Sample-Efficient Post-Training for LEGO Spatial-Physics Reasoning

Anonymous ACL submission

Abstract

LLM-based LEGO assembly generation requires both semantic grounding and physical feasibility. We identify a data-induced failure mode, *PhysHack*, in which the assemblies satisfy physical-validity constraints while producing structures that are geometrically misaligned, semantically inconsistent, or poorly calibrated. To address this challenge, we propose a model-based data selection approach that uses only a small fraction of the training data while improving physically grounded LEGO assembly generation. Building on the selected trajectories, we introduce PVPO, a sample-efficient reinforcement learning method that couples physical feasibility with voxel-space geometric rewards. Our results show that physical validity alone is an insufficient proxy for reliable physical reasoning: models can learn to generate valid structures without preserving semantic or geometric fidelity. Experiments across model backbones and test-time scaling settings demonstrate that PVPO improves structural and semantic alignment, physical validity, structural stability, and calibration, while reducing reliance on extensive post-hoc rejection sampling. In particular, results on calibration show that PVPO mitigates PhysHack by making test-time selection more predictive of semantic and structural quality.

1 Introduction

LEGO Brick Assembly (LBA) (Pun et al., 2025; Kulits and Schmid, 2026; Ahn et al., 2022) focuses on creating real-world compositional objects from modular LEGO bricks, challenging the creativity as well as the precise spatial and physical reasoning abilities of generative models (Kingma and Welling, 2013; Ho et al., 2020; Vaswani et al., 2017). Recent advances in large language models (LLMs) (Grattafiori et al., 2024; Qwen et al., 2024; Achiam et al., 2023) have reframed LBA into a program synthesis task, where models generate executable assembly programs through language modeling. However, how post-training data shapes the LBA capabilities of LLMs remains underexplored. Although existing LBA datasets contain more than 200,000 examples (Kulits and Schmid, 2026; Pun et al., 2025), they also contain substantial noise, redundancy, and annotation errors, making it difficult to understand the role of data in LBA post-training (See Figure 3). Performance may depend not only on data scale, but also on which examples provide useful supervision for spatial-physics reasoning. Indeed, our failure-case analysis suggests that models trained on the full-scale dataset from Pun et al. (2025) may exhibit unexpected behaviors, such as satisfying physical constraints but failing to preserve the intended object semantics. This motivates us to ask:

Research Question

What makes a training example valuable for LEGO spatial-physics reasoning, and can a compact set of high-value examples enable more effective post-training than full-scale noisy supervision?

In this work, we first study LBA post-training through the lens of data valuation (Koh and Liang, 2017; Jia et al., 2019; He et al., 2016; Du et al., 2024; Liu et al., 2024c). Given a large pool of noisy LEGO reasoning trajectories, we ask which examples provide effective supervision for learning spatial-physics reasoning. Rather than treating all training examples as equally useful, we estimate their value by measuring the semantic consistency between the text description and the rendered LEGO structure, while filtering out examples that violate basic physical constraints. This allows us to construct a compact but high-value training subset, and to isolate the role of data quality and composition from that of data scale. This further motivates a second question:

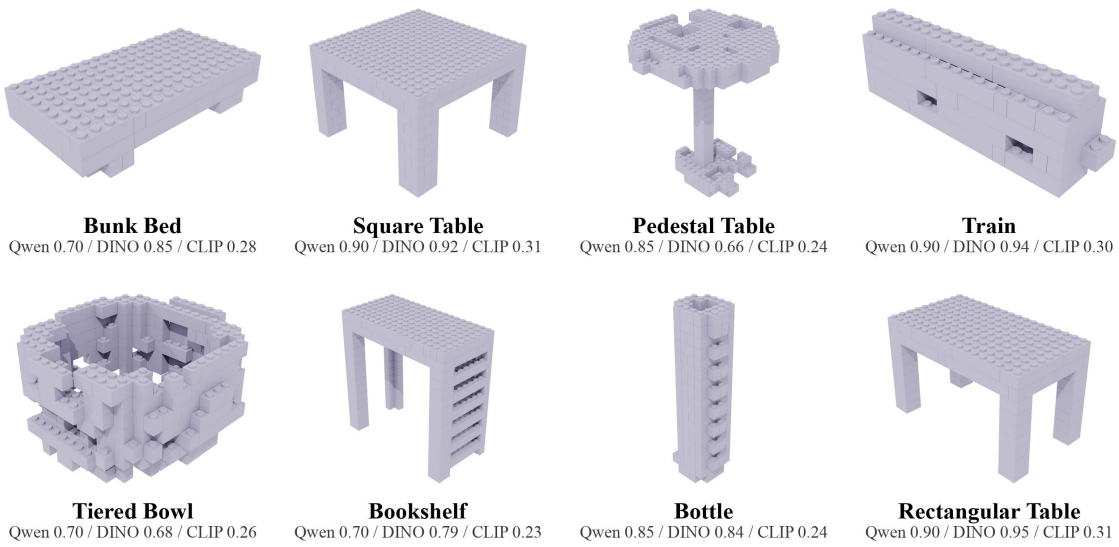


Figure 1: **Qualitative examples of PVPO-generated LEGO structures across diverse object categories.** The scores shown below each example correspond to Qwen-VL, DINOv3, and CLIP evaluations.

Research Question

If valuable data can improve post-training efficiency, how should the learning objective better exploit the verifiable spatial and physical structure of LEGO assemblies?

Data selection identifies useful supervision, but it does not by itself specify how a model should balance geometrical scene alignment and physical validity during generation. This is particularly important for LBA, where a generated program must satisfy low-level assembly constraints while still preserving the object semantics described by the prompt. To address this question, we introduce PVPO, a physics-informed reinforcement learning framework for LEGO spatial-physics reasoning. PVPO combines simulation-based physical-validity rewards with structure-aware geometric rewards, encouraging models to generate assemblies that are both physically feasible and semantically aligned with the intended 3D object structure. By coupling data valuation with verifiable post-training feedback, our framework provides a sample-efficient approach to improving LLM-based LEGO assembly generation (See qualitative examples in Figure 1). Our key contributions are as follows:

- **Data-Induced Physics-Reward Hacking.** We identify *PhysHack* phenonmenon in LBA. Models trained on noisy full-scale trajectories achieve high physical validity, e.g., 0.93 Validity@4 on Qwen, but remain weak in semantic alignment $0.59/0.67$ by Qwen-VL@4 and DINOv3@4 (See Table 1).
- **Data Valuation for Sample-Efficient Post-Training.** Motivated by *PhysHack*, we systematically study what makes LBA trajectories valuable for post-training. By comparing semantic, diversity, perplexity, and length-based selection signals, we find that VLM-based semantic valuation, together with domain diversity, identifies the most effective supervision. With only the top 5% examples, our method improves Qwen-VL@4 from 0.59 to 0.77, CLIP@4 from 0.26 to 0.28, and DINOv3@4 from 0.67 to 0.82 for Qwen (See Table 1).
- **Physics–Voxel Policy Optimization.** We further introduce PVPO, a physics-informed reinforcement learning framework that couples simulation-based physical-validity rewards with structure-aware geometric rewards. Under test-time scaling, PVPO consistently outperforms full-scale dataset on structural and semantic alignment (See Figure 2) and whole structure stability (See Figure 4). Figure 5 shows that PVPO improves confidence calibration, making high-confidence test-time selections more predictive of true semantic and structural quality.

