

---

# Can vision language models learn intuitive physics from interaction?

---

Luca M. Schulze Buschoff<sup>\*1</sup> Konstantinos Voudouris<sup>\*1</sup> Can Demircan<sup>1</sup> Eric Schulz<sup>1</sup>

## Abstract

Pre-trained vision language models do not have good intuitions about the physical world. Recent work has shown that supervised fine-tuning can improve model performance on simple physical tasks. However, fine-tuned models do not appear to learn robust physical rules that can generalize to new contexts. Based on research in cognitive science, we hypothesize that models need to interact with an environment to properly learn its physical dynamics. We train models that learn through interaction with the environment using reinforcement learning. While learning from interaction allows models to improve their within-task performance, it fails to produce models with generalizable physical intuitions. We find that models trained on one task do not reliably generalize to related tasks, even if the tasks share visual statistics and physical principles, and regardless of whether the models are trained through interaction.

## 1. Introduction

A central goal of machine learning research is to build machines that think and behave like people do. Lake et al. (2017) propose that human-like machine learning models must be capable of reasoning about their environment and its physical, social, and causal structure. These capabilities are often referred to as intuitive theories (Baillargeon et al., 1995; Spelke, 1990; Spelke & Kinzler, 2007). Here, we focus on *intuitive physics* — the ability to understand and predict the physical properties and interactions of objects (Battaglia et al., 2012; Piloto et al., 2022).

Recent work has established that vision language models (VLMs), models that receive visual and textual inputs, are still limited in their understanding of the physical world and its causal structure (Jin et al., 2023; Balazadeh et al., 2024). VLMs do not perform well on standard visual cogni-

tion tasks — such as tasks testing intuitive physics — and they do not show a good fit with human behavioral data (Schulze Buschoff et al., 2025a). While supervised fine-tuning enables models to perform well on the tasks they were fine-tuned on, they do not appear to learn generalizable intuitions about the physical world (Schulze Buschoff et al., 2025b).

A prominent idea in cognitive science is that humans learn a robust understanding of their world by interacting with it (Gibson, 1979; Merleau-Ponty, 1945; Varela et al., 1991). The key claim is that humans learn robust, generalizable concepts for explaining and predicting their world not merely from passive observation and symbolic abstraction, but from actively interacting with their environment’s dynamics (Barsalou, 1999; Clark, 1998). Some have argued that directly experimenting with the physical properties of objects in the environment allows children to test their hypotheses about their environment (Gopnik et al., 1999). In contrast to passively observing the interactions of other people with an environment, they learn much more from trying, and often failing, to predict how the environment will evolve given their own actions (Smith, 1982; Chu & Schulz, 2020; Nicolopoulou, 1993; Smith & Gasser, 2005; Schulz & Bonawitz, 2007). While the important role of interaction is slowly being recognized in generative model training (Silver & Sutton, 2025; Motamed et al., 2025), its merit for teaching vision language models visual cognitive abilities such as intuitive physics has not yet been explored.

In this paper, we present a first attempt at evaluating the role of interaction for learning intuitive physics in VLMs. Interaction can be operationalized in several ways (Shapiro & Spaulding, 2025), from one- and multi-step reinforcement learning (RL) to multi-sensory robotics. We operationalize interaction in the context of one-step RL, defining an *environment*, *action space*, and *reward function* (Sutton et al., 1998). VLMs are presented with an image of a stack of colored blocks generated by a physics engine. They must for example respond with an action sequence to move another block to build a taller, stable tower, receiving a reward that depends on the stability of the resulting tower.

---

<sup>\*</sup>Equal contribution <sup>1</sup>Institute for Human-Centered AI, Helmholtz Munich, Oberschleißheim, Germany. Correspondence to: Luca M. Schulze Buschoff <lucaschulze-buschoff@gmail.com>.

We compare models that are trained to build towers through trial-and-error (the *interactive* condition) with models that are shown examples of optimal action sequences to build stable towers (the *non-interactive* condition). Similarly to how children appear to learn generalizable physical intuitions by playing with objects (Piaget, 1952), we propose that learning to build towers through interaction with the physics of the environment will enable VLMs to learn those same intuitions.

Following this line of argument, we hypothesize the following:

1. Models in the interactive condition will generalize better to building new towers not seen in their training data, compared to the non-interactive condition.
2. Models in the interactive condition will generalize better to a new task, such as judging the stability of a tower, compared to the non-interactive condition.

