PhyBlock: A Progressive Benchmark for Physical Understanding and Planning via 3D Block Assembly

Liang Ma[1*], **Jiajun Wen**[2,4*], **Min Lin**[2*], **Rongtao Xu**[1*], **Xiwen Liang**[2*], **Bingqian Lin**[3], **Jun Ma**[2], **Yongxin Wang**[1], **Ziming Wei**[2], **Haokun Lin**[1], **Mingfei Han**[1], **Meng Cao**[1], **Bokui Chen**[4†], **Ivan Laptev**[1], **Xiaodan Liang**[1,2†]

[1]Mohamed bin Zayed University of Artificial Intelligence
[2]Sun Yat-sen University [3]Shanghai Jiao Tong University
[4]Tsinghua Shenzhen International Graduate School, Tsinghua University, China
[*]Authors contributed equally to this research. [†]Corresponding author.
https://phyblock.github.io/

Abstract

While vision-language models (VLMs) have demonstrated promising capabilities in reasoning and planning for embodied agents, their ability to comprehend physical phenomena, particularly within structured 3D environments, remains severely limited. To close this gap, we introduce PhyBlock, a progressive benchmark designed to assess VLMs on physical understanding and planning through robotic 3D block assembly tasks. PhyBlock integrates a novel four-level cognitive hierarchy assembly task alongside targeted Visual Question Answering (VQA) samples, collectively aimed at evaluating progressive spatial reasoning and fundamental physical comprehension, including object properties, spatial relationships, and holistic scene understanding. PhyBlock includes 2600 block tasks (400 assembly tasks, 2200 VQA tasks) and evaluates models across three key dimensions: partial completion, failure diagnosis, and planning robustness. We benchmark 23 state-of-the-art VLMs, highlighting their strengths and limitations in physically grounded, multi-step planning. Our empirical findings indicate that the performance of VLMs exhibits pronounced limitations in high-level planning and reasoning capabilities, leading to a notable decline in performance for the growing complexity of the tasks. Error analysis reveals persistent difficulties in spatial orientation and dependency reasoning. We position PhyBlock as a unified testbed to advance embodied reasoning, bridging vision-language understanding and real-world physical problem-solving.

1 Introduction

Understanding physical interactions and spatial relationships is crucial for embodied agents tasked with manipulating and navigating complex real-world environments. Recent Vision–Language Models (VLMs), such as GPT-4o [42], Claude-3.7 [5], and Gemini 2.0 [21], have made impressive strides in multimodal reasoning, yet their grasp of physical-world characteristics—such as object stability, spatial support, and realistic multi-step planning—remains limited. As illustrated in Figure 1, 3D block assembly tasks serve as an intuitive testbed for these capabilities, encapsulating fundamental physical concepts like gravity (e.g., stability of constructed blocks), structural dependencies (e.g., correct block structure should be determined based on the desired target image), and geometric constraints. Accurately evaluating whether VLMs internalize such physical priors is critical, especially when they serve as high-level planners in hierarchical agent systems (e.g., System 2 in GR00T-N1 [8] and Helix [3]). These systems rely on physical awareness to generate actionable plans for low-level controllers (System 1), bridging abstract reasoning with real-world execution.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Track on Datasets and Benchmarks.

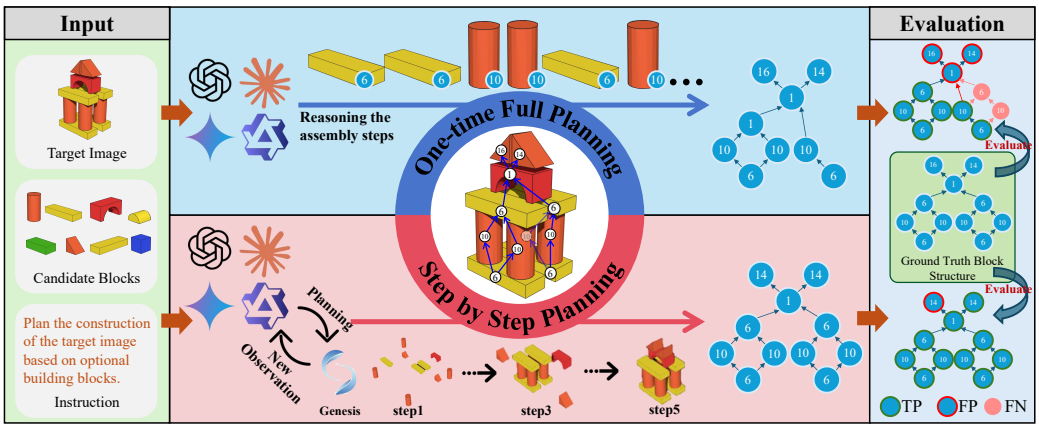


Figure 1: Assembly Planning Task in **PhyBlock**. Here shows inference setting of two planning strategies(one-time full planning and step-by-step planning).

Existing benchmarks [27, 36, 45, 22, 57] still suffer from two critical limitations[54]: 1) Perception dominance without planning capability. Current frameworks primarily emphasize perceptual understanding while neglecting long-horizon planning, resulting in models that excel at single-step reasoning but demonstrate inadequate inference capacity in complex scenarios; 2) Unrealistic physical assumptions. The prevailing assumption of objects existing in idealized states while ignoring real-world physical interactions significantly undermines their practical applicability. Consequently, we lack a rigorous yardstick that couples high-level language reasoning with the dynamic constraints of the physical world, leaving open whether modern VLMs truly understand how objects interact in three dimensions.

To benchmark physical understanding and planning capability, we adopt interactive 3D blocks, as they intuitively embody fundamental physical concepts, such as stability, support, and spatial relationships, in a clear and interpretable manner. Leveraging a physics-based simulator, we construct realistic 3D scenes that dynamically respond to interactions, enabling systematic evaluations of increasingly complex, multi-step tasks.

Building on this insight, we present **PhyBlock**, a comprehensive two-branch benchmark explicitly designed to assess the physical reasoning capabilities of modern VLMs. The first branch, **Hierarchical Assembly Planning** (shown in Figure 1), evaluates model's capacity to plan and reason about spatial arrangements through step-by-step interactions in a physics-aware simulator. This planning branch features 400 systematically constructed scenes across four ascending difficulty tiers (Basic, Simple Combinations, Complex Structures, and Advanced Spatial Planning), culminating in assemblies that involve up to 22 distinct blocks. The second branch, **Physical-Understanding VQA** (shown in Figure 2), measures model's explicit understanding of physical concepts. The VQA branch comprises 2,200 rigorously curated questions spanning 16 semantic categories including object attributes, relational reasoning, scene dynamics, and counterfactual inference.

Drawing inspiration from cognitive-development research, particularly the observation that structured block play enables children to internalize complex spatial and physical principles, we model eight LEGO-like block geometries in five distinct colors within the Genesis physics simulator, ensuring uniform and physically plausible interactions. This design not only captures key real-world regularities but also leaves headroom for future extensions that integrate low-level motor actions and control policies, thereby bringing the benchmark even closer to embodied deployment scenarios. To further guarantee dataset quality and rigor in evaluation, we encode essential dependencies and spatial relationships between blocks with an Activity-on-Vertex (AOV) graphs (detailed in Section 3.2), and construct manually verified Visual Question Answering (VQA) tasks through a robust, multi-stage process combining automated generation and rigorous human validation (detailed in Section 3.3). This careful design supports clear diagnostics, precise scoring, and reproducible analysis.

Building on PhyBlock, we conduct a comprehensive evaluation of 23 state-of-the-art open-source and closed-source vision-language models (VLMs) [1, 40, 6, 55, 29, 34], covering diverse architectures and scales. Empirical results uncover three consistent trends. First, a steep *performance cliff*: mean planning F_1 scores drop by more than half from the simplest to the most challenging assembly tier,

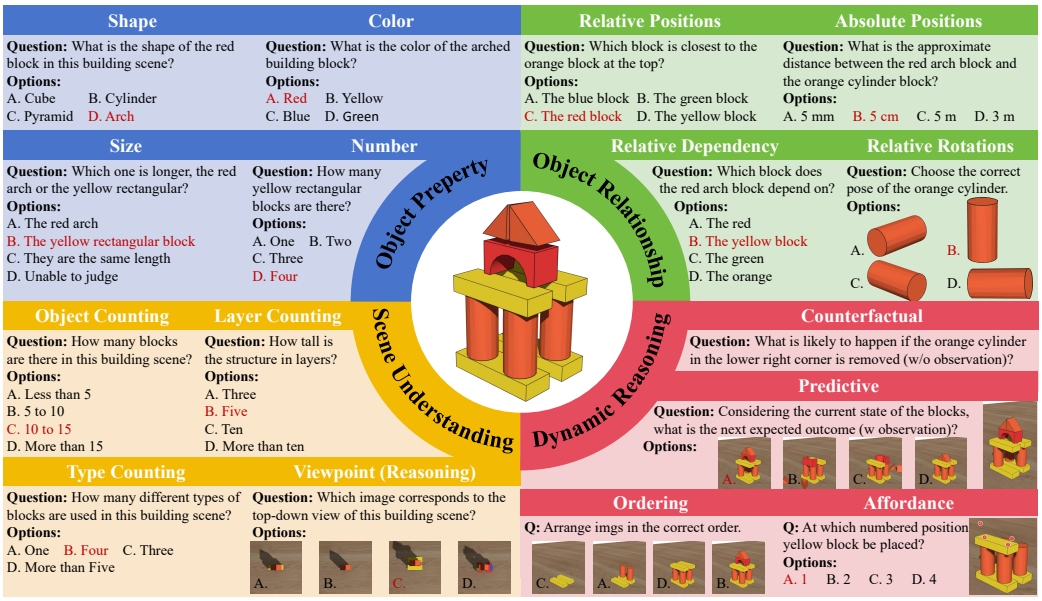


Figure 2: Physics Understanding VQA in **PhyBlock**. We construct a compact set of questions per 3D assembly scene, covering four key dimensions of physical and spatial reasoning to assess diverse aspects of the model's understanding of 3D block assembly.

and no model maintains high recall once tasks demand long-horizon sequencing or hidden-support reasoning. Second, a *perception–reasoning gap*: models answer low-level questions about colour or shape with high accuracy, yet accuracy collapses on counterfactual, causal, or affordance queries, mirroring the assembly failures that stem from unmodelled dynamics. Third, a pair of *universal error modes*: (i) mis-estimated the orientation of blocks that lead to systematically incorrect block poses, and (ii) ignored support dependencies that violate basic stability—together accounting for the majority of mistakes across architectures. Notably, enabling thinking mode prompting in larger models leaves these two error modes virtually unchanged, indicating that more text tokens alone cannot compensate for missing physical priors. Collectively, these findings expose a fundamental shortfall in current multimodal pre-training: while today's VLMs perceive objects well, they still lack the physical insight and sequential reasoning needed for reliable embodied planning. Bridging this gap will require architectures and training regimes that fuse rich visual embeddings with explicit physics reasoning and interactive feedback, charting a path toward truly capable embodied agents.

Our contributions are as follows:

- **PhyBlock benchmark.** We present a unified testbed for physical understanding and multi-step planning, built on interactive 3-D blocks in a high-fidelity physics simulator with strict guarantees on spatial precision and physical feasibility.
- **Progressive dataset.** We release a cognitively inspired dataset of 3D scenes, dependency graphs, step-wise plans, and 2,200 validated VQA pairs that scale smoothly from simple stacks to 22-block assemblies.
- **Comprehensive evaluation.** We assess 21 leading VLMs (13 proprietary, 8 open-source) and show that, despite strong perceptual skills, all models falter on complex spatial reasoning, physical inference, and long-horizon planning—exposing key challenges for future embodied intelligence.

2 Related Work

2.1 Benchmarks and Datasets

Current benchmarks for evaluating physical understanding in embodied AI systems exhibit critical limitations. While manipulation-focused benchmarks like RLBench and LIBERO [27, 30, 36] assess basic object interaction skills, they fail to systematically evaluate fundamental physical

comprehension including object stability, structural integrity, and spatial relationships. Navigation-oriented benchmarks [45, 46] emphasize environment interaction but neglect the precise physical reasoning required for structured assembly tasks. Most existing datasets either lack physical grounding [10] or suffer from limited scalability in 3D construction scenarios [39, 51]. Recent efforts like Reflective Planning [22] and VLABench [57] advance multi-stage reasoning but neglect physical constraints (e.g., gravity, structural dependencies) critical for assembly.

PhyBlock fills these gaps as the first benchmark evaluating 3D block assembly. It combines VQA and planning tasks to assess physical reasoning (stability, object properties) and multi-step planning, bridging perception and real-world understanding.

2.2 Vision-Language Models

Recent advances in VLMs have expanded multimodal reasoning capabilities and enabled their successful application across diverse domains[24, 25, 31, 32, 33, 16, 35], yet their understanding of physical phenomena remains limited. While foundational models like GPT-4 [1] and GPT-4V [40] demonstrate strong visual-language alignment, and open-source alternatives [19, 49] enable broader experimentation, these systems show fundamental gaps in physical comprehension. Specialized architectures like region-level VLMs [12, 17, 44] improve spatial awareness for object localization, while video-based models [34, 47, 53, 15] enhance temporal reasoning, yet neither adequately addresses the physical understanding required for structured 3D assembly. Current benchmarks [13, 23] primarily assess static perception or functional affordances rather than the core physical reasoning capabilities needed for tasks involving object properties, structural stability, and spatial dependencies.

This limitation persists despite architectural innovations in models like LLaVA [37, 58] and MiniGPT-4 [60], which integrate vision encoders with language models but lack explicit representations of physical constraints. Our analysis reveals that existing VLMs struggle with the intuitive physical reasoning that humans employ when manipulating objects - particularly in understanding how spatial relationships affect structural integrity, predicting physical outcomes of actions, and maintaining consistency across multi-step assembly sequences. PhyBlock addresses these gaps by providing the first systematic evaluation framework specifically designed to probe models' physical understanding through the lens of 3D block assembly, complementing existing benchmarks that focus on perception or functional reasoning.

2.3 Physical Understanding

Physical understanding [9, 26, 38] has widespread applications spanning visual reasoning [28, 56, 14], embodied AI [2, 11], *etc*. The primary works [7, 28**?**]focused on simple scenarios where visual primitives (*e.g*., spheres, cubes) are restricted to a limited set of interactions. The follow-up works [20, 59, 52] extend the scope to more realistic scenes with real-world objects and complex backgrounds. Wang *et.al* [52] introduces the 4D scene representation to simultaneously model dynamic properties of objects and multi-object interactions. PhysBench [20] is introduced to evaluate VLMs' physical world understanding capability across a more diverse and comprehensive tasks. Another stream of works focus on spatial intelligence, which requires the understanding the objects positions in 3D space and their relationships in-between.

Compared with existing benchmarks, our proposed PhyBlock offers several key advantages in evaluating physical understanding and planning. First, unlike prior datasets that are either limited to passive visual prediction or constrained to toy-like synthetic scenes, PhyBlock introduces a goal-conditioned block stacking task grounded in high-fidelity physics simulation. Second, PhyBlock supports interactive and constructive physical reasoning. Instead of merely recognizing or forecasting physical events, VLMs are required to plan and generate a sequence of physically plausible actions to achieve a specified structural goal, which aligns more closely with real-world embodied scenarios.

3 PhyBlock

In human cognitive and educational psychology, structured block play has been hypothesized to cultivate essential cognitive skills, including estimation, measurement, pattern recognition, part–whole relations, visualization, symmetry, transformation, and balance [50]. Given the cognitive benefits of

structured block play, we extend this framework to evaluate embodied agents in three fundamental capabilities: visual alignment and pose estimation, spatial reasoning, and long-horizon planning.

In this section, we introduce the hierarchical capability levels in Sec. 3.1. Then in Sec. 3.2, we detail the capability-oriented data collection process. Next, we present the construction of the Physics Understanding VQA dataset in Sec. 3.3, which enables fine-grained evaluation of scene perception and physical reasoning. Finally, the overall dataset construction process is demonstrated in Sec. 3.4.

3.1 Hierarchical Capability Levels

In human cognitive and educational psychology, structured block play has been hypothesized to cultivate essential cognitive skills. Following human cognitive skill, we propose **PhyBlock** to benchmark VLMs on Robotic 3D Block **Assembly Planing** task.

To systematically assess these core capabilities, we construct a hierarchical capability levels for embodied 3D Block Assembly inspired by the developmental stages of children's cognitive growth. Specifically, PhyBlock is curated in a hierarchical capability levels including basic perception, simple combinations, complex structures, and advanced spatial planning.

Level-1 Basic Perception. The model is required to identify and select correct blocks from a component library based on a reference diagram. Tasks involve up to four blocks with minimal variation in type and color, focusing on visual feature recognition and matching accuracy.

Level-2 Basic Simple Combinations. Building on Level-1, this stage evaluates elementary structural reasoning. The model must select fewer than six relevant blocks and generate a valid assembly sequence with up to three vertical layers, respecting basic support relations and spatial dependencies.

