From Diagnosis to Improvement: Probing Spatio-Physical Reasoning in Vision Language Models

Anonymous ACL submission

Abstract

Spatio-physical reasoning, which requires performing physical reasoning on the basis of spatial perception ability, is a foundation capability for understanding the real physics world. While recent vision language models (VLMs) have shown remarkable progress in specialized domains like multimodal mathematics and pure spatial understanding, their capability for spatio-physical reasoning remains largely unexplored. This paper provides a comprehensive diagnostic analysis of mainstream VLMs, revealing that current models perform inadequately on this crucial task. Further detailed analysis shows that this underperformance is largely attributable to biases caused by human-like prior and a lack of deep reasoning. To address these challenges, we apply supervised fine-tuning followed by rule-based reinforcement learning to Qwen2.5-VL-7B, resulting in significant improvements in spatio-physical reasoning capabilities and surpassing leading proprietary models. Nevertheless, despite this success, the model's generalization to new physics scenarios remains limited—underscoring the pressing need for new approaches in spatio-physical reasoning.

1 Introduction

The ability to intuitively reason about the physical world—predicting how objects interact under physical laws—is a cornerstone of intelligence. This capability, which we term **Spatio-Physical Reasoning**, involves both perceiving spatial information and reasoning with physical laws based on that perception. For example, when performing a block-stacking task (Groth et al., 2018), the agent must not only recognize the spatial relationships involved in stacking but also integrate this spatial information with the center-of-mass equilibrium principle that governs physical stability. The spatio-physical reasoning ability is a fundamental prerequisite for any agent to operate effectively in the real world.

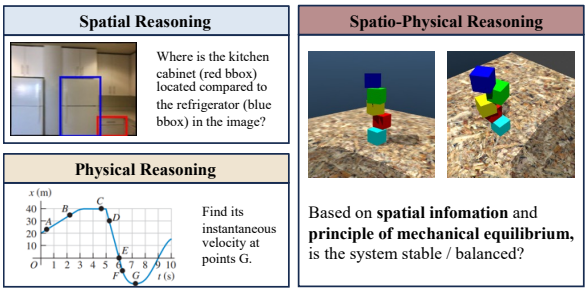

Figure 1: Spatio-physical reasoning tasks require the integration of spatial information with the laws of physics. Task examples are adopted from publicly available datasets (Zhang et al., 2025a; Shen et al., 2025; Groth et al., 2018).

Recent advances in Vision Language Models (VLMs) have shown remarkable capabilities in various tasks. However, existing research on their reasoning abilities has been largely confined to specific domains like multimodal mathematics (Meng et al., 2025) or spatial understanding (Ouyang et al., 2025; Liao et al., 2025), which involves only spatial relations and requires no physical reasoning. When physics is involved, it is often in well-defined, textbook-style contexts (Shen et al., 2025; Xiang et al., 2025; Dai et al., 2025; Zheng et al., 2025), leaving a critical gap in understanding VLMs' performance in spatio-physical reasoning tasks.

Bridging this gap is essential for ambitious AI goals like World Models, as a robust internal model of physical outcomes is a potential pathway toward AGI (Assran et al., 2025). Furthermore, strong physical cognition directly improves performance in downstream embodied AI tasks (Chow et al., 2025). In this work, we first conduct a diagnostic analysis of how mainstream VLMs behave in spatio-physical reasoning, and then investigate whether fine-tuning can remedy their deficiencies and improve generalization.

Model	Accuracy	Difficulty Bias		Height Bias					Duplicated Height Bias		
		Easy	Hard	2	3	4	5	6	2	4	6
InternVL3-8B	0.542	0.794	0.385	0.938	0.805	0.472	0.221	0.249	0.996	0.700	0.252
InternVL3-14B	0.547	0.711	0.405	0.984	0.819	0.505	0.013	−0.162	0.988	0.622	−0.354
InternVL3-38B	0.547	0.997	0.967	0.999	0.993	0.978	0.942	0.868	1.000	0.939	0.804
InternVL3-78B	0.575	0.430	−0.311	0.764	0.264	−0.144	−0.317	−0.421	1.000	0.798	0.076
Qwen2.5VL-7B	0.522	0.218	−0.115	0.059	0.281	−0.107	0.118	−0.094	−0.032	0.038	−0.179
Qwen2.5VL-32B	0.546	0.927	0.335	0.938	0.692	0.692	0.573	0.317	0.258	0.514	−0.036
Qwen2.5VL-72B	0.547	0.967	0.596	0.724	0.781	0.933	0.928	0.715	0.018	1.0	0.722
Gemma3-12B-it	0.507	−0.698	−0.714	−0.674	−0.681	−0.713	−0.735	−0.736	−0.653	−0.695	−0.748
Gemma3-27B-it	0.550	−0.533	−0.663	−0.297	−0.582	−0.640	−0.705	−0.733	−0.145	−0.524	−0.736
GPT-4o	0.560	-	-	-	-	-	-	-	-	-	-
o4-mini	0.593	-	-	-	-	-	-	-	-	-	-
o3	0.641	-	-	-	-	-	-	-	-	-	-
Human Expert	0.779	-	-	-	-	-	-	-	-	-	-

Table 1: Accuracy and behavioral modeling parameters. T_{pref} is used to quantify models' bias. A positive T_{pref} indicates the model tends to answer true, while a negative score indicates a preference for false.

Our main contributions are: (1) We conduct a comprehensive diagnosis of VLMs' spatio-physical reasoning, revealing that their underperformance stems not just from errors, but from systematic, human-like biases and a lack of deep reasoning. (2) We demonstrate the duality of current fine-tuning methods: while a supervised fine-tuning (SFT) + reinforcement learning (RL) pipeline can achieve state-of-the-art in-domain accuracy, its limited ability to generalize to new physical contexts suggests this learning is superficial. This highlights a critical limitation of current paradigms, which excel at pattern matching but fail to instill robust, generalizable physical laws.

2 Related Works

Enhancing Reasoning Ability in VLMs Recent efforts to enhance reasoning in VLMs have focused on specialized domains. In multimodal mathematical reasoning, approaches such as two-stage reinforcement learning (Peng et al., 2025) and online filtering (Meng et al., 2025) have achieved notable success. In parallel, spatial reasoning has been advanced through large-scale video or 3D datasets and tailored training schemes to improve understanding of spatial layouts (Ouyang et al., 2025; Daxberger et al., 2025).

Physical Reasoning Research in physical reasoning follows two main directions: evaluation and enhancement. Early evaluation benchmarks relied on physics simulators to test intuitive dynamics and statics (e.g., IntPhys (Riochet et al., 2020), ShapeStacks (Groth et al., 2018), Physion (Bear et al., 2021)), while recent benchmarks assess symbolic knowledge (Zhang et al., 2025b) or real-world physical understanding (Chow et al., 2025). In contrast, our work targets a more structured and physical fine-grained task: static stability analysis. Enhancement efforts either integrate external modules for contextual scene modeling (Balazadeh et al., 2025) or pursue end-to-end learning of general physical commonsense (NVIDIA et al., 2025), which may not prioritize fine-grained physical laws.

3 Modeling the Reasoning Behavior of VLMs

We analyze how VLM performs in spatio-physical reasoning on a prototypical task, i.e., judging the stability of stacked objects using the ShapeStacks benchmark (Groth et al., 2018) generated with MuJoCo engine (Todorov et al., 2012). The block-stacking task is essentially analogous to many important real-world tasks. For example, determining how to stack and store goods in a factory to ensure stability, or how to distribute mass in construction to maintain structural integrity. This benchmark features stacked objects of varying heights (2-6) and difficulty levels, requiring models to reason by combining the spatial information of the stacked system with the centre of mass criterion. The model receives 16 images of the same stacking system captured from different angles at the same moment and is asked to classify the system as balanced ('True') or unbalanced / will collapse ('False') based on multiple viewpoints. Our analysis, conducted on an 888-sample test subset, examines model performance, reasoning strategies, error types, and

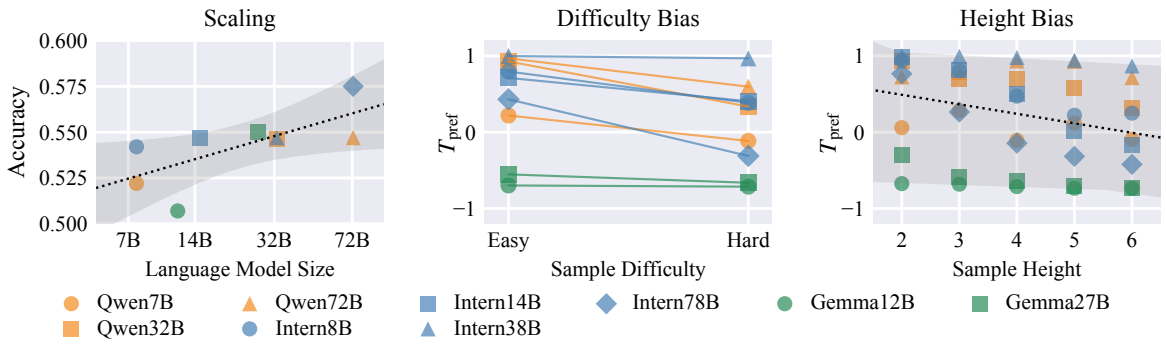

Figure 2: Left: Model accuracy exhibits a positive correlation with the log size of the language model component. Center: Model bias (T_{pref}) shifts towards predicting "False" on hard samples. Right: Models become increasingly biased towards "False" as stack height increases. In the left plot, trend line and 95% CI grey shadow are fitted using linear regression. In the right plot, the shaded envelope of trend lines for each model and dash line for overall trend is fitted with linear mixed-effect model.

inherent biases. Further details about the dataset are provided in Appendix A.