We test these hypotheses mainly by evaluating the textual outputs of VLMs. However, it is possible that models might have the knowledge required to solve the task, but cannot produce textual outputs in the right format. We explore this distinction between model *competence* and model *performance* (Chomsky, 1965) by decoding model activations layer-wise to see how predictive they are of key physical quantities. We thus further hypothesize that these quantities will be more decodable at later model layers in models trained in the interactive condition compared to the non-interactive condition.

We find no noticeable differences between the interactive and non-interactive conditions, both in and outside of the training tasks. Both methods yield models that perform at ceiling on the tasks they are trained on, but neither method produces models that reliably generalize to new physical tasks. While we find that physical quantities like tower stability are highly decodable from model activations, neither post-training method successfully converts this competence into reliable performance on new tasks.

## 2. Related Work

Despite recent advances in architectures and training methods, VLMs continue to struggle on simple visual tasks that are trivial for any human observer, such as counting objects in a scene or making judgements about their interactions (Rahmanzadehgervi et al., 2024; Schulze Buschoff et al., 2025a; Balazadeh et al., 2024). Campbell et al. (2024) suggest that these failures originate from the fact that the test images contain multiple objects whose higher-order relations must be tracked. There is evidence that pre-trained VLMs struggle to attend to and distinguish multiple objects at the same time (Frankland et al., 2021).

Supervised fine-tuning (SFT) has emerged as an efficient way to overcome limitations such as these through extensive post-training on specific problems (Han et al., 2024). These methods have also proven useful for aligning models towards more human-like outputs (Binz et al., 2024; Husain et al., 2024). However, SFT with VLMs appears to have a limited effect on their ability to learn generalizable physical intuitions (Schulze Buschoff et al., 2025b) and interact reliably with physical environments (Mecattaf et al., 2024). One plausible hypothesis is that SFT simply allows VLMs to learn useful shortcuts for specific tasks (Geirhos et al., 2020). Indeed, Motamed et al. (2025) argue that large generative video models are likely making predictions about the physical world without a proper understanding of its underlying physics. They suggest that a lack of active interaction with the physical world could be the limiting factor. Our study therefore seeks to explore whether models’ understanding of physics can be improved through active interaction with an environment.

In line with this proposal, online reinforcement learning (RL), a paradigm in which models learn through interaction with an environment, has recently been argued to generalize more robustly than SFT (Chu et al., 2025), an analogue of offline RL (Wu et al., 2025). Online RL refers to updating the model sequentially based on actions it has taken in an environment, whereas offline RL refers to updating the model based on a fixed set of state-action pairs collected using another policy (Levine et al., 2020). Online RL has been shown to outperform offline RL methods in some cases (Ostrovski et al., 2021), but this distinction has been under-explored in the context of VLMs. Chu et al. (2025) train VLMs on arithmetic reasoning and simple navigation tasks and find that online RL trained models generalize better than models trained with SFT. In contrast to Chu et al. (2025), our work focuses on established intuitive physics tasks in cognitive science: building and judging the stability of block towers (Lake et al., 2017).

## 3. Methods

### 3.1. Datasets

We construct two tower block datasets for our experiments, each consisting of stacks of 2-4 randomly colored cubes.  $256 \times 256$  pixel RGB images are taken from a fixed camera angle in the ThreeDWorld environment (Gan et al., 2020). We keep the camera angle and block sizes fixed throughout, so that the models are able to learn the mapping between pixel space and ground truth distance. Both datasets feature towers that consist of perfectly stacked blocks except for one block. In the first dataset, **top block**, this block is on top of the tower but displaced to the left or the right (see Fig. 1 and Fig. 7 in the Appendix).