2 PhysHack: Physical Validity as Hackable Proxy

We identify *PhysHack*, a misalignment phenomenon in LLM-based LEGO Brick Assembly

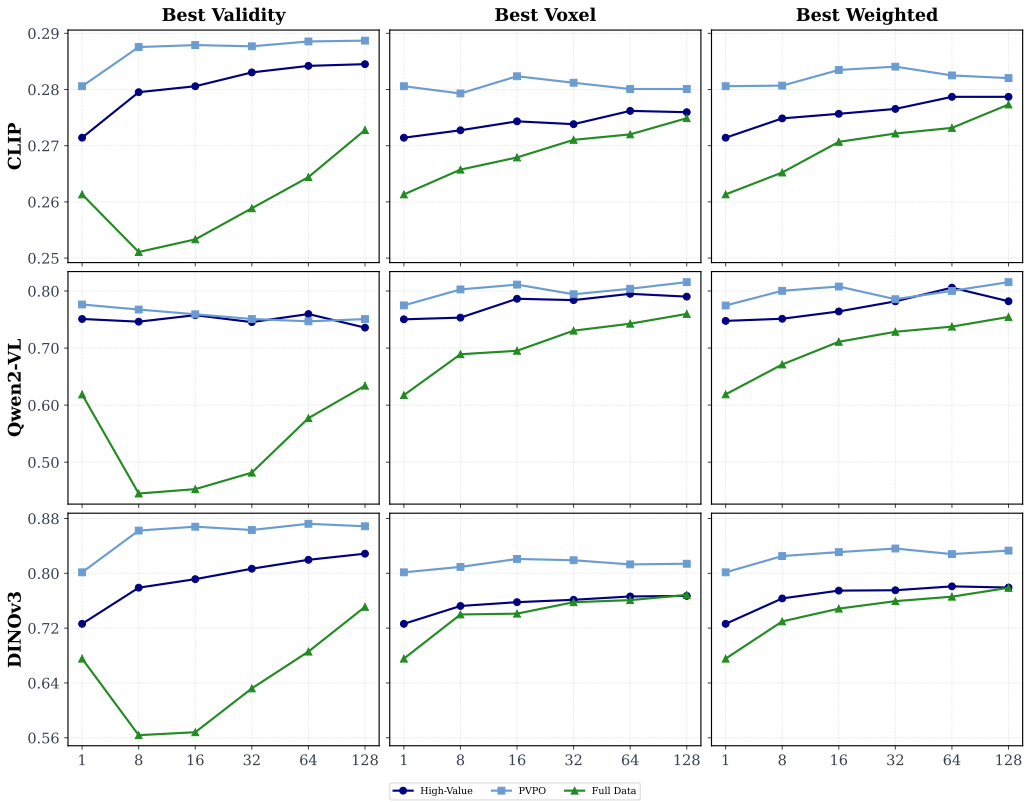


Figure 2: **Test-Time Scaling on Physics–Structure Alignment.** Best@K results evaluated by Qwen2-VL, CLIP, and DINOv3 under different selection metrics. PVPO consistently outperforms the full-dataset training baseline.

(LBA), where models achieve high measured physical validity by satisfying checkable assembly constraints while failing to preserve the intended object semantics and 3D structure.

2.1 Preliminaries

Language Modeling for Brick Assembly. Following Pun et al. (2025); Kulits and Schmid (2026), we represent each LEGO construction as an executable assembly program generated autoregressively by a language model. Given a text prompt x, the LLM policy π_θ produces a sequence of brick commands $o = (b_1, \ldots, b_T)$:

$$\pi_\theta(o \mid x) = \prod_{t=1}^{T} \pi_\theta\left(b_t \mid x, b_{<t}\right). \tag{1}$$

Each command b_t specifies the brick type and 3D voxel placement of one LEGO brick, and is serialized into 10 tokens. The generated program is then rendered into 3D brick structure via StableLego simulator Liu et al. (2024a).

Physical Validity. Physical validity measures whether a generated LEGO program can be instantiated as a feasible brick assembly. For bricks $o = (b_1, \ldots, b_T)$, each b_i must satisfy low-level constraints, including valid brick type, bounded placement, collision-free occupancy, etc. Let $P(o) \in [0, 1]$ denote the physical-validity score of o, where a higher value indicates that more generated bricks satisfy these constraints. However, physical validity alone does not guarantee semantic or structural correctness.

2.2 Measuring Physics-Structure Misalignment

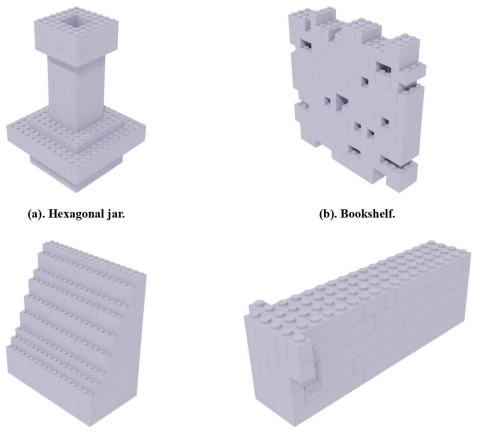
This gap is already visible in full-data training (See Table 1): Qwen achieves 0.93 on Validity@4, yet remain limited in semantic and structural alignment, with the full-data Qwen model reaching 0.59 Qwen-VL@4 and 0.67 DINOv3@4. We refer to this mismatch as *PhysHack*: models satisfy checkable physical constraints while producing structures that are misaligned with the target object. Figure 3 further illustrates this issue with low-alignment examples from the original dataset: although physically feasible, these assemblies often omit core object semantics, such as jars without container bodies, bookshelves as solid blocks, tables without clear tabletop-leg structures, and buses without wheels or windows. These cases suggest that *PhysHack* can be data-induced, arising from physically valid but semantically noisy supervision.

Setting	Qwen2.5-3B-Instruct						Llama-3.2-1B-Instruct					
	Qwen-VL ↑	CLIP ↑	DINOv3 ↑	Physics ↑	Voxel ↑	Bricks	Qwen-VL ↑	CLIP ↑	DINOv3 ↑	Physics ↑	Voxel ↑	Bricks
Full dataset	0.59	0.26	0.67	0.93	0.32	196	0.67	**0.27**	0.74	0.96	**0.35**	177
Diversity-only	0.58	0.26	0.66	**0.95**	0.32	163	0.55	0.25	0.66	0.94	0.31	199
Random subset	0.56	0.25	0.64	0.91	0.28	176	0.58	0.25	0.67	0.95	0.32	194
Low-value VLM	0.51	0.22	0.64	0.85	0.25	144	0.28	0.22	0.57	0.87	0.29	334
Shortest responses	0.50	0.24	0.65	0.91	0.22	33	0.49	0.24	0.62	0.89	0.22	38
Lowest perplexity	0.45	0.25	0.56	0.88	0.22	136	0.64	**0.27**	0.74	**0.97**	0.30	140
Longest responses	0.44	0.23	0.57	0.86	0.26	**346**	0.48	0.25	0.62	0.91	0.26	**351**
High-Value VLM	0.70	0.27	0.72	0.86	0.30	162	0.70	**0.27**	0.72	0.80	0.26	205
High-Value VLM + Diversity	0.72	0.26	0.70	0.86	0.31	184	**0.74**	**0.27**	**0.76**	0.89	0.32	181
PVPO	**0.77**	**0.28**	**0.80**	0.93	**0.35**	146	0.67	**0.27**	0.74	**0.97**	**0.35**	179

Table 1: **Comparisons of Data Selection**: Structure or semantic alignment (Qwen-VL/CLIP/DINOv3), physics validity, voxel alignment, and generated-brick statistics for models trained with different data-selection strategies.