3.1 Evaluation

We evaluated nine mainstream open-source VLMs (≥7B) and several proprietary models for reference (Table 1). All models underperformed significantly. Open-source models failed to surpass 0.6 accuracy, and even the leading proprietary model, o3, only reached 0.641. Both are well below the human expert baseline of 0.779 (details in Appendix B.1 & B.2).

Furthermore, we find that model performance scales with the size of its language model component. A linear regression analysis reveals a weak but statistically significant positive correlation between accuracy and the log-size of the language model (slope = 0.0352, $p = 0.0360$, 95% confidence interval (CI) $[0.0031, 0.0673]$; Figure 2, left). This suggests that scaling laws still apply to spatio-physical reasoning, and performance is more tightly coupled with the language model's capacity rather than the vision encoder. Full details are in Appendix B.4.1.

3.2 Cognitive Behavior and Error Cause

To understand the underlying causes of these underperformance, we conduct a manual analysis of their Chain-of-Thought (CoT) responses to characterize their logical steps, identify error sources, and examine the role of high-level cognitive behaviors.

3.2.1 Three Main Error Causes

Through manual analysis of the model's CoT, we observe a clear step-by-step reasoning pattern. The model first interprets the problem by identifying the task and analyzing the visual input, then proceeds to apply physical principles to reach a final conclusion.

Notably, the model often adopts a process of elimination approach: checking whether any object violates stability conditions and concluding stability only if none is found.

Based on the analysis of CoT, we identify three primary sources of failure: **(1) Visual Perception Errors**, where the model captures salient features like object count and color but fails on fine-grained spatial details such as layer misalignment; **(2) Physical Reasoning Errors**, which involve applying incorrect physical principles, like miscalculating the center of mass for a lying cylinder as if it were upright; and **(3) Causal Reasoning Errors**, where models make faulty logical leaps, most commonly by misinterpreting visual stillness as evidence of physical stability (i.e., "the system hasn't moved, so it must be stable"), despite the prompt explicitly stating that the images show a static moment.

Among these, visual perception errors are most critical, since incorrect visual inputs inevitably cause faulty reasoning. Causal errors, by contrast, mainly stem from flawed logical priors rather than perceptual or physical ones.

3.2.2 Rich High-level Cognitive Behaviors with Limited Utility

Inspired by previous research (Gandhi et al., 2025), which identifies high-level cognitive behaviors as key to performance in symbolic reasoning, we investigate whether these same behaviors can also boost VLMs' performance in spatio-physical reasoning. The four cognitive behaviors under consideration are: **Verification**: checking or compar-

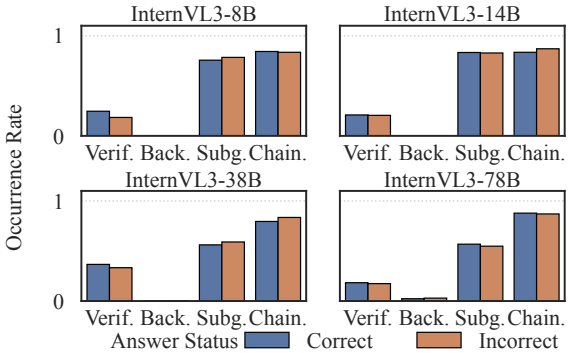

Figure 3: The proportion of advanced reasoning behaviors in correct and incorrect responses of the InternVL model series shows no significant difference.

ing a generated result to confirm its correctness. **Backtracking**: abandoning a failed reasoning path and explicitly attempting a new strategy. **Subgoal Setting**: decomposing a complex problem into a sequence of smaller, more manageable subproblems. **Backward Chaining**: reasoning backward from the desired goal to infer the necessary preconditions to achieve it.

To investigate whether the presence of these behaviors correlates with correctness, we used the o4-mini model to annotate the responses from the InternVL series. We then compared the distribution of these behaviors across correct and incorrect responses (details in Appendix C).

As shown in Figure 3, for all InternVL models, the distribution of cognitive behaviors does not significantly differ between correct and incorrect responses. The model is simply performing superficial rather than genuinely deep reasoning. This suggests that the quality and content of the reasoning process matter more than the mere presence of cognitive behavior patterns.

3.3 Bias with Human-like Prior

We next investigate if VLMs, pre-trained on human-generated data, exhibit human-like biases in spatio-physical reasoning. Specifically, we consider two priors: the *Difficulty Prior*, where greater inter-object displacement are considered as instability, and the *Height Prior*, where taller stacks are perceived as less stable. To quantify this, we analyze the confusion matrix of model predictions and define the preference score T_{pref}:

$$T_{\text{pref}} := \tanh(\psi), \quad \psi := \frac{\text{Recall} - \text{Specificity}}{\text{Specificity}} \tag{1}$$

where Recall $:= \frac{\text{TP}}{\text{TP+FN}}$ and Specificity $:= \frac{\text{TN}}{\text{TN+FP}}$ indicate the proportion of correctly predicted True and False samples, respectively. The metric T_{pref} reflects the model's preference: a positive score indicates a bias toward "True", while a negative score indicates a bias toward "False".

3.3.1 Difficulty Bias with Inter-object Displacement Prior

For the inter-object displacement prior, we analyze the model's preference score T_{pref} on easy and hard samples. As shown in Figure 5, hard samples involve large misalignment in both stable and unstable cases, which can mislead the model. We hypothesize that if the model has a human-like bias, it will assign lower T_{pref} scores to hard samples, mistakenly judging balanced stacks as unbalanced.

Analysis of the response data supports our hypothesis. As shown in Table 1 and Figure 2, all models yield lower T_{pref} scores on hard samples, indicating they are misled by large visual displacements. This consistent pattern suggests a human-like prior that equates noticeable inter-object displacement with instability. The models "elimination-based" reasoning process exacerbates this bias, as they often identify a single large misalignment in hard samples and prematurely conclude the structure is unbalanced.

3.3.2 Height Bias with Height Prior

Our analysis reveals a strong height bias in most VLMs, where taller stacks are more likely to be judged as unstable. As shown in Table 1 and Figure 2, T_{pref} scores consistently decrease with increasing stack height. A linear mixed-effects model confirms this observation with a statistically significant negative trend (slope = -0.1244, $p = 0.0018$,95% CI: $[-0.2025, -0.0463]$).

To test the robustness of this bias, we create duplicated samples: based on $h = 2$ cube samples, we generate $h = 4$ and $h = 6$ versions by vertically stacking identical cubes without altering horizontal positions (Figure 4a). Although these samples differ in height, their mechanical structure remains essentially the same.

On these duplicated samples, the height-related bias persists for most models, with T_{pref} scores decreasing as height increases (Figure 4b, Table 1). A linear mixed-effects model confirms this with a statistically significant negative correlation (slope = -0.1008, $p = 0.0389$, 95% CI: $[-0.1965, -0.0051]$). However, the Qwen series

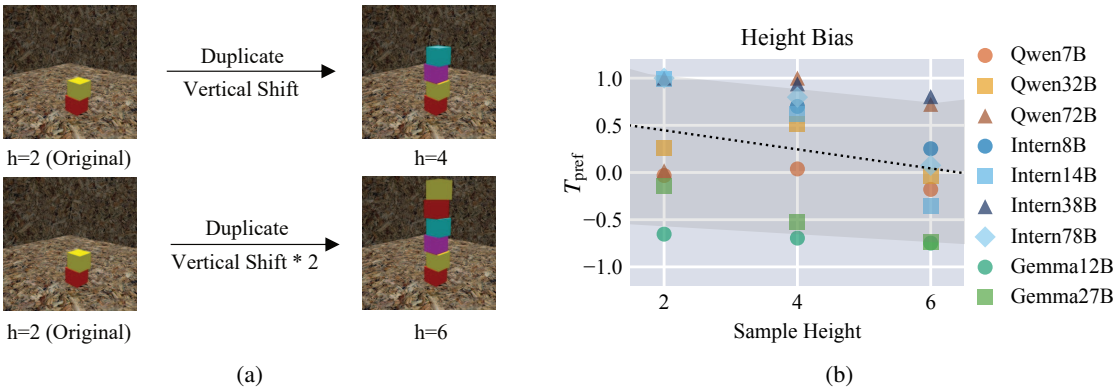

Figure 4: (a) Generate higher samples with highly similar mechanical structures through duplication and shift. (b) Bias still holds for most models on duplicated samples. Fitting with a linear mixed-effects model reveals a general decreasing trend as height increases. Although certain models show an increasing tendency to predict "true" with greater height, this dose not affect the overall trend. The grey shadow is the envelope of trend lines for each model.

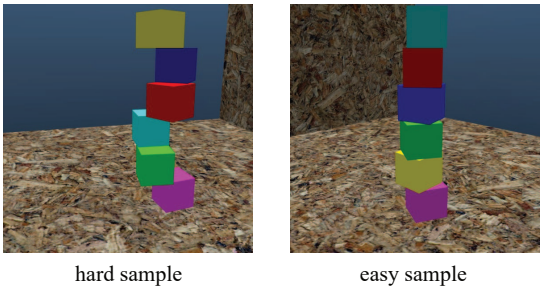

Figure 5: Easy and hard samples in the ShapeStacks benchmark. Although both examples shown are in equilibrium, the hard sample exhibits noticeable misalignment between different layers, which is deceptive.

is a notable exception, as its bias disappears on these controlled samples, suggesting its prior is more brittle and less systematic. Fitting details are in Appendix B.4.2.

