CHAIN of Consequences: A 3D Interactive Benchmark for Constraint-Aware Physical Planning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Physical intelligence requires more than recognizing the current scene or predicting isolated outcomes: agents must choose actions whose physical consequences determine what can be done next. Placing a large block too early, for example, can leave gaps that no remaining piece can fill. We study this ability as *constraint-aware sequential planning*: anticipating how each action reshapes which future actions remain feasible. Existing evaluations largely target physical perception, physical prediction, or end-to-end embodied execution, and rarely test this ability directly. We introduce **C**ausal **H**ierarchy of **A**ctions and **In**teractions (**CHAIN**), an interactive, physics-driven 3D benchmark in which models act through a fixed high-level action API and observe the physical consequences after every step. Its 200 task instances span five task families organized by three action–state coupling dimensions: constraint gating, spatial commitment, and structural propagation. Across 14 multimodal models, even the strongest solves fewer than half of the tasks (55.8% macro Pass@1). Without step-by-step interaction, top models drop to about 2% success, and longer histories or larger step budgets do not help. Current models can repair mistakes once feedback exposes them, but rarely foresee how an early action restricts what comes later.

1 Introduction

Physical intelligence is fundamentally about how actions reshape future feasibility [Sadhu et al., 2025, Lin et al., 2026, Wang et al., 2025]. An agent must not only recognize the current state or predict isolated outcomes, but also choose actions whose physical consequences determine what can be done next. An early placement may create support, use up limited space, block later access, or trigger a chain of contacts that propagates through the scene [Yang et al., 2025, Li et al., 2024b]. The key competence is therefore *constraint-aware sequential planning*: anticipating how each action changes what remains possible. Recent evidence suggests that current models are weak at this. VLMs struggle to produce multi-step plans even when perception is nearly perfect [He et al., 2025]; reasoning models break down on structured constraint-satisfaction problems beyond modest scales [Valmeekam et al., 2023]; and diffusion-based world models often generalize from cases rather than abstract the underlying rules [Kang et al., 2025, Meng et al., 2025].

Existing evaluations, however, do not directly diagnose this weakness (Fig. 1, left). Perception benchmarks ask what is true now, and prediction benchmarks ask what will happen under observed dynamics [Chow et al., 2025, Kun Xiang, 2025, Wang et al., 2023, Chen et al., 2022, Ma et al., 2025, Yi et al., 2019, Baradel et al., 2020, Riochet et al., 2018, Zheng et al., 2024, Bansal et al., 2024]; in neither case does the model act. Embodied benchmarks do require action, but they evaluate complete agents end to end, so planning errors are entangled with perception and execution errors [Li et al., 2024a, James et al., 2020, Gu et al., 2023]. As a result, current protocols cannot tell whether a model can choose an action, observe its physical consequence, and update which future actions remain

Submitted to 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Do not distribute.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

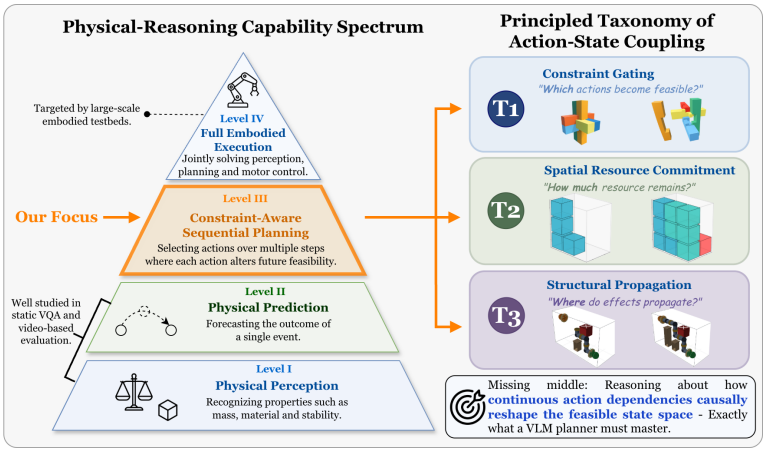


Figure 1: (Left) The physical-reasoning capability spectrum: **(I) Physical Perception**, **(II) Physical Prediction**, and **(III) Constraint-Aware Sequential Planning**, the setting our benchmark targets. (Right) Three action–state coupling dimensions that characterize how an action reshapes future feasibility: **constraint gating**, **spatial commitment**, and **structural propagation**.

feasible. This leads to our central question: *where and why does planning under physical constraints fail?*

To answer this question, we introduce **CHAIN** (**C**ausal **H**ierarchy of **A**ctions and **In**teractions), an interactive, physics-driven 3D benchmark for constraint-aware sequential planning. Models interact in a closed loop: they act through a fixed high-level action API and receive updated multi-view observations after every step. The API removes low-level motor control, so failures do not stem from grasping or trajectory errors, while the model must still infer the physical state from what it sees. The benchmark is organized around three action–state coupling dimensions (Fig. 1, right): **constraint gating**, where an action unlocks or blocks later actions; **spatial commitment**, where an action uses up limited space; and **structural propagation**, where a local action produces non-local physical effects. We instantiate these dimensions in **200 task instances** across five 3D task families: Puzzle, Stacking, FluidNet, PipeFit, and Domino Chain Reaction.

Our evaluation of 14 closed- and open-source multimodal models shows that **CHAIN** is far from solved. The strongest model, Gemini-3-Pro, reaches only 55.8% macro Pass@1 and leaves more than half of the 200 tasks unsolved, and the best open-source model reaches 37.0%. The per-family results reveal distinct failure patterns across the three coupling dimensions. On *Puzzle* (constraint gating), most models solve at most one of 32 instances, because they cannot identify the first move that unlocks the structure. On *Stacking* (spatial commitment), they make early placements that fragment the remaining space. On the propagation-oriented families (*FluidNet*, *PipeFit*, and *Domino*), top models succeed more often, but their remaining errors come from mispredicted non-local effects.

Further analyses show where these failures come from. Closed-loop interaction is essential: when models must produce a complete solution in one shot, the average success of three top models falls from about 50% to about 2%, and to 0% on the propagation-oriented families. Diagnostic controls on Stacking show that these failures are not explained by the short interaction history or the step budget; explicit symbolic state helps substantially but still leaves about half of the tasks unsolved. A qualitative stress test of generative video models on the same puzzles reveals similar breakdowns in structure and physical consistency. Overall, current models remain far from robust constraint-aware sequential planning: they can repair mistakes once feedback exposes them, but rarely foresee how an early action restricts what comes later.

Our contributions are:

1. We introduce **CHAIN**, an interactive 3D benchmark with **200 task instances** for constraint-aware sequential planning, bridging static physical QA and full embodied execution.
2. We organize the benchmark around **three recurring action–state coupling dimensions**, which provide a structured way to diagnose distinct planning failures.

3. We evaluate 14 multimodal models under a unified closed-loop protocol, complemented by one-shot, diagnostic-control, and video-model stress-test analyses, showing that even the strongest model leaves more than half of the tasks unsolved and that failures concentrate on early commitments and hidden constraints.

2 CHAIN Benchmark

This section describes **CHAIN**: the action–state coupling dimensions and task families (Section 2.1), the construction pipeline (Section 2.2), and the evaluation metrics (Section 2.3).

2.1 Task Overview

Task taxonomy. **CHAIN** is organized around three action–state coupling dimensions, each describing one way an action reshapes future feasibility. In **Constraint Gating** (T1), an action unlocks or blocks later actions, so the agent must find the prerequisite moves that make progress possible. In **Spatial Commitment** (T2), an action uses up limited space and is costly or impossible to undo, so early placements constrain every later one. In **Structural Propagation** (T3), a local action produces non-local physical effects through a connected structure, which may cause delayed success or failure. These dimensions are recurring and operationalized for diagnosis, not an exhaustive or orthogonal ontology; axes such as partial observability or tool use can co-occur with any of them. Since real planning tasks rarely involve a single coupling dimension in isolation, we complement single-dimension families with two mixed-dimension families to study their interaction. Based on these dimensions, **CHAIN** contains **200 task instances** across five task families: **32** Puzzle, **78** Stacking, and **30** instances each for FluidNet, PipeFit, and Domino, comprising three single-dimension families and two mixed-dimension families (for mixed families, the dominant dimension is listed first).

Puzzle (T1). Inspired by classical interlocking designs (*Kongming/Lu Ban locks*, burr puzzles), the agent assembles or disassembles multi-piece **mortise-and-tenon**[1] structures via pick/rotate/insert, where progress depends on executing steps in the correct order under kinematic and collision constraints. We manually curate **32 instances** (10 easy / 12 medium / 10 hard); success requires exact match to the target configuration.

Stacking (T2). The agent places multiple polycube blocks into a fixed container via select/rotate/place, where early placements irreversibly restrict the remaining free space. We programmatically generate and verify **78 instances** (Appendix A.4) with controlled container dimensions, block shapes, and difficulty; success requires exact volume coverage without overlap or gaps.

FluidNet 3D (T3). In a 3D pipe network, the agent opens one irreversible valve per step; fluid then propagates through connected sources, pipes, valves, and targets, and contact with protected cells causes failure. The challenge is to mentally simulate the global consequence of a local valve decision. We construct **30 instances** from network templates with scripted instantiation and verification; success requires filling all targets while avoiding protected regions.

PipeFit 3D (T2 $\times$ T3). The agent places a limited inventory of straight/elbow/T-junction pipes plus supports to connect entrances to exits in a 3D grid, where success demands both global connector-aligned reachability and local anchoring—so a locally plausible placement can break connectivity or support propagation. We build **30 instances** covering single/multi-source and obstacle settings.

Domino Chain Reaction 3D (T3 $\times$ T2). The agent places a small set of patch pieces, then triggers a push that launches a chain reaction; ramps, splitters, and placed patches determine how the cascade travels. The goal is to topple all designated pieces while avoiding protected ones. We create **30 instances** spanning straight chains, near-misses, ramps, splitters, and mixed configurations[2].

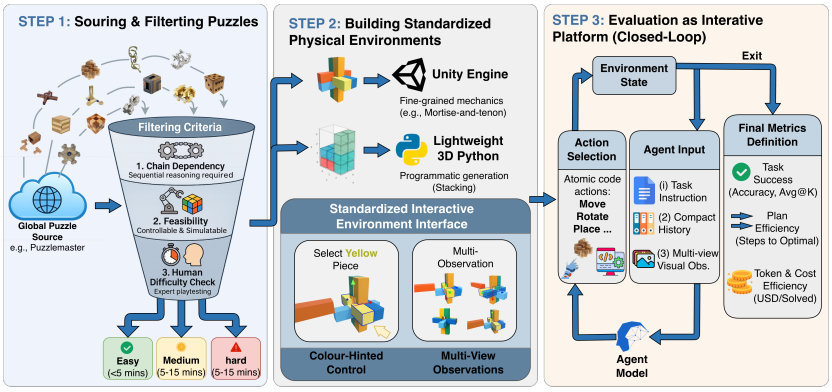


Figure 2: **Benchmark construction pipeline.** We illustrate the end-to-end process for building our benchmark, including task sourcing, manual design, programmatic generation, difficulty calibration, environment construction, and final evaluation setup. The pipeline is designed to ensure controlled difficulty, minimize parametric leakage, and enable fine-grained analysis of planning behavior.

2.2 Construction Pipeline

As illustrated in Fig. 2, we build **CHAIN** through three stages: task selection and generation, physical-environment construction, and closed-loop evaluation setup.

Step 1: Selecting and Generating Task Instances. We construct task instances through two modes. For interlocking puzzles, we manually curate candidates from classical puzzle designs and Puzzlemaster[3], then model and test them as interactive 3D environments. For stacking, FluidNet, PipeFit, and Domino, we use a hybrid process: humans specify task motifs, constraints, and difficulty settings, while scripts generate concrete layouts and verify validity. Across all task families, we apply three shared criteria: **action dependency**, requiring ordered or causally linked decisions; **simulation feasibility**, requiring stable state transitions and a controllable action space; and **difficulty calibration**. Human experts[4] solve representative instances to calibrate difficulty tiers as *easy* (< 5 minutes), *medium* (5–15 minutes), or *hard* (> 15 minutes).

Step 2: Building Physical Environments. To ensure experimental consistency, we unify the collected and generated tasks, spanning diverse structures and dynamics, into standardized interactive environments. We implement these environments using two complementary toolchains: **Unity** and **Python task environments with PyBullet physics backends**. Unity is used for puzzles with complex interlocking mechanics (e.g., Kongming and Lu Ban locks), where precise control over kinematic constraints and contact interactions is required. In contrast, the voxel packing, pipe-connection, flow-propagation, and domino-cascade tasks implement task logic, interaction, and verification in Python on top of PyBullet, for greater development efficiency and easier rule-based verification. To provide a uniform interaction interface across tasks, we adopt a **color-hinted control scheme**: each object is assigned a distinct color, and the color–object mapping is exposed to the agent as additional metadata. The agent specifies objects by color when selecting and manipulating pieces. This design avoids introducing an additional low-level controller that could confound the evaluation. Finally, we provide **multi-view observations**, allowing agents to inspect the scene from different viewpoints and reducing failures due to occlusion.

Step 3: Closed-Loop Evaluation Setup. We evaluate each instance under a closed-loop protocol rather than a single-turn question answering setting. For each episode, we initialize one task instance to its predefined state, including object poses, constraints, and the fixed action set. At step t, the

[1] A traditional interlocking joint that connects parts without nails or adhesives.

[2] Additional examples and generation details in Appendix A.2.

[3] `https://www.puzzlemaster.ca/`

[4] We recruit two human experts, who are administrators of the corresponding puzzle interest groups in the forum. Each expert independently solves sampled calibration instances; we record their completion times and use the average solve time to assign the difficulty label. This calibration step serves only to assign difficulty tiers and is separate from the human-baseline study in Appendix A.6.

agent receives (i) a **task instruction** specifying the goal state, (ii) a compact **interaction history** summarizing prior observations and actions, and (iii) the current **multi-view visual observations**. The agent selects an action from the predefined action space; the simulator executes it, updates the environment state, and returns observations for step $t+1$. Infeasible actions (e.g., blocked by contact) instead return execution-failure feedback for replanning. The loop continues until the instance is solved or a predefined **step budget** is reached. After the episode ends, we compute metrics from the full trajectory, including final success, action efficiency, and token efficiency.

