., Cheng, X., Liu, X., Xie, X., Liu, X., Yang, X., Li, X., Su, X., Lin, X., Li, X.Q., Jin, X., Shen, X., Chen, X., Sun, X., Wang, X., Song, X., Zhou, X., Wang, X., Shan, X., Li, Y.K., Wang, Y.Q., Wei, Y.X., Zhang, Y., Xu, Y., Li, Y., Zhao, Y., Sun, Y., Wang, Y., Yu, Y., Zhang, Y., Shi, Y., Xiong, Y., He, Y., Piao, Y., Wang, Y., Tan, Y., Ma, Y., Liu, Y., Guo, Y., Ou, Y., Wang, Y., Gong, Y., Zou, Y., He, Y., Xiong, Y., Luo, Y., You, Y., Liu, Y., Zhou, Y., Zhu, Y.X., Huang, Y., Li, Y., Zheng, Y., Zhu, Y., Ma, Y., Tang, Y., Zha, Y., Yan, Y., Ren, Z.Z., Ren, Z., Sha, Z., Fu, Z., Xu, Z., Xie, Z., Zhang, Z., Hao, Z., Ma, Z., Yan, Z., Wu, Z., Gu, Z., Zhu, Z., Liu, Z., Li, Z., Xie, Z., Song, Z., Pan, Z., Huang, Z., Xu, Z., Zhang, Z., Zhang, Z.: Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. Nature **645**(8081), 633–638 (2025). https://doi.org/10.1038/s41586-025-09422-z, accessed: 2026-06-29
16. Guo, G., Zhang, K., Hoo, B., Cai, Y., Lu, X., Peng, N., Wang, Y.: Structured outputs enable general-purpose llms to be medical experts (2025), https://arxiv.org/abs/2503.03194, accessed: 2026-06-29
17. Huang, D.A., Radhakrishnan, S., Yu, Z., Kautz, J.: Frag: Frame selection augmented generation for long video and long document understanding. arXiv preprint arXiv:2504.17447 (2025)
18. Huang, I., Bao, Y., Truong, K., Zhou, H., Schmid, C., Guibas, L., Fathi, A.: Fireplace: Geometric refinements of llm common sense reasoning for 3d object placement. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 13466–13476 (2025)
19. Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., et al.: A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. ACM Transactions on Information Systems **43**(2), 1–55 (2025)
20. Huang, T., Zhang, Z., Tang, H.: 3d-r1: Enhancing reasoning in 3d vlms for unified scene understanding. arXiv preprint arXiv:2507.23478 (2025)
21. Li, C., Xu, T., Guo, Y.: Reasoning-as-logic-units: Scaling test-time reasoning in large language models through logic unit alignment. In: International Conference on Machine Learning, ICML 2025, 13-19 July 2025, Vancouver, Canada. Proceedings of Machine Learning Research, PMLR (2025), https://www.arxiv.org/abs/2502.07803, accessed: 2026-06-29

22. Li, J., Li, G., Li, Y., Jin, Z.: Structured chain-of-thought prompting for code generation. ACM Trans. Softw. Eng. Methodol. **34**(2) (Jan 2025). https://doi.org/10.1145/3690635, https://doi.org/10.1145/3690635, accessed: 2026-06-29
23. Lin, C., Mu, Y.: Instructscene: Instruction-driven 3d indoor scene synthesis with semantic graph prior. arXiv preprint arXiv:2402.04717 (2024)
24. Ling, Z., Fang, Y., Li, X., Huang, Z., Lee, M., Memisevic, R., Su, H.: Deductive verification of chain-of-thought reasoning. In: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) Advances in Neural Information Processing Systems. vol. 36, pp. 36407–36433. Curran Associates, Inc. (2023), https://proceedings.neurips.cc/paper_files/paper/2023/file/72393bd47a35f5b3bee4c609e7bba733-Paper-Conference.pdf, accessed: 2026-06-29
25. Liu, A., Feng, B., Xue, B., Wang, B., Wu, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., et al.: Deepseek-v3 technical report. arXiv preprint arXiv:2412.19437 (2024)
26. Liu, Y., Chi, D., Wu, S., Zhang, Z., Hu, Y., Zhang, L., Zhang, Y., Wu, S., Cao, T., Huang, G., et al.: Spatialcot: Advancing spatial reasoning through coordinate alignment and chain-of-thought for embodied task planning. arXiv preprint arXiv:2501.10074 (2025)
27. Ma, W., Chou, Y.C., Liu, Q., Wang, X., de Melo, C., Xie, J., Yuille, A.: Spatialreasoner: Towards explicit and generalizable 3d spatial reasoning. arXiv preprint arXiv:2504.20024 (2025). https://doi.org/10.48550/arXiv.2504.20024, accessed: 2026-06-29
28. Ma, W., Ye, L., de Melo, C.M., Yuille, A., Chen, J.: Spatialllm: A compound 3d-informed design towards spatially-intelligent large multimodal models. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 17249–17260 (2025)