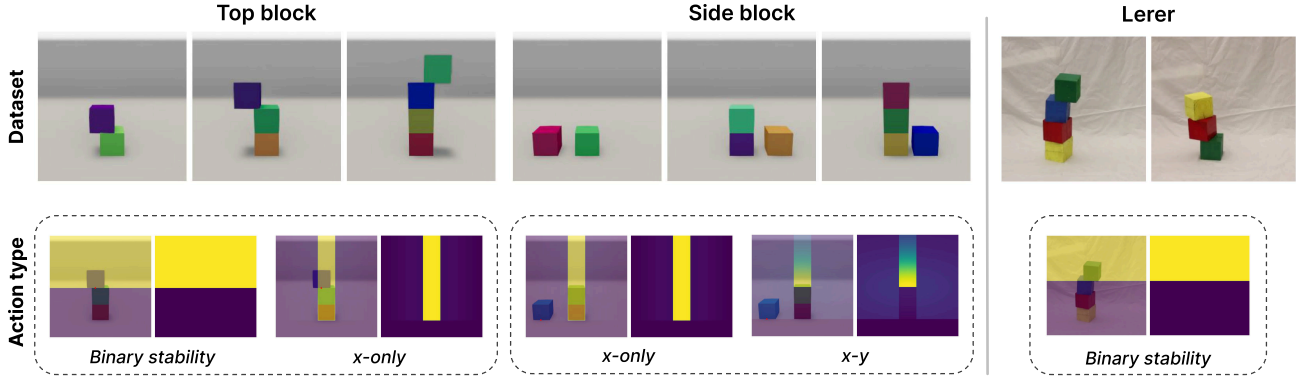


Figure 1. Overview of all combinations of datasets and action types. **Datasets:** We train models on two related datasets, one where the block on top of a tower is displaced, and one where a block is displaced on the ground next to a tower. **Action types:** For each dataset, we train models on two action types. *Binary stability* requires models to make a binary judgment on whether a given tower is stable. For the *x-only* task, models need to give a single value by which the displaced block should be moved to build a more stable or bigger tower. The *x-y* task requires the same action as *x-only*, but with an added dimension — here the block needs to be moved to the side and up. We train models on all four combinations of dataset and action types using GRPO to test whether models trained through interaction with an environment learn generalizable physical intuitions. We also evaluate models on an external dataset of real wooden block towers, taken from Lerer et al. (2016).

In the second dataset, **side block**, the block is on the floor next to the tower, also either to the left or the right (see Fig. 1 and Fig. 8 in the Appendix). As an additional evaluation dataset, we also use an independent dataset of real images of tower blocks from Lerer et al. (2016; see Appendix A.1).

### 3.2. Tasks

Using these datasets, we construct four tasks. Using the top block and Lerer et al. (2016) datasets, the **binary stability** task requires the model to give a judgment on whether a given tower is stable or not. In contrast, the **x-only** task requires the model to return a single integer that moves the block along the  $x$ -axis (see Fig. 1). Here, the goal is to improve the stability of the tower by moving the block closer to the centre. In both tasks, the model must attend to the displacement of the top block from the centre point of the tower, both to judge its stability and to choose an appropriate counter-displacement to stabilize the tower. However, the latter case is framed interactively.

With the side block dataset, we also construct an **x-only** task, again requiring the model to give an integer to move the block to the most central position. In contrast, the **x-y** task requires the model to give two integers to move the block to the most stable position in both the  $x$ - and  $y$ -dimensions (see Fig. 1). The  $x$ -only tasks are identical except that the range of correct integers is different due to the different block displacements. Moreover, models should be readily able to generalize from the  $x$ - $y$  task to the  $x$ -only tasks, thanks to their being identical problems on the  $x$ -dimension. The prompts for each task are included in Appendix A.4.

### 3.3. Fine-Tuning Methods

We fine-tune the 8B parameter 4-bit quantized version of the Qwen3-VL model (Yang et al., 2025) using the unsloth library (Han et al., 2023). We employ Parameter Efficient Fine-Tuning (PEFT; Han et al., 2024) — rather than updating all model weights we update small low-rank adapters inserted layer-wise in the model (QLoRA; Dettmers et al., 2024; Hu et al., 2021).

We are primarily interested in training models that learn from interaction. To implement this, we train models using reinforcement learning with Group-Relative Policy Optimization (GRPO). As a non-interactive baseline, we compare the GRPO models to models post-trained with Supervised Fine-Tuning (SFT). We outline each method in turn. Note that in Section 4.5 and Section A.8 in the Appendix, we also describe results from experiments with other models and a different RL algorithm.

**Group-Relative Policy Optimization** In the reinforcement learning setting, the set of all model and adapter weights ( $\theta$ ) is considered the policy,  $\pi_\theta$ . It takes the text prompt and image as input (observations of the state of the environment), and produces a token sequence as actions. For a batch of  $M < \text{prompt, image} >$  pairs,  $\{p_1, \dots, p_M\}$ , the model produces a set of  $M \times N$  completions  $\{c_{1,1}, \dots, c_{1,N}, \dots, c_{M,N}\}$ . These completions are rewarded using a reward function, giving a set of rewards  $\{r_{1,1}, \dots, r_{1,N}, \dots, r_{M,N}\}$ . We use  $N = 16$  in our experiments.