Level-3 Complex Structures. At this level, the model must not only select the necessary blocks but also plan an optimal assembly sequence, ensuring a logical and stable construction process. Compared to Level-2, the dependency relationships are significantly more complex. The scenarios in this level contain up to 12 blocks with a maximum of 8 layers, demanding advanced 3D spatial reasoning and multi-step decision-making capabilities.

Level-4 Advanced Spatial Planning. As the highest complexity level, this stage requires the model to execute systematic planning for assembling large-scale structures under complex spatial constraints. The scenarios involve up to 22 components, challenging the model's ability to develop a global understanding of intricate 3D structures and execute long-horizon spatial reasoning and planning.

3.2 AOV-Based Assembly Evaluation

Block assembly exhibits non-Markovian dependencies: placing a block on an upper layer requires proper support from lower layers. While inter-layer construction must follow strict temporal order, within-layer actions can often be executed in parallel. Final-state-only evaluation fails to disentangle errors in physical reasoning, planning, and control. To better analyze the hierarchical and sequential constraints inherent in physical assembly, we introduce the Activity-on-Vertex (AOV) network, which models blocks as vertices and their assembly dependencies as directed edges, as illustrated in Figure 1. This graph-based representation captures both inter-layer temporal dependencies and intra-layer parallelism, enabling fine-grained analysis of planning behaviors.

This dual-representation enables more rigorous evaluation by: (1) computing intermediate metrics to quantify partial completion, even in failure cases; and (2) diagnosing failure modes via systematic analysis of dependency violations, such as missing prerequisites or conflicting operations.

This AOV framework enables: (1) fine-grained assessment via intermediate completion metrics, (2) diagnostic analysis of failure modes based on violated dependencies, and (3) evaluation of planning robustness across valid sequence variations. Details of the AOV-based evaluation algorithm are provided in Appendix B.1.

3.3 Physics Understanding VQA

To evaluate an agent's capability for physical reasoning, we propose a comprehensive set of questions targeting diverse aspects of 3D block assembly understanding. As illustrated in Figure 2, the questions are grouped into four major categories: **Object Property**, **Object Relationship**, **Scene**

Understanding and **Dynamic Reasoning**, focusing on both static perception and dynamic physical reasoning. Details for each are provided below.

Category 1: Object Property. 1) Shape: Identify the geometric shape of a given block. 2) Color: Determine the color of a specified object. 3) Size: Compare the dimensions (e.g., length or height) of two blocks. 4) Number: Count how many blocks of a particular color or type are present. This category assesses the agent's ability to understand basic attributes of individual objects.

Category 2: Object Relationship. 1) Relative Positions: Analyze the relative positional relationships between blocks, such as proximity and distance. 2) Absolute Positions: Estimate the spatial relationships between blocks by providing concrete numerical values with physical units. 3) Relative Dependency: Identify which blocks depend on or support others. 4) Relative Rotations: Determine the relative rotational relationships between blocks. This category focuses on spatial and logical relationships among multiple blocks.

Category 3: Scene Understanding. 1) Object Counting: Estimate the number of blocks present in the scene. 2) Layer Counting: Infer how many vertical layers the construction consists of. 3) Type Counting: Count the number of distinct block types (e.g., cube, arch, cylinder). 4) Viewpoint: Match a given single-view image with its corresponding 3D scene configuration. This category assesses the agent's holistic perception of the environment, requiring recognition of object presence, spatial composition, and view-consistent scene interpretation.

Category 4: Dynamic Reasoning. 1) Counterfactual: Predict what will happen if a supporting block is removed. 2) Predictive: Anticipate the next step or possible continuation of the current assembly. 3) Ordering: Determine the correct temporal or structural sequence of subassemblies. 4) Affordance: Decide where a given block can be stably placed. This category evaluates the agent's understanding of physical dynamics, structural stability, and potential consequences of actions

These question types collectively establish a progressive and fine-grained benchmark for evaluating physical understanding in both VLMs and embodied agents. PhyBlock emphasizes grounded reasoning beyond visual recognition, targeting real-world generalization and planning competence.

3.4 Dataset Construction

Construction of Simulated Block Assets. We construct a parametric simulated block library inspired by global standards (e.g., LEGO®, Mega Bloks®), covering eight shapes and five colors. Detailed geometric specifications and texture mappings are provided in the Appendix A.1.

Construction of Block Assembly Scenes. We use the Genesis simulator to construct scenes with precise control, recording each block's pose and dependencies in structured JSON files for downstream analysis. Detailed procedures and examples are provided in the Appendix A.2.

Construction of Physics Understanding VQA. We introduce two data generation paradigms: LLM-based static VQA construction and simulation-driven dynamic VQA generation, targeting both perception and physically grounded reasoning tasks. Refer to Appendix A.4 for details.

Data Augmentation. The Level-4 block assembly scenes are generated by augmenting Level-3 scenes through compositional transformations. Due to the deliberate focus of PhyBlock as an *evaluation benchmark* rather than a large-scale training corpus, we adopt a carefully balanced dataset size—comprising 400 assembly tasks and 2,200 VQA samples—which we found sufficient to reliably assess the physical reasoning and planning capabilities of modern VLMs without introducing redundant or overlapping samples. Each scene is generated within a high-fidelity physics simulator and manually verified to ensure physical validity and uniqueness, making large-scale expansion computationally expensive and conceptually unnecessary for diagnostic evaluation.

Importantly, our minimalist design—with simple geometric shapes and plain colors—reduces visual confounds and isolates core physical reasoning capabilities, enabling a controlled and cognitively interpretable evaluation. However, leveraging our open-source code interface, users can easily scale the dataset to millions of scenes through automated compositional transformations (e.g., color combinations, shape variations, scene compositions, and lighting adjustments) without compromising quality or physical realism. This design philosophy balances rigorous benchmark fidelity with extensibility for future large-scale studies.

Table 1: Results (%) overview. Evaluation of 3D Block Assembly Planning (One-time Full Planning).

Model	Level-1			Level-2			Level-3			Level-4			Overall.		
	Rec	*Prec*	F_1	*Rec*	*Prec*	F_1	*Rec*	*Prec*	F_1	*Rec*	*Prec*	F_1	*Rec*	*Prec*	F_1
Claude-3.5 Haiku	59.1	41.2	48.6	39.8	27.8	32.7	31.1	20.5	24.7	28.8	16.6	21.1	32.5	20.6	25.3
Claude-3.5 Sonnet	74.9	72.4	73.7	58.4	56.5	57.4	44.7	41.8	43.2	**43.9**	38.9	41.2	**47.7**	44.1	45.8
Claude-3.7 Sonnet	75.6	**78.0**	**76.8**	57.6	59.5	58.6	**46.0**	**46.0**	**46.0**	42.7	**40.9**	**41.8**	47.6	**47.2**	**47.4**
Claude-3.7 Sonnet-Thinking	**75.9**	77.0	76.4	**59.6**	**60.5**	**60.1**	45.0	44.5	44.8	42.2	40.6	41.4	47.4	46.7	47.1
GPT-4o-mini	55.8	42.7	48.4	34.0	26.0	29.5	28.3	20.2	23.5	25.0	15.4	19.0	28.6	19.6	23.3
GPT-4o	69.7	67.5	68.6	50.2	48.6	49.4	39.2	36.5	37.8	35.1	31.8	33.4	40.3	37.5	38.8
GPT-o1	69.3	72.4	70.8	50.3	52.6	51.4	41.6	42.8	42.2	39.4	39.8	39.6	42.8	43.9	43.4
Gemini-1.5-flash-8b	52.0	47.0	49.4	31.4	28.4	29.8	23.1	20.9	22.0	25.4	19.6	22.1	26.0	22.1	23.9
Gemini-1.5-flash	62.6	54.7	58.4	38.5	33.7	35.9	30.9	25.5	28.0	31.1	22.8	26.4	33.0	26.4	29.3
Gemini-2.0-flash-lite	62.8	65.3	64.0	40.3	41.9	41.1	35.6	36.4	36.0	33.6	32.8	33.2	35.9	36.3	36.1
Gemini-2.0-flash	68.6	66.1	67.3	46.1	44.5	45.3	40.6	37.3	38.9	38.7	33.8	36.1	41.2	37.6	39.3
Gemini-2.0-flash-thinking-exp	69.3	60.5	64.6	47.2	41.2	44.0	36.8	32.3	34.4	36.0	29.2	32.2	39.0	33.2	35.8
Qwen-VL-Max	61.2	48.6	54.2	40.7	32.3	36.1	33.9	26.6	29.8	29.4	19.6	23.5	34.0	25.2	28.9
InternVL2.5-1B	5.3	7.3	6.2	3.0	3.8	3.3	19.3	2.5	4.4	2	2.1	2.1	4.6	7.6	5.8
InternVL2.5-8B	44.2	39.2	41.5	24.9	22.1	23.4	22.0	16.8	19.0	23.6	13.8	17.4	23.3	16.8	19.6
InternVL2.5-78B	60.1	42.1	49.5	37.6	26.3	31.0	29.2	18.4	22.6	28.3	15.1	19.7	31.2	19.0	23.6
Qwen2.5-VL-3B-Instruct	37.8	36.0	36.9	25.8	24.6	25.2	20.5	15.6	17.7	25.8	13.4	17.6	23.7	16.8	19.7
Qwen2.5-VL-7B-Instruct	43.8	46.5	45.1	23.4	24.8	24.1	20.5	22.0	21.2	19.9	14.6	16.9	21.1	19.7	20.4
LLaVa-OneVision-0.5B	43.7	16.0	23.4	29.2	10.7	15.6	19.4	3.7	6.3	24.5	6.6	10.4	24.7	6.5	10.3
LLaVa-OneVision-7B	38.9	21.2	27.4	24.3	13.2	17.1	19.5	12.3	15.1	22.0	9.0	12.7	21.4	11.2	14.7
Random	20.4	16.5	18.3	8.6	16.7	11.4	6.7	16.1	9.5	4.1	14.4	6.4	6.1	15.8	8.8

Table 2: Results (%) overview. Evaluation of Physical Understanding VQA.

Model	Object Property					Object Relationship					Scene Understanding					Dynamic Reasoning					Overrall.
	SH	*CO*	*SI*	*NU*	*Avg*	*RP*	*AP*	*RD*	*RR*	*Avg*	*OC*	*LC*	*TC*	*VP*	*Avg*	*CF*	*PD*	*OR*	*AD*	*Avg*	*Avg*
Claude-3.5 Haiku	59.3	27.3	40.7	40.0	41.8	60.0	40.0	43.3	30.0	43.3	42.0	35.3	46.7	40.7	41.2	19.3	28.0	0.0	18.0	16.3	35.7
Claude-3.5 Sonnet	**92.0**	77.3	51.3	20.0	60.2	76.0	65.5	88.0	**59.3**	**72.2**	56.7	48.7	45.3	52.0	50.7	38.0	21.3	54.0	52.0	41.3	56.1
Claude-3.7 Sonnet	91.3	77.3	52.0	44.0	66.2	79.3	59.3	85.3	47.3	67.8	52.7	54.0	53.3	35.3	48.8	23.3	14.0	50.0	40.0	31.8	54.4
GPT-4o-mini	52.7	31.3	46.7	58.7	47.4	58.0	**72.0**	36.0	44.7	52.7	43.3	38.0	33.3	45.3	40.0	31.3	26.0	28.0	26.0	27.8	42.0
GPT-4o	82.0	64.7	43.3	49.3	59.8	75.3	55.3	55.3	43.3	57.3	39.3	45.3	28.7	62.7	44.0	38.0	36.0	34.0	28.0	34.0	48.8
GPT-4.1	81.3	78.7	52.0	75.3	71.8	82.7	57.3	83.3	48.7	68.0	57.3	66.7	42.7	64.0	57.7	43.3	36.7	**74.0**	60.0	53.5	62.8
GPT-o3	88.0	**90.0**	**70.7**	**79.3**	**82.0**	86.0	55.3	**86.0**	54.7	70.5	41.3	**80.0**	63.3	**67.3**	**63.0**	**71.3**	**54.0**	52.0	**80.0**	**64.3**	**70.0**
Gemini-1.5-flash-8b	64.0	70.7	39.3	62.7	59.2	68.7	24.7	48.7	23.0	41.3	55.3	46.0	52.7	40.7	48.7	21.3	24.7	16.0	38.0	25.0	43.6
Gemini-1.5-flash	86.0	78.7	48.0	71.3	71.0	**86.7**	14.0	72.7	26.0	49.9	**70.0**	60.0	80.0	40.7	62.7	26.7	26.7	34.0	38.0	31.4	52.1
Gemini-2.0-flash-lite	81.3	87.3	54.7	77.3	75.2	78.0	52.0	79.3	29.3	59.7	52.0	54.7	60.7	40.7	52.0	38.7	21.3	46.0	34.0	35.0	55.5
Gemini-2.0-flash	84.7	86.0	68.7	78.0	79.4	82.7	56.0	83.3	35.3	64.3	65.3	59.3	**81.3**	40.7	61.7	39.3	24.7	70.0	38.0	43.0	62.1
Qwen-VL-Max	80.0	71.3	48.7	56.7	64.2	80.0	52.0	69.3	36.0	59.3	58.7	56.0	58.0	24.7	49.4	42.7	36.7	6.0	46.0	32.9	51.6
Qwen2.5-VL-72B-Instruct	82.7	58.0	50.7	42.0	58.4	70.7	42.7	74.3	46.0	58.4	28.0	40.0	34.0	21.3	30.8	40.7	30.0	40.0	48.0	39.7	47.1
Random	24.9	25.2	25.0	24.9	25.0	23.9	23.8	23.5	23.6	23.7	25.6	25.1	25.3	24.8	25.2	27.8	26.9	4.2	27.9	21.7	24.1

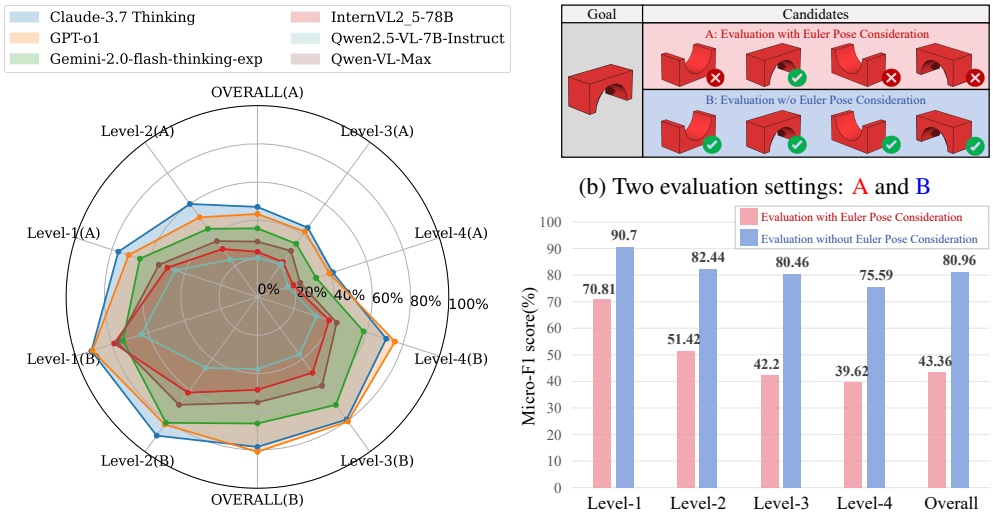
Note: The 16 abbreviations denote different task types, grouped into four categories. *SH*: Shape, *CO*: Color, *SI*: Size, *NU*: Number. *RP*: Relative position, *AP*: Absolute position, *RD*: Relative dependency, *RR*: Relative rotation. *OC*: Object counting, *LC*: Layer counting, *TC*: Type counting, *VP*: Viewpoint. *CF*: Counterfactual, *PD*: Predictive, *OR*: Ordering, *AD*: Affordance.

4 Experiment

4.1 Setup

Evaluated Models: We evaluate a range of state-of-the-art VLMs on the PhyBlock benchmark. Our evaluation includes fourteen proprietary models: GPT-O1 [43], GPT-4o [42], GPT-4o-mini [41], the Gemini-1.5 series [48], the Gemini-2.0 series [21], Qwen-VL-Max [6], the Claude 3.5 series [4], and the Claude 3.7 series [5]. Additionally, we assess eleven open-source models, including the LLaVA-OneVision series [29], the Qwen 2.5 series [55], and the InternVL 2.5 series [18].

Inference Setting: In our inference setup, models are tasked with generating structured block assembly plans given a goal image, a set of candidate blocks, and a text instruction describing the construction objective. As shown in Figure 1, we evaluate two distinct planning strategies:



(a) Task-level performance of mainstream models

(b) Two evaluation settings: A and B

(c) Performance of ***GPT-o1*** under two settings

Figure 3: Comprehensive Comparison of Mainstream Models Across Evaluation Dimensions. (a) We conduct a comprehensive comparison of six representative models under both A and B evaluation settings across all four task difficulty levels. (b) The differences between two Evaluation Setting are illustrated. For a detailed explanation, please refer to Appendix B.2. (c) A focused analysis on GPT-o1 reveals its performance under the two evaluation settings. Interestingly, we observe a significant performance boost when the strict constraint on pose alignment is relaxed, highlighting the model's potential under less rigid spatial requirements.