2.3 Metric

We report three metrics covering whether the agent solves the task, how efficiently it acts, and how much interaction context it consumes. Let the benchmark contain $F = 5$ families with sizes $N_f \in \{32, 78, 30, 30, 30\}$, $s_{f,i} \in \{0, 1\}$ the per-instance success, $\mathrm{Steps}_{f,i}$ the number of executed actions, $o_{f,i}$ the task-specific minimal plan length, and $\mathcal{S} = \{(f, i) : s_{f,i} = 1\}$ the set of solved instances; $\mathbb{I}(\cdot)$ is the indicator function.

(1) Task success. We report **Pass@1** as a *macro-average across task families*, ensuring that no single task family dominates the results regardless of its scale. We also report **Succ.Task**, which denotes the raw count of solved instances.

$$\mathrm{Pass@1}_{\mathrm{macro}} = \frac{1}{F} \sum_{f=1}^{F} \frac{1}{N_f} \sum_{i=1}^{N_f} \mathbb{1} s_{f,i} = 1. \tag{1}$$

(2) Plan efficiency (on solved tasks). To disentangle efficiency from task difficulty, we restrict plan-efficiency metrics to $\mathcal{S}$. **AvgSteps** is the mean of $\mathrm{steps}_{f,i}$ over $\mathcal{S}$. **Normalized Distance** measures avoidable overhead relative to the task-specific optimum, making tasks of different optimal length comparable:

$$\mathrm{NormDist} = \frac{1}{|\mathcal{S}|} \sum_{(f,i)\in\mathcal{S}} \frac{\max(0,\ \mathrm{steps}_{f,i} - o_{f,i})}{\max(1,\ o_{f,i})}. \tag{2}$$

(3) Token efficiency. We count all billable tokens (prompts, tool I/O, replies) over the entire interaction and report **Solved/1M Tokens**: solved-instance count divided by total tokens, scaled by 10^6. This captures differences in interaction length, observation history, and response verbosity.

3 Experiments

3.1 Experimental Setup

We evaluate 14 closed- and open-source multimodal models under a unified decoding and interaction protocol. Concretely, our evaluated pool includes 1) closed-source models: *GPT-5.2* and *GPT-5-mini*[Singh et al., 2025], *Claude-Opus-4.5*[Anthropic, 2025a], *Claude-Sonnet-4.5*[Anthropic, 2025b], *Gemini-3-Pro (prev.)*[Google DeepMind, 2025b], *Gemini-3-Flash (prev.)*[Google DeepMind, 2025a], *Seed-1.6-Flash* and *Seed-1.6*[Seed, 2025]; 2) open-source models: *Qwen3-VL-235B-Ins*, *Qwen3-VL-30B-A3B-Ins*, *Qwen3-VL-235B-Thinking*, *Qwen3-VL-30B-A3B-Thinking*, *Qwen3-VL-8B-Thinking*[Bai et al., 2025], and *Kimi-k2.5*[Team et al., 2025]; details are provided in Appendix A.5.

We use identical generation hyperparameters across all models: temperature is fixed to 0.6, and nucleus sampling is set to top-$p = 0.95$. For each task instance, the model interacts with the environment through the same action API, and all atomic actions are executed by the system proxy to factor out low-level control and isolate operation-level decision making. We additionally cap the interaction budget at 30–60 steps per instance, which is substantially larger than the optimal solution length for all tasks, ensuring that failures are not driven by insufficient rollout horizon. Finally, we set the interaction history window to 5 (i.e., the model conditions on only the most recent five interaction turns when making each decision) (ablated in Section 3.5).

Models	Task Success							Plan Efficiency (Solved)		Token Efficiency
	Pass@1 ↑	Succ. ↑	Puzzle ↑	Stacking ↑	FluidNet ↑	PipeFit ↑	Domino ↑	Steps ↓	N.Dist ↓	Solv./Tok. ↑
Closed-source Models										
GPT-5.2	53.5	95	3.1	**31.2**	66.7	**86.7**	80.0	10.2	0.90	4.81
GPT-5-mini	36.8	62	3.1	14.3	60.0	23.3	83.3	8.1	1.06	2.65
Claude-Opus-4.5	40.8	71	3.1	20.8	70.0	26.7	83.3	8.6	1.13	6.04
Claude-Sonnet-4.5	38.9	67	3.1	18.2	63.3	36.7	73.3	9.1	1.13	6.09
Gemini-3-Pro	**55.8**	**96**	3.1	26.0	**80.0**	83.3	**86.7**	8.9	0.86	**7.06**
Gemini-3-Flash	29.1	51	3.1	15.6	60.0	16.7	50.0	6.7	0.82	6.97
Seed-1.6	36.5	61	3.1	13.0	63.3	23.3	80.0	7.5	0.93	4.29
Seed-1.6-Flash	20.7	36	0.0	10.4	43.3	3.3	46.7	5.8	0.78	1.02
Open-source Models										
Kimi-k2.5	7.2	21	3.1	19.5	16.7	0.0	0.0	5.8	**0.41**	0.53
Qwen3-235B-Thk	37.0	61	3.1	11.7	63.3	36.7	70.0	6.6	0.63	2.66
Qwen3-30B-Thk	30.8	53	0.0	14.3	63.3	13.3	63.3	6.4	0.75	1.49
Qwen3-8B-Thk	27.7	47	0.0	11.7	56.7	3.3	66.7	6.1	0.89	1.20
Qwen3-235B-Inst	19.4	34	3.1	10.4	53.3	0.0	30.0	4.9	0.77	3.34
Qwen3-30B-Inst	6.7	12	3.1	3.9	20.0	0.0	6.7	**3.5**	0.53	0.72

Table 1: Main results on 200 task instances. Pass@1 is the macro-average of family-level success rates; Succ. is the raw solved count; Steps and N.Dist are computed on solved tasks; Solv./Tok. is solved tasks per million tokens.

3.2 Main Experiment

Table 1 reports the performance of a broad set of closed- and open-source models on **CHAIN**. In addition to macro-averaged Pass@1 and raw solved-task counts, the table reports success on each of the five task families: *Puzzle* ($N = 32$), *Stacking* ($N = 78$), *FluidNet* ($N = 30$), *PipeFit* ($N = 30$), and *Domino* ($N = 30$), for a total of 200 task instances. The macro Pass@1 is computed by averaging the five family-level success rates, so each task family contributes equally to the overall score regardless of the number of instances. This per-family view aligns the empirical analysis with the three action–state coupling dimensions in Section 2.1: Constraint Gating, Spatial Commitment, Structural Propagation, and their mixed forms. Overall, closed-source models dominate the leaderboard: Gemini-3-Pro achieves the best macro-averaged result (Pass@1=55.8, Succ.Task=96), while GPT-5.2 remains strongest on *Stacking* and *PipeFit*. Among open-source models, Qwen3-VL-235B-A22B-Thinking is the strongest overall (Pass@1=37.0, Succ.Task=61). Despite strong performance on propagation-oriented tasks such as *FluidNet* and *Domino*, **CHAIN** remains challenging: *Puzzle* success is consistently far below the other families (most models reach only $\approx$ 0.0–3.1). This gap suggests that even state-of-the-art models struggle with structural inference and constrained, sequential manipulation in human-designed 3D interlocking puzzles. As human experts also solve only 25%/10% of Medium/Hard puzzles (Appendix A.6), we treat these tiers as a stress-test split with a VLM floor effect, not a ranking signal; likewise, small gaps between adjacent models should not be over-interpreted.

Higher success does not always imply better interaction efficiency. Stronger models often *backtrack and revise* their plans, which increases Avg-Steps and NormDist and consumes more tokens per solved task. In contrast, lightweight "flash" models can be token-efficient but achieve much lower success. To visualize the quality–efficiency trade-off, Fig. 3 plots token usage against overall success; points closer to the *upper-left* represent better value. Notably, in *easy* regimes, extremely small models are not always the most efficient choice: low success can inflate tokens per solved task, whereas a moderately stronger model may solve more reliably with fewer failed attempts.

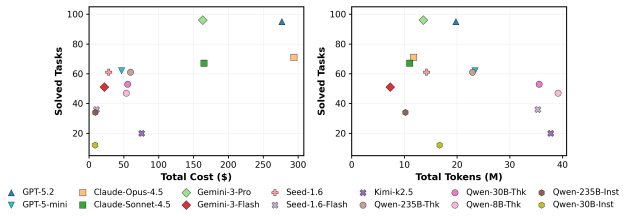
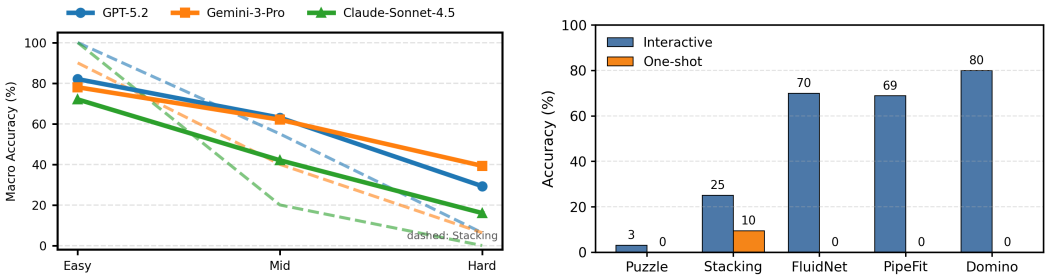


Figure 3: Estimated cost (left) and total tokens (right) vs. solved tasks. Cost is computed with OpenRouter completion (output) prices.

Finally, we analyze representative failure cases to pinpoint what most limits progress under different action–state couplings. For *Puzzle* (Constraint Gating), the primary obstacle is weak perception and utilization of *physical structure*: even when partial internal structure is revealed as a reference, many models still cannot reliably identify the *first critical move* that unlocks the puzzle. As a result, they



(a) Macro accuracy across Easy/Mid/Hard tiers.

(b) Interactive vs. one-shot accuracy per task family.

Figure 4: Visualization of difficulty effects and closed-loop interaction benefits on **CHAIN**. Both plots use representative top models. The difficulty plot reports Easy/Mid/Hard macro accuracy averaged over the five task families; dashed curves show the corresponding Stacking-only trend.

resort to extensive trial-and-error over candidate blocks, leading to largely random exploration with little constraint-guided progress. For *Stacking* (Spatial Commitment), two factors dominate failures. First is *object-set complexity and coupling*: as illustrated in Fig. 7, while easy instances can be solved with locally reasonable placements, mid and hard levels require tightly coupled packing decisions to maximize space utilization. Second is *global spatial planning*: most models greedily place "easy" items early, only to later discover that the remaining free volume is fragmented, forcing costly removals and replanning. The newly introduced propagation-oriented tasks expose a complementary failure mode: although success rates on *FluidNet*, *PipeFit*, and *Domino* are noticeably higher than on *Puzzle*/*Stacking* for top models, the residual errors are dominated by mispredicted *non-local consequences*—unsafe flow routing in *FluidNet*, support/connectivity violations in *PipeFit*, and broken or premature cascades in *Domino*. In short, failures on the propagation-oriented families mostly arise late in an otherwise workable plan, whereas failures on *Puzzle* and *Stacking* are rooted in early decisions: a missed unlocking move or a placement that silently rules out later ones. For a specific discussion of the factors contributing to success and failure, please refer to Appendix A.3.

3.3 Impact of Difficulty Stratification

Fig. 4a reports macro accuracy across the five task families on Easy/Mid/Hard for three representative top models. All three exhibit a clear monotonic degradation from Easy to Hard, indicating that higher difficulty consistently stresses constraint-aware sequential planning. The slope, however, differs across models: Gemini-3-Pro retains the strongest Hard-tier performance, GPT-5.2 starts highest on Easy but drops more on Hard, and Claude-Sonnet-4.5 degrades most sharply overall. The Stacking-only trend (dashed) drops faster than the macro trend, indicating that spatial-commitment difficulty escalates more aggressively with tier than the average across families. Together, these patterns suggest that difficulty in **CHAIN** is not a single uniform axis: tier transitions stress different action–state coupling abilities to different degrees across families[5].

3.4 One-shot Solving without Interaction

A key motivation for interactive evaluation is to distinguish *offline planning* from *closed-loop adaptation*. In physical problem solving, interaction can compensate for imperfect priors: a model may probe the environment, observe feasibility constraints, and revise its plan accordingly. To isolate how much success on **CHAIN** depends on such feedback, we construct a one-shot setting where the model receives only an initial fixed-view image and must output a complete solution with minimal or no intermediate feedback. We then compare one-shot accuracy with the interactive setting; the gap Δ (Interactive $-$ One-shot) quantifies the benefit brought by interaction.

Fig. 4b reports interactive vs. one-shot accuracy on each of the five task families, averaged over three representative top closed-source models (*GPT-5.2*, *Gemini-3-Pro*, and *Claude-Opus-4.5*). One-shot performance is uniformly lower than interactive across families, with the macro-averaged All score collapsing from roughly 50% in the interactive setting to about 2% in one-shot. The gap is most extreme on the propagation-oriented families: *FluidNet*, *PipeFit*, and *Domino* drop from 70%, 69%,

[5]Difficulty-level definitions are detailed in Appendix A.2.

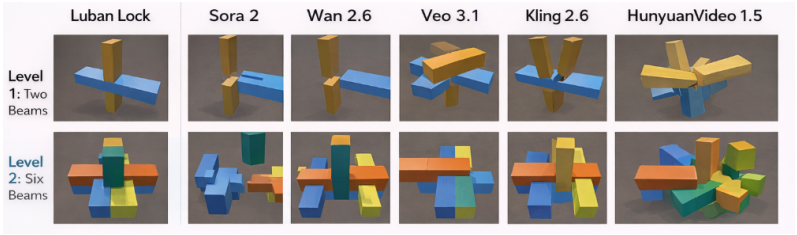


Figure 5: Qualitative results on the Luban puzzle subtask across world models. Top: **Level 1** (two beams). Bottom: **Level 2** (six beams). All models fail to produce a physically valid disassembly, either violating interlocking constraints or hallucinating (e.g., structural corruption and object insertion/removal), with failures worsening at higher complexity.

and 80% (interactive) all the way to 0% once feedback is removed, indicating that success on these tasks fundamentally relies on observing physical consequences after each action rather than on pre-computing the full plan. For *Puzzle*, one-shot accuracy collapses to 0% while interactive accuracy reaches only 3%, showing that even modest success requires iterative constraint discovery rather than a fully inferred disassembly plan. *Stacking* shows the smallest gap (25% $\rightarrow$ 10%), consistent with its more local placement structure, but interaction still nearly triples accuracy. Overall, success on **CHAIN** depends on observing consequences step by step rather than on one-shot recognition or static plan synthesis: models can repair mistakes once feedback exposes them, but rarely foresee how an early action restricts what comes later. Appendix A.9 compares RM reranking with verifier-based selection.