These two experiments confirm that a human-like height prior is a common VLM feature, though its strength varies across model families. We speculate that this height bias is influenced by real-world data distributions, where higher objects are often less stable than lower ones. Models trained on such data are likely to inherit this bias. Furthermore, the varying strength of this bias across different VLMs may be attributed to their distinct training methodologies.

3.4 Transfer of Prior Knowledge

To investigate prior knowledge transfer, we evaluate two specialized models: SpaceR (Ouyang et al., 2025) with a strong spatial prior, and MM-Eureka-Qwen-7B (Meng et al., 2025) with a strong physical prior. We compare their performance against their base model, Qwen2.5-VL-7B on ShapeStacks. As shown in Table 2, despite their strong priors, these models exhibit almost no accuracy improvement, indicating no performance gain. This suggests that fine-tuning on a single domain is insufficient for compositional reasoning.

Model	Accuracy
Qwen2.5-VL-7B	0.522
MM-Eureka-Qwen-7B	0.521
SpaceR	0.522

Table 2: Prior Knowledge Transfer: Possessing spatial or physical reasoning abilities alone makes it difficult to improve performance in spatio-physical reasoning.

4 Improvement through Fine-Tuning

Building on our diagnosis, this section investigates if fine-tuning can remedy the identified VLM limitations. We demonstrate that a two-stage supervised fine-tuning (SFT) following reinforcement learning (RL) training pipeline on Qwen2.5-VL-7B achieves state-of-the-art performance on ShapeStacks, surpassing leading proprietary models. However, when we test this improved model's generalization to novel physical scenarios (across dimensions, dynamics, and heights), we find its performance degrades as the domain gap widens, indicating limited generalization.

4.1 In-domain Performance Improvement

4.1.1 Experimental Setup

Our fine-tuning data is the ShapeStacks training set (13,618 samples). We construct the SFT dataset by

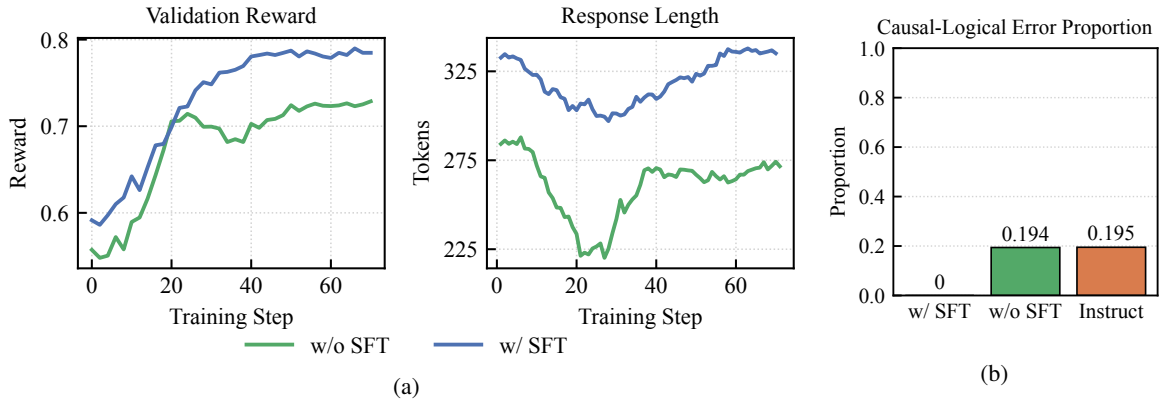

Figure 6: The benefits of SFT as an initialization for RL. We compare RL fine-tuning with SFT initialization (w/ SFT) against RL fine-tuning from the Instruct model (w/o SFT). (a) SFT initialization leads to more stable validation rewards and maintains longer responses during training. (b) By annotating model's response with o4-mini, we find that SFT also eliminates the causal-logical errors that are present without it. Details are provided in Appendix E.5.

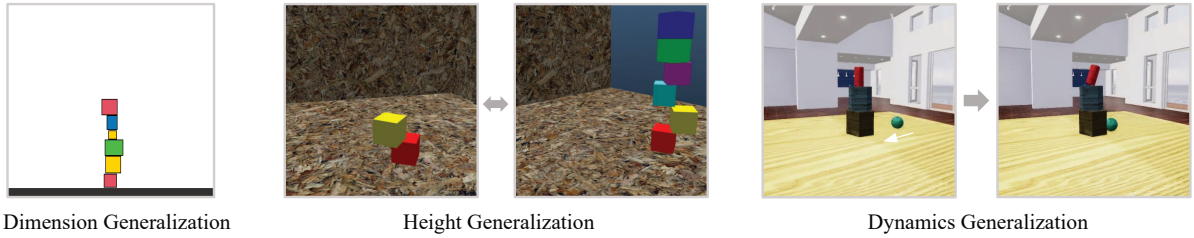

Figure 7: An illustration of the three physics-based generalization tests. Dimensional generalization tests transfer between 2D and 3D data. Height generalization tests transfer between low and high stacks. Dynamics generalization tests transfer from static stability to dynamic scenarios involving external forces (Physion).

using the o4-mini model to distill CoT-style reasoning responses for 3,000 randomly selected samples (seed=0). Remaining samples are formatted for RL with outcome supervision. Details of distillation are provided in Appendix D.2.4.

4.1.2 Training

We fine-tune Qwen2.5-VL-7B in a two-stage process. First, we perform supervised fine-tuning (SFT) using LoRA via the TRL framework (von Werra et al., 2020). Subsequently, we apply reinforcement learning (RL) with the GRPO algorithm (Shao et al., 2024) using veRL framework (Sheng et al., 2025) and full-parameter fine-tuning. The RL reward is a weighted sum of a 10% binary format score, which ensures the model separates its reasoning and final answer with special tokens, and a 90% binary correctness score based on the final answer. The resulting model is named Qwen-SFT-RL, with detailed hyperparameters provided in Appendix D.1.

4.1.3 Evaluation

We evaluate the Qwen-SFT-RL model on the ShapeStacks test set used in the previous section. The results are shown in Table 3. As the table indicates, SFT+RL significantly improves the model's performance, achieving a 47.7% gain over the Instruct model and outperforming the leading proprietary model o3. This demonstrates that the SFT+RL fine-tuning paradigm can effectively enhance the performance of VLMs on stacked-system stability tasks.

4.1.4 Ablation Study on SFT

We conduct an ablation study to understand SFT's role. We create a Qwen-RL model by applying RL directly to the base model without the SFT. We also evaluate the Qwen-SFT model, which is fine-tuned using SFT only. The hyperparameters are the same as those used for training Qwen-SFT-RL.

As shown in Table 3, Qwen-RL achieves substantial 42.7% accuracy improvement but remains below Qwen-SFT-RL's 47.7%, indicating RL alone is suboptimal. While Qwen-SFT shows minimal improvement (0.4%), training dynamics analysis (Fig-

Models	Test Samples				
	ShapeStacks	ShapeStacks high	ShapeStacks low	2d-towers	Physion Support
Qwen2.5VL-7B	0.522	0.580	0.531	0.510	0.504
Qwen-SFT-RL (ShapeStacks)	**0.771 / 47.7%**	**0.781 / 34.7%**	**0.799 / 50.5%**	0.658 / 29.0%	0.509 / 1.0%
Qwen-high (ShapeStacks high)	0.606 / 16.1%	**0.678 / 16.9%**	0.573 / 7.9%	0.670 / 31.4%	0.493 / -2.2%
Qwen-low (ShapeStacks low)	0.677 / 29.7%	0.683 / 17.8%	**0.751 / 41.4%**	0.661 / 29.6%	0.507 / 0.6%
Qwen-2d (2d-towers)	0.613 / 17.4%	0.674 / 16.2%	0.609 / 14.7%	**0.918 / 80.0%**	0.522 / 3.6%
Qwen-RL (ShapeStacks)	**0.745 / 42.7%**	**0.766 / 32.1%**	**0.792 / 49.2%**	0.520 / 2.0%	0.5 / -0.8%
Qwen-SFT (ShapeStacks)	**0.524 / 0.4%**	**0.532 / -8.3%**	**0.521 / -1.9%**	0.590 / 15.7%	0.493 / -2.2%

Table 3: Metrics in Generalization Ability Analysis. In the "Model" column, the dataset used for fine-tuning is indicated in brackets after the model name.The fine-tuned model evaluation results are presented in the format: "accuracy / improvement over the pre-fine-tuned Instruct model". Bolded numbers indicate in-domain results.

ure 6) reveals SFT initialization benefits. Qwen-RL exhibits overfitting (validation reward drops sharply around step 20), fails to maintain response length, and retains the base model's causal reasoning error rate, unlike error-free Qwen-SFT-RL.

This suggests that RL with outcome supervision alone is insufficient to guide reasoning. SFT serves two critical roles: (1) it provides a robust initialization for RL, enhancing training stability and final performance, and (2) its CoT data constrains the model to learn correct reasoning logic.

4.2 Generalization Ability Test

We investigate whether the model can generalize specific spatio-physical reasoning patterns through post-training.