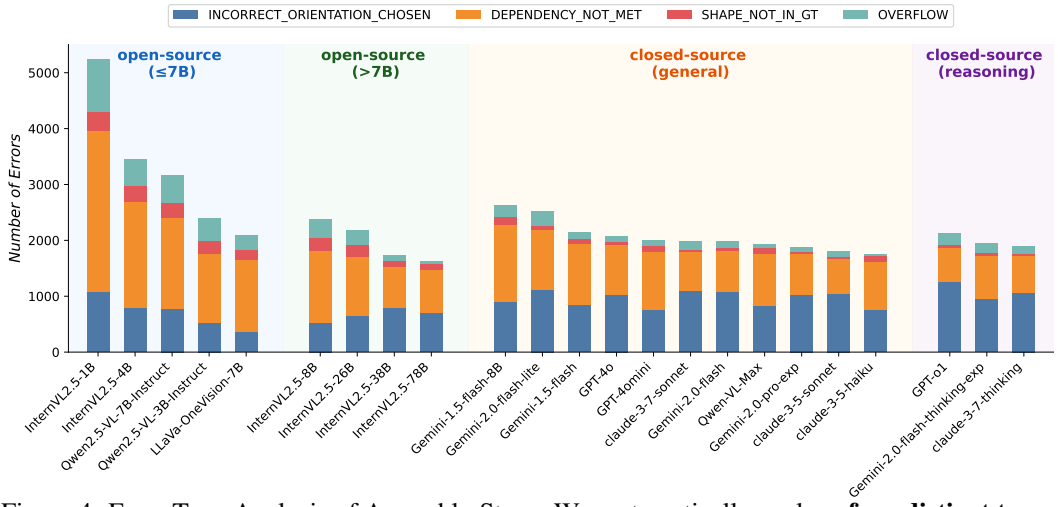


Figure 4: Error Type Analysis of Assembly Steps. We systematically analyze **four distinct types of errors** encountered during the planning process for each sample. A detailed definition and categorization of the four error types can be found in the Appendix B.3.

One-time Full Planning: Given an initial scene (RGB-D scans), a goal image, and a textual instruction, the model generates a complete block assembly plan in a single forward pass—without iterative feedback or action history. This tests the model's global planning ability.

Step-by-Step Planning: In an interactive simulator, the model receives step-wise visual observations and action histories, generating the next operation incrementally. This evaluates closed-loop planning grounded in environment feedback.

Physics Understanding VQA Inference: The model is queried with simple natural language questions based on block scene images from the Physics Understanding VQA dataset. Prompts target the question, testing the model's intuitive physical reasoning via direct visual grounding.

Evaluation Metrics: We evaluate performance using precision, recall, and F_1-score, based on step-wise correctness defined by the AOV constraints: correct steps as True Positives (TP), incorrect as False Positives (FP), and missing required steps as False Negatives (FN). Micro-F_1 is computed across all samples and difficulty levels for overall performance. In addition, to evaluate the agent's physical perception and reasoning capabilities, we adopt a simple yet effective metric for the Physics Understanding VQA dataset. Since each question follows a multiple-choice format, we report the **accuracy**—the proportion of correctly answered questions—as the primary evaluation metric.

Random Baseline. To contextualize model performance, we include a random baseline that simulates an unskilled agent acting without perception or reasoning. For the 3D Block Assembly Planning task, actions are sampled uniformly at random from the valid block and placement space within the Genesis simulator. These random plans are scored using the same AOV-based precision, recall, and F_1 metrics, yielding an overall F_1 of about **8.8%**, which represents the expected lower bound of purely stochastic assembly. For the Physical Understanding VQA branch, answers are sampled uniformly from all options. The Ordering subtask requires ranking rather than single-choice selection, lead to slightly lower scores, but the overall accuracy remains around **24%**.

Human Expert Upper Bound. To complement our quantitative evaluation of vision-language models, we additionally establish a human expert upper bound for reference. We conducted a controlled study involving **20 adult participants** with academic or engineering backgrounds related to embodied AI and robotics. Each participant was presented with **400 representative tasks** randomly sampled from our benchmark, covering both the assembly planning and physical understanding VQA branches. Participants were instructed to answer the questions or design task plans based on the same multimodal inputs (i.e., textual instructions and visual scenes) as used for model inference.

The collected responses were scored using the same evaluation metrics as the benchmark. On average, human experts achieved a **score of 378.83 out of 400**, corresponding to an overall accuracy of approximately **94.7%**. This result provides an empirical upper bound for interpreting model performance and highlights the considerable gap that remains between current VLM capabilities and human-level reasoning in physically grounded planning and understanding tasks.

4.2 Experiment Findings

Performance plummets with increasing task complexity.

As shown in Table 3, we evaluated a range of state-of-the-art vision-language models across four difficulty levels of vision-based block construction tasks. The results clearly demonstrate that current models struggle significantly as task complexity increases. The best-performing model, Claude 3.7 Sonnet, achieved the highest overall recall (47.15%) and F_1 score (47.36%), yet its performance still sharply declined from simpler (Level-1 recall 75.62%, F_1 76.78 %) to more complex tasks (Level-4 recall 40.93%, F_1 41.82%). This trend was consistently observed across all evaluated models, highlighting their limitations in handling complex spatial reasoning and multi-step planning tasks.

Current models excel at object properties but remain challenged by complex physical inference. We report the evaluation results of various models on our proposed Physical VQA benchmark, as shown in Table 2. Among the evaluated models, GPT-o3 achieves the highest overall accuracy (70.0%), demonstrating strong generalization across diverse physical reasoning tasks. It notably excels in Object Property (e.g., 90.0% in CO), Object Relationship (86.0% in RD), and Scene Understanding (80.0% in LC). Claude-3.5 Sonnet and Gemini-2.0-flash also show competitive performance, particularly in perceptual tasks such as Shape and Color, though their capabilities on reasoning-heavy tasks (e.g., Counterfactual and Affordance) remain more limited. These results highlight recent advances in multimodal models' abilities to perceive and reason about physical properties, while also indicating that complex causal and temporal reasoning remains a challenging frontier.

Incorrect Orientation Chosen dominate across models, highlighting universal spatial-reasoning gaps. Figure 4 categorizes assembly errors into four types: orientation, dependency, shape, and overflow errors. Smaller models ($\leq$7B parameters, e.g., InternVL2.5-1B) show high error rates, especially in orientation and dependency tasks, while larger open-source models reduce errors but retain dependency issues. Commercial models (GPT-4o, Claude) outperform but still struggle with

orientation chosen errors. Notably, reasoning-tuned models (GPT-o1, Claude-3.7-thinking) reduce dependency/overflow errors but not orientation chosen mistakes, underscoring spatial reasoning as a key challenge for future work. As an ablation study, we evaluate model performance under two evaluation paradigms and observe a significant performance boost when strict pose alignment constraints are relaxed, highlighting the model's limitations in universal spatial reasoning (Figure 3). Originally, we required models to directly predict absolute orientation or relative spatial coordinates, but this setting yielded near-zero success rates—even for tasks trivial to humans. After simplifying the formulation to high-level spatial reasoning, the tasks remained highly challenging.

Thinking Mode offers negligible benefit. Table 3 indicate that Claude 3.7 Sonnet performs nearly identically under normal inference and with thinking mode enabled, with a similar error distribution. This suggests that Thinking Mode reasoning provides little to no benefit on this benchmark. We posit that spatial understanding of block shapes relies more on the model's intuitive processing rather than the generation of an extensive reasoning chain. In failure cases, the model often misinterprets the number or structure of blocks at the outset, causing errors to propagate throughout the reasoning process and ultimately affecting the final output.

Table 3: Results (%) overview. Step-by-step interactive reasoning results with the environment.

Model	Overall Pref		
	$Prec$	Rec	F_1
GPT-4o	90.1	13.0	22.8
Qwen2.5-VL-72B-Instruct	94.9	35.9	52.1
Qwen-VL-Max	96.1	16.5	28.2
Claude-3.7 Sonnet-Thinking	69.7	23.9	35.6

Performance degrades steeply from perception-level tasks to strategic multi-step planning, exposing VLMs' limits in cross-modal reasoning and temporal integration. Experimental results reveal a typical hierarchical difficulty distribution across task levels, as shown in Figure 3, where Level 1 exhibits the lowest difficulty while Level 4 demonstrates the highest complexity. This progression highlights the limitations of current VLMs in tasks that transition from perceptual understanding to strategic planning. The performance degradation suggests that as tasks evolve from basic perception (e.g., object recognition) to advanced planning (e.g., multi-step reasoning), VLMs encounter challenges in effectively integrating multimodal information and executing systematic cognitive processes. Potential factors include insufficient contextual reasoning capacity, limited cross-modal alignment precision, and inadequate temporal dependency modeling in complex decision-making scenarios. In our experiments, the tasks are organized into four tiers of increasing complexity (Level-1 through Level-4). Each successive level introduces additional blocks, more intricate spatial relationships, and higher-order reasoning requirements, thereby posing a progressively greater challenge for vision-language models.

5 Conclusion

In this paper, we introduced PhyBlock, a novel benchmark for evaluating the scaling of cognitive skills in robotic 3D block assembly tasks. PhyBlock features a four-level difficulty scale, ranging from basic perception to advanced spatial planning, which allows for progressively challenging models in their cognitive abilities. The benchmark evaluates models based on three critical dimensions: partial task completion, failure diagnosis, and planning robustness. We applied PhyBlock to evaluate a range of state-of-the-art VLMs, providing a detailed analysis of their performance across these dimensions. Our results highlight both the strengths and weaknesses of these models, offering insights into their capabilities and areas for further improvement in handling complex, multi-step tasks.

6 Acknowledgments

This work is supported by National Key Research and Development Program of China (2024YFE0203100), National Natural Science Foundation of China (NSFC) under Grants No.62476293, National Postdoctoral Program for Innovative Talents under Grant Number BX20250379, China Postdoctoral Science Foundation under Grant Number 2025M771521, and General Embodied AI Center of Sun Yat-sen University.

References

[1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] P. Agrawal, A. V. Nair, P. Abbeel, J. Malik, and S. Levine. Learning to poke by poking: Experiential learning of intuitive physics. *Advances in neural information processing systems*, 29, 2016.

[3] F. AI. Helix: A vision-language-action model for generalist humanoid control, 2025.

[4] Anthropic. Claude 3.5 sonnet, 2024.

[5] Anthropic. Claude 3.7 sonnet, 2025.

[6] J. Bai, S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, Y. Fan, W. Ge, Y. Han, F. Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

[7] P. Battaglia, R. Pascanu, M. Lai, D. Jimenez Rezende, et al. Interaction networks for learning about objects, relations and physics. *Advances in neural information processing systems*, 29, 2016.

[8] J. Bjorck, F. Castañeda, N. Cherniadev, X. Da, R. Ding, L. Fan, Y. Fang, D. Fox, F. Hu, S. Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

[9] D. G. Bobrow. Qualitative reasoning about physical systems: an introduction. *Artificial intelligence*, 24(1-3):1–5, 1984.

[10] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[11] A. Byravan, F. Leeb, F. Meier, and D. Fox. Se3-pose-nets: Structured deep dynamics models for visuomotor planning and control. *arXiv preprint arXiv:1710.00489*, 2017.

[12] M. Cai, H. Liu, S. K. Mustikovela, G. P. Meyer, Y. Chai, D. Park, and Y. J. Lee. Vip-llava: Making large multimodal models understand arbitrary visual prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12914–12923, 2024.

[13] M. Cai, R. Tan, J. Zhang, B. Zou, K. Zhang, F. Yao, F. Zhu, J. Gu, Y. Zhong, Y. Shang, et al. Temporalbench: Benchmarking fine-grained temporal understanding for multimodal video models. *arXiv preprint arXiv:2410.10818*, 2024.

[14] M. Cao, H. Tang, H. Zhao, H. Guo, J. Liu, G. Zhang, R. Liu, Q. Sun, I. Reid, and X. Liang. Physgame: Uncovering physical commonsense violations in gameplay videos. *arXiv preprint arXiv:2412.01800*, 2024.

[15] M. Cao, T. Yang, J. Weng, C. Zhang, J. Wang, and Y. Zou. Locvtp: Video-text pre-training for temporal localization. In *European Conference on Computer Vision*, pages 38–56. Springer, 2022.

[16] M. Cao, H. Zhao, C. Zhang, X. Chang, I. Reid, and X. Liang. Ground-r1: Incentivizing grounded visual reasoning via reinforcement learning. *arXiv preprint arXiv:2505.20272*, 2025.

[17] K. Chen, Z. Zhang, W. Zeng, R. Zhang, F. Zhu, and R. Zhao. Shikra: Unleashing multimodal llm's referential dialogue magic. *arXiv preprint arXiv:2306.15195*, 2023.

[18] Z. Chen, W. Wang, Y. Cao, Y. Liu, Z. Gao, E. Cui, J. Zhu, S. Ye, H. Tian, Z. Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024.

[19] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez, I. Stoica, and E. P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023.

[20] W. Chow, J. Mao, B. Li, D. Seita, V. Guizilini, and Y. Wang. Physbench: Benchmarking and enhancing vision-language models for physical world understanding. *arXiv preprint arXiv:2501.16411*, 2025.

[21] G. DeepMind. Google gemini ai update - december 2024, 2024.

[22] Y. Feng, J. Han, Z. Yang, X. Yue, S. Levine, and J. Luo. Reflective planning: Vision-language models for multi-stage long-horizon robotic manipulation. *arXiv preprint arXiv:2502.16707*, 2025.

[23] C. Fu, Y. Dai, Y. Luo, L. Li, S. Ren, R. Zhang, Z. Wang, C. Zhou, Y. Shen, M. Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *arXiv preprint arXiv:2405.21075*, 2024.

[24] M. Han, L. Ma, K. Zhumakhanova, E. Radionova, J. Zhang, X. Chang, X. Liang, and I. Laptev. Roomtour3d: Geometry-aware video-instruction tuning for embodied navigation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 27586–27596, 2025.

[25] H. Hao, M. Han, C. Li, Z. Li, and X. Chang. Conav: Collaborative cross-modal reasoning for embodied navigation. *arXiv preprint arXiv:2505.16663*, 2025.

[26] S. J. Hespos, A. L. Ferry, E. M. Anderson, E. N. Hollenbeck, and L. J. Rips. Five-month-old infants have general knowledge of how nonsolid substances behave and interact. *Psychological science*, 27(2):244–256, 2016.

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

[28] A. Lerer, S. Gross, and R. Fergus. Learning physical intuition of block towers by example. In *International conference on machine learning*, pages 430–438. PMLR, 2016.

[29] B. Li, Y. Zhang, D. Guo, R. Zhang, F. Li, H. Zhang, K. Zhang, P. Zhang, Y. Li, Z. Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

[30] X. Li, K. Hsu, J. Gu, K. Pertsch, O. Mees, H. R. Walke, C. Fu, I. Lunawat, I. Sieh, S. Kirmani, et al. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024.

[31] X. Liang, L. Ma, S. Guo, J. Han, H. Xu, S. Ma, and X. Liang. Cornav: Autonomous agent with self-corrected planning for zero-shot vision-and-language navigation. *arXiv preprint arXiv:2306.10322*, 2023.

[32] B. Lin, Y. Nie, Z. Wei, J. Chen, S. Ma, J. Han, H. Xu, X. Chang, and X. Liang. Navcot: Boosting llm-based vision-and-language navigation via learning disentangled reasoning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[33] B. Lin, Y. Nie, K. L. Zai, Z. Wei, M. Han, R. Xu, M. Niu, J. Han, L. Lin, C. Lu, et al. Evolvenav: Self-improving embodied reasoning for llm-based vision-language navigation. *arXiv preprint arXiv:2506.01551*, 2025.

[34] B. Lin, Y. Ye, B. Zhu, J. Cui, M. Ning, P. Jin, and L. Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023.

[35] L. Lin, Z. Zhu, T. Zhang, and Y. Wen. Inframind: A novel exploration-based gui agentic framework for mission-critical industrial management. *arXiv preprint arXiv:2509.13704*, 2025.

[36] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

[37] H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

[38] M. McCloskey, A. Washburn, and L. Felch. Intuitive physics: the straight-down belief and its origin. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 9(4):636, 1983.

[39] S. Nasiriany, A. Maddukuri, L. Zhang, A. Parikh, A. Lo, A. Joshi, A. Mandlekar, and Y. Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. *arXiv preprint arXiv:2406.02523*, 2024.

[40] OpenAI. Gpt-4v(ision) system card, 2023.

[41] OpenAI. Gpt-4o mini - advancing cost-efficient intelligence, 2024.

[42] OpenAI. Hello gpt-4o, 2024.