Setting	SmolLM3-3B					
	Qwen-VL ↑	CLIP ↑	DINOv3 ↑	Physics ↑	Voxel ↑	Bricks
Full dataset	0.26	0.23	0.52	0.63	0.23	172
High-Value VLM + Diversity	0.67	0.26	0.68	0.68	0.26	236
PVPO	**0.77**	**0.28**	**0.78**	**0.86**	**0.28**	137

Table 2: Performance of SmolLM3-3B on semantic alignment, physics validity, voxel and brick number.



(a). Hexagonal jar. (b). Bookshelf. (c). Triangular table. (d). Bus with projecting front.

Score Interval	[0,0.5]	(0.5,0.8]	0.85	0.9	0.95	1.0
Percentage (%)	3.52	25.06	64.63	6.80	0.00	0.00

Figure 3: Examples of physically valid LEGO assemblies that fail to match the intended object semantics (with Qwen-VL score less than 0.4).

3 Value-Guided Data Selection for Sample-Efficient Post-Training

To investigate which data patterns mitigate *PhysHack* and enable efficient yet effective post-training, we treat data selection as a controlled testbed of how different supervision signals shape the learned policy.

3.1 Experimental Settings

Backbones and Baselines. We adopt following LLMs as backbones: Qwen2.5-3B-Instruct (Qwen et al., 2024), Llama-3.2-1B-Instruct (Grattafiori et al., 2024) and SmolLM3-3B (Bakouch et al., 2025). We compare against five baselines: using all raw data from Pun et al. (2025), **Perplexity**, selecting responses with the lowest perplexity under the full-data model, **Length**, selecting the longest or shortest responses, **Diversity-only**, sampling uniformly across domains, and **Low-value VLM**, selecting examples with the lowest VLM alignment scores, serves as a lower-bound ablation. All subset-based methods use the same budget, corresponding to 5% of the original training pool. We train all models with LoRA (Hu et al., 2022) using rank 32, a cutoff length of 4,096, and 12 epochs on 8× NVIDIA RTX 4090 GPUs.

Evaluation Metrics. For each prompt, we sample $K = 4$ LBA programs and report @4 metrics averaged over samples and evaluation prompts. We evaluate generated structures from three aspects. For semantic and visual-structural alignment, **Qwen-VL** (Bai et al., 2023) scores prompt-structure consistency in object identity, attributes, and spatial layout, **CLIP** (Radford et al., 2021) measures global image–text alignment, and **DINOv3** (Siméoni et al., 2025) measures visual structural similarity to the GT reference. **Voxel** measures voxel-space alignment between generated and reference structures, with details in Section 4, and **Bricks@4** records the average number of generated bricks. The evaluation set is identical from Pun et al. (2025).

3.2 Value-Guided Trajectory Selection

For each trajectory $\tau_i = (x_i, o_i)$, where x_i is the text description and o_i is the executable LEGO program, we render o_i into an image via Blender (See Section A.2) and use Qwen2.5-VL as a value model to score text–structure consistency:

$$V(\tau_i) = S_{\text{sem}}\big(x_i, \text{Render}(o_i)\big). \quad (2)$$

We formulate trajectory selection as an optimization problem to select a compact and diverse high-

value subset:

$$\mathcal{S}^\star = \arg\max_{\mathcal{S}\subseteq\mathcal{D}} \sum_{\tau_i\in\mathcal{S}} V(\tau_i) + \lambda \mathrm{Div}(\mathcal{S}) \quad \text{s.t.} \quad P(o_i) \geq \epsilon_{\text{phys}},\ \forall \tau_i \in \mathcal{S}, \quad |\mathcal{S}| = \rho|\mathcal{D}|. \tag{3}$$

Here, $P(o_i)$ denotes physical validity, $\rho = 0.05$ is the selection ratio, and $\mathrm{Div}(\mathcal{S})$ promotes domain coverage to avoid over-represented categories. In practice, we use domain-stratified top-K selection: within each domain, we rank physically valid trajectories by $V(\tau_i)$ and select the top examples under a fixed budget. This produces a $20\times$ smaller training set that preserves text–structure consistency, physical feasibility, and domain diversity.

3.3 Evaluation and Analysis

Small high-value subsets can outperform full-scale training. Tables 1 and 2 show that compact high-value subsets achieve stronger semantic and structural alignment than full-data training. Using only 5% of the original training pool, **High-Value VLM + Diversity** improves Qwen-VL@4 from 0.59 to 0.72 on Qwen, 0.67 to 0.74 on Llama, and 0.26 to 0.67 on SmolLM. It also improves DINOv3@4 from 0.67 to 0.70, 0.74 to 0.76, and 0.52 to 0.68, respectively. These results suggest that LBA post-training depends more on trajectory value than raw data scale.

Different selection signals reveal complementary data properties. The alternative selectors reveal complementary signals for LBA post-training. **Diversity-only** yields the strongest Qwen physical validity (0.95 Validity@4), but its Qwen-VL@4 and DINOv3@4 remain below High-Value VLM + Diversity, showing that coverage alone is insufficient. **Perplexity-based** selection captures syntactically regular programs and performs well on Llama, but transfers poorly to Qwen (0.45 Qwen-VL@4). **Length-based** selection shows that assembly complexity matters: shortest and longest responses produce very different brick counts (33 vs. 346 on Qwen), yet neither extreme yields strong semantic alignment. **Low-value VLM** serves as a negative ablation, dropping Llama Qwen-VL@4 to 0.28. Overall, useful LBA data requires both structural coverage and semantic alignment.

4 PVPO: Exploiting Physics–Structure Consistency via Reinforcement Learning

We introduce *Physics–Voxel Policy Optimization* (PVPO), a physics- and structure-aware reinforcement learning framework for LLM-based LEGO assembly generation. Although data selection removes explicitly misaligned trajectories, models may still exploit implicit dataset shortcuts, such as generic brick patterns, length distributions, base structures, or brick-type biases, that satisfy physical validity without preserving object-level semantic correctness (MacDiarmid et al., 2025; Cloud et al., 2025; Bowman et al., 2022).

4.1 Reward Modeling

Modeling Physical Validity as Reward. Given a generated program $o = (b_1, \ldots, b_T)$, let $v_t \in \{0, 1\}$ indicate whether brick b_t satisfies the required assembly constraints. We define physical validity as reward:

$$R_{\text{phys}}(o) = \frac{1}{T}\sum_{t=1}^{T} v_t, \tag{4}$$

to measure the fraction of illegal bricks.