We explore three types of generalization. **Dimensional generalization** tests the model's ability to transfer reasoning across dimensions, specifically between 3D (ShapeStacks) and 2D data. **Dynamics generalization** assesses whether the model, trained on static stacking, can generalize to dynamic scenarios from the Physion benchmark (Bear et al., 2021). **Height generalization** investigates if the model learns abstract principles rather than memorizing patterns, which we test by training on low- or high-stack data and evaluating on the unseen height category.

Three types of generalization are illustrated in Figure 7. In all generalization experiments, the fine-tuned models are based on Qwen2.5-VL-7B. Except for differences related to the training data—such as the content of the fine-tuning set, the step of the final checkpoint and number of epochs—all other hyperparameters remain the same as those used for Qwen-SFT-RL. Details are provided at Appendix D.1.

4.2.1 Dimension Generalization: Asymmetry Between 3D and 2D

Although 3D analysis involves an extra spatial dimension, the core physical principle of analyzing the center of mass projection remains the same. We investigate whether a model trained on data from one spatial dimension can generalize its reasoning capability across different dimensions.

For 2D stacked system data, we generate samples using the Box2D physics engine (Catto., 2009), with tower heights ranging from 3 to 6 (as 2D height-2 towers are overly simple), and refer to this dataset as *2d-towers*. The full dataset is split to training set and test set. Details are provided in Appendix D.2.1.

We first test the 3D-to-2D generalization by evaluating our 3D-trained Qwen-SFT-RL on the 2d-towers dataset, observing a significant 29.0% performance gain (Table 3). In the reverse direction, we fine-tune a model on 2d-towers (Qwen-2d). While Qwen-2d achieves a strong 80% in-domain accuracy boost on 2d-towers, its generalization gain to the 3D ShapeStacks is a more modest 17.4%. This highlights an asymmetry in dimensional generalization, with 3D-to-2D transfer being more effective.

This suggests that while dimensional generalization is possible, it is more effective from 3D to 2D. We hypothesize this is because 3D samples provide richer spatial information, making it easier for a 3D-proficient model to handle simpler 2D cases.

4.2.2 Dynamics Generalization: Hard to Generalize

We next explore the generalization from static to dynamic spatio-physical reasoning. Specifically, we evaluate Qwen-SFT-RL on the *support scenarios* of the *Physion* benchmark (Bear et al., 2021).

Physion is a dynamic classical mechanics benchmark that contains a variety of physical scenes. In the support scenario, a stacked geometry system is subjected to an external force, requiring models to predict whether the structure remains stable or collapses, and how the collapse unfolds if it does. The experiment aims to assess whether a model that has learned reasoning about physical stability can transfer the understanding to dynamic scenarios.

The model fails to generalize, showing only a 1% improvement (Table 3). This indicates that even with underlying static equilibrium principles present, the model cannot transfer its knowledge to dynamic domains. A large visual discrepancy between static and dynamic samples is the likely explanation, hindering the application of learned reasoning patterns to new visual contexts.

4.2.3 Height Generalization: Easier From Low to High

We investigate height generalization to determine if models learn abstract principles or memorize height-specific patterns. To do this, we partition the training and test sets into ShapeStacks-low (heights 2-3) and ShapeStacks-high (heights 4-6) subsets. We then fine-tune two specialized models, Qwen-low and Qwen-high, on their respective training subsets and cross-evaluate their performance on the unseen height category, using a 1,000-sample test set for each. Details of datasets are provided in Appendix D.2.2.

As shown in Table 3, Qwen-low achieves stronger in-domain performance and better height generalization than Qwen-high. The generalization from high to low yields only a 7.9% improvement, while generalization from low to high shows a 17.8% gain, which even surpassing the in-domain performance of Qwen-high (a 16.9% gain). On mixed-height ShapeStacks set, Qwen-low achieves a 29.7% performance increase, significantly higher than Qwen-high's 16.1%.

These findings indicate that VLMs are capable of height generalization in both directions, but are more effective from lower to higher samples. We attribute this asymmetry to the increased visual and physical complexity of high-stack samples, which can lead to challenging, uninformative gradients during training. In contrast, simpler low-stack samples facilitate the learning of correct reasoning strategies, which results in stronger performance and better generalization.

4.2.4 Performance Drop with Domain Shift

Our final analysis evaluates the models across all benchmark scenarios. The results are shown in Table 3, where performance improvements on the in-domain samples are highlighted in bold. We observe a clear trend: *the fine-tuned VLM is capable of meaningful generalization to closely related domains, but this capability decays as the domain gap between test samples and train samples widens.*

Specifically, our cross-model analysis reveals a clear pattern of performance decay as the domain gap widens. For instance, the significant in-domain gains of models like Qwen-SFT-RL and Qwen-low diminish on moderately different tasks (e.g., 2D generalization) and vanish almost entirely on distant tasks. Critically, generalization from static to dynamic scenarios (Physion) proves to be the hardest challenge, with nearly all fine-tuned models showing negligible to no improvement over the baseline, regardless of their training data.

In addition, the comparison shows that Qwen-SFT-RL consistently outperforms Qwen-RL across all generalization test sets, indicating that SFT also contributes to improving the generalization ability of the fine-tuned model.

5 Conclusion

Our comprehensive analysis reveals a critical disconnect between the apparent capabilities and the actual physical understanding of current open-source VLMs. We first identify their fundamental limitations, including weak performance, a lack of deep reasoning and human-like cognitive biases. We then demonstrate that while mainstream fine-tuning can offer significant in-domain performance gains, this improvement is largely superficial. The models show limited generalization to novel physical scenarios, especially as the domain gap increases, suggesting a reliance on statistical shortcuts rather than robust physical principles. This exposes a fundamental limitation of current paradigms: they excel at pattern matching but fail to instill the generalizable understanding required for world models.

Our findings suggest that the path forward lies not in merely scaling existing methods, but in developing new approaches. Future work should prioritize strategies, such as leveraging physically-grounded simulation data and designing novel training paradigms, that explicitly encourage the learning of causal and generalizable physical laws.

Limitations

We acknowledge three main limitations of this work.

First, our study is restricted to the task of judging static stability in block-stacking scenarios. While we have argued for the importance of this setting, it does not encompass the full range of real-world applications. Many spatio–physical reasoning tasks involve dynamics rather than purely static analysis. Moreover, our task requires models to passively observe and answer questions, rather than act as agents that interact with the environment. We leave the exploration of dynamic tasks and agent-based paradigms to future work.

Second, due to computational and time constraints, we are unable to evaluate all open-source VLMs. We focus on mainstream dense architectures with LLM backbones, while leaving out alternative paradigms such as mixture of experts (MoE). For proprietary models, whose internal architectures are unspecified, we limit our analysis to performance testing rather than deeper probing.

Finally, this paper is primarily diagnostic: we model the behavioral characteristics of existing models and analyze the generalization of current fine-tuning methods. We do not propose novel solution methods here, but rather aim to complement method-oriented research with a systematic analysis that highlights key challenges for future improvement. In future work, we will further investigate how to effectively enhance models' spatio-physical reasoning capabilities.

Ethic Statement

The datasets used in this work are either publicly available or synthetically generated through physics simulation engines, modeling only simple physical scenarios without involving any personal or sensitive information. In addition, our self-constructed 2D-tower dataset will be released upon publication to facilitate future research.

We recruite three experts with relevant academic backgrounds. Prior to their participation, each expert is verbally briefed on the study's objectives and the task involved. They are informed that their performance data will be collected and used in an anonymized format for academic comparison, and all provided verbal consent to participate. To ensure fair remuneration for their time and expertise, participants are compensated at a rate equivalent to five times the local minimum hourly wage.

This study is diagnostic in nature and does not involve the deployment of new models or algorithms. The tasks focus on spatio-physical reasoning over geometric objects, and therefore do not raise concerns related to social bias, discrimination, or fairness. It should be noted that indiscriminately generalizing our findings to all vision models may lead to misinterpretations and inaccurate conclusions.

Our findings show that current VLMs still exhibit limitations in spatio-physical reasoning. If directly applied to safety-critical downstream applications such as autonomous driving or robotics, these shortcomings may pose risks. We believe that identifying these gaps contributes to improving the robustness of VLMs and ultimately enables safer and more reliable applications.

References

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba, Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, and 11 others. 2025. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *Preprint*, arXiv:2506.09985.

Vahid Balazadeh, Mohammadmehdi Ataei, Hyunmin Cheong, Amir Hosein Khasahmadi, and Rahul G. Krishnan. 2025. Physics context builders: A modular framework for physical reasoning in vision-language models. *Preprint*, arXiv:2412.08619.

Daniel Bear, Elias Wang, Damian Mrowca, Felix Binder, Hsiao-Yu Tung, Pramod RT, Cameron Holdaway, Sirui Tao, Kevin Smith, Fan-Yun Sun, Fei-Fei Li, Nancy Kanwisher, Josh Tenenbaum, Dan Yamins, and Judith Fan. 2021. Physion: Evaluating physical prediction from vision in humans and machines. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.

Erin Catto. 2009. Box2d: Box2d is a 2d physics engine for games. https://github.com/erincatto/box2d. Accessed: 2025-5-30.

Wei Chow, Jiageng Mao, Boyi Li, Daniel Seita, Vitor Campagnolo Guizilini, and Yue Wang. 2025. Physbench: Benchmarking and enhancing vision-language models for physical world understanding. In *The Thirteenth International Conference on Learning Representations*.