[43] OpenAI. O1, 2024.

[44] Z. Peng, W. Wang, L. Dong, Y. Hao, S. Huang, S. Ma, and F. Wei. Kosmos-2: Grounding multimodal large language models to the world. *arXiv preprint arXiv:2306.14824*, 2023.

[45] M. Shridhar, J. Thomason, D. Gordon, Y. Bisk, W. Han, R. Mottaghi, L. Zettlemoyer, and D. Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10740–10749, 2020.

[46] A. Szot, A. Clegg, E. Undersander, E. Wijmans, Y. Zhao, J. Turner, N. Maestre, M. Mukadam, D. S. Chaplot, O. Maksymets, et al. Habitat 2.0: Training home assistants to rearrange their habitat. *Advances in neural information processing systems*, 34:251–266, 2021.

[47] R. Tan, X. Sun, P. Hu, J.-h. Wang, H. Deilamsalehy, B. A. Plummer, B. Russell, and K. Saenko. Koala: Key frame-conditioned long video-llm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13581–13591, 2024.

[48] G. Team, P. Georgiev, V. I. Lei, R. Burnell, L. Bai, A. Gulati, G. Tanzer, D. Vincent, Z. Pan, S. Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

[49] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[50] B. N. Verdine, R. M. Golinkoff, K. Hirsh-Pasek, N. S. Newcombe, A. T. Filipowicz, and A. Chang. Deconstructing building blocks: Preschoolers' spatial assembly performance relates to early mathematical skills. *Child development*, 85(3):1062–1076, 2014.

[51] H. R. Walke, K. Black, T. Z. Zhao, Q. Vuong, C. Zheng, P. Hansen-Estruch, A. W. He, V. Myers, M. J. Kim, M. Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning*, pages 1723–1736. PMLR, 2023.

[52] X. Wang, W. Ma, A. Wang, S. Chen, A. Kortylewski, and A. Yuille. Compositional 4d dynamic scenes understanding with physics priors for video question answering. *arXiv preprint arXiv:2406.00622*, 2024.

[53] Y. Weng, M. Han, H. He, X. Chang, and B. Zhuang. Longvlm: Efficient long video understanding via large language models. In *European Conference on Computer Vision*, pages 453–470. Springer, 2024.

[54] K. Xiang, T. J. Zhang, Y. Huang, J. He, Z. Liu, Y. Tang, R. Zhou, L. Luo, Y. Wen, X. Chen, et al. Aligning perception, reasoning, modeling and interaction: A survey on physical ai. *arXiv preprint arXiv:2510.04978*, 2025.

[55] A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[56] K. Yi, C. Gan, Y. Li, P. Kohli, J. Wu, A. Torralba, and J. B. Tenenbaum. Clevrer: Collision events for video representation and reasoning. In *International Conference on Learning Representations*, 2020.

[57] S. Zhang, Z. Xu, P. Liu, X. Yu, Y. Li, Q. Gao, Z. Fei, Z. Yin, Z. Wu, Y.-G. Jiang, et al. Vlabench: A large-scale benchmark for language-conditioned robotics manipulation with long-horizon reasoning tasks. *arXiv preprint arXiv:2412.18194*, 2024.

[58] Y. Zhang, B. Li, h. Liu, Y. j. Lee, L. Gui, D. Fu, J. Feng, Z. Liu, and C. Li. Llava-next: A strong zero-shot video understanding model, April 2024.

[59] Y. Zhao, L. Xie, H. Zhang, G. Gan, Y. Long, Z. Hu, T. Hu, W. Chen, C. Li, J. Song, et al. Mmvu: Measuring expert-level multi-discipline video understanding. *arXiv preprint arXiv:2501.12380*, 2025.

[60] D. Zhu, J. Chen, X. Shen, X. Li, and M. Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

A Detailed Dataset Collection Process

A.1 Construction of Modular 3D Block Simulation Assets

Current research in embodied intelligence faces significant limitations in constructing simulation environments for block assembly tasks. Although several prior works have investigated manipulation of block-like objects, most of them do not release the core simulation assets. Existing open-source implementations are constrained by fixed shapes and predefined connection mechanisms, limiting their applicability in studies requiring generalization. To systematically investigate embodied agents' capabilities in long-horizon planning and spatial reasoning within diverse block assembly scenarios, we develop a modular asset library of 3D blocks with geometrically extensible structures. This asset library is built using rigid-body modeling under physical constraints, supporting both compositional geometry and interactive simulation.

To ensure the constructed simulation environment aligns with real-world design patterns, we conducted a comprehensive survey of popular commercial block kits. As summarized in Table 4, we analyzed core attributes—such as shape categories, color variations, pattern types, and connection mechanisms—across different mainstream brands. These factors play a critical role in shaping the perception, manipulation, and strategy learning of embodied agents in complex block-building tasks.

Based on the above preliminary analysis, we adopt a hierarchical design principle in constructing the block asset library: **1) Geometric Primitives**: The library includes eight types of ISO-standard geometric shapes, such as cubes, cuboids, and triangular prisms, which together cover over 90% of the basic forms found in commercial kits. **2) Color System**: We adopt five highly distinguishable colors—red, yellow, blue, green, and orange—to ensure visual clarity and perceptual diversity.

We use Blender as the primary platform for building our simulation assets, consisting of two stages: **1) Geometric Modeling**: Parameterized models are constructed in Blender using a 5 cm as the base unit. Boolean operations are applied to generate the eight standard geometric shapes. **2) Physical Material Modeling**: The models are imported into Blender 3.4, where rigid-body dynamics are configured. We simulate ABS plastic properties by setting a friction coefficient of $\mu = 0.35$ and a density of $\rho = 1.04\ \mathrm{g/cm^3}$. To enhance surface detail and edge fidelity, normal mapping and edge subdivision techniques are applied. The final block asset style and size are shown in the figure 5.

A.2 Construction Pipeline of Block Assembly Scenes

Based on the aforementioned simulated block kit, we construct a variety of 3D block-building scenarios with different styles and levels of difficulty by composing individual blocks into diverse configurations. This process involves four key steps:

Table 4: Comparison of morphological characteristics across different commercial block brands. These features significantly influence the perception and interaction strategies of embodied agents in block-building tasks. "®" denotes registered trademarks of the respective companies. This research is not affiliated with or endorsed by the mentioned companies.

Brand	Shape Variety	Color Range	Pattern Types	Connection
LEGO®	100+	12	Letters / Numbers / Graphics	Slot
Mega Bloks®	50+	7	Letters / Numbers / Graphics	Slot
MAGFORMERS®	20+	8	None	Magnetic
BanBao®	100+	11	None	Slot / Screw
Gigo®	1	10	None	Slot
Learning Resources®	4	5	Graphics	Stack
TopBright®	6	10	Letters / Numbers / Graphics	Stack
Hape®	8	8	Letters / Numbers / Graphics	Stack
MuWanShiJia®	9	8	Letters / Numbers / Graphics	Stack
LeLeFish®	1	5	None	Stack

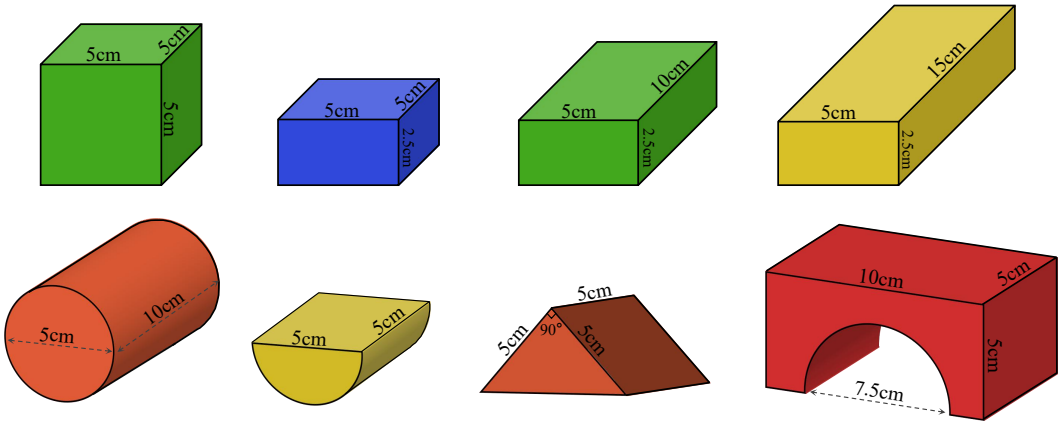


Figure 5: Basic Styles and Dimensional Specifications of 3D Simulated Block Models

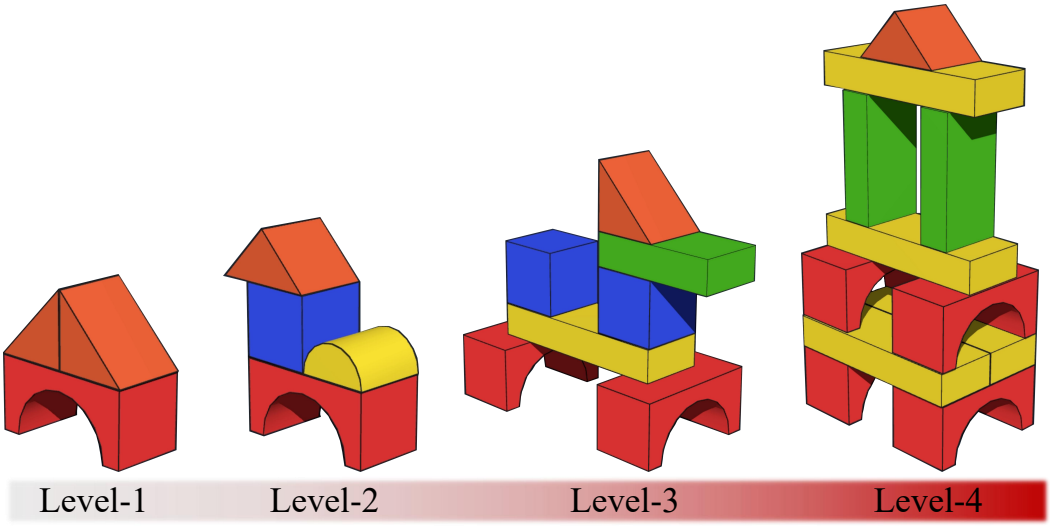


Figure 6: Illustration of Difficulty Levels in Block Assembly Tasks

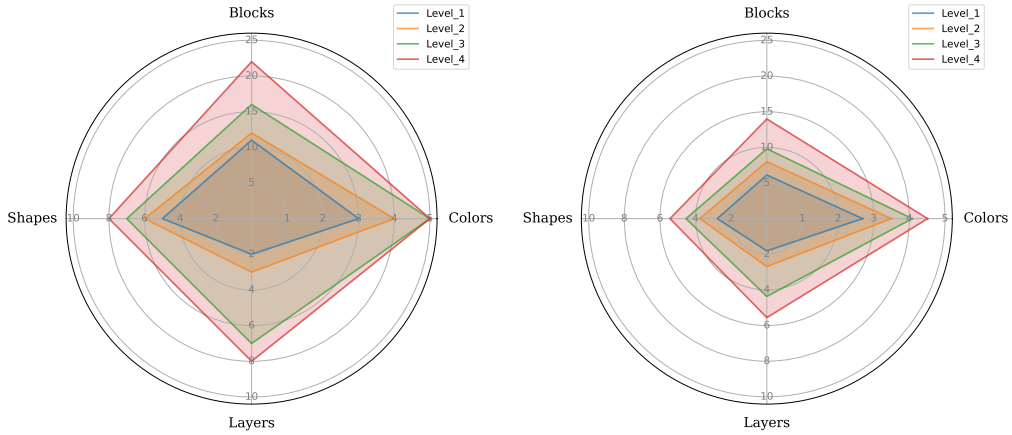
(1) Collecting Example Images of Scene Styles. We curated a diverse set of block assembly images from the internet as inspiration and references for designing our simulation scenarios.

(2) Manual Construction of Block Scenes. Each scene consists of multiple blocks of different types. To ensure high-quality dataset generation, it is essential to precisely annotate the spatial position, orientation, and topological dependencies among the blocks. We adopt a manual annotation pipeline to label the pose and relational structure of each block in a scene. All annotations are stored in structured `JSON` format. An example of such an annotation is illustrated in Figure 8.

(3) Data Augmentation and Difficulty-Level Classification. We first manually constructed 150 unique block scenes and then applied geometric augmentations such as rotation to expand the dataset to 400 scenes. The final dataset is categorized into four difficulty levels: Level-1, Level-2, Level-3, and Level-4, containing 36, 121, 138, and 119 scenes respectively, as illustrated in Figure 6. Note that Level-1 and Level-2 exhibit partial overlap.

(4) Simulation. Using the `Genesis` platform, we rendered the block-building scenes under 6 varying background environments and camera viewpoints. The resulting multi-view images serve as the basis for subsequent question-answering data design.

Moreover, the difficulty of each block assembly scene is significantly influenced by four key factors: the number of blocks involved, the diversity of block types, the variety of colors, and the depth of the final assembly hierarchy. As illustrated in Fig.7, we provide a quantitative analysis of these four



(a) Maximum Value Distribution

(b) Average Value Distribution

Figure 7: Distribution of Block Assembly Scenes Across Four Evaluation Levels

dimensions across different difficulty levels. Specifically, Fig.7(a) presents the maximum values observed in each dimension, while Fig. 7(b) reports the corresponding mean values. By comparing the two radar charts, we observe a clear upward trend across all dimensions as the difficulty level increases. This progressive pattern demonstrates the effectiveness and rationality of our dataset design in stratifying task complexity. Such a difficulty-aware structure is essential for benchmarking model performance across varying levels of planning and reasoning challenges.

A.3 Data Format and Structural Representation of Block Scenes

Each constructed block scene is stored in a structured `JSON` format, which encapsulates both the high-level scene attributes and the fine-grained block-wise specifications. This structured format ensures that the dataset can be easily parsed and utilized in simulation platforms or learning algorithms. An example is shown below:

- **"level"**: An integer indicating the difficulty level of the block scene, ranging from 1 (easiest) to 4 (most complex).
- **"shape_name"**: A unique identifier string assigned to each scene configuration.
- **"blocks"**: A list of dictionaries, each representing an individual building block in the scene. The detailed fields are:
  - **"order"**: A unique integer index indicating the ID of the block within the scene.
  - **"type"**: The geometric category of the block (e.g., `"cube"`, `"cuboid2"`, `"arch"`, `"triangle"`), consistent with the primitives defined in our asset library.
  - **"color"**: A categorical string indicating the block's color.
  - **"layer"**: An integer representing the vertical level or stacking layer of the block, where a higher value implies a physically higher placement in the structure.
  - **"depend"**: A list of integer indices referencing other blocks that this block is dependent on (i.e., those that must be placed before this block in the stacking process). These dependencies form a directed acyclic graph (DAG) that defines the scene's topological constraints.
  - **"position"**: A 3D vector `[x, y, z]` specifying the center position of the block in the world coordinate frame, expressed in meters.
  - **"orientation"**: A 3D vector `[roll, pitch, yaw]` defining the block's orientation using Euler angles in degrees, following the XYZ convention.

This format provides a comprehensive and interpretable representation of the scene configuration, facilitating reproducibility, rendering, and task reasoning. Notably, the inclusion of topological dependencies (`"depend"`) allows for accurate reconstruction of the assembly process, which is crucial for downstream embodied manipulation and reasoning tasks. Figure 8 illustrates an example of block scene annotation encoded in structured JSON format.