Voxel-Space Geometric Alignment Reward. We measure geometric alignment in voxel space using Chamfer distance. Let $\mathcal{V}(o)$ and $\mathcal{V}(o^\star)$ denote the occupied voxel sets of the generated program o and the target construction $o^\star$, respectively. We compute a symmetric voxel-space Chamfer distance:

$$D_{\text{vox}}(o, o^\star) = \frac{1}{2}\left(\frac{1}{|\mathcal{V}(o)|}\sum_{u\in\mathcal{V}(o)} \min_{v\in\mathcal{V}(o^\star)} \|u - v\|_2^2 + \frac{1}{|\mathcal{V}(o^\star)|}\sum_{v\in\mathcal{V}(o^\star)} \min_{u\in\mathcal{V}(o)} \|u - v\|_2^2\right) \tag{5}$$

We normalize this distance and convert it into a reward:

$$R_{\text{vox}}(o, o^\star) = 1 - \min\left(\frac{D_{\text{vox}}(o, o^\star)}{d_{\max}}, 1\right) \tag{6}$$

where $d_{\max}$ is a normalization constant. Although this reward does not directly measure text-level semantics, the target construction $o^\star$ is conditioned on the input prompt, thus voxel reward provides a geometry-level signal for recovering the intended object shape and layout.

Coupled PVPO Reward. PVPO combines physical feasibility and voxel-space structural alignment:

$$R_{\text{PVPO}}(o, o^\star) = (1 - \lambda)R_{\text{phys}}(o) + \lambda R_{\text{vox}}(o, o^\star) \tag{7}$$

where λ_{vox} controls the strength of the structure-aware reward. In our main experiments, we set $\lambda_{\text{vox}} = 0.5$, which empirically yields the best performance (see Figure 4).

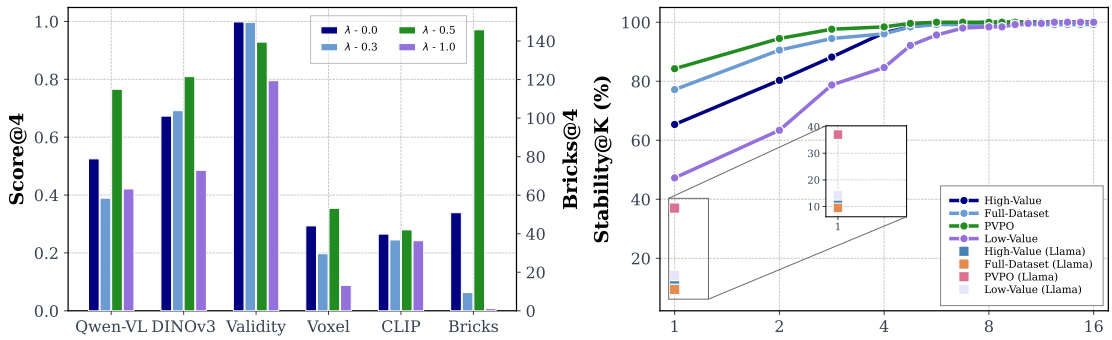


Figure 4: **Left**: Semantic alignment (Qwen-VL/CLIP/DINOv3), physics validity, voxel alignment, and generated-brick number under different voxel weight λ on Qwen2.5-3B-Instruct. **Right**: Stability@K (%) versus regeneration attempts K during the test-time inference on Qwen2.5-3B-Instruct (K=1–16) and Llama3.2-1B-Instruct (K=1).

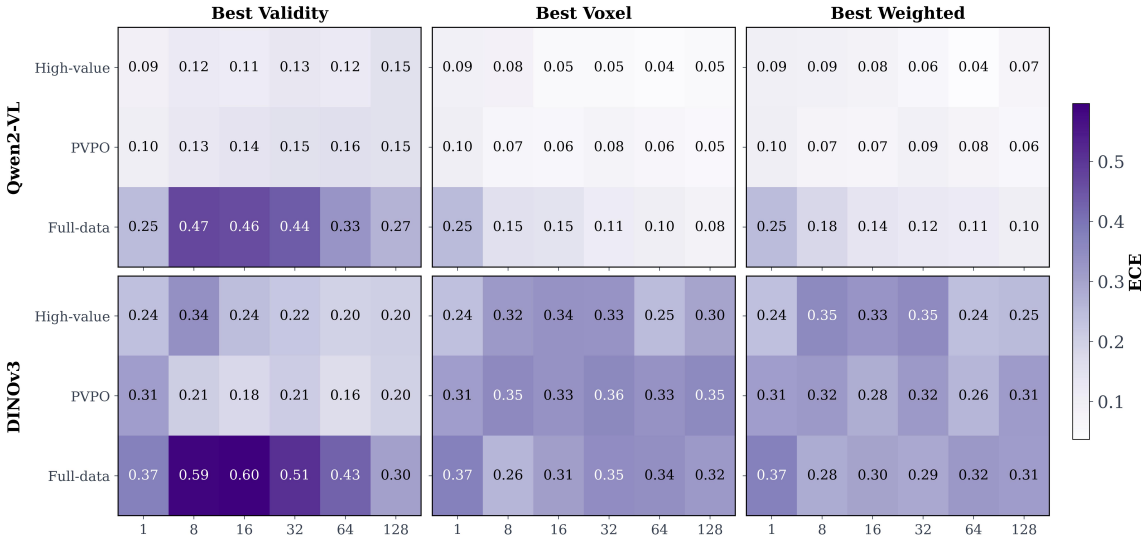


Figure 5: Confidence calibration measured by ECE under three best@k selection mechanisms. Row blocks correspond to confidence calibrated against Qwen2-VL and DINOv3 semantic alignment scores and columns correspond to Best Validity, Best Voxel, and Best Weighted selection. Within each block, rows compare High-value, PVPO, and Full-data methods across different inference test-time K (Qwen2.5-3B-Instruct).

4.2 RL Training Settings

We optimize PVPO with a GRPO-style training setup, following (Schulman et al., 2017; Yu et al., 2026; Shao et al., 2024; Liu et al., 2025; Cui et al., 2025). KL coefficient set to 0.001 to regularize the online policy against the frozen SFT reference policy. Entropy regularization is set to 0.0, we empirically find that larger entropy weights, such as 0.1 or 0.2, lead to policy collapse. We adopt token-level policy-gradient aggregation. To improve rollout utilization, each update batch mainly uses samples from the current policy, with a small portion reused from the previous policy via a replay buffer. For efficient training, we use LoRA with rank 32 and update only the adapter parameters. The RL stage uses the same dataset as SFT.

4.3 Evaluation and Analysis

Coupled rewards improve physics–structure consistency. Table 1 and Table 2 show that PVPO improves the balance among semantic alignment, geometric fidelity, and physical validity across model backbones. On Qwen, compared with full-data training, PVPO increases Qwen-VL@4 from 0.59 to 0.77, CLIP@4 from 0.26 to 0.28, and DINOv3@4 from 0.67 to 0.80. It maintains strong physical validity (0.93 Validity@4), improves Voxel@4 from 0.32 to 0.35, and reduces the average number of generated bricks from 196 to 146. It suggests that PVPO does not merely increase construction complexity, rather, the coupled