Song Dai, Yibo Yan, Jiamin Su, Dongfang Zihao, Yubo Gao, Yonghua Hei, Jungang Li, Junyan Zhang, Sicheng Tao, Zhuoran Gao, and Xuming Hu. 2025.

Physicsarena: The first multimodal physics reasoning benchmark exploring variable, process, and solution dimensions. *Preprint*, arXiv:2505.15472.

Erik Daxberger, Nina Wenzel, David Griffiths, Haiming Gang, Justin Lazarow, Gefen Kohavi, Kai Kang, Marcin Eichner, Yinfei Yang, Afshin Dehghan, and Peter Grasch. 2025. Mm-spatial: Exploring 3d spatial understanding in multimodal llms. *Preprint*, arXiv:2503.13111.

Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D. Goodman. 2025. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars. *Preprint*, arXiv:2503.01307.

Oliver Groth, Fabian B. Fuchs, Ingmar Posner, and Andrea Vedaldi. 2018. Shapestacks: Learning vision-based physical intuition for generalised object stacking. In *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part I*, page 724–739, Berlin, Heidelberg. Springer-Verlag.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Zhenyi Liao, Qingsong Xie, Yanhao Zhang, Zijian Kong, Haonan Lu, Zhenyu Yang, and Zhijie Deng. 2025. Improved visual-spatial reasoning via r1-zero-like training. *Preprint*, arXiv:2504.00883.

Fanqing Meng, Lingxiao Du, Zongkai Liu, Zhixiang Zhou, Quanfeng Lu, Daocheng Fu, Tiancheng Han, Botian Shi, Wenhai Wang, Junjun He, Kaipeng Zhang, Ping Luo, Yu Qiao, Qiaosheng Zhang, and Wenqi Shao. 2025. Mm-eureka: Exploring the frontiers of multimodal reasoning with rule-based reinforcement learning. *Preprint*, arXiv:2503.07365.

NVIDIA, :, Alisson Azzolini, Junjie Bai, Hannah Brandon, Jiaxin Cao, Prithvijit Chattopadhyay, Huayu Chen, Jinju Chu, Yin Cui, Jenna Diamond, Yifan Ding, Liang Feng, Francesco Ferroni, Rama Govindaraju, Jinwei Gu, Siddharth Gururani, Imad El Hanafi, Zekun Hao, and 35 others. 2025. Cosmos-reason1: From physical common sense to embodied reasoning. *Preprint*, arXiv:2503.15558.

Kun Ouyang, Yuanxin Liu, Haoning Wu, Yi Liu, Hao Zhou, Jie Zhou, Fandong Meng, and Xu Sun. 2025. Spacer: Reinforcing mllms in video spatial reasoning. *Preprint*, arXiv:2504.01805.

Yingzhe Peng, Gongrui Zhang, Miaosen Zhang, Zhiyuan You, Jie Liu, Qipeng Zhu, Kai Yang, Xingzhong Xu, Xin Geng, and Xu Yang. 2025. Lmm-r1: Empowering 3b lmms with strong reasoning abilities through two-stage rule-based rl. *Preprint*, arXiv:2503.07536.

Ronan Riochet, Mario Ynocente Castro, Mathieu Bernard, Adam Lerer, Rob Fergus, Véronique Izard, and Emmanuel Dupoux. 2020. Intphys: A framework and benchmark for visual intuitive physics reasoning. *Preprint*, arXiv:1803.07616.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Hui Shen, Taiqiang Wu, Qi Han, Yunta Hsieh, Jizhou Wang, Yuyue Zhang, Yuxin Cheng, Zijian Hao, Yuansheng Ni, Xin Wang, Zhongwei Wan, Kai Zhang, Wendong Xu, Jing Xiong, Ping Luo, Wenhu Chen, Chaofan Tao, Zhuoqing Mao, and Ngai Wong. 2025. Phyx: Does your model have the "wits" for physical reasoning? *Preprint*, arXiv:2505.15929.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2025. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, EuroSys '25, page 1279–1297, New York, NY, USA. Association for Computing Machinery.

Emanuel Todorov, Tom Erez, and Yuval Tassa. 2012. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. 2020. Trl: Transformer reinforcement learning. `https://github.com/huggingface/trl`.

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

Jiahui Zhang, Yurui Chen, Yanpeng Zhou, Yueming Xu, Ze Huang, Jilin Mei, Junhui Chen, Yu-Jie Yuan, Xinyue Cai, Guowei Huang, Xingyue Quan, Hang Xu, and Li Zhang. 2025a. From flatland to space: Teaching vision-language models to perceive and reason in 3d. *Preprint*, arXiv:2503.22976.

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. *Preprint*, arXiv:2502.12054.

Shenghe Zheng, Qianjia Cheng, Junchi Yao, Mengsong Wu, Haonan He, Ning Ding, Yu Cheng, Shuyue Hu, Lei Bai, Dongzhan Zhou, Ganqu Cui, and Peng Ye. 2025. Scaling physical reasoning with the physics dataset. *Preprint*, arXiv:2506.00022.

Appendices

A ShapeStacks Dataset

ShapeStacks (Groth et al., 2018) is a dataset of geometric stacking scenes generated using the MuJoCo (Todorov et al., 2012) physics engine. Each sample consists of 16 simultaneously rendered images of the same stacking system from different viewpoints. The task is to determine whether the depicted structure is stable or will collapse based on these images. The stack height in the samples ranges from 2 to 6, and each sample is labeled as either easy or hard based on the difficulty of stability judgment.

The original benchmark categorizes the samples according to the types of geometric objects involved: blocks_all (samples containing only cubes), ccs_all (samples containing cubes, cylinders, and spheres), and default, which is the union of the previous two splits. Each split is further divided into training, validation, and test sets. In our experiments, we use the test set from the default split as our evaluation set.

Constructing Evaluation Samples The original test set under the default split contains 2,916 samples, which is excessive for evaluating vision language models (VLMs). To ensure a balanced and manageable evaluation set, we perform downsampling based on the following criteria:

- For each stack height from 2 to 6, we select 100 easy and 100 hard samples.
- Within each difficulty group, we ensure an equal number of stable and unstable samples (50 each).
- However, due to class imbalance in the original test set—specifically, there are only 22 stable samples among hard samples with height 6—we adjust our strategy for this subset. For height-6 samples, we include 44 easy and 44 hard samples, each consisting of 22 stable and 22 unstable examples.

In total, this results in a curated evaluation set of 888 samples.

We convert the filtered samples into a VQA-style format suitable for evaluating VLMs. Since all samples correspond to the same task, we design a unified and appropriate prompt. The full set of images for each sample is provided to the model, which is then required to perform reasoning and answer whether the structure is stable.

B Evaluation Details

B.1 Open-Source Model

We use vLLM (Kwon et al., 2023) engine to perform inference with open-source models. During inference, we adopt the officially recommended generation hyperparameters specified in generation_config.json file in each VLM's repository. If no such configuration is provided, we fall back to vLLM's default settings. To mitigate the effects of randomness, each open-source model is evaluated three times using the same set of parameters, and the results are averaged.

When calculating metrics derived from confusion matrix and the preference score T_{pref}, we exclude responses that do not produce a valid answer (i.e., "true" or "false"). Since each model is evaluated three times, we retain only those samples for which all three runs yield valid responses. The number of valid samples used in our evaluation is shown in Table 4. As the table indicates, the number of invalid responses is very small and can be safely ignored.

Model	ShapeStacks (Total: 888)
InternVL3-8B	886
InternVL3-14B	886
InternVL3-38B	887
InternVL3-78B	885
Qwen2.5VL-7B	887
Qwen2.5VL-32B	887
Qwen2.5VL-72B	888
Gemma3-12B-it	887
Gemma3-27B-it	888
MM-Eureka-Qwen-7B	888
SpaceR	888

Table 4: The number of valid responses for each model evaluated on ShapeStacks. Invalid responses are extremely rare.

When analyzing the preference scores for duplicated height samples, the same strategy is applied. In this case, all models produced no invalid responses.

B.2 Human Expert Evaluation

For human expert evaluation, we recruite three participants who each hold a bachelor's degree in physics. Similar to the VLM evaluation setting, each participant is shown all 16 images of a sample and asked to predict whether the structure

is stable or will collapse. The individual accuracies achieve by the three participants are 79.5% (706/888), 77.7% (690/888), and 76.6% (680/888), respectively, resulting in an average accuracy of 77.9%.

The script for human evaluation is included in the code package.

B.3 Duplicated Height Sample Construction

To construct the duplicated height samples, we build upon the test set described in Section 2.1. We first select samples that contain only cubes and have a stack height of 2. While samples with a sphere on top can also be stable, duplicating such configurations may introduce instability, as the sphere would serve as a supporting layer in the middle of the stack. Therefore, we restrict our selection to block-only samples, resulting in a total of 84 base samples.

For each selected sample, we read its MuJoCo simulation file and duplicate the existing configuration by vertically stacking it without altering the horizontal positions of the objects. This results in duplicated samples with a height of 4. To generate samples with a height of 6, we apply the same vertical translation twice.

In total, this results in a curated duplicated height set of 252 samples.

B.4 Fitting Results

B.4.1 Linear Regression For Scaling

We use a linear regression model to analyze the scaling relationship between the logarithm of the language model size in each VLM and its corresponding accuracy on the ShapeStacks benchmark. The fitting is performed to assess how language model capacity correlates with VLM performance. The results is shown in Table 5.

Table 5: Results of the linear regression analysis for the scaling law, modeling the relationship between model accuracy and the logarithm of language model size.