JSON format content of block-based scenes


```
{
  "level": 3,
  "shape_name": "020",
  "lookat_pos": [
    0.0,
    0.0,
    0.0625
  ],
  "blocks": [
    {
      "order": 1,
      "type": "arch",
      "color": "red",
      "layer": 1,
      "depend": [
        0
      ],
      "position": [
        0.0,
        -0.075,
        0.025
      ],
      "euler": [
        0,
        0,
        90
      ]
    },
    {
      "order": 2,
      "type": "arch",
      "color": "red",
      "layer": 1,
      "depend": [
        0
      ],
      "position": [
        0.0,
        0.075,
        0.025
      ],
      "euler": [
        0,
        0,
        90
      ]
    },
    {
      "order": 3,
      "type": "cuboid3",
      "color": "yellow",
      "layer": 2,
      "depend": [
        1,
        2
      ],
      "position": [
        0.0,
        0.0,
        0.0625
      ],
      "euler": [
        0,
        0,
        90
      ]
    },
    {
      "order": 4,
      "type": "cube",
      "color": "blue",
      "layer": 3,
      "depend": [
        3
      ],
      "position": [
        0.0,
        -0.05,
        0.1
      ],
      "euler": [
        0,
        0,
        0
      ]
    },
    {
      "order": 5,
      "type": "cube",
      "color": "blue",
      "layer": 3,
      "depend": [
        3
      ],
      "position": [
        0.0,
        0.05,
        0.1
      ],
      "euler": [
        0,
        0,
        0
      ]
    },
    {
      "order": 6,
      "type": "cuboid2",
      "color": "green",
      "layer": 4,
      "depend": [
        5
      ],
      "position": [
        0.0,
        0.075,
        0.1375
      ],
      "euler": [
        0,
        0,
        90
      ]
    },
    {
      "order": 7,
      "type": "triangle",
      "color": "orange",
      "layer": 5,
      "depend": [
        6
      ],
      "position": [
        0.0,
        0.05,
        0.175
      ],
      "euler": [
        135,
        0,
        0
      ]
    }
  ]
}
```

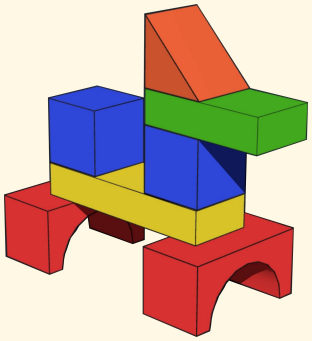



Figure 8: JSON File Content for a Sample Block Assembly Scene

A.4 Dataset Construction for Physical Understanding VQA

A.4.1 Large Language Model-based VQA Dataset Generation

To complement our block assembly scenes with question-answering tasks, we leverage a large language model (LLM) to generate a suite of Visual Question Answering (VQA) samples targeting fundamental perception skills such as color recognition and object counting. This automated data generation process consists of three main stages:

(1) Prompt Engineering. We first design a series of detailed prompts that instruct the LLM to generate high-quality, scene-aware VQA samples. Each prompt specifies the desired question type (e.g., *Color*, *Number*) and provides guidelines on phrasing, semantic clarity, and answer format. These prompts also include visual context descriptions to ensure the questions are relevant and grounded in the associated block scene images. Figure 9 illustrates a prompt example for a *Color*-type question.

(2) LLM-Based Data Generation. Using the curated prompts, we interact with a commercial LLM API—specifically, the latest GPT-4o model—to produce candidate VQA entries for each block scene. Each entry includes a scene ID, a natural language question, four multiple-choice answer options, and the correct answer label.

(3) Human Validation and Quality Assurance. To ensure the reliability and fairness of the generated dataset, we conduct a rigorous manual verification process. Three human annotators independently review each generated question three times, correcting semantic inaccuracies, verifying visual consistency, and rebalancing answer distributions. This intensive validation effort spans over 1,000 hours, resulting in a high-quality VQA dataset with accurate answers, diverse question formulations, and well-balanced choice distributions across scenes.

A.4.2 Simulation Engine-based VQA Data Generation

While large language models (LLMs) are well-suited for generating simple, static visual question answering (VQA) samples, more complex question types—such as *Predictive* and *Counterfactual* reasoning—require dynamic scene understanding grounded in physical interactions. To this end, we develop a simulation-driven VQA data engine based on the `Genesis` platform to support physically grounded question generation. The process involves three main components:

(1) Question Template Design. We first construct diverse and flexible question templates with the assistance of GPT-4o, enabling coverage across various reasoning scenarios. For instance, for the *Predictive* question category, we design multiple paraphrased templates to elicit responses about scene evolution following a specific perturbation. Sample templates include:

- What is likely to happen if the {color} {type} block is taken away?
- How will the scene change if the {color} {type} block is removed?
- What consequences might follow the removal of the {color} {type} block?
- Suppose the {color} {type} is taken out—what happens then?

Here, `color` spans *Red, Blue, Green, Yellow, Orange*, and `type` spans *Rectangular Prism, Cube, Triangular Prism, Half Cylinder, Cylinder, Arch*.

(2) Simulation-based Scene Perturbation. Based on existing JSON scene annotations, we randomly apply physically plausible perturbations—such as removing a block—to instantiate specific question templates. Each perturbed scene is then reconstructed and simulated within the Genesis engine. The resulting scene evolution is rendered as a short video (in `.mp4` format), capturing the dynamic changes. To support VQA input-output formatting, selected video frames are exported as `.png` images and used either as question prompts or as visual multiple-choice options.

(3) Human Verification. To ensure high data quality, three annotators performed three rounds of thorough manual verification, spending over 500 cumulative hours. This process ensured correctness of the VQA samples, balanced question-type and option-type distributions, and eliminated potential annotation noise or ambiguities.

This simulation-driven pipeline enables us to systematically generate physically grounded VQA samples, extending beyond the static-image domain to support robust understanding of cause-effect dynamics in structured environments.

Task Overview:
You are an intelligent VQA data generator. Given an image of a physical scene containing colored block objects of various shapes, your task is to generate a multiple-choice question that asks about the **color(s)** of a specific type of block in a scene image containing multiple colored blocks of different shapes.

Output only a valid JSON object with the following format:

```
{
  "question": "...",
  "options": ["...", "...", "...", "..."],
  "answer": "<Correct option letter: A / B / C / D>"
}
```

Guidelines:
1. The question should ask about the color(s) of blocks of a **specific shape**, using natural and varied language. Examples include:
  - "What color is the cube block in this scene?"
  - "What colors are the arch-shaped blocks at the top?"
  - "Which of the following best describes the colors of the triangular prism blocks?"

2. You may refer to multiple objects (plural form) or a single object (singular form), depending on what is shown in the image.

3. The correct answer may consist of **a single color** (e.g., "Red") or **a combination of multiple colors** (e.g., "Red & Blue", "Green & Yellow & Orange").

4. Use only the following standard color list for both single and multi-color answers:
  - Red, Blue, Green, Yellow, Orange, Purple, Pink, Brown, Gray, Black, White

5. All four answer options should be valid plausible combinations of 1–3 colors from the list.
  - Separate colors with "&" and a space: e.g., `"Red & Blue"`, `"Green & Yellow & Orange"`
  - Only one option should match the correct answer.

6. Shuffle the answer options and assign the correct one to `"answer"` using "A", "B", "C", or "D".

7. Use only the following standardized shape names in the question:
  - Rectangular Prism, Cube, Triangular Prism, Half Cylinder, Cylinder, Arch

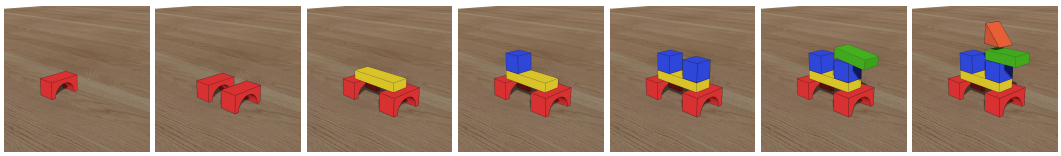
Only return the final JSON output — no explanations, no comments.

Figure 9: Example Prompt Design for the *Color* Question Type

A.5 Data Format and Structural Design for VQA Tasks

To support various types of Visual Question Answering (VQA) tasks in our block assembly benchmark, we design a unified and lightweight data format that facilitates model training, evaluation, and interpretability. Each VQA sample is stored in a structured JSON format, as shown in Figure 11, containing the following key elements:

- **scene_id**: A unique identifier for the associated block scene.
- **question**: A natural language question grounded in the visual content of the scene.
- **options**: A list of four candidate answers, typically labeled A–D. The modality of the options varies across different subtasks. For standard tasks such as *Color*, *Number*, or *Shape*, the options are textual descriptions. However, for more complex reasoning subtasks—including *dynamic*, *counterfactual*, and *ordering*—the answer choices are rendered as images corresponding to possible scene evolutions or structural outcomes. In such cases, the options are denoted as `<img1>`, `<img2>`, `<img3>`, and `<img4>`, referring to the paths of the candidate image files.
- **answer**: The correct answer's index or label, aligned with one of the provided options.



(a) Step-by-Step Construction Rendered by the Simulation Engine

(b) Dynamic Perturbations Rendered by the Simulation Engine

Figure 10: Visual Data Generated by Simulation Engine-based Methods

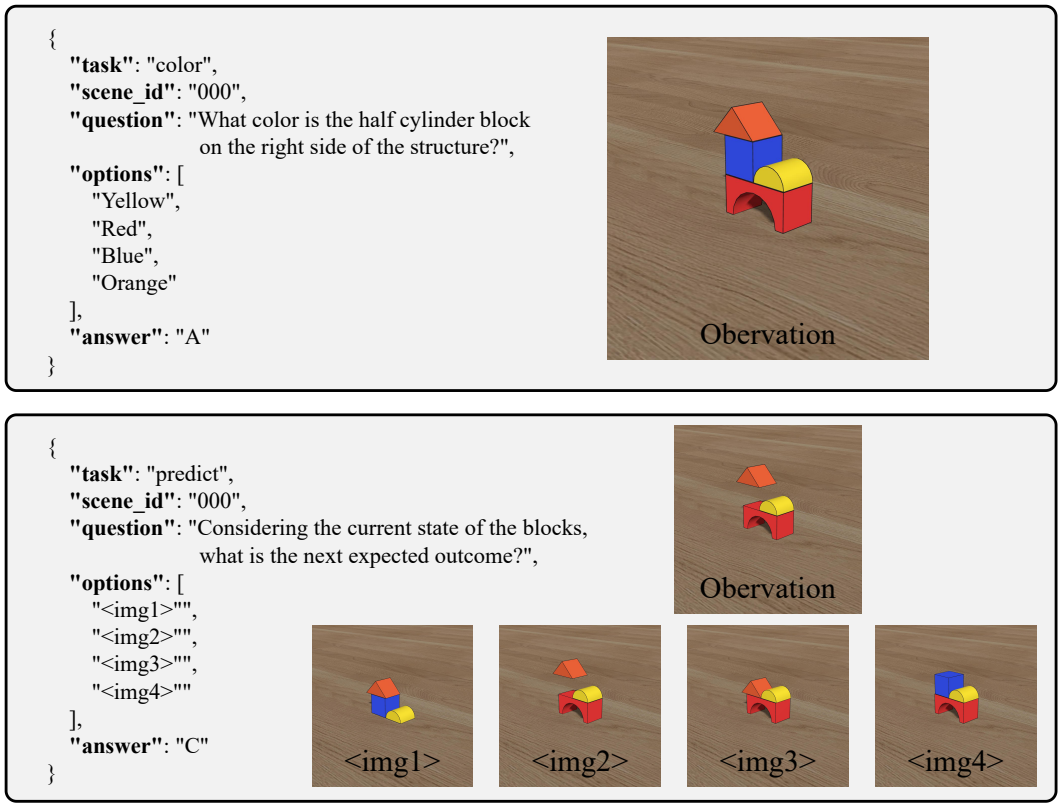


Figure 11: VQA Format Examples for *Color* and *Predict* Subtasks

This design ensures extensibility across a wide range of question categories—such as object attributes, spatial relations, numerical counting, and dynamic reasoning—while maintaining consistency across annotations. This multi-modal design across 16 subtasks enables our dataset to capture a broad spectrum of physical and semantic reasoning challenges.

B More Details On The Setup

B.1 Assembly Step Evaluation via AOV Network

Core Idea: In this work, we design a model that performs multimodal reasoning over a combination of inputs: a target reference image, candidate block images, and a natural language instruction describing the assembly process. The model is expected to integrate both visual and linguistic information to identify the correct building components from the candidates and generate a complete, ordered sequence of assembly actions.

To systematically assess the model's performance on this task, we propose an evaluation algorithm based on the *Assembly Order Validation* (AOV) network. The goal of this algorithm is to verify the correctness of each predicted assembly step by comparing it with a ground-truth assembly sequence. Specifically, the AOV network determines whether each predicted operation adheres to the reference construction order, thus evaluating the model's ability to understand and respect the sequential dependencies inherent to the assembly process. By aligning the predicted sequence with the ground truth, the method enables the computation of standard evaluation metrics, including True Positives (TP), False Positives (FP), and False Negatives (FN), thereby quantifying the model's reasoning accuracy, structural awareness, and robustness in task execution.

Algorithmic Procedure: To comprehensively evaluate a model's reasoning ability in the block assembly task, we develop a sequence-matching algorithm that compares the predicted assembly sequence against the ground-truth order, as detailed in Algorithm 1. The algorithm begins by initializing all ground-truth blocks with an "unplaced" status and resetting the match flags for all predicted blocks. It then iterates over each predicted block placement, searching for a matching target in the ground-truth sequence that: (1) has not yet been placed, (2) has exactly the same geometric properties (e.g., position and pose), and (3) satisfies topological feasibility in the assembly structure.

Once a valid match is found, the corresponding ground-truth block is marked as placed, and the predicted block's match index is recorded for subsequent metric computation. Upon completion of the matching process, the algorithm computes the following key metrics: - **TP** (True Positives): the number of correctly matched predicted blocks, - **FP** (False Positives): the number of unmatched or incorrectly matched predicted blocks, and - **FN** (False Negatives): the number of ground-truth blocks not matched by any prediction.

Based on the proposed evaluation algorithm, the computed TP, FP, and FN quantify the prediction performance for each individual block assembly scene. These values are further used to calculate the **precision**, **recall**, and **F1-score** at the scene level. While these metrics effectively capture local reasoning performance for individual samples or task-level instances, we also adopt the **micro-averaged F1-score** (Micro-F1) to aggregate performance across all samples, thereby providing a comprehensive evaluation of the model's global assembly reasoning capability.

B.2 Two Evaluation Settings for 3D Block Assembly Step Planning

We propose a two-dimensional evaluation framework to systematically assess a model's scene understanding and 3D block assembly capabilities. As illustrated in Fig. 1, the model is required to solve two hierarchical tasks: (1) select the necessary blocks (in terms of shape, color, and pose) from a candidate pool based on a reference image, and (2) generate a step-by-step assembly plan. To rigorously quantify model performance, we introduce two complementary evaluation paradigms:

A. Pose-Constrained Evaluation (with orientation Consideration). Under this strict setting, the model must produce an assembly sequence that exactly matches the ground truth orientation, as detailed in Algorithm 1. Each predicted step is considered correct only if the selected block's shape, color, and orientation all match the corresponding ground-truth attributes. This paradigm emphasizes the model's geometric reasoning ability, particularly its precision in understanding 3D rotational configurations. As shown in Fig. 12, even a minor pose error in the yellow cuboid leads to structural failure in subsequent steps, highlighting the setting's sensitivity to long-range planning inconsistencies. This strict constraint enables a fine-grained evaluation of the model's robustness in complex spatial reasoning tasks.

Algorithm 1: Evaluation Algorithm for Predicted Block Assembly Sequences

Input: Ground truth assembly sequence of blocks GT,
Predicted assembly sequence of blocks P

Output: Evaluation metrics: True Positives (TP), False Positives (FP), False Negatives (FN)

```
/* Step 1:  Initialization                                          */
```
foreach $block_{GT} \in GT$ **do**
  $block_{GT}.placed \leftarrow$ False;
foreach $block_P \in P$ **do**
  $block_P.matched\_order \leftarrow 0$;

```
/* Step 2:  Match each predicted block to the earliest valid GT block  */
```
foreach $block_P \in P$ **do**
  foreach $block_{GT} \in GT$ **do**
    if $block_{GT}.placed = False$ ***and***
      $block_{GT}.is\_place\_legal$ ***and***
      $block_P.type = block_{GT}.type$ ***and***
      $block_P.pose = block_{GT}.pose$ **then**
      $block_{GT}.placed \leftarrow$ True;
      $block_P.matched\_order \leftarrow block_{GT}.order$;
      break;

```
/* Step 3:  Count evaluation metrics                                */
```
$TP \leftarrow$ number of $block_P$ where $matched\_order \neq 0$;
$FP \leftarrow$ number of $block_P$ where $matched\_order = 0$;
$FN \leftarrow |GT| - TP$;
return TP, FP, FN

B. Topology-Oriented Evaluation (without orientation Consideration). To decouple pose sensitivity from structural planning performance, this relaxed setting ignores the orientation differences. A predicted step is deemed correct if the shape and color of the selected block match those of the ground truth, regardless of its pose. This paradigm focuses on the model's structural planning and task decomposition capabilities, effectively mitigating the impact of local pose inaccuracies on global evaluation. As shown in Fig. 12, although the yellow cuboid has an incorrect pose, it still functions as a supporting structure, allowing the assembly to proceed correctly. This setting is particularly well-suited for evaluating the model's high-level reasoning and planning competence.