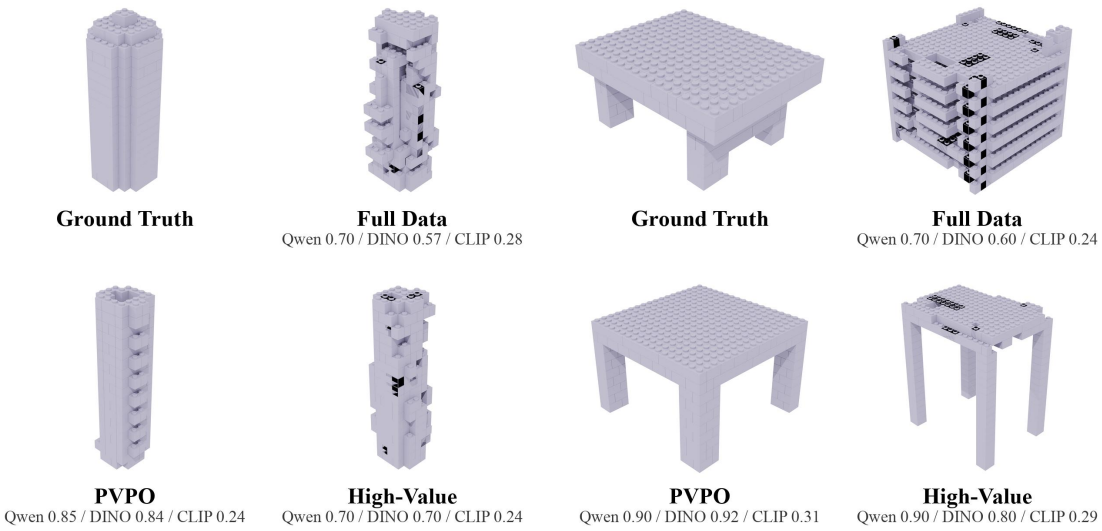


Figure 6: **Qualitative comparison on two representative LEGO generation tasks**. Bottle (left) and square table (right) show the generated structures from Full Data, PVPO, High-Value training, and ground truth. PVPO and High-Value produce cleaner geometry and better visual alignment than Full Data, black bricks indicate collisions.

reward promotes more compact and structurally faithful assemblies.

Figure 2 shows that the alignment advantage persists under test-time scaling with Best@K. Under best-by-validity selection, full-data training still exhibits weaker semantic alignment, with Qwen2-VL and DINOv3 scores remaining substantially below PVPO across most K values. By contrast, PVPO maintains strong alignment across different selection criteria: in the best-by-voxel and best-weighted settings, its Qwen2-VL and DINOv3 curves stay consistently above the full-data baseline. This suggests that PVPO improves correlation between test-time selection proxies and the desired physics–structure alignment. Figure 6 provides qualitative comparison for this trend.

Balanced reward coupling outperforms single-proxy optimization. Optimizing either reward alone leads to clear failure modes. With $\lambda = 0$, physics-only training reaches near-perfect Validity@4 (1.00), but yields much lower Qwen-VL@4, DINOv3@4, and Voxel@4 scores (0.52, 0.67, and 0.29), suggesting that physical feasibility alone can amplify PhysHack phenomenon. With $\lambda = 1.0$, voxel-only training also underperforms: Validity@4 drops to 0.79, while Qwen-VL@4, DINOv3@4, and Voxel@4 reach only 0.42, 0.49, and 0.08, respectively. This suggests that voxel reward becomes unreliable without physical-validity constraints. In contrast, the balanced setting $\lambda = 0.5$ achieves the best overall trade-off, with the strongest Qwen-VL@4 (0.76), DINOv3@4 (0.80), Voxel@4 (0.35), and CLIP@4 (0.28), while preserving high Validity@4 (0.93). These results show that PVPO benefits from coupling physical feasibility with geometric alignment, rather than maximizing either component in isolation.

PVPO improves stability under test-time scaling. Figure 4 shows that PVPO achieves high structural stability with fewer rejection samples. Structural stability is a more holistic metric than basic physical validity, as it evaluates whether the entire assembly remains physically stable (Details in Section A.3). While such stability typically requires extensive rejection sampling, PVPO starts at roughly 85% Stability@1 on Qwen, improves to about 95% at Stability@2, and reaches near-saturated stability by $K = 4$. In contrast, the low-value model starts much lower, around 40% Stability@1, and needs substantially more samples to reach comparable stability. Structural stability emerges as an additional capability induced by PVPO, even though it is not directly optimized as the primary objective.

5 Intriguing Insights and Discussion

PVPO Calibrates LLMs for Reliable Physical Reasoning. Figure 5 evaluates whether different test-time selection criteria are calibrated with downstream visual-structural quality. We compute the expected calibration error (ECE) (Guo et al., 2017) between each selection proxy and two evaluation

metrics, Qwen2-VL and DINOv3, under three selection rules: best-by-validity, best-by-voxel, and best-weighted selection. Lower ECE indicates that the proxy more reliably identifies candidates that are also semantically and structurally aligned. The results reveal a clear miscalibration pattern in full-data training. Under best-by-validity selection, the full-data model has much larger ECE than high-value SFT or PVPO: for Qwen2-VL, it reaches 0.47–0.46 at $K = 8$–16, while PVPO stays around 0.13–0.14, for DINOv3, full-data reaches 0.59–0.60, whereas PVPO remains around 0.21–0.18. This indicates that physical validity is poorly calibrated with semantic and structural correctness under full-data training. Voxel-based selection improves calibration for Qwen2-VL, reducing full-data ECE from 0.25 at $K = 1$ to 0.08 at $K = 128$, but remains less reliable for DINOv3, where ECE fluctuates across models. Best-weighted selection is more balanced: high-value SFT and PVPO keep low Qwen2-VL ECE across K, with PVPO decreasing from 0.10 to 0.06. These results suggest that combining physical and structural signals is more reliable than optimizing either proxy alone. Overall, this calibration study shows that *PhysHack* arises not only from physically valid but semantically wrong structures, but also from miscalibrated selection or reward proxies. PVPO mitigates this by coupling physical feasibility with voxel-space feedback, making test-time selection more predictive of semantic and structural quality.

6 Related Works and Concluding Remarks

LLM for Symbolic, Vision, and Physics Reasoning. LLMs, agentic workflows (Yao et al., 2022; Wei et al., 2022; Muennighoff et al., 2025), and post-training techniques (Shao et al., 2024; Yu et al., 2026; Schulman et al., 2017; Ouyang et al., 2022) have been increasingly adopted for general symbolic, vision, and physics reasoning (Johnson et al., 2017; Zhang et al., 2025a; Alrashedy et al., 2025; Chen et al., 2025; Qiu et al., 2025; Wang et al., 2023; Zheng and Bordes, 2026; Yu et al., 2025; Rodriguez et al., 2026; Zhang et al., 2025b; Verma et al., 2024; Lilienthal et al., 2026; Yang et al., 2025, 2024; Melnik et al., 2023; Bakhtin et al., 2019; Xue et al., 2023; Cherian et al., 2024; Liang et al., 2023). LEGO-Brick Assembly (LBA) is one representative setting that jointly tests these capabilities, requiring models to interpret high-level intents, reason over discrete 3D structures, and satisfy physical constraints during generation.

Generative Modeling and Physics Reasoning Pipeline for Brick Assembly. LBA poses a challenging task for generative models (Vaswani et al., 2017; Ho et al., 2020; Veličković et al., 2017), as it requires precise geometric understanding and physics-aware reasoning to synthesize intent-conditioned brick constructions that are both structurally aligned and physically stable (Wen et al., 2026; Guo et al., 2024; Ahn et al., 2022; Ge et al., 2024, 2025; Wang et al., 2022; Thompson et al., 2020; Tang et al., 2025). In particular, recent works (Pun et al., 2025; Kulits and Schmid, 2026; Xu et al., 2025; Guo et al., 2024) adopt pretrained autoregressive language models (Radford et al., 2019; Grattafiori et al., 2024; Qwen et al., 2024) and formulate LBA as a 3D program synthesis problem under a language modeling framework. Despite promising progress, several challenges remain in precise physics and vision requirement. First, physical stability often depends on costly post-hoc rejection sampling (Liu et al., 2024b), which does not improve the model's internal understanding of structural feasibility. Second, existing methods require substantial human effort for data generation and curation, yet still suffer from frequent structure–text misalignment. Moreover, how dataset-level factors affect learning, generalization, and physical reasoning remains poorly understood. Finally, reward design for RL-based physical reasoning remains challenging: rewards must support efficient rendering while balancing physical validity, structural feasibility, and downstream quality.