Parameter	Estimate	Std. Error	t-value	p-value
Regression Coefficients				
Intercept	0.4949	0.0190	26.0389	<0.0001
Slope (log size)	0.0352	0.0136	2.5893	0.0360

B.4.2 Linear Mixed-Effect Model

Linear Mixed-Effects (LME) models are a class of statistical models used to analyze data with a hierarchical or clustered structure, where observations are not independent. In our study, this structure arises because multiple data points (e.g., performance on different sample heights) are collected from the same model or model family. A standard linear regression would violate the assumption of independence in such cases, potentially leading to incorrect inferences. Results for analysing height bias is shown in Table 6 and Table 7.

Table 6: Full Results of the Linear Mixed-Effects Model for the Height Bias Analysis. The model assesses the relationship between preference score (T_{pref}) and stack height, accounting for variations across different models (groups).

Parameter	Estimate	Std. Error	z-value	p-value
Fixed Effects (Overall Trend)				
Intercept	0.7382	0.2571	2.8712	0.0041
Slope (height)	-0.1244	0.0398	-3.1226	0.0018
Random Effects (Variance Components)				
Var(Intercept) | Group	0.5744			
Var(height) | Group	0.0131			

Table 7: Full Results of the Linear Mixed-Effects Model for the Height Bias Analysis on **Duplicated Samples**. The model assesses the relationship between preference score (T_{pref}) and stack height on mechanically similar structures.

Parameter	Estimate	Std. Error	z-value	p-value
Fixed Effects (Overall Trend)				
Intercept	0.6485	0.2784	2.3293	0.0198
Slope (height)	-0.1008	0.0488	-2.0650	0.0389
Random Effects (Variance Components)				
Var(Intercept) | Group	0.5672			
Var(height) | Group	0.0145			

C Cognitive Behaviors Analysis

For the analysis of cognitive behaviors, we employ the o4-mini model to annotate the VLM responses. The prompt includes carefully selected few-shot examples to ensure the annotator model fully understood the task requirements.

The InternVL3 series is selected for this specific analysis for the following reasons: (1) The Qwen2.5-VL series has only three models larger than 7B (7B, 32B, and 72B), which are not released simultaneously. The 32B variant is released last, potentially indicating different training methodologies. (2) The Gemma3 series offers only two models of comparable size (12B and 27B), providing an insufficient sample for a robust comparison across scales. The InternVL3 series includes four models larger than 7B (excluding the 9B model, as only its

language component is based on InternLM3, while the others use language models based on Qwen2.5). This series contains more models than the Qwen and Gemma series, and all were released at the same time, making it suitable for analysis.

We show one of the annotated response here to show the effectiveness of our model-based annotation method.


```
"model response": "<think>\nTo
determine if the system is stable,
let's analyze the arrangement and
potential for motion in the stack.\n
\n1. **Top Object (Red Cylinder):**
\n   - The red cylinder is balanced
on the cyan sphere. \n   - The
center of gravity of the red
cylinder should be directly over the
 surface area of the cyan sphere for
 stability. In these images, the red
 cylinder appears to be balanced
correctly.\n\n2. **Middle Object (
Cyan Sphere):**\n   - The cyan
sphere is balanced on the green
cylinder.\n   - Similar to the red
cylinder, the center of gravity of
the cyan sphere needs to be over the
 green cylinder. Observing these
images, it appears balanced.\n\n3.
**Bottom Object (Green Cylinder and
Blue Cube):**\n   - The green
cylinder is on top of a blue cube.\n
   - The green cylinder must be
centered on the blue cube for
stability in the stack. Both objects
 seem to be correctly balanced here
.\n\n4. **Support and Materials:**\n
   - The stack is on a checkered
floor, which is a flat surface
without noticeable slant or
unevenness.\n   - Assuming uniform
material and equal sizes of the
objects, the distances from the
center of mass of each component to
the edge of the support should be
sufficient for stability.\n\
nConsidering all these observations,
 the stack is properly aligned such
that each block's center of gravity
rests directly over the base beneath
 it, and there are no external
forces that would cause tipping or
toppling. Hence, this configuration
is stable.\n\n</think>\n<answer>\
nTrue\n</answer>",

"o4-mini annotation":{
            "Verification": "The
center of gravity of the red
cylinder should be directly over the
 surface area of the cyan sphere for
 stability. In these images, the red
 cylinder appears to be balanced
correctly.",
            "Backtracking": "",
            "Subgoal Setting": "1.
**Top Object (Red Cylinder):**",
            "Backward Chaining": ""
        }
```


D Fine-Tuning

D.1 Hyperparameters

- The hyperparameters for SFT on ShapeStacks (include high and low subset) is shown in Table 8.
- The hyperparameters for SFT on 2d-towers is shown in Table 9.
- The hyperparametrs for RL is shown in Table 10

It should be noted that when creating validation sets from our training data, we designated a 10% split for validation purposes. For the SFT dataset, this split is performed with a random seed of 0. For the RL dataset, the split is performed with a random seed of 42.

For the SFT models, since the validation loss consistently decreased without signs of overfitting (Figure 8), we selected the last training step as the checkpoint for subsequent RL across all models.

The number of epochs in the RL stage and the step of the final checkpoint for each model are shown in Table 11.

Table 8: Key hyperparameters used for Supervised Fine-Tuning (SFT) with LoRA.

Parameter Category	Value
Training & Optimizer Parameters	
Model	Qwen2.5-VL-7B
Epochs	40
Learning Rate	1e-4
LR Scheduler	Cosine
Warmup Ratio	0.05
Optimizer	AdamW
Batch Size	48
Gradient Accumulation Steps	16
Precision	bfloat16
Random Seed	0
LoRA Parameters	
Rank (r)	32
Alpha (α)	64
Dropout	0.05
Target Modules	q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj

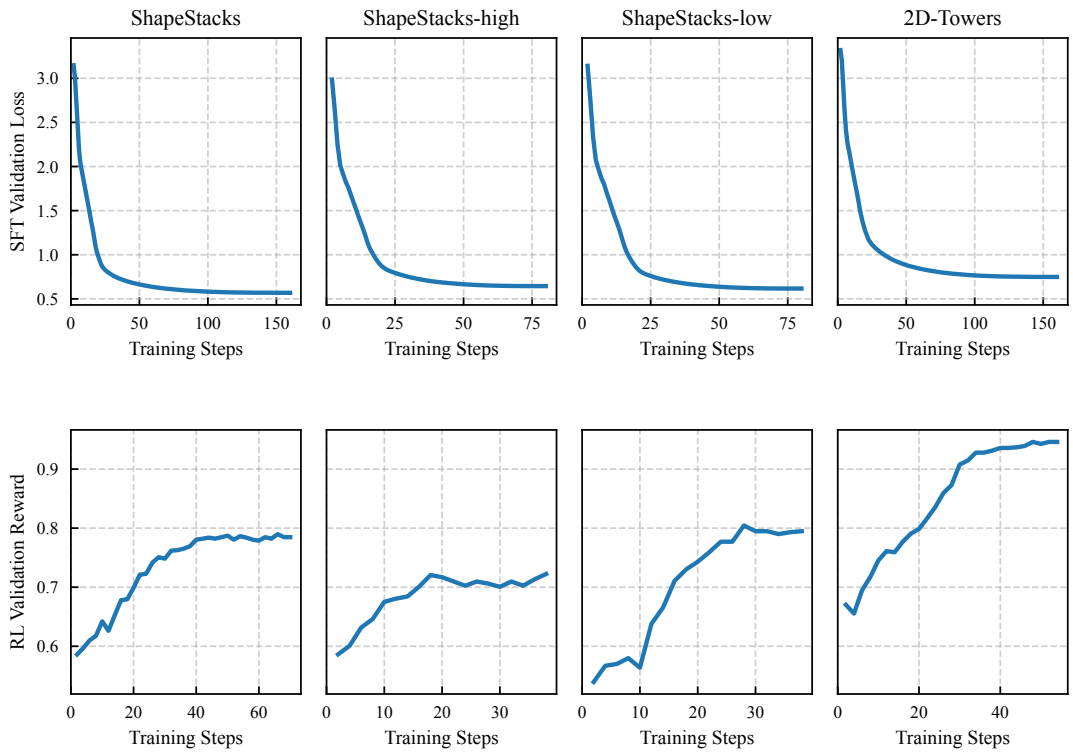

Figure 8: Validation loss in SFT and validation reward in RL for fine-tuning on 4 datasets.

Table 9: Key hyperparameters for Supervised Fine-Tuning (SFT) on the **2D Towers** dataset.

Parameter Category	Value
Training & Optimizer Parameters	
Model	Qwen2.5-VL-7B
Epochs	40
Learning Rate	1e-4
LR Scheduler	Cosine
Warmup Ratio	0.05
Optimizer	AdamW
Batch Size	32
Gradient Accumulation Steps	16
Precision	bfloat16
Random Seed	0
LoRA Parameters	
Rank (r)	32
Alpha (α)	64
Dropout	0.05
Target Modules	q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj

Table 10: Key hyperparameters for Reinforcement Learning (RL) fine-tuning using the GRPO algorithm.