3.5 Diagnostic Controls: Memory, Budget, and Perception

We run matched controls with *Kimi-k2.5* on a fixed 77-task Stacking subset (Appendix A.8). **Memory and budget.** Full history or a 100-step budget gives no consistent gain over the 5-turn/50-step default (Pass@1 14.3–18.2% vs. 16.9%, overlapping 95% CIs); with 100 steps, the model stops early rather than exhausting the budget. **Perception.** Symbolic coordinates instead of images (no legal actions, constraint graph, or solution) raise Pass@1 from 20.8% to 50.6% and cut illegal actions from 72.6% to 15.7%, so visual state understanding matters; yet 49.4% of tasks (70.2% of Hard) stay unsolved, so a planning bottleneck persists. **Stability.** Two runs agree on 93.5% of tasks; the CI of their difference includes zero. These controls cover one model and one family.

3.6 Qualitative Stress Test of Generative Video Models

Beyond multimodal models acting through the action API, we use **CHAIN** as a stress test for the alternative paradigm of **generative world models** that aim to simulate physical futures directly in pixel space. We focus on the **puzzle subtask** and ask whether such models can render a **step-by-step disassembly** consistent with contact, interlocking, and collision constraints, rather than only producing visually plausible motion. We emphasize that this evaluation is intentionally *qualitative and complementary*: video generators do not consume our action API, so the comparison with VLM agents in Table 1 is not head-to-head but a probe into whether current world models exhibit object permanence and constraint consistency strong enough to plug into a downstream planner.

Concretely, given a reference image of a Luban lock, models are prompted to disassemble all pieces while respecting rigidity, no-interpenetration, axis-aligned sliding, and a realistic unlocking sequence (full prompt in Appendix A.11). We evaluate state-of-the-art video generation models (SORA 2 [Li et al., 2025], WAN 2.6 [Wan et al., 2025], VEO 3.1 [Wiedemer et al., 2025], KLING 2.6 [Kuaishou Technology, 2025], and HUNYUANVIDEO 1.5 [Team, 2025]) on two configurations: **Level 1** (two beams) and **Level 2** (six beams). Despite the highly specified prompts, none of the models completes the disassembly; instead, they exhibit systematic failures with severe hallucinations (Fig. 5), and these failures intensify with structural complexity. A stepwise variant, generating one disassembly step at a time, shows the same errors, so the failures are not solely an artifact of one-shot generation.

First, some models show **superficial instruction-following with physics violations**. SORA 2 and WAN 2.6 often extract a target beam by direct translation, even when such motion is physically infeasible due to interlocking. This pattern worsens at Level 2, where the models increasingly abandon the prescribed step-by-step procedure and produce random or under-specified motions

that remain physically invalid. Second, other models undergo a more severe **representational collapse**, losing consistency of both object structure and object identity. At Level 1, VEO 3.1, KLING 2.6, and HUNYUANVIDEO 1.5 already generate distorted geometry or spurious components. At Level 2, this escalates into full hallucination: beams are added, removed, or merged, and the puzzle is sometimes transformed into an unrecognizable structure—a clear breakdown of object permanence and constraint consistency under multi-part coupling. Overall, modern world models can generate visually plausible motion or simple event transitions, but remain **unreliable for structured, constraint-driven interaction** in this stress test, which we keep outside the VLM leaderboard. An important future direction is to turn world models into *usable forward dynamics modules* for VLM planners on **CHAIN**-style closed-loop tasks, rather than standalone solvers.

4 Related Work

Physical Understanding versus Applied Physical Reasoning. A long line of work probes whether models can *recognize* physical attributes and *predict* short-horizon outcomes via intuitive physics, commonsense QA, and synthetic compositional reasoning [Riochet et al., 2018, Rajani et al., 2020, Yi et al., 2019, Wang et al., 2024b, Zheng et al., 2024, Tung et al., 2023, Lu et al., 2022, He et al., 2024, Baradel et al., 2020]. Recent VLM benchmarks such as PhysBench [Chow et al., 2025] and CausalVQA [Foss et al., 2025] push toward anticipation and counterfactual reasoning, yet remain *observational*: the model never *intervenes*. **CHAIN** targets this missing axis—turning physical knowledge into closed-loop, constraint-aware decisions.

Interactive Benchmarks for Constraint-Aware Sequential Planning. Existing interactive benchmarks each leave a structural blind spot (six-axis comparison in Appendix A.1, Table 2): PHYRE [Bakhtin et al., 2019] and I-PHYRE [Li et al., 2024c] are 2D; CausalWorld [Ahmed et al., 2021] entangles planning with low-level RL control; DeepPHY [Xu et al., 2026] is largely 2D and is not organized by action–state coupling; PhyBlock [Ma et al., 2025] and ViPlan [Merler et al., 2026] are restricted to narrow regimes (stacking, or pick–place–navigate). **CHAIN** differs along all axes: fully 3D, closed-loop, low-level actuation factored out by a fixed action API, and organized by **three action–state coupling dimensions** (constraint gating, spatial commitment, structural propagation) instantiated by five task families.

Constraint-Aware Sequential Planning for Predictive Action. Parallel work asks how such planning is *built into* VLA pipelines: 3D grounding and spatial perception [Chen et al., 2024, Lyu et al., 2024, Wang et al., 2024a, Zhao et al., 2025, Lu et al., 2025, Ma et al., 2026, Wang et al., 2026], reachability-augmented VLMs [Zhou et al., 2025], and reasoning-enhanced VLA policies via grounded thought, visual goals, or latent planning [Zawalski et al., 2025, Zhao et al., 2025, Huang et al., 2025, Driess et al., 2025], but they are evaluated on static reconstruction or one-shot inference. What remains untested is whether models can maintain and update an implicit physical state across multi-step chains where each action depends on the consequences of prior ones—precisely the regime **CHAIN** targets.

5 Conclusion

We introduced **CHAIN**, an interactive 3D benchmark for *constraint-aware sequential planning* organized by three action–state coupling dimensions. Across 200 task instances over five families, even the strongest of 14 multimodal models leaves more than half unsolved, with distinct failures per dimension: missed unlocking moves (Puzzle), early placements that fragment the remaining space (Stacking), and mispredicted non-local effects (FluidNet/PipeFit/Domino). Removing closed-loop interaction collapses performance, while longer interaction histories and larger step budgets do not help. Current models can repair mistakes once feedback exposes them, but rarely foresee how an early action restricts what comes later; **CHAIN** offers a controlled setting for measuring and closing this gap.

Broader Impact Statement

This paper presents work whose goal is to advance machine learning by enabling more rigorous evaluation of interactive, physics-grounded vision reasoning through an open benchmark and reproducible baselines. There are many potential societal consequences of improved interactive perception-and-action systems, but we do not anticipate any direct negative impacts from releasing this evaluation benchmark beyond standard concerns for open research artifacts.

References

Ossama Ahmed, Frederik Träuble, Anirudh Goyal, Alexander Neitz, Manuel Wüthrich, Yoshua Bengio, Bernhard Schölkopf, and Stefan Bauer. CausalWorld: A robotic manipulation benchmark for causal structure and transfer learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=Q2iaAc-4I1v.

Anthropic. Claude opus 4.5 system card. Technical report, Anthropic, November 2025a. URL https://www.anthropic.com/claude-opus-4-5-system-card.

Anthropic. Claude sonnet 4.5 system card. Technical report, Anthropic, October 2025b. URL https://www.anthropic.com/claude-sonnet-4-5-system-card.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025. URL https://arxiv.org/abs/2511.21631.

Anton Bakhtin, Laurens van der Maaten, Justin Johnson, Laura Gustafson, and Ross Girshick. PHYRE: A new benchmark for physical reasoning. In *Advances in Neural Information Processing Systems*, 2019. URL https://papers.neurips.cc/paper/2019/hash/4191ef5f6c1576762869ac49281130c9-Abstract.html.

Hritik Bansal, Zongyu Lin, Tianyi Xie, Zeshun Zong, Michal Yarom, Yonatan Bitton, Chenfanfu Jiang, Yizhou Sun, Kai-Wei Chang, and Aditya Grover. VideoPhy: Evaluating physical commonsense for video generation. *arXiv preprint arXiv:2406.03520*, 2024. URL https://arxiv.org/abs/2406.03520.

Fabien Baradel, Natalia Neverova, Julien Mille, Greg Mori, and Christian Wolf. Cophy: Counterfactual learning of physical dynamics. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SkeyppEFvS.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14455–14465, June 2024.

Zhenfang Chen, Kexin Yi, Yunzhu Li, Mingyu Ding, Antonio Torralba, Joshua B. Tenenbaum, and Chuang Gan. Comphy: Compositional physical reasoning of objects and events from videos. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=PgNEYaIc81Q.

Wei Chow, Jiageng Mao, Boyi Li, Daniel Seita, Vitor Guizilini, and Yue Wang. Physbench: Benchmarking and enhancing vision-language models for physical world understanding, 2025. URL https://arxiv.org/abs/2501.16411.

Danny Driess, Jost Tobias Springenberg, Brian Ichter, Lili Yu, Adrian Li-Bell, Karl Pertsch, Allen Z. Ren, Homer Walke, Quan Vuong, Lucy Xiaoyang Shi, and Sergey Levine. Knowledge insulating vision-language-action models: Train fast, run fast, generalize better, 2025. URL https://arxiv.org/abs/2505.23705.

Aaron Foss, Chloe Evans, Sasha Mitts, Koustuv Sinha, Ammar Rizvi, and Justine T. Kao. Causalvqa: A physically grounded causal reasoning benchmark for video models, 2025. URL https://arxiv.org/abs/2506.09943.

Google DeepMind. Gemini 3 flash: Model card. Technical report, Google DeepMind, December 2025a. URL https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-Flash-Model-Card.pdf.

Google DeepMind. Gemini 3 pro: Model card. Technical report, Google DeepMind, November 2025b. URL https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-Pro-Model-Card.pdf.

Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, Xiaodi Yuan, Pengwei Xie, Zhiao Huang, Rui Chen, and Hao Su. Maniskill2: A unified benchmark for generalizable manipulation skills, 2023. URL https://arxiv.org/abs/2302.04659.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv*, 2024.

Muyu He, Yuxi Zheng, Yuchen Liu, Zijian An, Bill Cai, Jiani Huang, Lifeng Zhou, Feng Liu, Ziyang Li, and Li Zhang. Vision language models cannot plan, but can they formalize? *arXiv preprint arXiv:2509.21576*, 2025. URL https://arxiv.org/abs/2509.21576. Under review.

Chi-Pin Huang, Yueh-Hua Wu, Min-Hung Chen, Yu-Chiang Frank Wang, and Fu-En Yang. Thinkact: Vision-language-action reasoning via reinforced visual latent planning, 2025. URL https://arxiv.org/abs/2507.16815.

Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J. Davison. RLBench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 2020. doi: 10.1109/LRA.2020.2974707. URL https://arxiv.org/abs/1909.12271.

Bingyi Kang, Yang Yue, Rui Lu, Zhijie Lin, Yang Zhao, Kaixin Wang, Gao Huang, and Jiashi Feng. How far is video generation from world model: A physical law perspective. In *Proceedings of the International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=DLlVjZQ7vD.

Kuaishou Technology. Kling ai 2.6: Next-generation video generation model. https://klingai.com/, dec 2025. Accessed: 2026-01-28.

Terry Jingchen Zhang Yinya Huang Zirong Liu Peixin Qu Jixi He Jiaqi Chen Yu-Jie Yuan Jianhua Han Hang Xu Hanhui Li Mrinmaya Sachan Xiaodan Liang Kun Xiang, Heng Li. Seephys: Does seeing help thinking?–benchmarking vision-based physics reasoning. *arXiv preprint arXiv:2505.19099*, 2025.

Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Wensi Ai, Benjamin Martinez, Hang Yin, Michael Lingelbach, Minjune Hwang, Ayano Hiranaka, Sujay Garlanka, Arman Aydin, Sharon Lee, Jiankai Sun, Mona Anvari, Manasi Sharma, Dhruva Bansal, Samuel Hunter, Kyu-Young Kim, Alan Lou, Caleb R Matthews, Ivan Villa-Renteria, Jerry Huayang Tang, Claire Tang, Fei Xia, Yunzhu Li, Silvio Savarese, Hyowon Gweon, C. Karen Liu, Jiajun Wu, and Li Fei-Fei. Behavior-1k: A human-centered, embodied ai benchmark with 1,000 everyday activities and realistic simulation, 2024a. URL https://arxiv.org/abs/2403.09227.

Harold Li, Dmytro Okhonko, Avi Verma, Eric Zhang, Ricky Wang, Troy Luhman, Eric Luhman, Bram Wallace, Eric Mintun, Michael Chang, et al. Sora 2. https://openai.com/zh-Hans-CN/index/sora-2/, sep 2025. Full author list includes the OpenAI Sora 2 Research and Product teams. Accessed: 2026-01-28.

Manling Li, Shiyu Zhao, Qineng Wang, Kangrui Wang, Yu Zhou, Sanjana Srivastava, Cem Gokmen, Tony Lee, Li Erran Li, Ruohan Zhang, Weiyu Liu, Percy Liang, Li Fei-Fei, Jiayuan Mao, and Jiajun Wu. Embodied agent interface: Benchmarking LLMs for embodied decision making. In *Advances in Neural Information Processing Systems*, 2024b. URL https://openreview.net/forum?id=iSwK1YqO7v. Datasets and Benchmarks Track.

Shiqian Li, Kewen Wu, Chi Zhang, and Yixin Zhu. I-PHYRE: Interactive physical reasoning. In *International Conference on Learning Representations*, 2024c. URL https://openreview.net/forum?id=1bbPQShCT2.

Xiaopeng Lin, Shijie Lian, Bin Yu, Ruoqi Yang, Zhaolong Shen, Changti Wu, Yuzhuo Miao, Yurun Jin, Yukun Shi, Jiyan He, Cong Huang, Bojun Cheng, and Kai Chen. Physbrain: Human egocentric data as a bridge from vision language models to physical intelligence, 2026. URL https://arxiv.org/abs/2512.16793.

