A. Training Details

Here, we provide the training details, including the hyperparameters, for both QA fine-tuning and PCB training tasks. All training was performed on NVIDIA A100 DGX systems.

A.1. QA fine-tuning

To fine-tune PaliGemma-3B on the question-answer datasets, we apply the LoRA fine-tuning scheme by targeting the attention weights in both the vision and language modules, as well as the fully connected MLP layers, multi-modal projector layers, embedding tokens, patch embedding, and positional embedding. Tab. 7 (left) shows other hyperparameters used for QA fine-tuning.

Table 7. Hyperparameters used to fine-tune the PaliGemma-3B models on the QA datasets (left) and as PCB modules (right).

Hyperparameter	**Falling Tower**	**CLEVRER**
LoRA rank	16	16
Learning rate	5e-5	5e-5
Batch size	32	32
Epochs	10	3
Trainable parameters	1.24 %	1.24 %
Number of frames	1	8
Compute time	∼ 3.5 hours	∼ 37 hours

Hyperparameter	**Falling Tower**	**CLEVRER**
LoRA rank	16	16
Learning rate	5e-5	5e-5
Batch size	32	64
Epochs	50	10
Trainable parameters	1.24 %	1.24 %
Number of frames	1	8
Compute time	∼ 1 hour	∼ 2.3 hours

A.2. PCB training

Descriptions used for training PCBs. We first discuss the two types of descriptions we considered for training PCBs:

1. Human-Narration (HN), which generates a summary of all the collisions that occurred in the scene.

```
Scene History:
In this scene, there are 3 collisions occurring in sequence.
Here are the relevant observations prior to the 1st collision:
Object 0 (the blue rubber sphere) enters the scene and moves toward the 1st collision
site.
Object 1 (the gray rubber sphere) is moving toward the 1st collision site.
Object 2 (the cyan metal cube) enters the scene and is moving in the rest of the
scene but does not participate in the collision.
Object 3 (the purple rubber sphere) remains stationary in the scene and does not
participate in the collision.
Object 4 (the blue metal sphere) remains stationary in the scene and does not
participate in the collision.
Finally, Object 0 collides with Object 1.
Here are the relevant observations prior to the 2nd collision:
...
```

2. Structured-Physics (SP), which describes each provided video frame separately as follows, while adding physical properties of the objects, including their discretized and normalized locations and velocities. We also include the locations of collisions that occurred up to a certain frame.

```
[FRAME] [OBJECTS] [OBJ] SHAPE COLOR MATERIAL [LOC] LOC [/LOC] [VEL] VEL [/VEL] [/OBJ]
[OBJ] SHAPE COLOR MATERIAL [LOC] LOC [/LOC] [VEL] VEL [/VEL] [/OBJ] ...  [/OBJECTS]
[COLLISION_PAST] [COLLISION] [OBJ] SHAPE COLOR MATERIAL [/OBJ] [OBJ] SHAPE COLOR
MATERIAL [/OBJ] [LOC] LOC [/LOC] [/COLLISION] ...  [/COLLISION_PAST] [/FRAME]
```

Training details. We use the pre-trained PaliGemma-3B model for training the PCB modules and apply the LoRA fine-tuning scheme, similar to the approach used for QA fine-tuning. Tab. 7 (right) provides the hyperparameters used to train PCB modules for both Falling Tower and CLEVRER datasets.

B. Ablations

B.1. The Effect of Framing Multi-Choice Questions as Multiple Binary Questions

As discussed in the main paper, framing the multi-choice questions as multiple binary questions in CLEVRER can yield significant improvement in the accuracy of the models. In Tab. 8, we provide a comparison between the performance of fine-tuned PaliGemma-3B models with and without this change. As demonstrated, we observe improvement in almost all categories, except for the per question predictive accuracy. We posit that this is because the predictive questions in CLEVRER are always binary questions with exactly one correct choice. Framing the predictive questions as two independent binary questions can result in a model choosing both options as correct or wrong.

Table 8. The performance of the fine-tuned PaliGemma-3B model on question answer pairs for the CLEVRER benchmark based on framing the multi-choice questions as binary questions. Both models are trained for three epochs.

Multi-Choice as Binary?	**Descriptive**	**Explanatory**		**Predictive**		**Counterfactual**	
		per ques.	per opt.	per ques.	per opt.	per ques.	per opt.
False	89.3	69.0	86.6	**83.6**	83.6	41.0	74.0
True	**92.9**	**94.7**	**98.2**	77.9	**88.2**	**68.4**	**88.7**

B.2. The Effect of Training Epochs

We run an ablation study to assess the effect of training for smaller vs. larger number of epochs on the accuracy of CLEVRER in the QA fine-tuning task. Tab. 9 demonstrates a large improvement in training for more epochs.

Table 9. The performance of the fine-tuned PaliGemma-3B model on question answer pairs for the CLEVRER benchmark based on the number of trained epochs. Here, multi-choice questions are asked as they are (without framing them as multiple binary questions).