Parameter Category	Value
Core Algorithm & Training Setup	
Algorithm	GRPO
Train Batch Size	1024
Mini-batch Size	256
Tensor Parallel Size	4
Actor Model Optimizer	
Optimizer	AdamW
Learning Rate	2e-6
LR Scheduler	Cosine
Warmup Ratio	0.1
GRPO Policy Parameters	
KL Coefficient (λ)	0.008
KL Loss Type	K3 estimator
Entropy Coefficient	0
Clip Ratio (ϵ)	0.2
Rollout & Reward	
Rollout number	6
Max Response Length	2048

D.2 Data Preparation

D.2.1 2d-towers

The 2D Towers dataset is developed using the Box2D physics engine (Catto., 2009) for simulation. The generation process is designed to mirror the principles of the ShapeStacks dataset. Stable samples are constrained to consist of either only rectangular objects or configurations with a single circular object placed at the top (as curved surfaces cannot serve as stable intermediate supports). Unstable samples could contain either only rectangles or a circle at any position.

Table 11: Total epochs and steps of final checkpoint for different model in RL fine-tuning.

Model	Epoch	Final Checkpoint Step
Qwen-SFT-RL	8	54
Qwen-RL	8	70
Qwen-high	10	38
Qwen-low	10	28
Qwen-2d	8	54

The stability label for each sample is predetermined before generation. We construct stacks layer by layer from the bottom up, calculating the system's equilibrium after each added object based on the center of mass theorem. Upon completion of the stack, we run a physics simulation. If the displacement of all objects in the system remained below a certain threshold for a period of time, the sample is confirmed as stable; otherwise, it is deemed unstable. Due to the limited precision of the simulation, fixing the random seed could often lead to getting stuck: if a structure generated from a specific seed did not match its predetermined label (e.g., an intended stable sample proved unstable in simulation), the generation process for that sample would fail and halt the program. Therefore, we do not fix the random seed during data generation. The full dataset will be made publicly available upon publication of this paper.

For the test set, we generate 1,000 samples with stack heights ranging from 3 to 6. This includes 250 samples for each height, comprising 125 stable samples (60 with a circle, 65 without) and 125 unstable samples (60 with a circle, 65 without). The training set is generated with the same distribution but scaled to 10,000 total samples, with 2,500 samples for each height level.

D.2.2 ShapeStacks Subset Sampling for Height Generalization

To investigate height generalization, we create two distinct subsets from the official ShapeStacks splits: ShapeStacks-high and ShapeStacks-low. The sampling process for both follows a similar two-step procedure: filtering by height, followed by random sampling with a fixed seed for reproducibility. The specific parameters for each subset are detailed below.

- **ShapeStacks-high:**
  - **Height Range:** 4, 5, and 6.
  - **Test Set:** From the full test set (2,916 samples), we filtered by height and sampled 1,000 items (seed=0).
  - **Training Set:** From the full training set (13,618 samples), we filtered by height to get a pool of 6,454 candidates, from which we sampled 1,500 items (seed=0) for SFT data distillation.
- **ShapeStacks-low:**
  - **Height Range:** 2 and 3.
  - **Test Set:** From the full test set (2,916 samples), we filtered by height and sampled 1,000 items (seed=0).
  - **Training Set:** From the full training set (13,618 samples), we filtered by height to get a pool of 7,164 candidates, from which we sampled 1,600 items (seed=0) for SFT data distillation.

D.2.3 Physion

For our dynamics generalization test, we evaluate the model exclusively on the "support" scenarios from the Physion benchmark (Bear et al., 2021). Following the original paper's protocol, we present the model with a single mask identifying the target objects, along with the first 45 frames of the video (recorded at 30 fps, equivalent to 1.5 seconds of footage). The model is then tasked with predicting the subsequent outcome and determining whether the target object would make contact with the support surface.

D.2.4 Distilled Data

We employ the o4-mini model to distill SFT cold-start data, requiring the model to reason in the order of visual observation → physical stability analysis → final conclusion when given an image and a question. To guarantee label fidelity, the ground-truth label of each instance is explicitly revealed to the model during distillation. A representative distilled sample is provided below.

```
    <think>\n1. Identify Objects and
 their State/Orientation:\n
Object 1: Purple cube (bottom)
resting flat on the wooden floor,
faces aligned horizontally and
vertically.\n   Object 2: Cyan cube
(top) resting flat on the purple
cube, faces also aligned.\n\n2.
System Stability Analysis:\n
Contact surfaces: Both cubes have
flat faces in full contact; no
curved surfaces are involved.\n
```

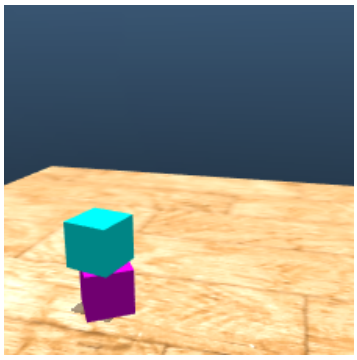

Figure 9: Sample Image

```
Center of mass and support: The cyan
 cubes center of mass projects
vertically down into the interior of
 the purple cubes top face. The
purple cubes center of mass
projects well within its base on the
 floor.\n   Other physical stability
 features: Large contact area and
frictional support between flat
faces prevent sliding or tipping.
The height-to-width ratio is
moderate, and the supporting base is
 wide relative to the stack height,
ensuring a low overall center of
mass.\n\n3. Overall Stability
Conclusion:\n   Since both cubes
have full, flat contact surfaces,
and the combined center of mass of
the two-cube stack falls squarely
within the footprint of the bottom
cube, the configuration satisfies
static equilibrium criteria with a
stable base of support. Therefore,
the stack is stable.\n</think>\n<
answer>True</answer>
```

D.3 Training Curve

Training curve for validation loss in SFT and validation reward in RL is shown in Figure 8.

E Prompt Used

E.1 ShapeStacks

E.1.1 Evlauation

The system prompt for evaluation on ShapeStacks, including ShapeStacks-high and ShapeStacks-low is:

```
You are an AI assistant specializing
 in visual physical reasoning.
Analyze the provided image sequence
and question.
```

The user prompt is:

```
You are provided with several **
static images of a single stacked
geometric system, all captured at
the exact same moment in time but
from different viewpoints**.  Assume
 uniform density for all objects and
 no external forces other than
gravity. Analyze the image sequence:
 Is the system stable / in
mechanical equilibrium? Base your
reasoning on physical laws. Output
your reasoning process based on
relevent physical law between `\
textless think\textgreater ` and `\
textless /think\textgreater `, and
your final answer ('True' for stable
, 'False' for unstable) between `\
textless answer\textgreater ` and `\
textless /answer\textgreater `.
```

E.1.2 Training

We use the same system prompt as that used in evaluation. For the user prompt, we use a more looser version:

```
These images show a stacked
geometric system from different
angles. Assume uniform density for
all objects and no external forces
other than gravity. Analyze the
image sequence: Is the system stable
 / in mechanical equilibrium? Base
your reasoning on physical laws.
Output your reasoning process based
on relevent physical law between `\
textless think\textgreater ` and `\
textless /think\textgreater `, and
your final answer ('True' for stable
, 'False' for unstable) between `\
textless answer\textgreater ` and `\
textless /answer\textgreater `.
```

E.2 2d-towers

We use the same system and user prompt for 2d-towers dataset in evaluation and fine-tuning.

The system prompt is:

```
You are an AI assistant specializing
 in visual physical reasoning.
Analyze the provided image and
question.
```

The user prompt is:

```
The image shows a stacked geometric
system. Assume that all objects have
 the same density and there is no
external torque or force other than
```

```
gravity. Analyze the image and
answer: is the system stable / in
mechanical equilibrium? Base your
reasoning progress on relevant
physical laws. Output your reasoning
 process based on relevent physical
law between `\textless think\
textgreater ` and `\textless /think\
textgreater `, and your final answer
 ('True' for stable, 'False' for
unstable) between `\textless answer\
textgreater ` and `\textless /answer
\textgreater `.
```

E.3 Physion

We only conduct evaluation on Physion support scenario. The system prompt is:

```
You are an AI assistant specializing
 in visual physical reasoning.
Analyze the provided image sequence
and question.
```

The user prompt is:

```
The input image sequence represents
a video with a frame rate of 30fps.
The first frame shows a visual guide
/mask indicating the red and yellow
target objects. Please first
identify the red and yellow objects
based on the mask, observe the video
 content, try to predict the
subsequent development after the
video ends and answer the question:
Will the red and yellow objects make
 contact after the video ends?
Output your reasoning process based
on relevent physical law between `\
textless think\textgreater ` and `\
textless /think\textgreater `, and
your final answer ('True' if contact
 and 'False' otherwise) between `\
textless answer\textgreater ` and `\
textless /answer\textgreater `.
```

E.4 Cognitive Behaviors Analysis

We use o4-mini through API to label the cognitive behaviors in VLMs' responses. The system prompt is:

```
You are an expert AI Cognitive
Analyst. Your task is to analyze a
given model's response and identify
the presence of four specific
cognitive reasoning behaviors. You
must be precise and only cite direct
 evidence from the text.
```