Meng Lu, Ran Xu, Yi Fang, Wenxuan Zhang, Yue Yu, Gaurav Srivastava, Yuchen Zhuang, Mohamed Elhoseiny, Charles Fleming, Carl Yang, Zhengzhong Tu, Yang Xie, Guanghua Xiao, Hanrui Wang, Di Jin, Wenqi Shi, and Xuan Wang. Scaling agentic reinforcement learning for tool-integrated reasoning in vlms, 2025. URL https://arxiv.org/abs/2511.19773.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521, 2022.

Ruiyuan Lyu, Tai Wang, Jingli Lin, Shuai Yang, Xiaohan Mao, Yilun Chen, Runsen Xu, Haifeng Huang, Chenming Zhu, Dahua Lin, et al. Mmscan: A multi-modal 3d scene dataset with hierarchical grounded language annotations. *arXiv*, 2024.

Liang Ma, Jiajun Wen, Min Lin, Rongtao Xu, Xiwen Liang, Bingqian Lin, Jun Ma, Yongxin Wang, Ziming Wei, Haokun Lin, Mingfei Han, Meng Cao, Bokui Chen, Ivan Laptev, and Xiaodan Liang. Phyblock: A progressive benchmark for physical understanding and planning via 3D block assembly. *arXiv preprint arXiv:2506.08708*, 2025. URL https://arxiv.org/abs/2506.08708.

Wenxin Ma, Chenlong Wang, Ruisheng Yuan, Hao Chen, Nanru Dai, S. Kevin Zhou, Yijun Yang, Alan Yuille, and Jieneng Chen. Causalspatial: A benchmark for object-centric causal spatial reasoning, 2026. URL https://arxiv.org/abs/2601.13304.

Fanqing Meng, Jiaqi Liao, Xinyu Tan, Quanfeng Lu, Wenqi Shao, Kaipeng Zhang, Yu Cheng, Dianqi Li, and Ping Luo. Towards world simulator: Crafting physical commonsense-based benchmark for video generation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 43781–43806. PMLR, 2025. URL https://proceedings.mlr.press/v267/meng25c.html.

Matteo Merler, Nicola Dainese, Minttu Alakuijala, Giovanni Bonetta, Pietro Ferrazzi, Yu Tian, Bernardo Magnini, and Pekka Marttinen. Viplan: A benchmark for visual planning with symbolic predicates and vision-language models, 2026. URL https://arxiv.org/abs/2505.13180.

Brandon Ong, Tej Deep Pala, Vernon Toh, William Chandra Tjhi, and Soujanya Poria. Training vision-language process reward models for test-time scaling in multimodal reasoning: Key insights and lessons learned, 2025. URL https://arxiv.org/abs/2509.23250.

Nazneen Fatema Rajani, Rui Zhang, Yi Chern Tan, Stephan Zheng, Jeremy Weiss, Aadit Vyas, Abhijit Gupta, Caiming Xiong, Richard Socher, and Dragomir Radev. Esprit: Explaining solutions to physical reasoning tasks. *arXiv*, 2020.

Ronan Riochet, Mario Ynocente Castro, Mathieu Bernard, Adam Lerer, Rob Fergus, Véronique Izard, and Emmanuel Dupoux. Intphys: A framework and benchmark for visual intuitive physics reasoning. *arXiv*, 2018.

Tanmana Sadhu, Yanan Chen, and Ali Pesaranghader. VestaBench: An embodied benchmark for safe long-horizon planning under multi-constraint and adversarial settings. In Saloni Potdar, Lina Rojas-Barahona, and Sebastien Montella, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 2122–2145, Suzhou (China), November 2025. Association for Computational Linguistics. ISBN 979-8-89176-333-3. doi: 10.18653/v1/2025.emnlp-industry.149. URL https://aclanthology.org/2025.emnlp-industry.149/.

ByteDance Seed. Seed1. 6 tech introduction. *Accessed on September*, 28:2025, 2025.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, Akshay Nathan, Alan Luo, Alec Helyar, Aleksander Madry, Aleksandr Efremov, Aleksandra Spyra, Alex Baker-Whitcomb, Alex Beutel, Alex Karpenko, Alex Makelov, Alex Neitz, Alex Wei, Alexandra Barr, Alexandre Kirchmeyer, Alexey Ivanov, Alexi Christakis, Alistair Gillespie, Allison Tam, Ally Bennett, Alvin Wan, Alyssa Huang, Amy McDonald Sandjideh, Amy Yang, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrei Gheorghe, Andres Garcia Garcia, Andrew Braunstein, Andrew Liu, Andrew Schmidt, Andrey Mereskin, Andrey Mishchenko, Andy Applebaum, Andy Rogerson, Ann Rajan, Annie Wei, Anoop Kotha, Anubha Srivastava, Anushree Agrawal, Arun Vijayvergiya, Ashley Tyra, Ashvin Nair, Avi Nayak, Ben Eggers, Bessie Ji, Beth Hoover, Bill Chen, Blair Chen, Boaz Barak, Borys Minaiev, Botao Hao, Bowen Baker, Brad Lightcap, Brandon McKinzie, Brandon Wang, Brendan Quinn, Brian Fioca, Brian Hsu, Brian Yang, Brian Yu, Brian Zhang, Brittany Brenner, Callie Riggins Zetino, Cameron Raymond, Camillo Lugaresi, Carolina Paz, Cary Hudson, Cedric Whitney, Chak Li, Charles Chen, Charlotte Cole, Chelsea Voss, Chen Ding, Chen Shen, Chengdu Huang, Chris Colby, Chris Hallacy, Chris Koch, Chris Lu, Christina Kaplan, Christina Kim, CJ Minott-Henriques, Cliff Frey, Cody Yu, Coley Czarnecki, Colin Reid, Colin Wei, Cory Decareaux, Cristina Scheau, Cyril Zhang, Cyrus Forbes, Da Tang, Dakota Goldberg, Dan Roberts, Dana Palmie, Daniel Kappler, Daniel Levine, Daniel Wright, Dave Leo, David Lin, David Robinson, Declan Grabb, Derek Chen, Derek Lim, Derek Salama, Dibya Bhattacharjee, Dimitris Tsipras, Dinghua Li, Dingli Yu, DJ Strouse, Drew Williams, Dylan Hunn, Ed Bayes, Edwin Arbus, Ekin Akyurek, Elaine Ya Le, Elana Widmann, Eli Yani, Elizabeth Proehl, Enis Sert, Enoch Cheung, Eri Schwartz, Eric Han, Eric Jiang, Eric Mitchell, Eric Sigler, Eric Wallace, Erik Ritter, Erin Kavanaugh, Evan Mays, Evgenii Nikishin, Fangyuan Li, Felipe Petroski Such, Filipe de Avila Belbute Peres, Filippo Raso, Florent Bekerman, Foivos Tsimpourlas, Fotis Chantzis, Francis Song, Francis Zhang, Gaby Raila, Garrett McGrath, Gary Briggs, Gary Yang, Giambattista Parascandolo, Gildas Chabot, Grace Kim, Grace Zhao, Gregory Valiant, Guillaume Leclerc, Hadi Salman, Hanson Wang, Hao Sheng, Haoming Jiang, Haoyu Wang, Haozhun Jin, Harshit Sikchi, Heather Schmidt, Henry Aspegren, Honglin Chen, Huida Qiu, Hunter Lightman, Ian Covert, Ian Kivlichan, Ian Silber, Ian Sohl, Ibrahim Hammoud, Ignasi Clavera, Ikai Lan, Ilge Akkaya, Ilya Kostrikov, Irina Kofman, Isak Etinger, Ishaan Singal, Jackie Hehir, Jacob Huh, Jacqueline Pan, Jake Wilczynski, Jakub Pachocki, James Lee, James Quinn, Jamie Kiros, Janvi Kalra, Jasmyn Samaroo, Jason Wang, Jason Wolfe, Jay Chen, Jay Wang, Jean Harb, Jeffrey Han, Jeffrey Wang, Jennifer Zhao, Jeremy Chen, Jerene Yang, Jerry Tworek, Jesse Chand, Jessica Landon, Jessica Liang, Ji Lin, Jiancheng Liu, Jianfeng Wang, Jie Tang, Jihan Yin, Joanne Jang, Joel Morris, Joey Flynn, Johannes Ferstad, Johannes Heidecke, John Fishbein, John Hallman, Jonah Grant, Jonathan Chien, Jonathan Gordon, Jongsoo Park, Jordan Liss, Jos Kraaijeveld, Joseph Guay, Joseph Mo, Josh Lawson, Josh McGrath, Joshua Vendrow, Joy Jiao, Julian Lee, Julie Steele, Julie Wang, Junhua Mao, Kai Chen, Kai Hayashi, Kai Xiao, Kamyar Salahi, Kan Wu, Karan Sekhri, Karan Sharma, Karan Singhal, Karen Li, Kenny Nguyen, Keren Gu-Lemberg, Kevin King, Kevin Liu, Kevin Stone, Kevin Yu, Kristen Ying, Kristian Georgiev, Kristie Lim, Kushal Tirumala, Kyle Miller, Lama Ahmad, Larry Lv, Laura Clare, Laurance Fauconnet, Lauren Itow, Lauren Yang, Laurentia Romaniuk, Leah Anise, Lee Byron, Leher Pathak, Leon Maksin, Leyan Lo, Leyton Ho, Li Jing, Liang Wu, Liang Xiong, Lien Mamitsuka, Lin Yang, Lindsay McCallum, Lindsey Held, Liz Bourgeois, Logan Engstrom, Lorenz Kuhn, Louis Feuvrier, Lu Zhang, Lucas Switzer, Lukas Kondraciuk, Lukasz Kaiser, Manas Joglekar, Mandeep Singh, Mandip Shah, Manuka Stratta, Marcus Williams, Mark Chen, Mark Sun, Marselus Cayton, Martin Li, Marvin Zhang, Marwan Aljubeh, Matt Nichols, Matthew Haines, Max Schwarzer, Mayank Gupta, Meghan Shah, Melody Huang, Meng Dong, Mengqing Wang, Mia Glaese, Micah Carroll, Michael Lampe, Michael Malek, Michael Sharman, Michael Zhang, Michele Wang, Michelle Pokrass, Mihai Florian, Mikhail Pavlov, Miles Wang, Ming Chen, Mingxuan Wang, Minnia Feng, Mo Bavarian, Molly Lin, Moose Abdool, Mostafa

Rohaninejad, Nacho Soto, Natalie Staudacher, Natan LaFontaine, Nathan Marwell, Nelson Liu, Nick Preston, Nick Turley, Nicklas Ansman, Nicole Blades, Nikil Pancha, Nikita Mikhaylin, Niko Felix, Nikunj Handa, Nishant Rai, Nitish Keskar, Noam Brown, Ofir Nachum, Oleg Boiko, Oleg Murk, Olivia Watkins, Oona Gleeson, Pamela Mishkin, Patryk Lesiewicz, Paul Baltescu, Pavel Belov, Peter Zhokhov, Philip Pronin, Phillip Guo, Phoebe Thacker, Qi Liu, Qiming Yuan, Qinghua Liu, Rachel Dias, Rachel Puckett, Rahul Arora, Ravi Teja Mullapudi, Raz Gaon, Reah Miyara, Rennie Song, Rishabh Aggarwal, RJ Marsan, Robel Yemiru, Robert Xiong, Rohan Kshirsagar, Rohan Nuttall, Roman Tsiupa, Ronen Eldan, Rose Wang, Roshan James, Roy Ziv, Rui Shu, Ruslan Nigmatullin, Saachi Jain, Saam Talaie, Sam Altman, Sam Arnesen, Sam Toizer, Sam Toyer, Samuel Miserendino, Sandhini Agarwal, Sarah Yoo, Savannah Heon, Scott Ethersmith, Sean Grove, Sean Taylor, Sebastien Bubeck, Sever Banesiu, Shaokyi Amdo, Shengjia Zhao, Sherwin Wu, Shibani Santurkar, Shiyu Zhao, Shraman Ray Chaudhuri, Shreyas Krishnaswamy, Shuaiqi, Xia, Shuyang Cheng, Shyamal Anadkat, Simón Posada Fishman, Simon Tobin, Siyuan Fu, Somay Jain, Song Mei, Sonya Egoian, Spencer Kim, Spug Golden, SQ Mah, Steph Lin, Stephen Imm, Steve Sharpe, Steve Yadlowsky, Sulman Choudhry, Sungwon Eum, Suvansh Sanjeev, Tabarak Khan, Tal Stramer, Tao Wang, Tao Xin, Tarun Gogineni, Taya Christianson, Ted Sanders, Tejal Patwardhan, Thomas Degry, Thomas Shadwell, Tianfu Fu, Tianshi Gao, Timur Garipov, Tina Sriskandarajah, Toki Sherbakov, Tomer Kaftan, Tomo Hiratsuka, Tongzhou Wang, Tony Song, Tony Zhao, Troy Peterson, Val Kharitonov, Victoria Chernova, Vineet Kosaraju, Vishal Kuo, Vitchyr Pong, Vivek Verma, Vlad Petrov, Wanning Jiang, Weixing Zhang, Wenda Zhou, Wenlei Xie, Wenting Zhan, Wes McCabe, Will DePue, Will Ellsworth, Wulfie Bain, Wyatt Thompson, Xiangning Chen, Xiangyu Qi, Xin Xiang, Xinwei Shi, Yann Dubois, Yaodong Yu, Yara Khakbaz, Yifan Wu, Yilei Qian, Yin Tat Lee, Yinbo Chen, Yizhen Zhang, Yizhong Xiong, Yonglong Tian, Young Cha, Yu Bai, Yu Yang, Yuan Yuan, Yuanzhi Li, Yufeng Zhang, Yuguang Yang, Yujia Jin, Yun Jiang, Yunyun Wang, Yushi Wang, Yutian Liu, Zach Stubenvoll, Zehao Dou, Zheng Wu, and Zhigang Wang. Openai gpt-5 system card, 2025. URL `https://arxiv.org/abs/2601.03267`.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, Zhuofu Chen, Jialei Cui, Hao Ding, Mengnan Dong, Angang Du, Chenzhuang Du, Dikang Du, Yulun Du, Yu Fan, Yichen Feng, Kelin Fu, Bofei Gao, Hongcheng Gao, Peizhong Gao, Tong Gao, Xinran Gu, Longyu Guan, Haiqing Guo, Jianhang Guo, Hao Hu, Xiaoru Hao, Tianhong He, Weiran He, Wenyang He, Chao Hong, Yangyang Hu, Zhenxing Hu, Weixiao Huang, Zhiqi Huang, Zihao Huang, Tao Jiang, Zhejun Jiang, Xinyi Jin, Yongsheng Kang, Guokun Lai, Cheng Li, Fang Li, Haoyang Li, Ming Li, Wentao Li, Yanhao Li, Yiwei Li, Zhaowei Li, Zheming Li, Hongzhan Lin, Xiaohan Lin, Zongyu Lin, Chengyin Liu, Chenyu Liu, Hongzhang Liu, Jingyuan Liu, Junqi Liu, Liang Liu, Shaowei Liu, T. Y. Liu, Tianwei Liu, Weizhou Liu, Yangyang Liu, Yibo Liu, Yiping Liu, Yue Liu, Zhengying Liu, Enzhe Lu, Lijun Lu, Shengling Ma, Xinyu Ma, Yingwei Ma, Shaoguang Mao, Jie Mei, Xin Men, Yibo Miao, Siyuan Pan, Yebo Peng, Ruoyu Qin, Bowen Qu, Zeyu Shang, Lidong Shi, Shengyuan Shi, Feifan Song, Jianlin Su, Zhengyuan Su, Xinjie Sun, Flood Sung, Heyi Tang, Jiawen Tao, Qifeng Teng, Chensi Wang, Dinglu Wang, Feng Wang, Haiming Wang, Jianzhou Wang, Jiaxing Wang, Jinhong Wang, Shengjie Wang, Shuyi Wang, Yao Wang, Yejie Wang, Yiqin Wang, Yuxin Wang, Yuzhi Wang, Zhaoji Wang, Zhengtao Wang, Zhexu Wang, Chu Wei, Qianqian Wei, Wenhao Wu, Xingzhe Wu, Yuxin Wu, Chenjun Xiao, Xiaotong Xie, Weimin Xiong, Boyu Xu, Jing Xu, Jinjing Xu, L. H. Xu, Lin Xu, Suting Xu, Weixin Xu, Xinran Xu, Yangchuan Xu, Ziyao Xu, Junjie Yan, Yuzi Yan, Xiaofei Yang, Ying Yang, Zhen Yang, Zhilin Yang, Zonghan Yang, Haotian Yao, Xingcheng Yao, Wenjie Ye, Zhuorui Ye, Bohong Yin, Longhui Yu, Enming Yuan, Hongbang Yuan, Mengjie Yuan, Haobing Zhan, Dehao Zhang, Hao Zhang, Wanlu Zhang, Xiaobin Zhang, Yangkun Zhang, Yizhi Zhang, Yongting Zhang, Yu Zhang, Yutao Zhang, Yutong Zhang, Zheng Zhang, Haotian Zhao, Yikai Zhao, Huabin Zheng, Shaojie Zheng, Jianren Zhou, Xinyu Zhou, Zaida Zhou, Zhen Zhu, Weiyu Zhuang, and Xinxing Zu. Kimi k2: Open agentic intelligence, 2025. URL `https://arxiv.org/abs/2507.20534`.