Concluding Remarks This work identifies *PhysHack* as a key bottleneck in LLM-based LEGO assembly generation. We address this challenge with a data-efficient learning framework that combines model-based data selection with PVPO, a sample-efficient RL method coupling physical feasibility and voxel-space structural alignment. Our approach calibrates the policy distribution toward generations that are physically valid, stable, compact, and faithful to the prompt.

7 Limitations

While our framework improves sample efficiency and physics–structure alignment for LEGO brick assembly, several limitations remain. First, due to limited computational resources, our experiments

focus on relatively small instruction-tuned backbones, including 3B-scale models such as Qwen2.5-3B-Instruct and SmolLM3-3B, as well as the 1B-scale Llama-3.2-1B-Instruct. Evaluating whether the same data-selection and PVPO trends hold for larger models remains an important direction for future work.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Seokjun Ahn, Jungtaek Kim, Minsu Cho, and Jaesik Park. 2022. Budget-aware sequential brick assembly with efficient constraint satisfaction. *arXiv preprint arXiv:2210.01021*.

Kamel Alrashedy, Pradyumna Tambwekar, Zulfiqar Haider Zaidi, Megan Langwasser, Wei Xu, and Matthew Gombolay. 2025. Generating cad code with vision-language models for 3d designs. In *International Conference on Learning Representations*, volume 2025, pages 52236–52262.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Anton Bakhtin, Laurens van der Maaten, Justin Johnson, Laura Gustafson, and Ross Girshick. 2019. Phyre: A new benchmark for physical reasoning. *Advances in Neural Information Processing Systems*, 32.

Elie Bakouch, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, Lewis Tunstall, Carlos Miguel Patiño, Edward Beeching, Aymeric Roucher, Aksel Joonas Reedi, Quentin Gallouédec, Kashif Rasul, Nathan Habib, Clémentine Fourrier, Hynek Kydlicek, Guilherme Penedo, Hugo Larcher, Mathieu Morlon, Vaibhav Srivastav, Joshua Lochner, and 4 others. 2025. SmolLM3: smol, multilingual, long-context reasoner. https://huggingface.co/blog/smollm3.

Samuel R Bowman, Jeeyoon Hyun, Ethan Perez, Edwin Chen, Craig Pettit, Scott Heiner, Kamilė Lukošiūtė, Amanda Askell, Andy Jones, Anna Chen, and 1 others. 2022. Measuring progress on scalable oversight for large language models. *arXiv preprint arXiv:2211.03540*.

Yamei Chen, Haoquan Zhang, Yangyi Huang, Zeju Qiu, Kaipeng Zhang, Yandong Wen, and Weiyang Liu. 2025. Symbolic graphics programming with large language models. *arXiv preprint arXiv:2509.05208*.

Anoop Cherian, Radu Corcodel, Siddarth Jain, and Diego Romeres. 2024. Llmphy: Complex physical reasoning using large language models and world models. *arXiv preprint arXiv:2411.08027*.

Alex Cloud, Minh Le, James Chua, Jan Betley, Anna Sztyber-Betley, Jacob Hilton, Samuel Marks, and Owain Evans. 2025. Subliminal learning: Language models transmit behavioral traits via hidden signals in data. *arXiv preprint arXiv:2507.14805*.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, and 1 others. 2025. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*.

Xinrun Du, Zhouliang Yu, Songyang Gao, Ding Pan, Yuyang Cheng, Ziyang Ma, Ruibin Yuan, Xingwei Qu, Jiaheng Liu, Tianyu Zheng, and 1 others. 2024. Chinese tiny llm: Pretraining a chinese-centric large language model. *arXiv preprint arXiv:2404.04167*.

Jiahao Ge, Mingjun Zhou, and Chi-Wing Fu. 2024. Learn to create simple lego micro buildings. *ACM Transactions on Graphics (TOG)*, 43(6):1–13.

Jiahao Ge, Mingjun Zhou, Hanyou Zheng, Hao Xu, and Chi-Wing Fu. 2025. Lego®-maker: Autoregressive image-conditioned lego® model creation. *ACM Transactions on Graphics (TOG)*, 44(6):1–15.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR.

Mengqi Guo, Chen Li, Yuyang Zhao, and Gim Hee Lee. 2024. Treesba: Tree-transformer for self-supervised sequential brick assembly. In *European Conference on Computer Vision*, pages 35–51. Springer.

Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. 2016. Dual learning for machine translation. *Advances in neural information processing systems*, 29.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3.

Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gürel, Bo Li, Ce Zhang, Dawn Song, and Costas J Spanos. 2019. Towards efficient data valuation based on the shapley value. In *The 22nd international conference on artificial intelligence and statistics*, pages 1167–1176. PMLR.

Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. 2017. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2901–2910.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pages 1885–1894. PMLR.

Peter Kulits and Cordelia Schmid. 2026. Bricknet: Graph-backed generative brick assembly. *arXiv preprint arXiv:2604.22984*.

Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. 2023. Code as policies: Language model programs for embodied control. In *2023 IEEE International conference on robotics and automation (ICRA)*, pages 9493–9500. IEEE.

Derek Lilienthal, Manisha Mukherjee, and Sameera Horawalavithana. 2026. Reward design for physical reasoning in vision-language models. *arXiv preprint arXiv:2604.13993*.

Ruixuan Liu, Kangle Deng, Ziwei Wang, and Changliu Liu. 2024a. Stablelego: Stability analysis of block stacking assembly. *IEEE Robotics and Automation Letters*, 9(11):9383–9390.

Tianqi Liu, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter Liu, and Jialu Liu. 2024b. Statistical rejection sampling improves preference optimization. In *International conference on learning representations*, volume 2024, pages 54605–54624.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2024c. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. In *International Conference on Learning Representations*, volume 2024, pages 22353–22373.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*.

Monte MacDiarmid, Benjamin Wright, Jonathan Uesato, Joe Benton, Jon Kutasov, Sara Price, Naia Bouscal, Sam Bowman, Trenton Bricken, Alex Cloud, and 1 others. 2025. Natural emergent misalignment from reward hacking in production rl. *arXiv preprint arXiv:2511.18397*.

Andrew Melnik, Robin Schiewer, Moritz Lange, Andrei Muresanu, Mozhgan Saeidi, Animesh Garg, and Helge Ritter. 2023. Benchmarks for physical reasoning ai. *arXiv preprint arXiv:2312.10728*.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori B Hashimoto. 2025. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 20286–20332.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Ava Pun, Kangle Deng, Ruixuan Liu, Deva Ramanan, Changliu Liu, and Jun-Yan Zhu. 2025. Generating physically stable and buildable brick structures from text. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14798–14809.

Zeju Qiu, Weiyang Liu, Haiwen Feng, Zhen Liu, Tim Xiao, Katherine Collins, Joshua B Tenenbaum, Adrian Weller, Michael J Black, and Bernhard Schölkopf. 2025. Can large language models understand symbolic graphics programs? In *International Conference on Learning Representations*, volume 2025, pages 26265–26311.