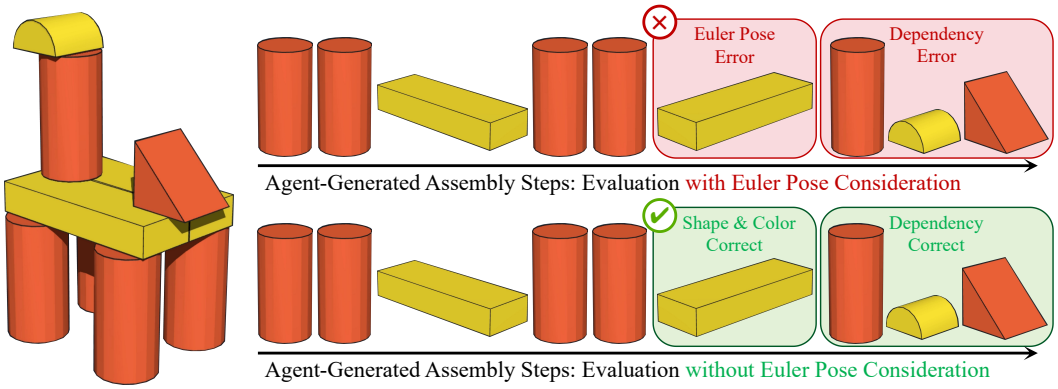


Figure 12: The situation under two evaluation settings A and B

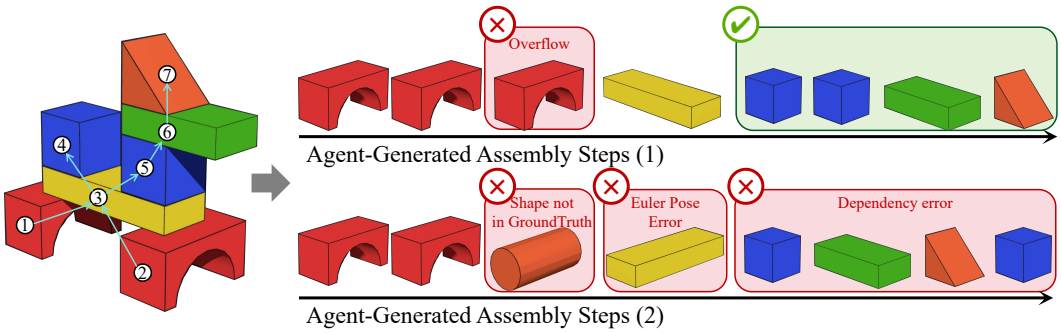


Figure 13: Four types of errors arising from the reasoning process

Together, these two evaluation protocols form a layered diagnostic framework that isolates and analyzes different dimensions of model ability. The pose-constrained evaluation probes geometric accuracy, while the topology-oriented evaluation targets structural reasoning. This hierarchical design provides a more nuanced and interpretable assessment of model performance, facilitating the identification of potential bottlenecks in assembly planning algorithms.

B.3 Four Error Type Classifications for 3D Block Assembly Step Analysis

To better understand the limitations of multimodal large models in multi-step reasoning tasks for 3D block assembly, we introduce a structured error analysis framework that categorizes step-level mistakes into four types: **Incorrect orientation chosen**, **Dependency Not Met**, **Shape Not in GT**, and **Overflow**.

Incorrect Orientation Chosen. This error occurs when the model correctly identifies the block type but predicts an incorrect orientation. Such rotation errors lead to misaligned placements despite the correct choice of block.

Dependency Not Met. This type of error arises when the placement of a block fails due to missing or incorrectly positioned prerequisite blocks from earlier steps. As a result, spatial or structural dependencies required for proper placement are violated.

Shape Not in GT. This error indicates that the model selects a block that does not belong to the ground-truth target set. It reflects a misidentification in shape or type, diverging from the intended assembly goal.

Overflow. Overflow errors represent redundant actions, where the model attempts to place a block that has already been correctly positioned. These unnecessary placements lead to structural overbuild or duplication.

Figure 13 presents illustrative examples for each of the defined error types. Specifically, in Assembly Trace (1), Step 3 involves a redundant placement of the red arch block, which is categorized as an **Overflow Error**. In Assembly Trace (2), Step 3 incorrectly selects an unnecessary orange cylinder that does not belong to the target block set, constituting a **Shape Not in GT** error. In Step 4, although the yellow cuboid is the correct block, its predicted orientation does not match the ground-truth pose, thus falling under the **Incorrect Orientation Chosen** category. Moreover, the failure to properly place the yellow cuboid in the second layer results in a cascade of **Dependency Not Meet** errors, as blocks in the third, fourth, and fifth layers rely on it for support and cannot be placed successfully.

This fine-grained evaluation paradigm not only pinpoints the root causes of performance discrepancies across models, but also provides a more interpretable perspective for understanding the underlying mechanisms of reasoning failures.

B.4 Prompt Design for 3D Block Assembly Step Planning

Prompts for Full Planning in a Single Step

You are an intelligent agent who can understand the spatial relationship between objects and assist me in the task planning of building blocks. I will give you a image of the target block and a image of the blocks to be used. Please choose the blocks that match the target and assemble them in the correct order based on your current observation.

Task Overview:
You must construct a structure using building blocks based on visual input. Two images are provided:

1. Main Target Image:
- Displays the desired structure composed of various building blocks.
- Shows the overall layout, including how many blocks, the number of layers, and the spatial relationships between blocks.

2. Block Dictionary Image:
- Contains a collection of sub-images, each showing a building block in a specific euler orientation.
- Each sub-image has a block index marked at the lower right corner.

Steps to Complete the Task:
1. Analyze and Describe the Main Target Image:
- Count & Layers: Determine how many blocks are used and how many layers the structure has.
- Spatial Relationships: Describe how the blocks are arranged relative to each other (e.g., which blocks are adjacent, above, or below one another).

2. Select the Required Blocks from the Dictionary:
- Identify Needed Blocks: Based on your analysis, decide which blocks from the dictionary are required to match the target structure.
- List Block Indices: Provide a list of the corresponding block indices.
- Matching Criteria: The chosen blocks must match the target image exactly in shape, color, and rotational orientation.
- Important: You cannot rotate any block. Use the blocks exactly as they appear in the dictionary.
- The same block index may be used multiple times if needed.
- Example: If the main image shows a green rectangular block in a horizontal orientation, you must select the horizontally oriented green rectangular block from the dictionary—not the vertical one.

3. Determine the Assembly Order:
- Inference of Order: Based on the selected blocks, infer a step-by-step assembly sequence.
- Output Format: Use the following template for your instructions:

```
{
  Step1: Move block with index {idx}
  Step2: Move block with index {idx}
  ...
  Stepn: Move block with index {idx}
}
```

Additional Guidelines:
- Detailed Analysis: Ensure your description of the main target image is thorough and covers all key details (block count, layers, spatial arrangement).
- Exact Matching: Carefully match each block's rotational orientation as shown in the dictionary; no adjustments or rotations beyond the provided images are allowed.
- Clear Assembly Instructions: Your final output must be organized and follow the given step-by-step format precisely.

Figure 14: Prompts for Full Planning in a Single Step

Prompts for Step-by-Step Planning

You are an intelligent agent who can understand the spatial relationship between objects and assist me in the task planning of building blocks. I will give you a image of the target block and a image of the blocks to be used. Please choose the blocks that match the target and assemble them in the correct order based on your current observation.
Please do not start working before I say "start working!".
Instead, just output the message "waiting for next input".

Task Overview:
You must construct a structure using building blocks based on visual input. Two images are provided:

1. Main Target Image:
- Displays the desired structure composed of various building blocks.
- Shows the overall layout, including how many blocks, the number of layers, and the spatial relationships between blocks.

2. Block Dictionary Image:
- Contains a collection of sub-images, each showing a building block in a specific euler orientation.
- Each sub-image has a block index marked at the lower right corner.

Action list for completing the task:
Move(id): Use the blocks with candidate label id in the given block dictionary image.
Done: Have finished the task.

The texts above are part of the overall instruction. Do not start working yet.

Steps to Complete the Task:
1. Analyze and Describe the Main Target Image:
- Count & Layers: Determine how many blocks are used and how many layers the structure has.
- Spatial Relationships: Describe how the blocks are arranged relative to each other (e.g., which blocks are adjacent, above, or below one another).

2. Select the Required Blocks from the Dictionary:
- Identify Needed Blocks: Based on your analysis, decide which blocks from the dictionary are required to match the target structure.
- List Block Indices: Provide a list of the corresponding block indices.
- Matching Criteria: The chosen blocks must match the target image exactly in shape, color, and rotational orientation.
- Important: You cannot rotate any block. Use the blocks exactly as they appear in the dictionary.
- The same block index may be used multiple times if needed.
- Example: If the main image shows a green rectangular block in a horizontal orientation, you must select the horizontally oriented green rectangular block from the dictionary—not the vertical one.

3. Determine the Assembly Order:
- Inference of Order: Based on the selected blocks, infer a step-by-step assembly sequence. Also pay attention to the order of building blocks, you need to start from the basic level.
- Output Format: Previously defined actions must be used. Use the following template for your instructions:

Next plan: ...

4. Note the actions that have been executed:
- In the execution action, all actions that have been executed will be attached with feedback related to the action execution.
- For example: **Execution failed** represents execution failure; **Execution successful** represents execution success.

The texts above are part of the overall instruction. Do not start working yet.

Figure 15: Prompts for Step-by-Step Planning

Prompts for Step-by-Step Planning

##Rules:
1. You can only use the actions I specified and the identification ID in the block dictionary image.
2. Only need to perform a single plan each time, do not output the complete plan.
3. Please note that the block you select must be consistent with the target image in color and posture. Please do not select the same block but in a different posture!
4. Only need to perform a single plan each time, do not output the complete plan. This means that you can only use one object at a time. For example, Next plan: Move(3).
5. You need to note that Executed Plan includes all the plans you have executed. Please do not repeat the previous step if the previous step failed to execute!

The texts above are part of the overall instruction. Do not start working yet.

The main target image you need to build is shown in the following picture.
<target goal image>

The block dictionary Image you need to use is shown in the following picture.
<block dictionary image>

Now please output your next plan based on the main target image, block dictionary image and your observation.

Observation:
<observation image>

Executed Plan:
{HISTORY}

start working!