Tencent Hunyuan Foundation Model Team. Hunyuanvideo 1.5 technical report, 2025. URL `https://arxiv.org/abs/2511.18870`.

Fish Tung, Mingyu Ding, Zhenfang Chen, Daniel M. Bear, Chuang Gan, Joshua B. Tenenbaum, Daniel L. K. Yamins, Judith Fan, and Kevin A. Smith. Physion++: Evaluating physical scene understanding that requires online inference of different physical properties. *arXiv*, 2023.

Karthik Valmeekam, Matthew Marquez, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. Planbench: An extensible benchmark for evaluating large language models on planning and reasoning about change. In *Advances in Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=YXogl4uQUO.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models, 2025. URL https://arxiv.org/abs/2503.20314.

Siyin Wang, Zhaoye Fei, Qinyuan Cheng, Shiduo Zhang, Panpan Cai, Jinlan Fu, and Xipeng Qiu. World modeling makes a better planner: Dual preference optimization for embodied task planning, 2025. URL https://arxiv.org/abs/2503.10480.

Tai Wang, Xiaohan Mao, Chenming Zhu, Runsen Xu, Ruiyuan Lyu, Peisen Li, Xiao Chen, Wenwei Zhang, Kai Chen, Tianfan Xue, Xihui Liu, Cewu Lu, Dahua Lin, and Jiangmiao Pang. Embodiedscan: A holistic multi-modal 3d perception suite towards embodied ai. In *Conference on Computer Vision and Pattern Recognition*, 2024a.

Xingrui Wang, Wufei Ma, Angtian Wang, Shuo Chen, Adam Kortylewski, and Alan Yuille. Compositional 4d dynamic scenes understanding with physics priors for video question answering. *arXiv*, 2024b.

Yi Wang, Jiafei Duan, Dieter Fox, and Siddhartha Srinivasa. Newton: Are large language models capable of physical reasoning? In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9743–9758, 2023. doi: 10.18653/v1/2023.findings-emnlp.652. URL https://aclanthology.org/2023.findings-emnlp.652/.

Zirui Wang, Junyi Zhang, Jiaxin Ge, Long Lian, Letian Fu, Lisa Dunlap, Ken Goldberg, XuDong Wang, Ion Stoica, David M. Chan, Sewon Min, and Joseph E. Gonzalez. Visgym: Diverse, customizable, scalable environments for multimodal agents, 2026. URL https://arxiv.org/abs/2601.16973.

Thaddäus Wiedemer, Yuxuan Li, Paul Vicol, Shixiang Shane Gu, Nick Matarese, Kevin Swersky, Been Kim, Priyank Jaini, and Robert Geirhos. Video models are zero-shot learners and reasoners, 2025. URL https://arxiv.org/abs/2509.20328.

Xinrun Xu, Pi Bu, Ye Wang, Börje F. Karlsson, Ziming Wang, Tengtao Song, Qi Zhu, Jun Song, Zhiming Ding, and Bo Zheng. DeepPHY: Benchmarking agentic VLMs on physical reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 34160–34168, 2026. URL https://ojs.aaai.org/index.php/AAAI/article/view/40711.

Rui Yang, Hanyang Chen, Junyu Zhang, Mark Zhao, Cheng Qian, Kangrui Wang, Qineng Wang, Teja Venkat Koripella, Marziyeh Movahedi, Manling Li, Heng Ji, Huan Zhang, and Tong Zhang. EmbodiedBench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents. *arXiv preprint arXiv:2502.09560*, 2025. URL https://arxiv.org/abs/2502.09560.

Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B Tenenbaum. Clevrer: Collision events for video representation and reasoning. *arXiv*, 2019.

Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning, 2025. URL https://arxiv.org/abs/2407.08693.

Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, Ankur Handa, Ming-Yu Liu, Donglai Xiang, Gordon Wetzstein, and Tsung-Yi Lin. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1702–1713, 2025. URL https://api.semanticscholar.org/CorpusID:277435005.

Zhicheng Zheng, Xin Yan, Zhenfang Chen, Jingzhou Wang, Qin Zhi Eddie Lim, Joshua B Tenenbaum, and Chuang Gan. Contphy: Continuum physical concept learning and reasoning from videos. In *International Conference on Machine Learning*. PMLR, 2024.

Weijie Zhou, Manli Tao, Chaoyang Zhao, Haiyun Guo, Honghui Dong, Ming Tang, and Jinqiao Wang. Physvlm: Enabling visual language models to understand robotic physical reachability, 2025. URL https://arxiv.org/abs/2503.08481.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction clearly state that CHAIN is a benchmark for Level III constraint-aware sequential planning, describe the 200 task instances across five task families, and report that even the strongest model achieves only $\sim$55% task success. These claims are fully supported by the experimental results in Section 3.

   Guidelines:

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

2. **Limitations**

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

Answer: [Yes]

Justification: Limitations are discussed in Appendix A.1. We acknowledge (1) limited scale of interlocking puzzle instances due to the manual engineering effort required per Unity environment, and (2) that we primarily report Pass@1 rather than multi-run statistics due to the high API cost of closed-loop evaluation.

Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [N/A]

Justification: This paper introduces an evaluation benchmark and reports empirical results; it contains no theoretical results, theorems, or proofs.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: Section 3.1 specifies all evaluation hyperparameters (temperature 0.6, top-p 0.95, step budget 30–60, history window 5) and the full list of evaluated models. Appendix A.2 describes the benchmark release plan including task definitions, simulator code, and model evaluation scripts sufficient to reproduce all reported results.

Guidelines:

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

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

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

Answer: [Yes]

Justification: As described in Appendix A.2, we will release an anonymized benchmark repository containing task definitions, simulator code, trajectory logs, and evaluation scripts for all reported experiments.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).

- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Section 3.1 provides the full experimental setup: the evaluated model pool, decoding parameters (temperature 0.6, top-p 0.95), step budget, and trajectory history window. Per-model context window and parameter scale are listed in Appendix A.4 (Table 4). No training is performed; all models are evaluated in zero-shot closed-loop inference.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: We primarily report Pass@1 without error bars due to the high cost of closed-loop multi-step interaction (each episode involves up to 30–60 API calls per instance). As discussed in Appendix A.1, preliminary analysis in Section 3.5 shows that Pass@1 and Avg@4 exhibit consistent trends, indicating that run-to-run variance is not dominant in practice.

   Guidelines:

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

- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [No]

   Justification: All models are evaluated via APIs. We report token counts per model via the Solved/Tok. metric (Table 1) as a proxy for inference cost, but do not report wall-clock time or specific hardware configurations, as these vary by provider and deployment environment.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: This work releases an open evaluation benchmark for physical reasoning. It does not involve sensitive personal data, does not enable harmful applications, and the human baseline study (Appendix A.9) was conducted with voluntary participants under standard academic practice.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: A Broader Impact Statement is provided after the Conclusion. It discusses the positive impact of enabling more rigorous evaluation of interactive physical reasoning and notes that no direct negative societal impacts are anticipated beyond standard concerns for open research artifacts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: CHAIN is an evaluation benchmark consisting of 3D physics puzzle environments and interaction logs. It does not release a generative model, scraped dataset, or any artifact with meaningful misuse potential.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All evaluated models are cited with their original technical reports. Puzzle designs are drawn from classical interlocking designs (Kongming/Lu Ban locks) in the public domain, and the Puzzlemaster reference is cited. Simulation environments are built with Unity and custom Python engines; no third-party datasets are incorporated.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets`

has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: The CHAIN benchmark is the primary new asset. Appendix A.2 describes the release structure (task JSON manifests, simulator code, evaluation scripts, trajectory logs). Task construction is detailed in Section 2.2 and Appendix A.7. The repository will be released under a permissive open-source license.

    Guidelines:

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

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

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

    Answer: [Yes]

    Justification: The human baseline study (Appendix A.9) involved voluntary participants asked to solve benchmark instances under the same closed-loop protocol as the model evaluations. Task instructions are identical to those provided to models (Section 2.2). Participant recruitment and compensation details are described in Appendix A.9.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

    Answer: [N/A]

    Justification: The human baseline study involved only voluntary puzzle-solving tasks with no sensitive data collection, no physical risk, and no deception; it does not meet the threshold requiring formal IRB review at our institution.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: LLMs and VLMs are the *subjects* of evaluation in this benchmark paper, not components of the research methodology. Benchmark construction relied on programmatic generation and human expert design. LLMs were not used as a core method in designing or constructing CHAIN.

    Guidelines:

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

Benchmark	Interaction	3D	Fixed Action API	Action–State Taxonomy	# Families	Primary Domain
PHYRE [Bakhtin et al., 2019]	Single-shot	✗	✗	✗	1	2D physics puzzle
I-PHYRE [Li et al., 2024c]	Real-time	✗	✓	✗	1	2D physics puzzle
CausalWorld [Ahmed et al., 2021]	Closed-loop (RL)	✓	✗ (joint ctrl)	✗	8	Robotic manipulation
PhyBlock [Ma et al., 2025]	Closed-loop (step-by-step)	✓	✓	✗	1 (+VQA)	3D block assembly
ViPlan [Merler et al., 2026]	Closed-loop	✓	✓	✗	2	Blocksworld / Household
DeepPHY [Xu et al., 2026]	Closed-loop (agentic)	✗ (mostly 2D)	✓	✗	6	Mixed physics sims
CHAIN (Ours)	**Closed-loop**	✓	✓	**✓ (T1/T2/T3)**	**5**	**Geometry-rich 3D physics**

Table 2: Comparison of representative physical-reasoning benchmarks along six axes: (i) *Interaction protocol* (single-shot, real-time, or closed-loop interactive); (ii) whether the environment is fully *3D*; (iii) whether evaluation uses a *fixed action API* that abstracts away low-level actuation; (iv) whether tasks are organized by a *explicit taxonomy of action–state coupling*; (v) the *number of task families* (excluding pure VQA); (vi) the *primary domain*. While several recent benchmarks combine closed-loop interaction with 3D scenes and a fixed action API, none provides an explicit action–state coupling taxonomy, and most are restricted to a single domain (block assembly, household, or 2D physics). **CHAIN** uniquely combines closed-loop interaction in geometry-rich 3D environments with a fixed action API and an explicit action–state coupling taxonomy (constraint gating T1, spatial commitment T2, structural propagation T3) instantiated by five task families (Puzzle, Stacking, FluidNet, PipeFit, Domino). Counts/flags are based on the original publications: PHYRE (single ball placement) [Bakhtin et al., 2019]; I-PHYRE (40 interactive 2D games, real-time block elimination) [Li et al., 2024c]; CausalWorld (8 manipulation distributions on TriFinger, joint/Cartesian control) [Ahmed et al., 2021]; PhyBlock (400 assembly tasks + 2,200 VQA, Genesis simulator, high-level block placement) [Ma et al., 2025]; ViPlan (Blocksworld + iGibson Household, PDDL-grounded high-level actions) [Merler et al., 2026]; DeepPHY (six 2D-leaning physics simulators) [Xu et al., 2026].