A Yang Qwen, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengpeng Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2. 5 technical report. *arXiv preprint*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and 1 others. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Juan Rodriguez, Haotian Zhang, Abhay Puri, Rishav Pramanik, Aarash Feizi, Pascal Wichmann, Arnab Mondal, Mohammad R Samsami, Rabiul Awal, Perouz Taslakian, and 1 others. 2026. Rendering-aware reinforcement learning for vector graphics generation. *Advances in Neural Information Processing Systems*, 38:60496–60534.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, and 1 others. 2025. Dinov3. *arXiv preprint arXiv:2508.10104*.

Kexian Tang, Junyao Gao, Yanhong Zeng, Haodong Duan, Yanan Sun, Zhening Xing, Wenran Liu, Kaifeng Lyu, and Kai Chen. 2025. Lego-puzzles: How good are mllms at multi-step spatial reasoning? *arXiv preprint arXiv:2503.19990*.

Rylee Thompson, Elahe Ghalebi, Terrance DeVries, and Graham W Taylor. 2020. Building lego using deep generative models of graphs. *arXiv preprint arXiv:2012.11543*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Aayush Atul Verma, Amir Saeidi, Shamanthak Hegde, Ajay Therala, Fenil Denish Bardoliya, Nagaraju Machavarapu, Shri Ajay Kumar Ravindhiran, Srija Malyala, Agneet Chatterjee, Yezhou Yang, and 1 others. 2024. Evaluating multimodal large language models across distribution shifts and augmentations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5314–5324.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*.

Ruocheng Wang, Yunzhi Zhang, Jiayuan Mao, Chin-Yi Cheng, and Jiajun Wu. 2022. Translating a visual lego manual to a machine-executable plan. In *European Conference on Computer Vision*, pages 677–694. Springer.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Haowei Wen, Ruixuan Liu, Weiyi Piao, Siyu Li, and Changliu Liu. 2026. Bricksim: A physics-based simulator for manipulating interlocking brick assemblies. *arXiv preprint arXiv:2603.16853*.

Hao Xu, Yuqing Zhang, Yiqian Wu, Xinyang Zheng, Yutao Liu, Xiangjun Tang, Yunhan Yang, Ding Liang, Yingtian Liu, Yuanchen Guo, and 1 others. 2025. Legoace: Autoregressive construction engine for expressive lego® assemblies. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pages 1–11.

Cheng Xue, Vimukthini Pinto, Chathura Gamage, Ekaterina Nikonova, Peng Zhang, and Jochen Renz. 2023. Phy-q as a measure for physical reasoning intelligence. *Nature Machine Intelligence*, 5(1):83–93.

Rui Yang, Hanyang Chen, Junyu Zhang, Mark Zhao, Cheng Qian, Kangrui Wang, Qineng Wang, Teja Venkat Koripella, Marziyeh Movahedi, Manling Li, and 1 others. 2025. Embodiedbench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents. *arXiv preprint arXiv:2502.09560*.

Yandan Yang, Baoxiong Jia, Peiyuan Zhi, and Siyuan Huang. 2024. Physcene: Physically interactable 3d scene synthesis for embodied ai. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16262–16272.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2026. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244.

Zhouliang Yu, Yuhuan Yuan, Tim Z Xiao, Fuxiang Frank Xia, Jie Fu, Ge Zhang, Ge Lin, and Weiyang Liu. 2025. Generating symbolic world models via test-time scaling of large language models. *arXiv preprint arXiv:2502.04728*.

Wenqian Zhang, Weiyang Liu, and Zhen Liu. 2025a. Agentic design of compositional machines. *arXiv preprint arXiv:2510.14980*.

Xinyu Zhang, Yuxuan Dong, Yanrui Wu, Jiaxing Huang, Chengyou Jia, Basura Fernando, Mike Zheng Shou, Lingling Zhang, and Jun Liu. 2025b. Physreason: A comprehensive benchmark towards physics-based reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16593–16615.

Yan Zheng and Florian Bordes. 2026. Voxelcodebench: Benchmarking 3d world modeling through code generation. *arXiv preprint arXiv:2604.02580*.

A Appendix

A.1 Text-Image Alignment Evaluation

Qwen-VL Text-Image Alignment: For VLM-based semantic alignment, we use **Qwen2.5-VL-7B-Instruct** as the evaluator, which is released under the Apache-2.0 license. The model is loaded with the HuggingFace Transformers interface. Given a rendered LEGO structure and its text description, Qwen-VL is prompted to output a scalar semantic alignment score in $[0, 1]$. Qwen-VL as a vision-language evaluator to assess how well each generated LEGO structure matches the input text description. The Qwen-VL based prompt template used to evaluate semantic alignment is shown below:

Prompt Template

Qwen2.5-VL Prompt. You are an expert evaluator specialized in assessing how well a text description matches a LEGO brick structure image.
Given the image of a LEGO model and the following text description, carefully evaluate the semantic alignment on a scale from 0.0 to 1.0:

- **1.0**: Perfect match, every important detail in the text is accurately and completely represented in the image.
- **0.9–0.99**: Excellent match, almost perfect, with only very minor visual differences.
- **0.7–0.89**: Good match, main shape, structure, and key features match, but some small details are missing or simplified.
- **0.4–0.69**: Moderate match, core idea is present, but with noticeable discrepancies.
- **0.0–0.39**: Poor match, significant mismatches, missing major elements, or irrelevant content.

Text description: `{text}`
Output **ONLY** the following JSON object, nothing else before or after:

```
{{
"score": 0.xx,
"reason": "One or two clear sentences explaining the main strengths and weaknesses of the match."
}}
```

CLIP-Based Text-Image Alignment: We use CLIP (Radford et al., 2021) to measure global text–image alignment. For each rendered LEGO image, the corresponding text prompt is encoded directly without any additional instruction template. The image and text representations are then compared with a CLIP scoring API, which returns a normalized similarity score between the visual representation and the textual description.

DINOv3 Image Similarity: In the DINOv3 text-alignment branch, we use **openai/clip-vit-base-patch32** as the CLIP text encoder. For DINOv3-based visual similarity, we use DINOv3 ViT-B/16 (Siméoni et al., 2025). Rendered LEGO images are encoded with DINOv3, and image-level similarity is computed using cosine similarity between normalized image features. When the text branch is enabled, the text description is encoded by the CLIP text encoder, and the DINOv3 image feature is compared with the CLIP text feature after dimensional alignment. Each rendered LEGO image is encoded by a pretrained DINOv3 vision encoder, and the resulting feature vector is L2-normalized. The score is computed as the cosine similarity between the generated image feature and the reference image feature. When multiple reference images are available, we average their DINOv3 features to obtain a single reference representation.

A.2 Rendering Details

All LEGO structures are rendered from LDraw (`.ldr`) files using Blender with the ImportLDraw plugin. We use the Cycles renderer with 64 samples and render each image at a resolution of 2048×2048. The camera field of view is fixed to $35°$. We use a studio-style setup with a pure white background and no ground plane. All bricks are rendered with a uniform light-purple material, with RGB value $(0.58, 0.48, 0.86)$ and roughness 0.55. Stud logos are disabled, brick gaps are enabled, and bevels are applied to brick edges with bevel width 0.5. The LDraw import scale is set to 0.02. Lighting is provided by two directional sun lights: a key light with energy 2.5 and a fill light with energy 1.2. We use Filmic/AgX tone mapping for the bricks and composite the transparent render onto a pure white background. These rendering settings are used only for visualization and do not affect quantitative evaluation.