Figure 16: Prompts for Step-by-Step Planning

C More Examples

C.1 Examples under Six Background Conditions

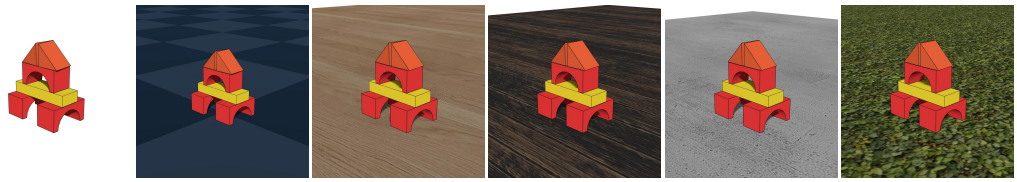


Figure 17: Block Assembly Scenes Across Six Environmental Backgrounds

C.2 Examples from 3D Block Assembly Scenes

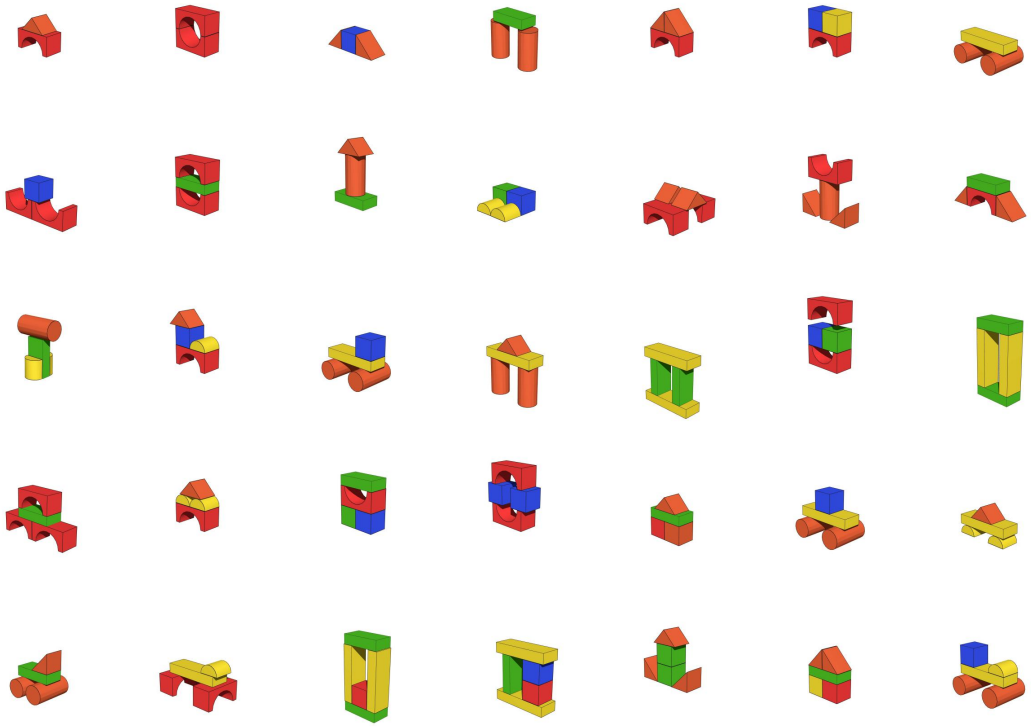


Figure 18: Partial Block Assembly Scenes at **Level-1** Difficulty

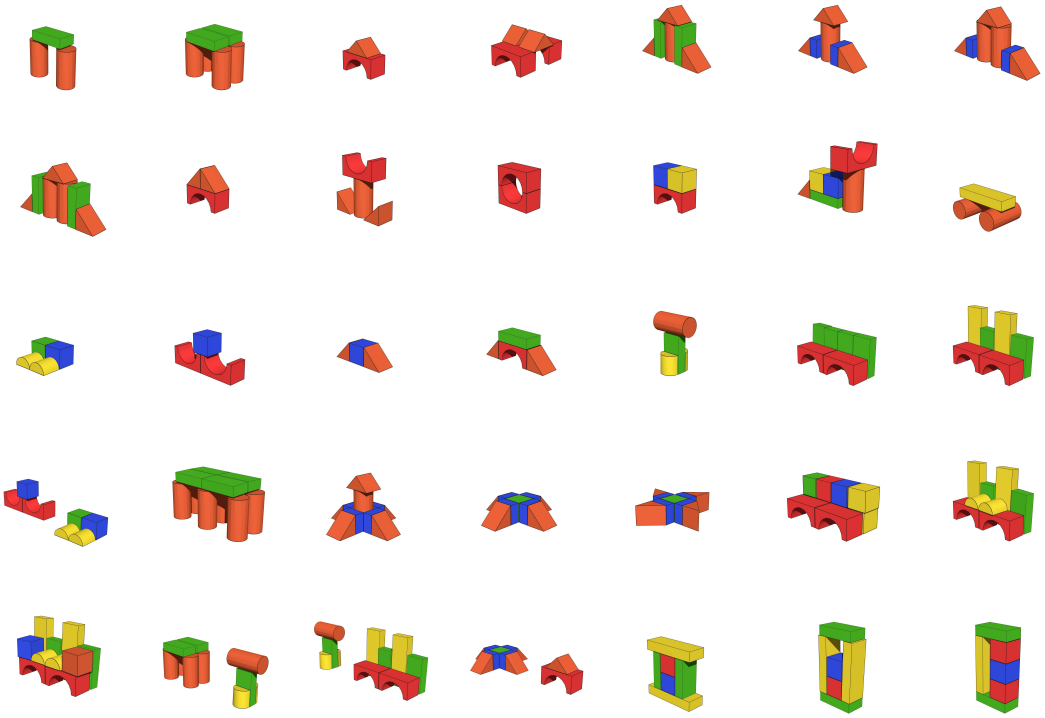


Figure 19: Partial Block Assembly Scenes at **Level-2** Difficulty

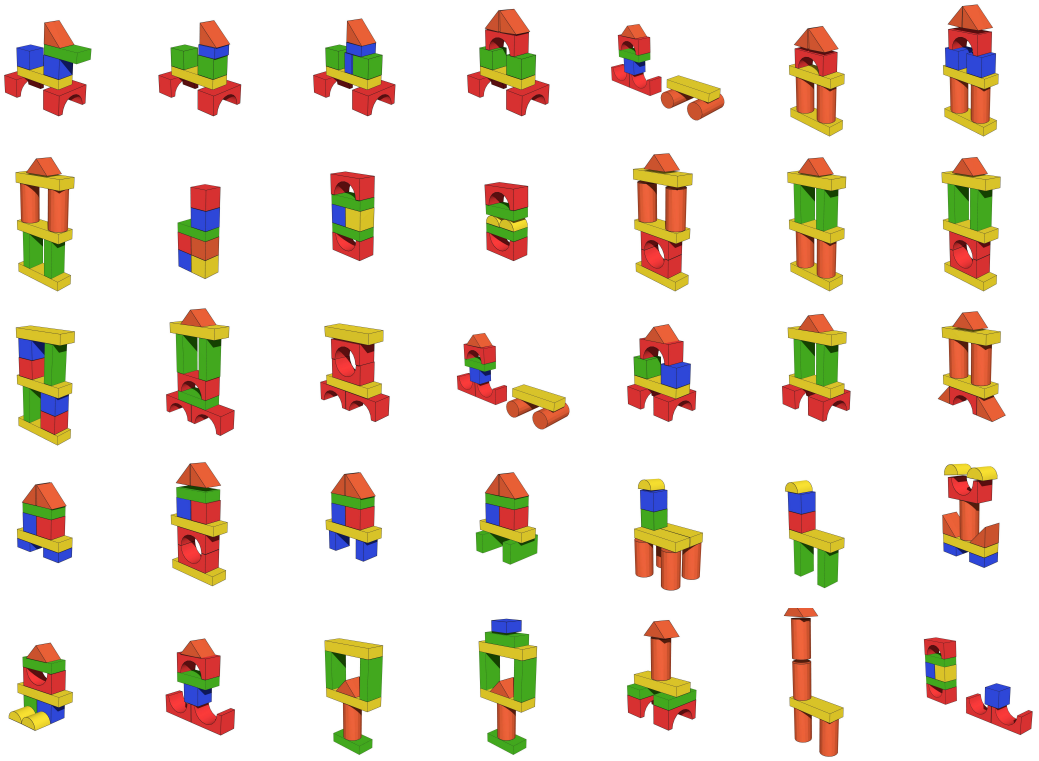


Figure 20: Partial Block Assembly Scenes at **Level-3** Difficulty

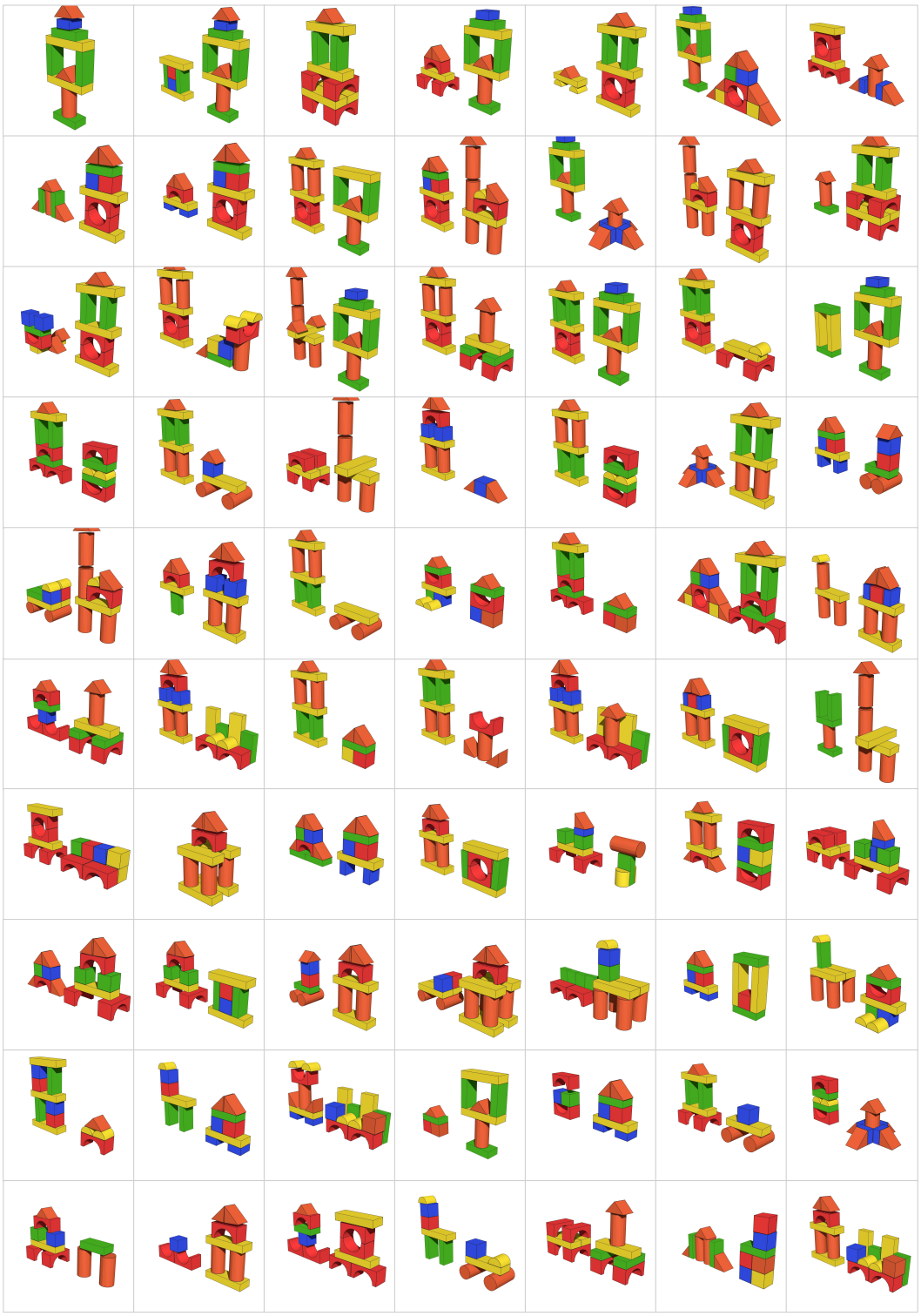


Figure 21: Partial Block Assembly Scenes at **Level-4** Difficulty

C.3 Examples of Visual Question Answering (VQA) Types

Question: What is the shape of the yellow block in this scene?

Answer : A. Cube B. Rectangular Prism C. Half Cylinder D. Sphere

Question: What is the shape of the red block in the center of this scene?

Answer : A. Arch B. Triangular Prism C. Half Cylinder D. Sphere

Question: What is the shape of the red block at the bottom of this scene?

Answer : A. Cube B. Arch C. Triangular Prism D. Sphere

Figure 22: VQA Examples for the ***Shape*** Subtask

Question: What color are the rectangular prism blocks in this scene?

Answer : A. Yellow & Blue B. Red & Blue C. Blue D. Green

Question: What colors are the arch blocks in this scene?

Answer : A. Yellow & Blue B. Orange C. Blue D. Red

Question: What colors are the triangular prism block at the top of the structure?

Answer : A. Orange & Blue B. Orange C. Green D. Red

Figure 23: VQA Examples for the ***Color*** Subtask

Question: Which block is taller, the red arch block or the blue rectangular prism?

Answer : A. The red arch B. The blue cube C. They are the same size D. Unknown

Question: Which block is taller, the red arch block or the orange triangular prism?

Answer : A. The red arch B. The orange triangular prism
C. The yellow rectangular prism D. They are equally tall in size

Question: Which block is longer, the orange triangular prism or the green block?

Answer : A. The orange triangular prism B. The green rectangular prism
C. The red arch block D. They are equally tall in size

Figure 24: VQA Examples for the ***Size*** Subtask

Question: How many red arch blocks are there in the scene?

Answer : A. One B. Two C. Three D. Four

Question: How many orange triangular prisms are there?

Answer : A. 1 B. 2 C. 3 D. 4

Question: How many blue cube blocks are in the scene?

Answer : A. Zero B. One C. Two D. Three

Figure 25: VQA Examples for the ***Number*** Subtask

Question: Which block is directly above the red arch block?

Answer : A. The yellow rectangular prism block B. The orange triangular prism
C. The blue cube block D. The yellow half cylinder

Question: Which block is directly below the orange triangular prism block?

Answer : A. The yellow rectangular prism block B. The orange triangular prism
C. The red arch block D. The yellow half cylinder

Question: Which blocks are sandwiched between the orange block and the yellow block in the middle?

Answer : A. The red arch block B. the green rectangular prism block
C. The blue cube block D. The yellow rectangular prism block

Figure 26: VQA Examples for the ***Relative Position*** Subtask

Question: What is the approximate distance between the blue cube block and the orange triangular prism block?

Answer : A. 15mm B. 25cm C. 15cm D. 2m

Question: What is the approximate distance between the yellow rectangular prism and the orange triangular prism block?

Answer : A. 25cm B. 0.5m C. 15cm D. 80mm

Question: What is the approximate distance between the blue cube block and the red arch block on the right?

Answer : A. 5mm B. 10cm C. 1m D. 50cm

Figure 27: VQA Examples for the ***Absolute Position*** Subtask

Question: Which block is the red arch block resting on?

Answer : A. The blue cube block B. The orange triangular prism
C. The red rectangular prism D. The yellow rectangular prism

Question: Which block is the orange triangular prism block supported by?

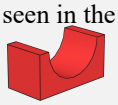
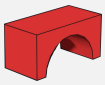
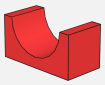
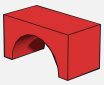
Answer : A. The red arch block B. The yellow rectangular prism block
C. The red cube block D. The red rectangular prism block

Question: Which block is the blue cube block resting on?

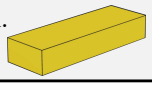
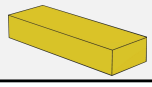
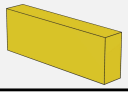
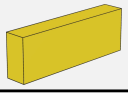
Answer : A. The red arch B. The green rectangular prism
C. The orange triangular prism D. The yellow rectangular prism

Figure 28: VQA Examples for the ***Relative Dependency*** Subtask

Question: Which candidate shows the correct orientation of the red arch block as seen in the scene?

Answer : A.  B.  C.  D. 

Question: Which candidate shows the correct orientation of The yellow rectangular prism block as seen in the scene?

Answer : A.  B.  C.  D. 

Question: Which candidate shows the correct orientation of the orange triangular prism block as seen in the scene?

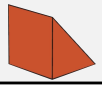
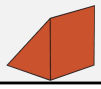
Answer : A.  B.  C.  D. 

Figure 29: VQA Examples for the ***Relative Rotation*** Subtask

Question: How many individual building blocks are present in the scene?

Answer : A. Less than 5 B. 5 to 9 C. 10 to 14 D. More than 14

Question: Can you count all the blocks used in this structure?

Answer : A. One B. Three C. Six D. Eight

Question: What's the total block count for this construction?

Answer : A. 4 B. 7 C. 6 D. 10

Figure 30: VQA Examples for the ***Object Counting*** Subtask

Question: Determine how many tiers are present in this 3D block scene.

Answer : A. 8 B. 4 C. 3 D. 2

Question: Can you count the number of stacked layers in this block setup?

Answer : A. 8 B. 4 C. 3 D. 2

Question: How many layers are there in this building scene?

Answer : A. 5 B. 3 C. 8 D. 2

Figure 31: VQA Examples for the ***Layer Counting*** Subtask

Question: How many different shape-color combinations of blocks are used in this building scene?

Answer : A. two B. three C. four D. five

Question: Determine the total count of different block variants by shape and color used in this setup.

Answer : A. two B. three C. four D. six

Question: What is the total number of distinct block types, considering both shape and color?

Answer : A. 4 B. 5 C. 6 D. 9

Figure 32: VQA Examples for the ***Type Counting*** Subtask

Question: Select the image that corresponds to a top-down view of the scene.

Answer : A.  B.  C.  D. 

Question: Choose the image that shows the same layout as seen from above.

Answer : A.  B.  C.  D. 

Question: Select the image that corresponds to a front view of the current scene.

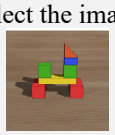
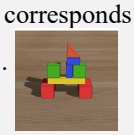
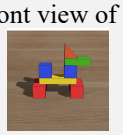
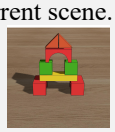
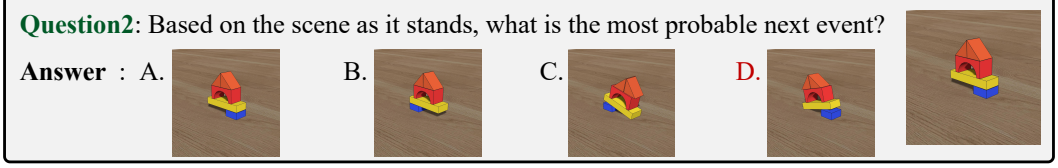
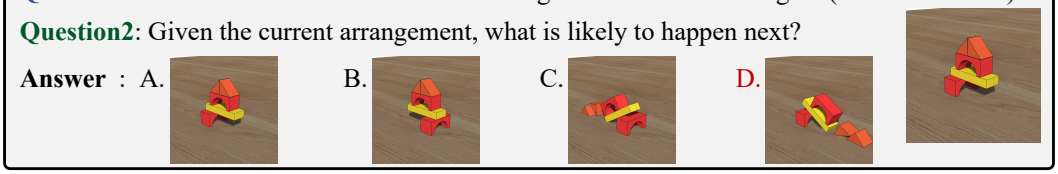
Answer : A.  B.  C.  D. 

Figure 33: VQA Examples for the ***Viewpoint*** Subtask

Question1: What will happen if the blue block on the left is taken away? (w/o observatio)
Question2: Based on the scene as it stands, what is the most probable next event?
Answer : A. B. C. D.



Question1: What would be the result of removing the red arch on the right? (w/o observation)
Question2: Given the current arrangement, what is likely to happen next?
Answer : A. B. C. D.



Question1: What would be the result of removing the red arch on the left? (w/o observation)
Question2: Given the current setup of the blocks, what will happen next?
Answer : A. B. C. D.

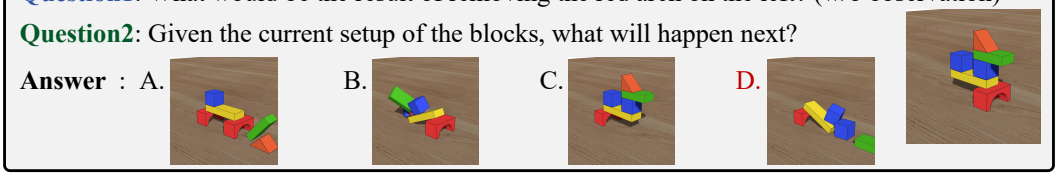
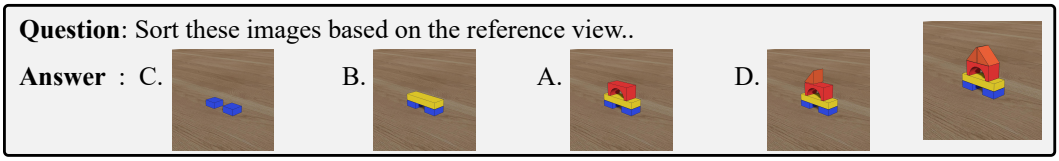
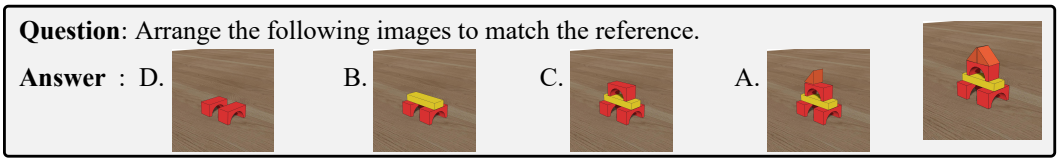


Figure 34: VQA Examples for the ***Counterfactual*** and ***Predictive*** Subtask

Question: Sort these images based on the reference view..
Answer : C. B. A. D.



Question: Arrange the following images to match the reference.
Answer : D. B. C. A.



Question: Put the images in the correct order according to the target.
Answer : B. C. A. D.

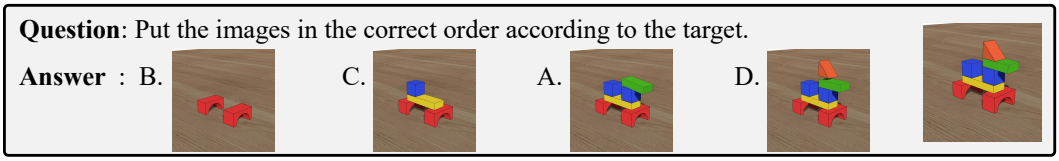
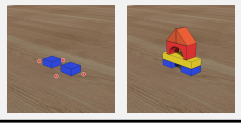
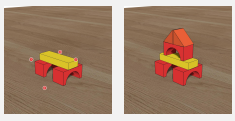


Figure 35: VQA Examples for the ***Ordering*** Subtask

Question: To match the reference scene, where should the yellow block be placed now?
Answer : A. Position1 B. Position2 C. Position3 D. Position4



Question: Where should the red arch block be placed next, according to the target structure?
Answer : A. Position1 B. Position2 C. Position3 D. Position4



Question: Where should the blue cube block be placed next, according to the target structure?
Answer : A. Position1 B. Position2 C. Position3 D. Position4

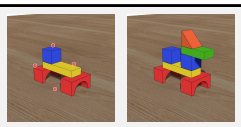


Figure 36: VQA Examples for the ***Affordance*** Subtask

C.4 Examples of One-time Comprehensive Planning Outputs

Analysis of the Main Target Image:

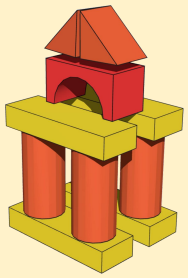
Block Count and Layers:
- **First Layer (Bottom):** 1 yellow rectangular block placed horizontally.
- **Second Layer:** 4 orange cylindrical blocks placed vertically, standing on the yellow block.
- **Third Layer:** 2 yellow rectangular blocks, one on each side, placed horizontally on top of the cylinders.
- **Fourth Layer:** 1 red semicircle block placed centrally on top of the yellow blocks.
- **Fifth Layer (Top):**
  - 1 red square block with an arch cut out, placed on the semicircle block.
  - 2 red triangular blocks form a roof on top of the square block.