A Appendix

A.1 Detailed Comparison with Existing Physical-Reasoning Benchmarks

To complement the related-work discussion in the main text, Table 2 compares **CHAIN** with a representative set of physical-reasoning benchmarks—PHYRE [Bakhtin et al., 2019], I-PHYRE [Li et al., 2024c], CausalWorld [Ahmed et al., 2021], PhyBlock [Ma et al., 2025], ViPlan [Merler et al., 2026], and DeepPHY [Xu et al., 2026]—along six axes: interaction protocol, 3D vs. 2D, fixed action API, action–state coupling taxonomy, number of task families, and primary domain. The flags and counts are based directly on each benchmark's original publication; we briefly summarize the key facts below. PHYRE is a single-shot 2D contextual bandit in which the agent places a single ball and observes the rollout. I-PHYRE extends this to real-time intervention with discrete block-elimination actions, but remains 2D. CausalWorld places a TriFinger robot in front of eight manipulation task distributions in 3D, but the action space is joint or Cartesian control, so planning is entangled with low-level actuation. PhyBlock evaluates VLMs on 3D block assembly via Genesis-based step-by-step or one-time planning with high-level placement actions, but is restricted to a single task family (with an additional VQA branch). ViPlan provides two visual-planning domains—a photorealistic Blocksworld renderer and an iGibson Household environment with PDDL-grounded high-level actions—both effectively 3D. DeepPHY aggregates six existing physics simulators (PHYRE, I-PHYRE, Kinetix, Pooltool, Cut the Rope, Angry Birds) into a unified agentic VLM evaluation, but these environments are predominantly 2D and the suite is not organized by an action–state coupling taxonomy. By contrast, **CHAIN** combines closed-loop interaction, geometry-rich 3D physics, a fixed action API, and an explicit taxonomy of action–state coupling (T1: constraint gating; T2: spatial commitment; T3: structural propagation), instantiated by five task families that include both single-type and mixed-type couplings.

A.2 Task Examples

This section showcases some cases of puzzles and stacking. As shown in Fig. 6, we divide the puzzle task into three difficulty levels according to (i) the number of pieces and (ii) the complexity of interlocks:

- **Easy Level:** the object consists of a small number of pieces with simple interfaces and weak interlocking. Each piece has a distinctive placement, and the disassembly can be completed with few steps and low risk of dead-ends.
- **Middle Level:** the object contains more pieces and more complex contact surfaces. Interlocking features become more frequent, requiring more careful selection of the moving direction and intermediate ordering.
- **Hard Level:** the object consists of many pieces with strong mutual interlocks (e.g., multiple pieces jointly constrain removal), often necessitating long-horizon planning and reasoning over constraint propagation across the entire disassembly.

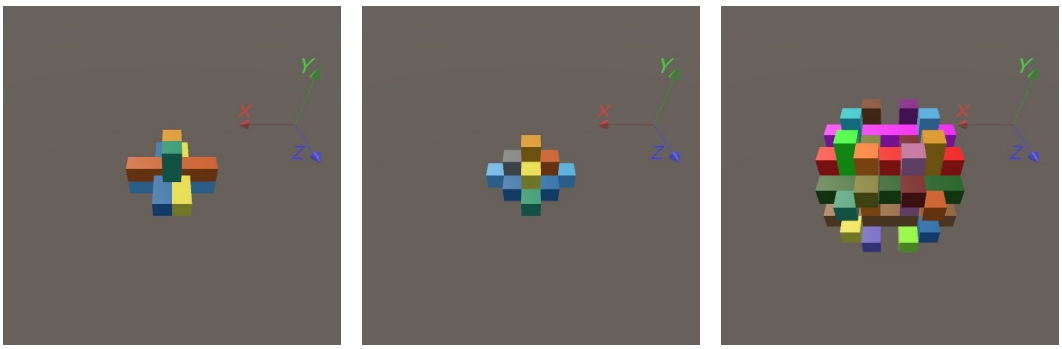
As shown in Fig. 7, we categorize stacking difficulty primarily by the container size and the number and shape diversity of pieces, which jointly determine the combinatorial search space:

- **Easy Level:** a small container (e.g., $2\times2\times3$) with relatively few pieces and limited shape complexity. Feasible placements are constrained and the solution space is small.
- **Middle Level:** a medium container (e.g., $2\times3\times3$) with increased spatial freedom and more placement permutations. The agent must reason more carefully about partial fills to avoid creating unreachable cavities.
- **Hard Level:** a large container (e.g., $3\times3\times4$) and a larger set of pieces. This setting introduces substantial ambiguity in placement order and orientation, making the task highly combinatorial and sensitive to early decisions.

A.3 Case Analysis of Stacking Task

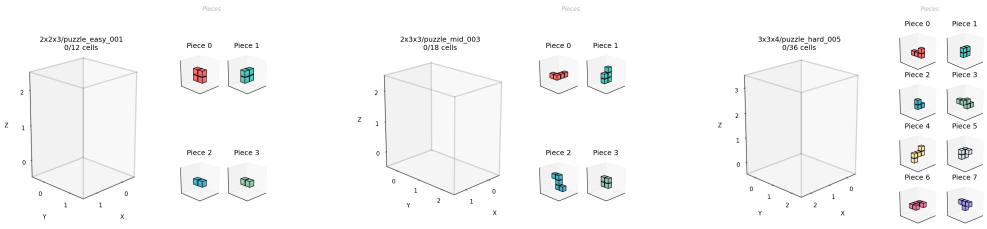
Fig. 8 and Fig. 9 demonstrate a successful and a failed trajectory of the stacking task, respectively. In the successful case (Fig. 8), the agent follows a structure-first strategy. It starts by placing compact blocks to form a stable, flat base (Steps 1–2), and then stacks the upper layers layer by layer (Step 3). After the base is expanded (Step 4), subsequent placements (Steps 5–7) are used to systematically fill the remaining volume without creating unsupported overhangs or isolated cavities. This ordering preserves sufficient free space for later insertions and keeps the target shape "fillable" at every intermediate state, enabling the final completion.

In contrast, the failure case (Fig. 9) is mainly caused by an early irreversible commitment. The agent places a tall vertical column at the beginning (Step 1), before establishing a complete foundation. This choice constrains where later pieces can go, forcing the following placements to wrap around the column (Steps 2–5). Even though the agent attempts to roll back some actions and try new placements (Steps 6–8), the partially built structure encloses awkward residual space and reduces the clearance needed to insert the remaining piece. Although each intermediate placement is locally valid, the resulting configuration becomes a dead-end: the remaining piece cannot fit due to geometric mismatch, so the episode ends incomplete.



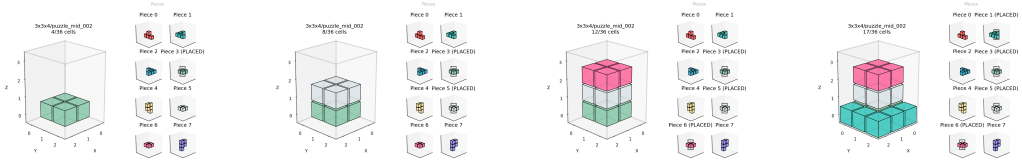
(a) Easy Level (b) Middle Level (c) Hard Level

Figure 6: Examples of different levels of puzzle task.

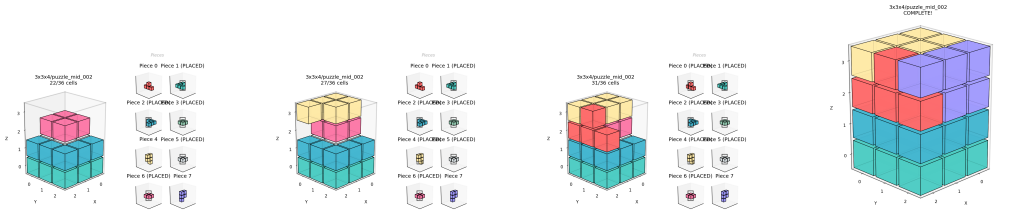


(a) Easy Level (b) Middle Level (c) Hard Level

Figure 7: Examples of different levels of stacking task.

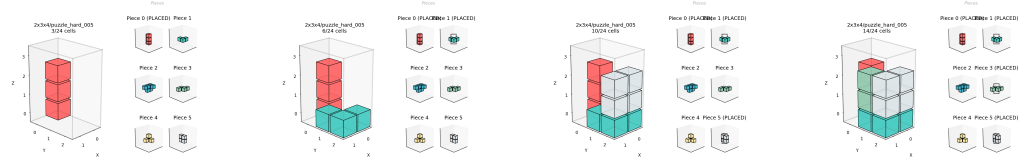


(a) Step 1 (b) Step 2 (c) Step 3 (d) Step 4

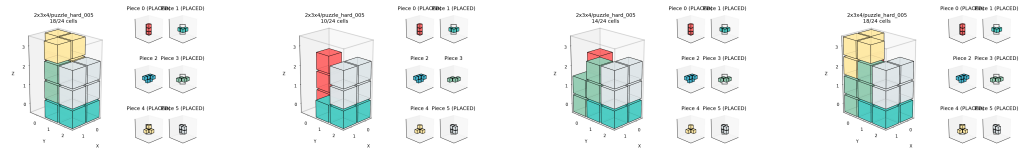


(e) Step 5 (f) Step 6 (g) Step 7 (h) Step 8

Figure 8: A case of successfully solving the stacking task.



(a) Step 1 (b) Step 2 (c) Step 3 (d) Step 4



(e) Step 5 (f) Step 6 (g) Step 7 (h) Step 8

Figure 9: A case of failed to solve the stacking task.

A.4 Polycube Stacking Puzzle Generation Pipeline

This appendix describes how we generate stable, solvable, and de-duplicated 3D polycube *stacking/assembly* puzzles inside an $a \times b \times c$ voxel box. The generator follows a *sample–verify* design: we first sample candidate partitions into connected pieces, then apply a sequence of hard validity checks (exact-cover solvability, structural constraints, and (optional) linear assembly feasibility). We also enforce strong de-duplication by canonicalizing each piece shape up to rigid rotations and hashing the resulting multiset.

A.4.1 Objectives and Core Constraints

Given a box $\mathcal{B} = \{0, \ldots, a-1\} \times \{0, \ldots, b-1\} \times \{0, \ldots, c-1\}$, we generate a set of pieces $\mathcal{P} = \{P_1, \ldots, P_K\}$ such that:

1. **Exact cover (solvable packing).** There exists a placement of all pieces (allowing 24 rigid rotations and translations within the box) that covers each voxel in $\mathcal{B}$ *exactly once*.
2. **Connectivity.** Each piece is 6-neighbor connected.
3. **Piece size bounds.** Each piece satisfies $|P_i| \in [\texttt{min\_piece}, \texttt{max\_piece\_cells}]$.
4. **Structural rule: no 2×3 full face.** No piece contains a filled 2×3 rectangle in any axis-aligned plane (XY/YZ/XZ), which avoids large flat plates that often reduce interaction richness.
5. **No isolated piece.** In the final packed solution, each piece must touch at least one other piece via 6-neighbor adjacency (preventing "floating" blocks).
6. **(Optional) Linear assembly.** If enabled, the packed solution must admit a collision-free *linear* disassembly/assembly sequence where each step removes one piece by translating it along one of $\{\pm x, \pm y, \pm z\}$ without intersecting remaining pieces.

A.4.2 Candidate Partition Sampling by Difficulty

We support three difficulty modes by varying how candidate pieces are sampled before verification:

Easy (regular cuboids). We sample a multiset of axis-aligned cuboids whose volumes sum to abc and whose dimensions satisfy an "easy-shape" rule: at least two dimensions are equal (e.g., $1\times1\times2$, $1\times2\times2$, $2\times2\times1$). We then verify whether these cuboids can be packed (with rotations) to exactly fill the box via the same exact-cover solver as other modes.

Mid (flat connected growth). We grow connected pieces by seeded BFS-style expansion, but constrain each piece to a plane by fixing one coordinate (e.g., $x = x_0$). This produces "sheet-like" structures while remaining connected and bounded in size.

Hard (free connected growth + uniqueness). We grow connected pieces without planar constraints and further enforce that all pieces have *distinct* shapes up to rigid rotations. If duplicates occur, we attempt to improve uniqueness via limited voxel transfers between adjacent pieces while maintaining connectivity; otherwise we reject the sample. We also use solver search effort (visited nodes) as a difficulty proxy and enforce a mode-specific threshold.

A.4.3 Exact-Cover Solvability via DLX

We validate packability by reducing the problem to an *exact cover* instance. Each candidate placement of piece i corresponds to a row that covers: (i) the voxels occupied by the placement, and (ii) a "use piece i" column that enforces each piece is used exactly once. We solve the resulting sparse exact-cover matrix using Algorithm X with dancing links (DLX), augmented with: (1) caching all placements per canonical shape and box size, (2) an *anchor piece* (the piece with fewest placements) constrained to cover the origin voxel to reduce symmetries, and (3) an MRV-style column choice heuristic.

Algorithm 1 Linear Assembly Sequence Extraction (via Greedy Disassembly)

Require: Placed pieces $\{C_i\}_{i=1}^{K}$, where $C_i \subseteq \mathcal{B}$ is the set of voxels occupied by piece i
Ensure: Assembly order π and removal directions $\{\mathbf{d}_t\}$, or FAIL

$R \leftarrow \{1, 2, \ldots, K\}$ ▷ Indices of remaining pieces
$removal \leftarrow [\,]$ ▷ Disassembly sequence as a list of $(i, \mathbf{d})$
while $R \neq \emptyset$ **do**
  $U \leftarrow \bigcup_{j \in R} C_j$ ▷ All voxels occupied by remaining pieces
  $found \leftarrow$ **false**
  for $i \in R$ **do**
    $O \leftarrow U \setminus C_i$ ▷ Voxels occupied by other pieces
    for $\mathbf{d} \in \{+\vec{x}, -\vec{x}, +\vec{y}, -\vec{y}, +\vec{z}, -\vec{z}\}$ **do**
      if REMOVABLE$(C_i, O, \mathbf{d})$ **then**
        Append $(i, \mathbf{d})$ to $removal$
        $R \leftarrow R \setminus \{i\}$
        $found \leftarrow$ **true**
        break
      end if
    end for
    if $found$ **then**
      break
    end if
  end for
  if not $found$ **then**
    return FAIL
  end if
end while
$\pi \leftarrow$ REVERSE$(removal)$
return π

A.4.4 Physical Feasibility and Linear Assembly Extraction

After DLX returns a packed solution (a concrete set of voxels for each placed piece), we apply two physical-feasibility checks.