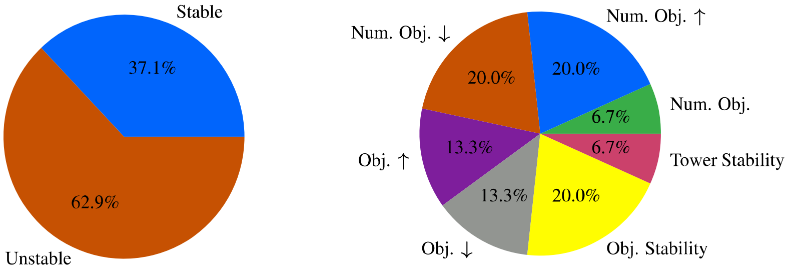
Epochs	**Descriptive**	**Explanatory**		**Predictive**		**Counterfactual**	
		per ques.	per opt.	per ques.	per opt.	per ques.	per opt.
3	**89.3**	**69.0**	**86.6**	**83.6**	**83.6**	**41.0**	**74.0**
2	87.2	66.5	85.7	82.3	82.3	38.3	72.9
1	78.1	52.9	77.3	73.5	73.5	15.3	51.0

B.3. Evaluating the Importance of Vision Module

We illustrate the importance of the vision module in a VLM for physical reasoning by conducting the following experiment. Here, we QA-fine-tune only the language model part of PaliGemma-3B while freezing the vision module. The results in Tab. 10 shows that the performance across all categories drops slightly for the language model-only setting. Therefore, jointly fine-tuning both the vision and language modules is essential for optimal performance, as it enables the model to better align visual features with linguistic representations.

Table 10. Performance drop due to freezing the vision module on the PaliGemma-3B-base model for the QA fine-tuning over CLEVRER.

Descriptive	**Explanatory**		**Predictive**		**Counterfactual**	
	per ques.	per opt.	per ques.	per opt.	per ques.	per opt.
-1.6	-1.8	-0.6	-8.9	-1.0	-2.8	-1.2



(a)

(b)

Figure 5. Falling Tower dataset: (a) distribution of stacked objects in terms of their stability, and (b) distribution of question types, including both descriptive and stability categories.

C. Falling Tower Dataset

The Falling Tower dataset is a benchmark for stability detection of stacked objects, inspired by the ShapeStacks benchmark [15]. It includes 4864 unique scenes, 72,775 questions, and detailed simulation-generated annotations to support training Vision-Language Models (VLMs) for spatial and physical reasoning. Each simulation instance is represented as a JSON file containing:

- **Scene Description:** A list of objects stacked from bottom to top with their respective offsets, e.g., "Scene description: Here are the parts stacked from bottom to top: purple cube, yellow cylinder. Offsets for each part, from bottom to top, are: (-0.03, -0.05), (0.0, 0.02)."
- **Simulation Metadata:** Physical and rendering settings, including stability status (`stable: true/false`), the number of objects, gravity parameters, and camera settings.
- **Objects:** Detailed information about each object, including its type (e.g., cube, cylinder), dimensions, colors (both RGBA and HEX), rigid body properties (e.g., mass, friction), initial and final positions, and positional offsets. Rigid body properties used for simulation were fine-tuned to reflect real-world dynamics, enabling us to achieve 89% accuracy in a human evaluation of 50 examples for stability detection.
- **Questions and Answers:** A variety of descriptive and stability QAs aimed at assessing spatial and physical reasoning, e.g:
  - **Descriptive Questions:** "How many objects are in the scene?" (Answer: 2), "What is the shape/color of the object above the purple cube?" (Answer: yellow cylinder).
  - **Stability Questions:** "Will this collection of objects stay stationary?" (Answer: False), "Will the yellow cylinder stay stationary?" (Answer: False).

Fig. 5 shows the distribution of object stacks in terms of their stability, as well as the distribution of question types.

The Sim2Real dataset consists of 20 images. Seven stable and seven unstable cases were captured against a clean background, while six additional stable cases were captured with a varying background for testing the robustness of a vision model. Additionally, the dataset includes 100 human-generated questions, with five questions per image. The objects are 3D-printed using a J55™ Prime 3D Printer.

Dataset Links:

- Falling Tower Dataset
- Sim2Real Dataset

D. Additional Experiments

D.1. Specialized Baselines for CLEVRER

Here, we compare the fine-tuned PaliGemma-3B model on the CLEVRER QA dataset to specialized architectures designed specifically for CLEVRER. Although the fine-tuned model does not outperform all benchmarks, its comparable performance highlights the potential benefits of generalist models over bespoke baselines.

Table 11. Per-question performance of fine-tuned PaliGemma-3B compared to specialized methods on CLEVRER.

Category	**Model**	**Descriptive**	**Explanatory**	**Predictive**	**Counterfactual**
Specialized Methods	VRDP [12]	89.80	82.40	83.80	75.70
	DCL [9]	90.70	82.80	82.00	46.50
	CRCG [20]	95.55	99.81	76.64	78.31
	Aloe [11]	94.00	96.00	87.50	75.60
Fine-tuned QA	PaliGemma-3B-Fine-Tuned	92.90	94.70	83.60	68.40

D.2. The Effect of PCBs on the InternVL 3.0 Model

Table 12. Performance of InternVL 3.0 (8B parameters), augmented with Physics Context Builders (PCBs), compared to its zero-shot version on the Falling Tower benchmark. HN refers to the Human Narration-style PCB. The second value after the slash indicates the Sim2Real accuracy, and the third value represents the F1 score on Sim2Real.

Model	**Descriptive** [sim acc. / real acc.]			**Stability** [sim acc. / real acc. / real F1]	
	num. obj.	num. obj. ↑↓	obj. ↑↓	obj. stable	tower stable
InternVL3-8B	81.57 / 78.95	52.77 / **78.95**	53.85 / 84.21	52.71 / **84.21 / 80.19**	46.42 / 73.68 / 68.64
InternVL3-8B-PCB	**95.94 / 88.24**	**66.29** / 70.58	**70.01 / 94.12**	**69.21** / 76.47 / 73.39	**83.07 / 76.47 / 73.73**