29. Qwen, :, Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., Lin, H., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Lin, J., Dang, K., Lu, K., Bao, K., Yang, K., Yu, L., Li, M., Xue, M., Zhang, P., Zhu, Q., Men, R., Lin, R., Li, T., Tang, T., Xia, T., Ren, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Wan, Y., Liu, Y., Cui, Z., Zhang, Z., Qiu, Z.: Qwen2.5 technical report (2025), https://arxiv.org/abs/2412.15115, accessed: 2026-06-29
30. Ran, X., Li, Y., Xu, L., Yu, M., Dai, B.: Direct numerical layout generation for 3d indoor scene synthesis via spatial reasoning. arXiv preprint arXiv:2506.05341 (2025)
31. Sarch, G., Saha, S., Khandelwal, N., Jain, A., Tarr, M.J., Kumar, A., Fragkiadaki, K.: Grounded reinforcement learning for visual reasoning. arXiv preprint arXiv:2505.23678 (2025)
32. Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Zhang, M., Li, Y., Wu, Y., Guo, D.: Deepseekmath: Pushing the limits of mathematical reasoning in open language models (2024), https://arxiv.org/abs/2402.03300, accessed: 2026-06-29
33. Song, C.H., Blukis, V., Tremblay, J., Tyree, S., Su, Y., Birchfield, S.: Robospatial: Teaching spatial understanding to 2d and 3d vision-language models for robotics. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 15768–15780 (2025), accessed: 2026-06-29
34. Sun, F.Y., Liu, W., Gu, S., Lim, D., Bhat, G., Tombari, F., Li, M., Haber, N., Wu, J.: Layoutvlm: Differentiable optimization of 3d layout via vision-language models. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 29469–29478 (2025)

35. Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., Zheng, C., Liu, D., Zhou, F., Huang, F., Hu, F., Ge, H., Wei, H., Lin, H., Tang, J., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Zhou, J., Lin, J., Dang, K., Bao, K., Yang, K., Yu, L., Deng, L., Li, M., Xue, M., Li, M., Zhang, P., Wang, P., Zhu, Q., Men, R., Gao, R., Liu, S., Luo, S., Li, T., Tang, T., Yin, W., Ren, X., Wang, X., Zhang, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Zhang, Y., Wan, Y., Liu, Y., Wang, Z., Cui, Z., Zhang, Z., Zhou, Z., Qiu, Z.: Qwen3 technical report. arXiv preprint arXiv:2505.09388 (2025)
36. Yang, J., Yang, S., Gupta, A.W., Han, R., Fei-Fei, L., Xie, S.: Thinking in space: How multimodal large language models see, remember, and recall spaces. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 10632–10643 (2025)
37. Yang, Y., Luo, Z., Ding, T., Lu, J., Gao, M., Yang, J., Sanchez, V., Zheng, F.: Optiscene: Llm-driven indoor scene layout generation via scaled human-aligned data synthesis and multi-stage preference optimization. In: The Thirty-ninth Annual Conference on Neural Information Processing Systems (2025)
38. Yang, Y., Sun, F.Y., Weihs, L., VanderBilt, E., Herrasti, A., Han, W., Wu, J., Haber, N., Krishna, R., Liu, L., et al.: Holodeck: Language guided generation of 3d embodied ai environments. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16227–16237 (2024)
39. Yu, Q., Zhang, Z., Zhu, R., Yuan, Y., Zuo, X., Yue, Y., Dai, W., Fan, T., Liu, G., Liu, L., et al.: Dapo: An open-source llm reinforcement learning system at scale. arXiv preprint arXiv:2503.14476 (2025)
40. Zheng, C., Liu, S., Li, M., Chen, X.H., Yu, B., Gao, C., Dang, K., Liu, Y., Men, R., Yang, A., Zhou, J., Lin, J.: Group sequence policy optimization. arXiv preprint arXiv:2507.18071 (2025)
41. Zheng, D., Huang, S., Wang, L.: Video-3d llm: Learning position-aware video representation for 3d scene understanding. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 8995–9006 (2025)
42. Zheng, K., Chen, X., He, X., Gu, J., Li, L., Yang, Z., Lin, K., Wang, J., Wang, L., Wang, X.E.: Editroom: Llm-parameterized graph diffusion for composable 3d room layout editing. arXiv preprint arXiv:2410.12836 (2024)
43. Zhou, E., An, J., Chi, C., Han, Y., Rong, S., Zhang, C., Wang, P., Wang, Z., Huang, T., Sheng, L., et al.: Roborefer: Towards spatial referring with reasoning in vision-language models for robotics. arXiv preprint arXiv:2506.04308 (2025)
44. Zhou, J., Li, X., Qi, L., Yang, M.H.: Layout-your-3d: Controllable and precise 3d generation with 2d blueprint. arXiv preprint arXiv:2410.15391 (2024)
45. Zhu, C., Wang, T., Zhang, W., Pang, J., Liu, X.: Llava-3d: A simple yet effective pathway to empowering lmms with 3d capabilities. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4295–4305 (2025)