No isolated piece. We reject solutions where any piece has no 6-neighbor contact with other pieces. This avoids degenerate instances that are technically solvable but physically uninformative for assembly reasoning.

Linear disassembly/assembly sequence. If linear assembly is required, we attempt to find a sequence in which pieces can be removed one-by-one by a pure translation along one of the six axis directions, without colliding with remaining pieces. A piece P is *removable* along direction $\mathbf{d}$ if translating each voxel of P by one unit along $\mathbf{d}$ does not enter any voxel occupied by other pieces that are still present (within box bounds). We greedily remove any currently removable piece; if we can remove all pieces, reversing this removal order yields a valid assembly sequence.

Algorithm 1 gives the extraction procedure used in our pipeline.

The predicate REMOVABLE checks whether translating C_i by one voxel step along $\mathbf{d}$ causes any voxel of the translated set to intersect O (within bounds). If no direction works for any remaining piece at some iteration, the instance does not admit a linear assembly and is rejected when linear assembly is required.

A.4.5 Strong De-duplication by Shape Multiset Canonicalization

To avoid repeated puzzle templates, we de-duplicate instances by canonicalizing each piece shape up to rigid rotation. For each piece P_i, we: (i) translate it to a normalized origin (minimum coordinate at zero), then (ii) enumerate its 24 rotation-equivalent shapes, and (iii) select the lexicographically smallest voxel list as its canonical representation. We then sort the list of canonical piece representa-

Algorithm 2 Batch Generation with Staged Sampling, Verification and De-duplication

Require: Set of box sizes $\mathcal{D}$,
Per-size configuration (max pieces, difficulty thresholds),
Target number of puzzles per size/mode T
Ensure: A dataset of accepted puzzles,
Summary metadata (counts, rejection reasons, generation statistics)

for all $(a, b, c) \in \mathcal{D}$ **do**
  for each mode $\in \{\texttt{easy}, \texttt{mid}, \texttt{hard}\}$ **do**
    for $t = 1$ to T **do**
      gen $\leftarrow$ GENERATEONESTAGED($a, b, c, mode, cfg$)
      if *gen* $= \texttt{Fail}$ **then**
        continue
      end if
      $s \leftarrow$ PUZZLESIGNATURE(*gen.pieces*)
      if $\neg$ RESERVESIGNATUREATOMICALLY(s) **then**
        continue
      end if
      SAVEPUZZLE(*gen*, s) ▷ Store as JSON + visual renders in per-puzzle folder
    end for
  end for
end for
return $(dataset, metadata)$

tions to form a *multiset signature* for the puzzle and hash it (e.g., SHA1). In multi-processing, we reserve signatures with an atomic marker file to ensure thread-safe de-duplication.

A.4.6 End-to-End Batch Generation and Outputs

For each box size (a, b, c) and difficulty mode (easy/mid/hard), we generate a fixed number of puzzles by repeatedly invoking the staged sampler with verification and de-duplication. Each accepted puzzle is stored in its own folder with: (i) a JSON manifest (pieces, packed solution, assembly order, difficulty stats, signature), and (ii) rendered 3D voxel visualizations of the full assembly and individual pieces.

Algorithm 2 summarizes the full pipeline.

Staged min-piece fallback. GenerateOneStaged implements a staged attempt schedule: it first searches with a stricter minimum piece size (e.g., $\texttt{min\_piece} = 4$) and, if unsuccessful within budget, relaxes to $\texttt{min\_piece} = 3$ to improve yield for small volumes or tight constraints. Each sampled partition is accepted only if it passes the full chain of checks: *(i) structural rules, (ii) DLX solvability, (iii) no isolated piece, (iv) (optional) linear assembly, and (v) difficulty threshold (hard).*

A.5 Model List

Evaluated Models. We evaluate 14 multimodal models, covering both closed-source APIs and open-source checkpoints. Our *closed-source* set includes GPT-5.2, GPT-5-mini, Claude-Opus-4.5, Claude-Sonnet-4.5, Gemini-3-Pro (preview), Gemini-3-Flash (preview), Seed-1.6, and Seed-1.6-Flash. Our *open-source* set consists of Kimi-k2.5, Qwen3-VL-235B-A22B (Thinking and Instruct), Qwen3-VL-30B-A3B (Thinking and Instruct), and Qwen3-VL-8B-Thinking. Table 3 summarizes each model's release date, maximum context window, and parameter scale (when disclosed). For MoE models, we additionally report the active parameter count per forward pass in the form of `AxxB`. Models whose parameter sizes are not publicly released are marked as -.

A.6 Human Baseline on CHAIN

To better contextualize the difficulty of **CHAIN**, we additionally report human performance on both the **Stacking** and **Puzzle** tasks. We restrict the human study to these two families because they are the most constraint-heavy and long-horizon families in **CHAIN**, and thus serve as the most

Models (abbr.)	Model Specs		
	Release Date	Context (tokens)	Model Size
Closed-source Models			
GPT-5.2	2025-12-11	400,000	–
GPT-5-mini	2025-08-07	400,000	–
Claude-Opus-4.5	2025-11-24	200,000	–
Claude-Sonnet-4.5	2025-09-29	1,000,000	–
Gemini-3-Pro (preview)	2025-11-18	1,048,576	–
Gemini-3-Flash (preview)	2025-12-17	1,000,000	–
Seed-1.6	2025-12-23	262,144	–
Seed-1.6-Flash	2025-12-23	262,144	–
Open-source Models			
Kimi-k2.5	2026-01-27	262,144	1T (A32B)
Qwen3-VL-235B-A22B-Thinking	2025-09-23	262,144	235B (A22B)
Qwen3-VL-235B-A22B-Instruct	2025-09-23	262,144	235B (A22B)
Qwen3-VL-30B-A3B-Thinking	2025-10-04	262,144	30B (A3B)
Qwen3-VL-30B-A3B-Instruct	2025-10-04	262,144	30B (A3B)
Qwen3-VL-8B-Thinking	2025-10-15	262,144	8B

Table 3: Model release time, context window, and parameter scale for the 14 evaluated models. For MoE models, `AxxB` denotes active parameters per forward pass when available. Unknown values are marked as –.

Table 4: Human baseline on Stacking tasks.

Group	Easy		Medium		Hard	
	Pass Rate	Avg Steps/Pass	Pass Rate	Avg Steps/Pass	Pass Rate	Avg Steps/Pass
Amateur	100%	4.10	88.88%	6.68	52.08%	14.72
Expert	100%	4.10	100%	6.06	72.91%	16.34

informative reference points for human-vs.-model comparison; the propagation-oriented families (FluidNet, PipeFit, Domino) are largely tractable for humans and are therefore omitted to keep the study focused. In our experiments, human players were divided into two groups: **amateurs**, who had limited prior experience with such tasks, and **experts**, who had substantial prior experience and demonstrated higher proficiency. For each difficulty level, we report the **Pass Rate**, i.e., the percentage of successfully completed tasks, and the **Avg Steps/Pass**, i.e., the average number of interaction steps among successful trials.

Experimental Setup. Human participants were asked to solve the same task instances used in **CHAIN** under the three predefined difficulty levels: *Easy*, *Medium*, and *Hard*. We separately evaluate performance on the **Stacking** tasks and the **Puzzle** tasks. The comparison between amateurs and experts is intended to measure not only the absolute difficulty of the benchmark, but also whether the benchmark is able to distinguish between different levels of human spatial and procedural reasoning ability. Participants use exactly the same interface as the models, with the same multi-view observations and execution feedback and no privileged tactile or force feedback. Human time (10 minutes, covering observation, reasoning, and interface operation) is not resource-equivalent to the model's discrete action budget, so we treat these results as a non-resource-matched reference rather than a strictly controlled comparison, and we limit human–AI comparisons to the two families studied here.

Analysis. The human results show a clear difference in difficulty between the two task families. **Stacking** tasks are generally manageable for humans: both amateurs and experts achieve 100% pass rates on easy instances, experts also reach 100% on medium instances, and both groups retain substantial success rates on hard tasks. This suggests that stacking mainly tests spatial planning and sequential placement, while remaining largely accessible to human reasoning.

By contrast, **Puzzle** tasks are considerably more challenging. Even experts achieve only 70%, 25%, and 10% pass rates on easy, medium, and hard puzzles, respectively, while amateurs fail on all

Table 5: Human baseline on Puzzle tasks.

Group	Easy		Medium		Hard	
	Pass Rate	Avg Steps/Pass	Pass Rate	Avg Steps/Pass	Pass Rate	Avg Steps/Pass
Amateur	30%	13.00	8.33%	28.00	0%	–
Expert	70%	14.14	25%	29.33	10%	49.00

hard instances. This sharp drop indicates that puzzle solving in **CHAIN** requires stronger structural reasoning, longer-horizon planning, and better inference of hidden physical constraints.

The difference between amateurs and experts is also more pronounced on puzzle tasks than on stacking tasks, suggesting that prior experience and procedural intuition are especially important for solving interlocking structures. Step statistics further support this trend: successful puzzle trials require substantially more steps, and the average number of steps increases with difficulty, reflecting their long-horizon nature.

Overall, the human baseline shows that **CHAIN** is both challenging and discriminative. While stacking remains relatively tractable for humans, puzzle tasks constitute a much harder test of spatial, structural, and long-horizon interactive reasoning.

A.7 Symbolic and Heuristic Search Baselines

To further contextualize VLM performance on **CHAIN** and to address the natural question of how *classical search and constraint-satisfaction solvers* fare on our tasks, we additionally implement a suite of representative symbolic / heuristic search baselines on the stacking family. We focus on stacking because it admits a clean, fully-specified symbolic state representation (voxel occupancy of the container together with the multiset of remaining pieces), which makes it directly amenable to classical search.

Reference plan computation. Before describing the baselines, we first clarify how task-specific reference plans are obtained, which is independent of the search baselines below. For *puzzles*, the reference plans are derived from the underlying solution instructions / manual decompositions, which already specify a valid step-by-step disassembly sequence; we use these reference sequences to define plan-efficiency metrics and to verify candidate disassembly orders. For *stacking*, the reference number of steps is primarily determined by the number of blocks, since under any valid solution each block is placed exactly once. More generally, both task families can equivalently be viewed as search problems over valid intermediate states and actions, and a constrained BFS-style formulation can in principle recover shortest feasible plans under the task constraints.

Baselines and protocol. We implement six representative non-VLM baselines that operate directly on the symbolic state (privileged code-level access to voxel occupancy and remaining pieces), in contrast to VLMs which must infer the same information from raw visual observations:

- **BFS** – exhaustive breadth-first search over placement sequences.
- **DFS-Heuristic** – depth-first search guided by a simple geometric heuristic (e.g., preferring placements that minimize unreachable cavities).
- **Greedy** – one-step greedy placement using the same geometric heuristic.
- **SimAnnealing** – simulated annealing over placement orders.
- **Genetic** – a genetic / evolutionary search over candidate placement orders.
- **AntColony** – ant-colony-style population-based search over placement orders.

For a fair comparison, all symbolic baselines are restricted to the same per-instance interaction budget as VLMs (50 placement decisions per instance), matching the closed-loop step cap used in our main evaluation. Human baselines are reported at a 10-minute time budget with raw visual observations, identical to the protocol in Appendix A.6; as noted there, this time budget is not resource-matched to the 50-step budget. GPT-5.2 is included as a representative VLM, also evaluated under the 50-step budget on visual observations.

Table 6: Representative non-VLM and human baselines on **CHAIN** stacking under matched, restricted budgets. Symbolic baselines are given privileged code-level / tabular state access, while VLMs and humans operate on raw visual observations. "Step 50" denotes a 50-step interaction budget; "10 min" denotes a 10-minute time budget for humans.

Method	Type	Input modality	Budget	ACC
Human-Expert	Human	Visual observation	10 min	83.0
Human-Amateur	Human	Visual observation	10 min	64.0
GPT-5.2	VLM	Visual observation	Step 50	31.2
DFS-Heuristic	Deterministic search	Code / tabular state	Step 50	37.5
AntColony	Population-based search	Code / tabular state	Step 50	28.6
Genetic	Evolutionary search	Code / tabular state	Step 50	21.4
SimAnnealing	Local search	Code / tabular state	Step 50	19.6
Greedy	Greedy heuristic	Code / tabular state	Step 50	10.7
BFS	Exhaustive search	Code / tabular state	Step 50	1.8

Results. Table 6 reports stacking accuracy under this restricted-budget protocol. Several observations are worth noting:

1. Under matched step budgets, even strong symbolic baselines do *not* trivially solve stacking. Pure exhaustive BFS performs poorly (1.8%) because the branching factor of valid placements quickly exceeds the 50-step budget; greedy heuristics (Greedy, SimAnnealing, Genetic, AntColony) achieve modest accuracies in the 10–30% range; and only DFS-Heuristic, which exploits explicit geometric heuristics on the privileged symbolic state, surpasses GPT-5.2.
2. GPT-5.2 (31.2%) operating only on visual observations is competitive with or stronger than several symbolic baselines that have direct access to the underlying tabular state, indicating that VLM scores on **CHAIN** are not merely a reflection of trivially solvable instances.
3. Both human-expert (83.0%) and human-amateur (64.0%) substantially outperform all symbolic baselines and the strongest VLM, which is consistent with our human-baseline findings in Appendix A.6 and confirms that there is significant room for improvement on this benchmark.

Caveats. We emphasize that these symbolic / heuristic baselines should be interpreted as **contextual references** rather than directly comparable multimodal baselines. They are built on *code-level / tabular state information*, and do *not* need to infer structure from raw visual input. By contrast, VLMs in **CHAIN** rely only on images to understand the scene, infer constraints, and make sequential decisions. This makes the comparison inherently imperfect: symbolic search methods are solving a different and, in some sense, easier problem because they are given privileged structured state access that VLMs do not have. In particular, if the search budget is sufficiently relaxed, the strongest symbolic methods can in principle eventually recover the correct solution through direct state access and backtracking. The human baseline is more meaningful as a multimodal reference, since both humans and VLMs must reason from visual observations under physical constraints. Following this view, we present the symbolic baselines below primarily as *calibration points* for interpreting absolute VLM accuracies, not as drop-in substitutes for multimodal evaluation.

A.8 Diagnostic Controls: Memory, Budget, Perception, and Stability

This section details the controls summarized in Section 3.5. All controls use *Kimi-k2.5* on the same fixed 77-task Stacking subset and change only the stated factor. Each condition is a single run unless noted; 95% confidence intervals use 10,000 task-level bootstrap samples.