The user prompt is:

```
# COGNITIVE BEHAVIOR DEFINITIONS
Here are the definitions and
examples for the four cognitive
behaviors you need to identify. Pay
close attention to the examples and
counter-examples, as they clarify
the boundaries of each definition.

1.  **Verification**: The act of
checking or comparing a result
against a standard. This is a
distinct, two-stage process: a
result is first produced, and then
it is actively evaluated. It is NOT
simply stating a conclusion.
    *   **Good Example (Numerical
Check)**: "The calculation is 84 +
72 - 83 - 34, which equals 39. This
matches the target, so it is the
correct solution."
    *   **Good Example (Qualitative
Check)**: "The combined center of
mass appears to be located directly
over the red cylinder. This position
 suggests the system is stable."
    *   **Bad Example / Counter-
example (Just a Conclusion)**: "
Therefore, the system is unstable."
(This is a final conclusion, not an
act of checking a previously
calculated or stated result).
    *   **Bad Example / Counter-
example (Just an Observation)**: "
The blue sphere is on top of the
green cube." (This is an observation
, not a verification of a prior
claim or calculation).

2.  **Backtracking**: The act of
explicitly abandoning a failed or
unpromising reasoning path to try a
new, different one. This signals a
non-linear, corrective thought
process.
    *   **Good Example**: "That
calculation resulted in 91, which is
 not our target. Okay, that approach
 didn't work. Let's try a different
combination, starting with
subtraction this time."
    *   **Bad Example / Counter-
example (Sequential Attempts)**: "
Let's try 1+2, that's 3. Now let's
try 1+3, that's 4." (This is just
trying different options
sequentially, not explicitly
identifying a path as 'failed' and
then changing course).

3.  **Subgoal Setting**: The act of
strategically breaking down a large,
 complex problem into a sequence of
smaller, more manageable steps or
intermediate goals.
    *   **Good Example**: "To solve
this, we need to do two things.
First, we must calculate the center
```

```
of mass for each individual object.
Second, we will use those results to
 find the combined center of mass
for the entire system."
    *   **Bad Example / Counter-
example (Restating the Goal)**: "My
goal is to determine if the system
is stable." (This simply restates
the overall objective, it does not
break it down into smaller,
actionable steps).

4.  **Backward Chaining**: Reasoning
 by starting from the final goal and
 working backward to determine the
necessary preceding conditions to
achieve that goal. The reasoning
direction is from `Goal -> Condition
`.
    *   **Good Example**: "For the
system to be stable [Goal], its
combined center of mass must lie
vertically above its base of support
 [Necessary Condition]. Therefore,
our task is to determine the
location of the center of mass."
    *   **Bad Example / Counter-
example (Forward/Causal Reasoning)
**: "The center of mass is located
above the base of support, therefore
 the system is stable." (This is a
standard forward-chaining or causal
conclusion: `Condition -> Result`).
    *   **Bad Example / Counter-
example (Stating a Fact)**: "
Stability depends on the center of
mass." (This is a statement of a
physical principle, not a
directional reasoning process
working backward from a specific
goal).

# TASK
Analyze the following 'MODEL
RESPONSE'. For each of the four
cognitive behaviors, determine if it
 is present in the text based on the
 strict definitions and examples
provided.

# OUTPUT INSTRUCTIONS
You MUST return ONLY a single JSON
object. Do not include any
explanation or text before or after
the JSON.
The JSON object must have exactly
four keys: "Verification", "
Backtracking", "Subgoal Setting", "
Backward Chaining".
- If a behavior IS PRESENT, the
value for its key should be a DIRECT
 and an EXACT quote from the
response that best demonstrates this
 behavior.
- If a behavior IS NOT PRESENT, the
value for its key MUST be an empty
string `""`.

Example of a valid output format:
{{
"Verification": "The resulting
position of the COM is (x, y), which
 is within the support base. This
confirms the system is stable.",
"Backtracking": "",
"Subgoal Setting": "First, I'll
calculate the COM of the top two
blocks. Then, I'll combine that with
 the bottom cylinder.",
"Backward Chaining": "To achieve
stability, the vertical line from
the system's COM must pass through
the area of the base."
}}

# MODEL RESPONSE TO ANALYZE
{MODELS_RESPONSE}
```

E.5 Causal Error Analysis

To identify the proportion of causal reasoning errors in incorrect responses, we use the o4-mini model via API calls, with the following system prompt:

```
 You are a highly focused Text
Auditor. Your sole task is to check
if a given text response violates a
specific instruction about the
nature of the input data.
```

The user prompt is:

```
# ROLE
You are a highly focused Text
Auditor. Your sole task is to check
if a given text response violates a
specific instruction about the
nature of the input data.

# THE CORE INSTRUCTION GIVEN TO THE
ORIGINAL MODEL
The original model is explicitly
told the following about its input:
"You are provided with several **
static images of a single stacked
geometric system, all captured at
the exact same moment in time**.
These are snapshots, **NOT a video
sequence**. There is no movement or
temporal change between these images
."

# YOUR TASK & JUDGMENT CRITERIA
Read the provided 'MODEL'S RESPONSE
'. Your only goal is to determine if
 the model's reasoning (**the text
inside the `<think>` tags**)
violates the core instruction above.

A violation occurs if the model's
reasoning:
1.  **Implies or describes motion/
dynamics**: Uses words like 'rolling
```

```
', 'tilting', 'shifting', 'moving',
'velocity', 'acceleration' to
describe the relationship *between*
the provided static images.
2.  **Uses "static" as evidence for
stability**: Explicitly argues that
because the images are static, the
system must be stable (e.g., "Since
the scene is stationary, it is in
equilibrium.").

# OUTPUT INSTRUCTIONS
You MUST return ONLY a single JSON
object with no extra text. The JSON
object must have one exactly key: '
Causal_logical_error'.

The value for each key must follow
these rules:
- If a violation IS PRESENT, its
value MUST be the **direct and exact
 quote** from the response that best
 demonstrates this violation.
- If a violation IS NOT PRESENT, its
 value MUST be an **empty string
`""`**.

**Example of a valid output if
causal logical error is detected:**
{{
    "Causal_logical_error":"Because
the frames are static, we can
conclude the system is in
equilibrium."
}}

**Example of a valid output if there
 is no causal logical error:**
{{
    "Causal_logical_error":""
}}

## MODEL'S RESPONSE TO ANALYZE:
{MODELS_RESPONSE}
```

E.6 SFT Data Distillation

The prompt used to distill sft cold-start CoT data with o4-mini is:

```
I will now provide you with an
images sequence, each pertaining to
a system of stacked geometric
objects, along with a ground truth
label ('true' for stable and 'false'
 otherwise) indicating whether that
system is stable. Your task is to
complete the physics reasoning
process that leads to this ground
truth, strictly following the CoT
logic provided below. Then, combine
your reasoning with the ground truth
 answer to form the final response.
Please only provide the final
response structure. Place your
completed physics reasoning process
within `<think></think>` tags. Place
 the provided ground truth answer
within `<answer></answer>` tags.

**CoT Logic to Follow:**
1.  **Identify Objects:**
Sequentially identify all objects
and their state/orientation (e.g.,
upright, lying flat, tilted) in the
described system (e.g., from bottom
to top).
2.  **System Stability Analysis:**
    *   Identify if there are any
curved surfaces at the contact
points between objects.
    *   Identify if any object is
significantly offset from the center
 of mass of the supporting structure
 below it, or if the combined center
 of mass of an upper stack is not
well-supported by the object beneath
 it.
    *   Other physical stability
features: This is where you can
expand. Think about base area,
height-to-width ratios, friction (if
 implied or inferable), potential
energy considerations, types of
equilibrium (stable, unstable,
neutral), etc., as appropriate for
the described system.
3.  **Overall Stability Conclusion
:**
    Based on the principles of
mechanical equilibrium (e.g., the
vertical projection of the combined
center of mass of all objects
resting on a supporting object must
fall within the supporting area of
that object; or for objects resting
on a curved surface, the combined
center of mass of all objects above
it should ideally be on the contact
line/point of that curved surface
for stability; ...), analyze and
conclude whether the system is
stable. This conclusion should
logically lead to the ground truth
answer provided.

**Example of your expected response
structure (demonstrating one
possible way to apply the logic):**
<think>
1.  **Identify Objects and their
State/Orientation:**
    *   Object 1: Large blue cube,
bottom, upright.
    *   Object 2: Small yellow
sphere, on top of the blue cube.
    *   Object 3: Medium-sized red
cylinder, on top of the yellow
sphere, standing upright on its
circular base.
2.  **System Stability Analysis:**
    *   Contact points: Yes, curved
surfaces are involved. The yellow
sphere has a curved surface in
contact with the blue cube (bottom)
and the red cylinder (top). The
cylinder's flat base contacts the
```

```
sphere's curved top.
    *   Center of mass and support:
The red cylinder, standing upright,
has its center of mass relatively
high. It's supported by the yellow
sphere, which offers a very small,
essentially point-like, contact area
. This makes the configuration
sensitive to the alignment of the
cylinder's center of mass over this
point.
    *   Other physical stability
features: The sphere provides
minimal frictional resistance to
rolling or pivoting for the cylinder
. The stability relies entirely on
precise balancing. This is a case of
 unstable equilibrium; a tiny
displacement will lead to toppling.
3.  **Overall Stability Conclusion
:**
    The system involves a red
cylinder standing upright on a
yellow sphere. Principles of static
equilibrium dictate that for an
object to be stable on a curved
surface like a sphere, its center of
 mass must be vertically aligned
directly above the point of contact,
 and ideally, the supporting surface
 should provide restoring forces/
torques if tilted. A sphere provides
 a point contact and offers no
restoring torque for an upright
cylinder balanced on it. Any slight
horizontal displacement of the
cylinder's center of mass, or a
slight perturbation, will create a
torque that causes the cylinder to
tilt further and fall. The base of
support for the cylinder is
effectively a point, making the
system inherently unstable.
Therefore, the system is unstable.
</think>
<answer>False</answer>
```