Why not image-based rewards? We also explored using rendered images as references for reward modeling, such as computing visual alignment between generated and target assemblies. However, this is computationally prohibitive for reinforcement learning: rendering a single LEGO structure with Blender takes roughly one minute on our hardware, making rollout-time image rendering impractical. Therefore, PVPO uses voxel-space geometric alignment as a lightweight structure-aware reward, which can be computed directly from the

generated brick occupancy without invoking the rendering pipeline.

A.3 Physics-Guided Rejection Sampling and Regeneration

To evaluate the physical validity of generated LEGO structures, we use a two-stage stability-aware inference procedure following the computation protocol of Pun et al. (2025), with summary statistics reported in table 3. The first stage performs individual-brick-level rejection sampling, while the second stage applies full-structure-level stability-guided regeneration.

During autoregressive generation, the model predicts one LEGO brick at a time in the format $h \times w\,(x, y, z)$. Each candidate brick is checked for format validity, library membership, grid bounds, collisions, and duplicate invalid proposals. Invalid bricks are rejected, the model state is rolled back, and a new candidate is sampled, with a budget of up to 200 rejections per generated brick.

After a full structure is generated, we evaluate its physical stability using the BrickGPT stability analyzer. The analyzer produces a voxel-level stability score over the occupied volume of the structure, where larger values indicate greater stability and non-positive values indicate unstable regions. For each brick, we define its brick-level stability as the minimum stability score over all voxels occupied by that brick:

$$s_i = \min_{v \in \mathcal{V}_i} S(v), \tag{8}$$

where $S(v)$ denotes the voxel-level stability score and $\mathcal{V}_i$ is the set of voxels occupied by brick i. We summarize the structure using the mean and minimum brick stability:

$$S_{\text{mean}} = \frac{1}{N}\sum_{i=1}^{N} s_i, \qquad S_{\text{min}} = \min_i s_i, \tag{9}$$

where N is the number of generated bricks. A structure is considered physically stable if $S_{\text{min}} > 0$, meaning that even the weakest brick has positive stability.

If the generated structure is unstable, we apply stability-guided regeneration. Specifically, we identify the first unstable brick in the generation order, remove that brick and all subsequent bricks, and keep the remaining stable prefix. The model then continues generation from this stable prefix. This rollback-and-regeneration process can be repeated up to a predefined maximum number of regenerations. In the reported setting, we allow one structure-level regeneration. We also use a tiered regeneration protocol in which samples that already satisfy $S_{\text{min}} > 0$ are frozen and excluded from later regeneration rounds.

This procedure combines local syntactic and geometric filtering with global physical stability checking. Rejection sampling prevents invalid bricks from entering the structure, while regeneration corrects higher-level instability that only becomes apparent after evaluating the assembled model.

Gurobi-based Stability Optimization. We compute physical stability using a Gurobi force-equilibrium solver, students can get Gurobi license free. Each generated LEGO structure is converted into a voxelized brick assembly, and contact-force variables are introduced at brick interfaces. The optimizer enforces action-reaction consistency between contacting bricks and minimizes the total residual force and torque imbalance:

$$\mathcal{L}_{\text{eq}} = \sum_i \left(|\Delta F_{x,i}| + |\Delta F_{y,i}| + |\Delta F_{z,i}| + |\Delta \tau_{1,i}| + |\Delta \tau_{2,i}|\right). \tag{10}$$

We further add small regularization terms on the maximum downward contact force per brick and the total downward contact force:

$$\mathcal{L} = \mathcal{L}_{\text{eq}} + \alpha \sum_i F^{\text{max}}_{\text{down},i} + \beta \sum_j F_{\text{down},j}, \tag{11}$$

with $\alpha = 10^{-3}$ and $\beta = 10^{-6}$. The solver uses $g = 9.8$, LEGO unit height 0.0096, unit length 0.0078, and contact threshold $T = 100$, converted to $F_T = Tg/1000$.

After optimization, each occupied voxel is assigned a stability score. If force or torque equilibrium is violated, or if the contact-force margin is non-positive, the voxel score is set to zero. Otherwise, the score is the normalized margin

$$S(v) = \frac{F_T - D_{\text{max}}}{F_T}, \tag{12}$$

where D_{max} is the maximum downward contact force. The brick-level stability is the minimum voxel score over the brick, and the structure is considered stable when the minimum brick stability is greater than zero.

A.4 Dataset

Training dataset is brick-text paired, where each input is a natural-language description of an object and each output is an executable LEGO brick program. A brick program is represented as a list of bricks in the format `<brick dimension> (x,y,z)`, using a fixed library of 14 brick types: `1x1`, `1x2`, `2x1`, `1x4`, `4x1`, `1x6`, `6x1`, `1x8`, `8x1`, `2x2`, `2x4`, `4x2`, `2x6`, and `6x2`. We consider a full-data SFT setting with 213,020 text-structure pairs

Setting	Qwen2.5-3B-Instruct				Llama-3.2-1B-Instruct			
	% Valid	% Stable@1	Mean Stability@1	Min Stability	% Valid	% Stable@1	Mean Stability@1	Min Stability
Full dataset	99.61	77.17	0.9986	0.9739	9.45	9.45	0.9934	0.8890
Low-value VLM	100.00	47.24	0.9983	0.9598	100.00	14.17	0.9980	0.5952
High-Value VLM	100.00	65.35	0.9989	0.9736	18.90	12.60	0.9954	0.8837
PVPO	100.00	**84.25**	0.9992	0.9802	100.00	**37.01**	0.9970	0.3329

Table 3: Physical validity and stability statistics under rejection sampling (N=200) and stability evaluation (K=1).

Model	Usage	License
Qwen2.5-3B-Instruct	SFT / GRPO	Qwen Research
Llama-3.2-1B-Instruct	SFT / GRPO	Llama 3.2
SmolLM3-3B	SFT / GRPO	Apache-2.0
Qwen2.5-VL-7B-Instruct	VLM evaluation	Apache-2.0
CLIP ViT-B/32	VLM evaluation	MIT
DINOv3 ViT-B/16	VLM evaluation	dinov3-license

Table 4: Model usage and licenses.

from the open-source BrickGPT dataset (Pun et al., 2025), which is released under the MIT license. All subset-based SFT settings use 11k examples selected from this full dataset. These subsets are curated with different selection mechanisms, including high-value VLM, high-value VLM + Diversity, low-value VLM, diversity-only, random, longest-response, and shortest-response selection. For physics-guided reinforcement learning, we use 2k prompts in GRPO format. These prompts are deduplicated and selected as the highest-scoring 2k samples from the 11k high-value subset. Evaluation is conducted on a deduplicated set of diverse objects designed to emphasize physically meaningful construction principles.

A.5 Models

We use Qwen2.5-3B-Instruct (Qwen et al., 2024), Llama-3.2-1B-Instruct (Grattafiori et al., 2024), and SmolLM3-3B (Bakouch et al., 2025) for supervised fine-tuning and GRPO-based reinforcement learning. For evaluation, we use Qwen2.5-VL-7B-Instruct, CLIP, and DINOv3. The corresponding model licenses are summarized in table 4.

A.6 Language Model Disclosure

We use LLM to assist with minor manuscript polishing and LaTeX formatting. All technical content, experimental design, analysis, and final revisions were reviewed and verified by the authors.