History length and action budget. We vary only the history window and the maximum number of steps (Table 7). The confidence intervals overlap substantially, and neither full history nor 100 steps yields a consistent gain. With the larger budget, no episode reaches the step limit; the model terminates early rather than exhausting its steps. This rules out a dominant memory or budget effect in this setting, but not for every **CHAIN** condition.

Visual vs. symbolic-coordinate input. To separate visual state understanding from planning, we give the same model either the standard images or an explicit text list of object geometry and poses

Table 7: History-window and action-budget ablation (Kimi-k2.5, 77 Stacking tasks).

History	Max steps	Pass@1 (95% CI)	Repeated-action rate	Reached step limit
Last 5 turns	50	16.9% (9.1–26.0)	15.3%	18.2%
Full history	50	14.3% (6.5–22.1)	22.7%	0.0%
Last 5 turns	100	16.9% (9.1–26.0)	16.7%	0.0%
Full history	100	18.2% (10.4–27.3)	22.0%	0.0%

(Table 8). The symbolic input provides no answers, legal-action list, or constraint graph. Both conditions use a history window of 3 and a 30-step budget, so their absolute numbers differ slightly from Table 7. Symbolic input more than doubles Pass@1 and sharply reduces illegal actions, showing that visual state understanding is an important difficulty. It does not solve the task, however: 49.4% of tasks remain unsolved, including 70.2% of Hard tasks. Moreover, tasks involving continuous contact, structural propagation, or dynamics cannot all be faithfully reduced to coordinate tables, so we treat the symbolic condition as a diagnostic control rather than a replacement for visual-world evaluation.

Table 8: Matched visual vs. symbolic-coordinate input (Kimi-k2.5, 77 Stacking tasks, history 3, 30 steps).

Input	Pass@1 (95% CI)	Illegal-action rate	Easy	Mid	Hard
Visual	20.8% (13.0–29.9)	72.6%	100.0%	25.0%	2.1%
Symbolic-coordinate	50.6% (39.0–62.3)	15.7%	100.0%	75.0%	29.8%

Two-run stability. We repeat the visual condition (history 3, 30 steps) twice with identical settings (Table 9). The runs are close and the paired-bootstrap CI of their difference includes zero. Two runs do not establish full-leaderboard stability; we present this as descriptive evidence only.

Table 9: Two-run stability of Kimi-k2.5 on 77 Stacking tasks.

Metric	Run 1	Run 2	Comparison
Pass@1	20.8% (16/77)	16.9% (13/77)	3.9 pp difference
Paired-bootstrap 95% CI of difference	–	–	-1.3 to 9.1 pp
Task-level agreement	–	–	93.5% (72/77)
Flip rate	–	–	6.5% (5/77)

A.9 Reward Models vs. Verifier-based Checking

Prior work on RLVR shows verification greatly aids domains with deterministic checking (e.g., math/code). We question if this applies to constraint-aware sequential planning—where success relies on long action sequences and errors may emerge late. Supervision is tougher here: reward models often favor locally plausible moves, whereas verifier-style checks are more conservative but environment-based.

We therefore compare RM-based selection [Ong et al., 2025] with verifier-based checking for choosing candidate actions/plans (e.g., execution-grounded validity/consistency), keeping the generator fixed, and testing lightweight strategies like reranking and pairwise judging and using Kimi-k2.5 as action model (Table 10).

For tractability, this study focuses on the two task families (*Puzzle* and *Stacking*) that most directly stress constraint gating and spatial commitment, where selection signals are hardest to design; we report the macro-averaged All score over these two families. Results show Pass@1 and Avg@4 are nearly identical, implying limited sampling variance. Multi-sample pass selection helps but saturates, while Pass@4 adds no further gain. RM reranking yields smaller improvements (+0.6), whereas a stronger VLM pairwise judge does better (+1.3) but still trails Pass@2. Beam search (batch size=2) with Reward also cannot match Pass@2, suggesting the bottleneck is selection-signal quality rather than decoding compute. Overall, current vision RMs offer limited leverage for long-horizon

Strategy	Puzzle(%) ↑	Stacking(%) ↑	All Macro(%) ↑	Δ vs. Avg@4 ↑
Avg@4	3.1	15.5	9.3	–
Pass@1	3.1	15.6	9.4	+0.1
Pass@2	3.1	19.4	11.2	+1.9
Pass@4	3.1	19.4	11.2	+1.9
VLM judge	3.1	18.1	10.3	+1.3
Reward model	3.1	16.8	9.9	+0.6

Table 10: Comparison of multi-sample selection and reward-model (RM) reranking by kimi-k2.5. All strategies use the same base generator and differ only in candidate selection; Puzzle and Stacking contain 32 and 78 task instances, respectively. Δ is computed w.r.t. Avg@4 on the macro-averaged All score.

interactive planning, and more reliable verifier-style signals—which are naturally available on the propagation-oriented tasks via execution-grounded checks—appear necessary for consistent gains.

A.10 Robustness to Shortcut Solutions and Benchmark Hacking

Closed-form puzzles and stacking configurations could in principle be susceptible to shortcut solutions—for example, memorizing a small set of canonical disassembly orders, or exploiting recurring visual patterns—that would inflate accuracy without reflecting genuine constraint-aware sequential planning. We design two complementary stress tests to probe whether such shortcuts dominate model behavior under the **CHAIN** protocol.

Test A: Inference-time selection with stronger reward signals. If a model already knows a high-quality answer but is bottlenecked by sampling noise, then stronger inference-time selection (best-of-K, reward-model reranking, or VLM-judge reranking) should produce large accuracy gains. As reported in Appendix A.9 (Table 10), we evaluate Pass@1, Pass@2, Pass@4, reward-model reranking, and VLM-judge reranking on *Puzzle* and *Stacking* with a fixed Kimi-K2.5 generator. The macro-averaged All score moves only from 9.3 (Avg@4) to at most 11.2 (Pass@2/Pass@4), and reward-model reranking yields just $+0.6$ over Avg@4. The fact that even strong selection signals cannot substantially raise accuracy indicates that failures on **CHAIN** are *not* concentrated in a small population of "almost-correct" samples that a verifier could rescue, but reflect a more fundamental difficulty in producing a single coherent multi-step plan.

Test B: Privileged final-answer hint. We additionally evaluate a stronger probe in which the **final goal answer is directly provided in the input prompt** (e.g., the target piece-removal order for puzzles, or the canonical placement layout for stacking), so that the remaining task is reduced to executing the prescribed action sequence faithfully rather than discovering it. Under this privileged setting, accuracy improves only by approximately **+3** percentage points relative to the standard closed-loop setting. The marginal gap between "with the answer in the input" and "without the answer in the input" indicates that the dominant difficulty in **CHAIN** lies in the *execution-time* reasoning required to translate a given target into a valid sequence of action–observation–update steps under physical constraints—perception of the current state, prediction of intermediate effects, and recovery from infeasible moves—rather than in inferring the high-level goal.

Implications. Together, the two tests place a tight envelope on the contribution of shortcut behavior to the reported scores: neither stronger selection over multiple samples (Test A) nor disclosing the goal answer at the input (Test B) is sufficient to close the gap to either human performance (Appendix A.6) or symbolic search baselines with privileged state access (Appendix A.7). This suggests that current model accuracies on **CHAIN** reflect persistent failures in closed-loop, constraint-aware sequential planning rather than artifacts of pattern memorization or limited sampling—an important property for a diagnostic benchmark of moderate size.

A.11 World-Model Prompt

Fig. 10 shows the content of our world model prompt.

Figure 10: World-model prompt for generating physically valid disassembly videos of Kongming/Luban locks (interlocking burr puzzles). Constraints include rigidity, collision avoidance, continuous motion, and axis-aligned sliding.

```
Generate a video showing the disassembly of a Kongming/Luban lock
(interlocking burr puzzle) from the exact configuration in the provided
reference image.
The video must be physically valid and obey these constraints:
• All parts are rigid wooden blocks (no bending or deformation).
• No interpenetration: contacts and collisions are respected; parts never
  pass through each other.
• No teleportation: motion is continuous over time.
• Each piece moves only by axis-aligned sliding along its allowed rail; no
  rotation until it is fully freed.
• Use a realistic unlocking sequence: remove the key piece first, then
  remove any newly unblocked pieces until fully separated.
• Match the reference geometry and part count exactly; do not add, remove,
  resize, or alter pieces.
```

A.12 Scope and Relation to Household Manipulation

CHAIN is a diagnostic microbenchmark, not a claim of direct transfer to household robotics. Full household tasks add perception, grasping, control, and hardware errors, which make planning failures harder to attribute. The relationship to household and VLA settings is structural: each coupling dimension captures a planning demand that recurs in everyday manipulation (Table 11). Household manipulation, tool use, and end-to-end control are not covered, and we leave a validated household extension to future work.

Table 11: Correspondence between **CHAIN** coupling dimensions and household/VLA planning demands.

CHAIN coupling	**Household / VLA analogue**
T1: Constraint Gating	Unlock or remove an obstruction before opening a drawer or cabinet
T2: Spatial Commitment	Preserve space and reachability during rearrangement or packing
T3: Structural Propagation	Reason about support, flow paths, or delayed contact effects

A.13 Limitations

We acknowledge the following limitations of the current CHAIN benchmark. **(1) Limited scale of interactive environments, especially for interlocking puzzles.** Our benchmark currently contains a finite set of *Puzzle* and *Stacking* instances (32 and 78, respectively), together with 30 instances each for FluidNet, PipeFit, and Domino, for a total of 200 task instances. While stacking instances are *programmatically generated* and can be scaled to near-unlimited complexity, high-fidelity interlocking puzzles require *careful environment engineering* to faithfully capture kinematic feasibility and contact-rich constraints, and we implement them in Unity for precise control of interlocking mechanics. As a result, each new puzzle environment incurs substantial manual modeling and debugging effort, leading to significant time and cost overhead, which currently limits benchmark scale.

(2) Current evaluation mainly reports macro-averaged Pass@1 due to high multi-step interaction cost. We primarily report **Pass@1** as our main success metric, computed as a macro-average over the five task families. This choice is partly driven by the evaluation cost of closed-loop interaction: each episode involves a non-trivial number of rounds, and we cap the interaction budget at **30–60 steps per instance**. We recognize that interactive tasks can exhibit run-to-run variability; however, our preliminary analysis in Appendix A.9 suggests that multi-sample evaluation (e.g., Avg@4) and Pass@1 show consistent trends, and a two-run stability analysis (Appendix A.8) finds 93.5%

task-level agreement. Two runs of one model do not establish full-leaderboard stability, so small differences between adjacent models should not be read as statistically meaningful. Looking forward, once sufficient API budget is available, we will report more robust best-of-K results (e.g., **Pass@4**) following the standard multi-run protocol.

(3) Controlled interface rather than realistic manipulation. CHAIN deliberately abstracts low-level actuation and simplifies object identification so that evaluation focuses on decision making, while models must still read the physical state from images. Objects carry distinct colors as stable identifiers, which reduces object-correspondence ambiguity; colors do not encode legal actions, blocking relations, action order, or solutions. Low-level contact physics is also abstracted: agents receive multi-view images and execution-failure feedback, but no tactile or force signals such as partial sliding or contact resistance. The benchmark therefore does not include a noisy-perception mode, and results should not be read as predictions of real-robot performance.

(4) Scope of the human baseline. Human results currently cover only Puzzle and Stacking, and the human time budget (10 minutes) is not resource-matched to the model's discrete action budget. We therefore use the human results as a reference point rather than as a controlled human–AI comparison, and do not claim a benchmark-wide human–AI gap.

(5) Scope of the diagnostic controls. The memory, budget, perception, and stability controls in Appendix A.8 use one model (Kimi-k2.5) on one family (Stacking). They rule out dominant protocol artifacts in that setting but do not characterize every model–family combination.

A.14 Benchmark Release and Reproducibility

To support transparent evaluation and reproducible follow-up research, we will publicly release the benchmark repository. The release is designed to make both the benchmark construction and the reported model evaluations independently inspectable. Specifically, the repository will include:

- **Task definitions.** Machine-readable task files for all benchmark instances, including task-family labels, difficulty labels, initial states, goal specifications, object metadata, action budgets, and reference solution information when available.
- **Simulator code and environment versions.** Source code for the Python task environments used by the voxel packing, flow, pipe-connection, and domino tasks, which implement task logic and interaction in Python on a PyBullet physics backend, together with their configurations and dependencies; and the Unity project or exported runtime assets used for interlocking puzzle environments. We will report the exact Python package versions, Unity editor/runtime version, operating-system assumptions, and rendering settings used in our experiments.
- **Unity setup instructions.** The Unity Puzzle environment is built and launched on Linux or Windows with an NVIDIA GPU, after which the Python client connects to it as a server. The README will specify supported systems, NVIDIA driver requirements, the Unity version, build and launch commands, the server/client start order, host/port configuration, and a single-instance smoke test.
- **Action API schema.** A complete specification of the fixed high-level action interface for each task family, including valid action formats, object identifiers, coordinate conventions, rotation conventions, illegal-action handling, and step-budget termination rules.
- **Evaluation scripts and verifiers.** Scripts for replaying model trajectories, executing actions in the corresponding simulator, computing task success, plan-efficiency metrics, token-efficiency metrics, and per-family breakdowns. We will also release deterministic verifier implementations and document numerical tolerances or rule-based tie-breaking whenever applicable.
- **Prompts and model outputs.** The prompts, system instructions, observation formatting, multi-view image packaging protocol, and parsing rules used for all evaluated models, along with the raw model outputs and executed trajectories supporting the reported tables.
- **Human baseline protocol.** The instructions, task interface description, grouping criteria for amateur and expert participants, recorded metrics, and aggregation procedure used for the human baseline reported in Appendix A.6.
- **Licensing and asset documentation.** A release license for the benchmark code and generated task assets, together with documentation of third-party assets, simulator dependencies, and any usage restrictions inherited from external sources.

For closed-source model evaluations, exact model weights cannot be redistributed; instead, we will provide the model identifiers, access dates, decoding parameters, interaction logs, and post-processing scripts needed to verify the reported results under the same protocol. This release plan is intended to separate three reproducibility layers: (i) reproducing benchmark instances and simulator transitions, (ii) replaying and verifying recorded trajectories, and (iii) rerunning model evaluations when the corresponding model APIs or checkpoints are available.