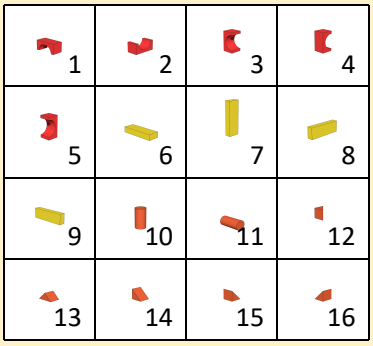
Spatial Relationships:
- The yellow rectangular base supports the four orange cylinders.
- The cylinders support the two yellow rectangular blocks.
- The red semicircle block is centrally placed on the yellow blocks.
- The red square block rests on the semicircle.
- The two red triangular blocks are placed as a roof on the red square block.

Selection of Required Blocks from the Dictionary:
1. **Yellow Rectangular Block:** (used 3 times) **Index 6** and **Index 9** (both are yellow rectangles).
2. **Orange Cylindrical Block:** (used 4 times) **Index 10**.
3. **Red Semicircle Block:** (used 1 time) **Index 1**.
4. **Red Square Block with Arch:** **Index 4**.
5. **Red Triangular Block for Roof:** (used 2 times) **Index 13**.

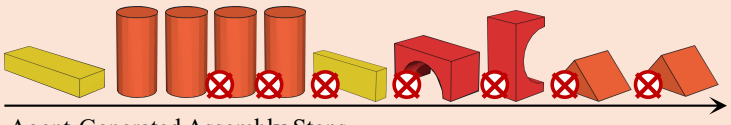
Assembly Instructions:

```json
{
 "Step1": "Move block with index 6",
 "Step2": "Move block with index 10",
 "Step3": "Move block with index 10",
 "Step4": "Move block with index 10",
 "Step5": "Move block with index 10",
 "Step6": "Move block with index 9",
 "Step7": "Move block with index 1",
 "Step8": "Move block with index 4",
 "Step9": "Move block with index 13",
 "Step10": "Move block with index 13"
}
```





Assembly Goal and Candidate Blocks



F1-Score: 0.286

Agent-Generated Assembly Steps

Figure 37: Example inference from ***GPT-4o*** on scene No.029

Figure 37 presents the inference outputs of the GPT-4o on scene No.029. The Assembly Goal and Candidate Blocks illustrate the target assembly configuration image and the set of available building blocks, respectively. Based on the Generated Assembly Steps, we observe that only the first three steps are correct. The two orange cylinders are misclassified as incorrect due to a failure in spatial dependency: they were not placed on top of the yellow cuboid as required. This violation of the dependency constraint led to the collapse of the second layer, causing all subsequent upper-layer blocks to be incorrectly assembled. As a result, the evaluation metrics for this scene are: True Positives (TP) = 3, False Positives (FP) = 7, and False Negatives (FN) = 8, yielding a final F1 score of **0.286** for the predicted assembly steps.

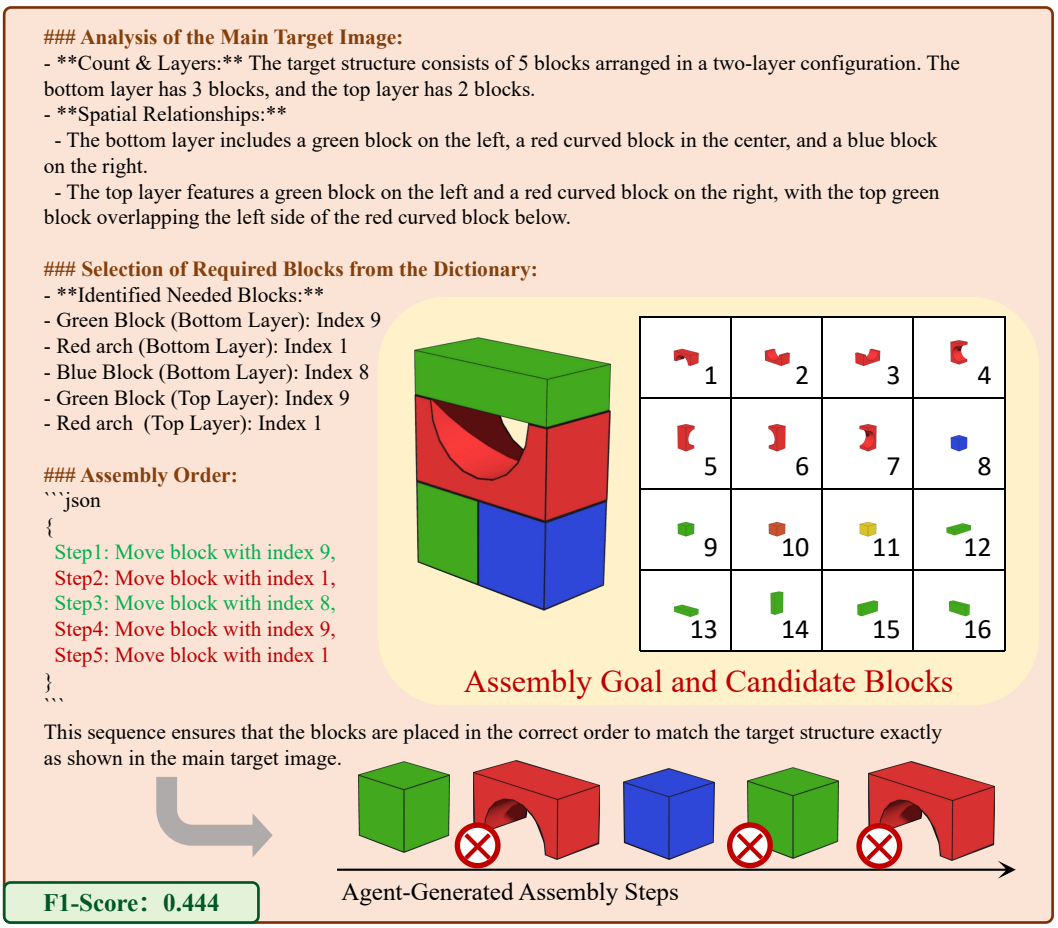


Figure 38: Example inference from ***Qwen-VL-Max*** on scene No.069

Figure 38 presents the inference outputs of the Qwen-VL-Max model on scene No.069. The Assembly Goal and Candidate Blocks denote the reference image of the target structure and the set of available building blocks, respectively. From the Generated Assembly Steps, it can be observed that only two steps were executed correctly. Specifically, the red arch block in step two was placed with an incorrect pose, the block selected in step four was a spillover error (i.e., exceeding the necessary components), and the final red arch block was also misaligned due to a pose error. As a result, the evaluation metrics for this scene are: True Positives (TP) = 2, False Positives (FP) = 3, and False Negatives (FN) = 2, yielding a final F1 score of **0.444** for the predicted assembly trajectory.

D More Results and Analysis

Repeated evaluations with five random seeds were conducted to ensure robust and reliable model performance. This approach guarantees that the reported outcomes are not influenced by random fluctuations and provides a more consistent measure of the models' performance. For each model, the mean and standard deviation (±) of the F1 score were computed across various difficulty levels, from Level 1 to Level 4, as well as the overall performance. The evaluation includes both closed-source models, such as GPT-4.1, and open-source models, such as Qwen2.5-VL-7B-Instruct and Cosmos-Reason1-7B. The performance of these models is summarized in Table 5.

As shown in Table 5, GPT-4.1 consistently outperforms the open-source models across all levels, with notably low standard deviations (±2.3 to ±1.1), indicating that its performance is stable and reproducible. In contrast, the open-source models exhibit larger variations in their results, suggesting less consistency across different runs. These low error margins for GPT-4.1 confirm the reliability of its performance and demonstrate that the observed differences are not due to random fluctuations.

Table 5: F1 Scores for Model Evaluation across Difficulty Levels.

Model	Difficulty Level				Overall
	Level 1	*Level 2*	*Level 3*	*Level 4*	
GPT-4.1	94.42 ± 2.3	46.26 ± 2.1	39.13 ± 2.4	36.07 ± 1.3	39.73 ± 1.1
Qwen2.5-VL-7B-Instruct	44.13 ± 1.3	24.12 ± 1.1	21.12 ± 2.1	16.71 ± 1.7	20.40 ± 1.2
Cosmos-Reason1-7B	43.45 ± 1.8	29.42 ± 2.1	21.53 ± 1.7	20.42 ± 1.5	23.27 ± 1.2

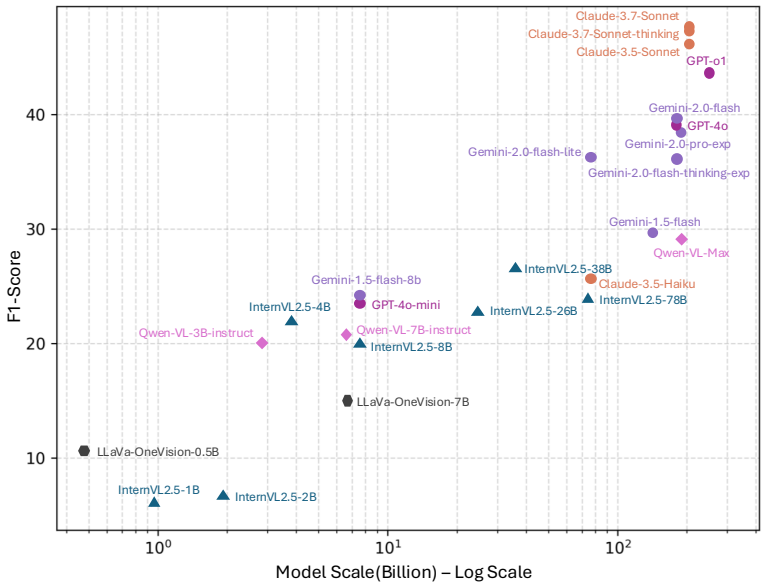


Figure 39: The impact of model size on F_1 Score

This reinforces the robustness of our evaluation framework and underscores the superior reasoning ability of GPT-4.1, particularly when compared to the open-source models.

The comparable F1 scores observed between Level 3 and Level 4 indicate a strong transfer of reasoning patterns across structural complexities. Upon deeper inspection, our extendable data creation logic involves constructing (partially) Level 4 structures by horizontally merging two different Level 3 structures, while maintaining the same vertical depth. This design allows the reasoning patterns learned for Level 3 to transfer effectively to Level 4, enabling the model to reuse similar inference chains. However, the merging process also introduces spatial reasoning challenges, such as partial visual occlusions and extended reasoning chains, which may compound earlier errors and cause minor inconsistencies. These subtle factors likely explain the observed performance plateau and slight variations across Levels 3 and 4.

Robust multi-step reasoning emerges only in >100B-parameter models, whereas smaller models falter. As shown in Figure 39, models with over 100 billion parameters consistently achieve an F_1 score above 30, demonstrating strong reasoning and planning abilities in the block building task. As parameter size decreases, performance declines significantly, with InternVL2.5-1B achieving only $F_1 = 5$, highlighting the challenges smaller models face in handling multi-step reasoning and spatial constraints. The LeiDA Graph further illustrates performance variations across task difficulty levels, where Claude 3.7 Thinking and GPT-4 maintain relatively strong results across all levels, while smaller models like InternVL2 5-78B and Qwen-VL-Max show inconsistencies, particularly in more complex tasks. These findings emphasize the crucial role of model scale in structured reasoning and multi-step decision-making.

Models misjudge block poses, prioritizing color over precise spatial alignment. In our experiments as shown in Figure 4, we found that current vision-language models often exhibit inaccuracies

in understanding the spatial poses of building blocks. In our setup, the model must accurately select blocks with spatial poses corresponding to those in the target image to ensure successful assembly. However, existing models tend to prioritize color and block type during selection while neglecting precise spatial alignment. This discrepancy leads to incorrect placements during assembly, compromising structural stability and overall task performance.

E Known Limitations and Future Directions of PhyBlock

E.1 Limited Inclusion of VLA and Affordance-Centric Models

While our benchmark comprehensively evaluates 25 of the most powerful vision-language models (VLMs) to date, it does not yet include a systematic assessment of emerging Vision-Language-Action (VLA) models and affordance-centric architectures. This omission is primarily due to the current limitations of these models in performing our 3D block assembly task under a strict zero-shot setting. Nevertheless, we acknowledge the critical importance of these model families in embodied reasoning and real-world interaction. At this stage, we prioritize enhancing the reasoning capabilities of VLM-based models on our task, establishing a strong foundation upon which our evaluation and experimentation can be progressively extended to VLA and affordance-centric models.

E.2 Limitations in 3D Spatial Coverage and Viewpoint Diversity

To assess the model's understanding of physical spatial reasoning in 3D block assembly, we design a series of VQA tasks targeting key dimensions such as **Counting**, **Rotation**, **Viewpoint**, **Ordering**, and **Affordance**—all closely tied to 3D perception and reasoning. While these tasks aim to comprehensively reflect the model's 3D reasoning and planning capabilities, we acknowledge that our current multi-view setting is limited to four canonical views: front, side, top, and oblique. Although this already distinguishes our benchmark from traditional 2D reasoning tasks, it still falls short for research specifically focused on 3D-awareness. In future work, we plan to incorporate a broader range of viewpoint relationships to better approximate complex and diverse 3D environments.

E.3 Dataset Scale, Augmentation Potential, and Future Expansion

Our evaluation dataset comprises 400 distinct 3D block assembly scenarios, from which we construct 2,200 high-quality VQA samples. Notably, thanks to detailed annotations of 150 core scenes—each capturing rich inter-block dependencies—we can readily scale the dataset to millions of configurations through systematic augmentations such as recombination, mirroring, and rotation.

On one hand, we believe that the 400 curated scenarios already cover a broad spectrum of spatial reasoning challenges encountered in 3D block assembly tasks, providing a strong foundation for benchmarking key model capabilities. On the other hand, we have explored more challenging levels involving deeper and denser structures composed of more blocks. Our preliminary experiments on such levels using ***GPT-o1*** reveal a substantial drop in performance, indicating that current models are not yet robust to increased structural complexity.

Therefore, we argue that our current set of 400+ evaluation scenarios is sufficient to probe critical reasoning bottlenecks. Nevertheless, in future iterations of PhyBlock, we plan to extend the scenario set at scale, enabling partitioning into training and evaluation subsets and facilitating the inclusion of fine-tuned models to further advance task-specific performance.

E.4 Pose Estimation as a Bottleneck for 3D Assembly Tasks

Our 3D block assembly task is designed to require the model to identify blocks from a candidate set that match the type and orientation of those shown in the target image, and to infer the correct sequence of assembly steps. While this process is relatively straightforward for humans, it remains highly challenging for current vision-language models (VLMs).

A natural question arises: why does our task not require models to predict the exact 3D pose (i.e., position and orientation) of each block in space? In fact, we initially considered this more demanding setting when designing the benchmark. However, through extensive pilot experiments with models such as the GPT and Claude series, we found that current VLMs still struggle significantly with

accurate 3D spatial reasoning. Their inability to predict precise poses results in zero completion rates for all block assembly tasks that require pose-level precision, which represents a major performance bottleneck.

Due to this limitation, we opted to simplify the task setting, focusing on type, orientation, and order reasoning while deferring exact pose prediction. Nevertheless, we consider 3D pose estimation a critical frontier and plan to extend our task in future work to include fine-grained pose reasoning.

E.5 Scope of Evaluation and Sim-to-Real Considerations

PhyBlock is intentionally designed to evaluate two core competencies of Vision-Language Models: (i) physical and spatial perception, and (ii) high-level assembly planning. Our evaluation setup does not currently include real-robot experiments. Incorporating real-world manipulation would introduce additional challenges, such as grasp synthesis, trajectory optimization, calibration, and hardware reliability. These factors could obscure the benchmark's diagnostic clarity, making it difficult to determine whether performance limitations stem from reasoning or from actuation. By isolating perception and planning within a physics-accurate simulator, PhyBlock provides precise and reproducible measurements of a model's reasoning abilities. Nonetheless, we recognize that bridging the sim-to-real gap remains an important future direction. Notably, the action sequences generated in Genesis are already compatible with standard robotic manipulation stacks (e.g., MoveIt and Cartesian impedance controllers), facilitating potential transfer to real-world robotic systems.

F Ethics Statement

Our study focuses on the development and evaluation of a 3D block assembly benchmark (PhyBlock) designed to assess the spatial reasoning and planning capabilities of vision-language models (VLMs). All data used in this benchmark, including rendered scenes and associated VQA questions, were synthetically generated without involving human subjects, sensitive personal data, or real-world environments.

To ensure the transparency and reproducibility of our research, we will make the dataset, benchmark suite, and evaluation protocols publicly available under an appropriate open license. We commit to following best practices in responsible dataset sharing and algorithmic evaluation to support the community in further research while minimizing potential misuse.

We believe this work adheres to the ethical standards outlined by NeurIPS and contributes positively to the development of interpretable and robust multimodal AI systems.