We compute the loss for some prompt  $p$  as:

$$\mathcal{L}(\theta) = -\frac{1}{\sum_{i=1}^N |c_i|} \sum_{i=1}^N \sum_{t=1}^{|c_i|} \left[ \min \left( \frac{\pi_{\theta}(c_{i,t}|q, c_{i,<t})}{\pi_{\theta_{\text{old}}}(c_{i,t}|q, c_{i,<t})} \hat{A}_{i,t}, \right. \right. \\ \left. \left. \text{clip} \left( \frac{\pi_{\theta}(c_{i,t}|q, c_{i,<t})}{\pi_{\theta_{\text{old}}}(c_{i,t}|q, c_{i,<t})}, 1 \pm \eta \right) \right) \hat{A}_{i,t} \right]$$

Where  $|c_i|$  is the length of the completion, in tokens, and  $\hat{A}_{i,t}$  is the normalized reward (advantage) for  $|c_i|$ :

$$\hat{A}_{i,t} = \frac{r_i - \text{mean}(\{r_1, \dots, r_n\})}{\text{std}(\{r_1, \dots, r_n\})}$$

Following common practice (Hu et al., 2025; Liu et al., 2025; Yu et al., 2025), we exclude the original KL-divergence term used in (Shao et al., 2024). We update the adapter weights with gradient ascent over  $\mathcal{L}(\theta)$ .

**Supervised Fine-Tuning** Using a labelled dataset, model weights are updated using batch gradient descent over the token-level cross-entropy loss:

$$\mathcal{L}(\theta) = -\sum_{t=1}^T \log p_{\theta}(y_t|y_{<t})$$

where  $\theta$  is the set of model and adapter weights,  $T$  is the ordered set of target completion tokens given a prompt,  $y_t$  is the target token at step  $t$ , and  $y_{<t}$  is the set of ordered target completion tokens prior to  $t$ . Only the adapter weight subset of  $\theta$  are actually updated.

**PEFT Hyperparameters** We keep the hyperparameters across both fine-tuning methods as consistent as possible. All adapters are the same size and injected in all layers of the model. Specifically, we inject a matrix  $W_a$  at each layer, which is the product of two low-rank matrices,  $M_1 \in \mathbb{R}^{d \times r}$ ,  $M_2 \in \mathbb{R}^{r \times k}$ , where  $d, k$  are the input and output dimensionalities respectively, and  $r \ll d, k$ . During the forward pass, the outputs of the full weight matrix for that layer are summed with the outputs of the adapter, subject to some scaling factor  $\frac{r}{\alpha}$ . We use  $r = \alpha = 16$  everywhere. We use stochastic gradient descent with the Adam optimizer. We train all models for 10,000 steps on single 80GB A100 GPUs.

### 3.4. Reward Functions

We use specific reward functions for each task both for training models with GRPO and for evaluating models in the results below. For the binary stability task, where models have to give a binary response judging the stability of a given block tower, we use three distinct reward values:  $-1$  for

non-parseable answers, 0 for legal but incorrect answers, and 1 for legal and correct answers.

For the x-only task where models reply with a single integer, we set the reward to  $-5$  for non-parseable completions. For answers that are parsed correctly we use two different Gaussian functions based on the distance to the center. As answers get closer to the center, they are rewarded more. For answers that result in an unstable tower, we calculate a weaker function as  $2 \cdot e^{(-d^2)} - 2$ , where  $d$  is the distance on  $x$  from the optimal position. For answers that result in a stable tower, we compute the reward as  $20 \cdot e^{(-d^2)}$  (see Fig. 1 and Fig. 7 in the Appendix for a visualization).

For the x-y task where models must reply with two integers, we again set the reward to  $-5$  for non-parseable answers. For answers that move the block below the floor, we set the reward to  $-4$ . For all other parseable answers, we again compute Gaussian reward functions depending on the euclidean distance between the final position of the moved block and the optimal position on top of the tower. For answers that are above ground but do not result in a stable bigger tower, we calculate the reward as  $2 \cdot e^{(-d^2)} - 2$ . For answers that are within the tower, we compute  $2 \cdot e^{(-d^2)} - 4$ . And for answers that result in a stable bigger tower, we compute the reward as  $20 \cdot e^{(-d^2)}$  (see Fig. 1 and Fig. 8 in the Appendix for a visualization).

## 4. Results

We evaluate performance on held-out instances from the post-training task (4.1), generalization to the other tasks (4.2), and generalization to the binary stability task with real images of block towers (4.3).

### 4.1. Post-training performance improvement

GRPO improves performance of the pre-trained model on all post-training tasks (see the diagonal in Fig. 2). On the binary stability top block task, where models have to give a binary judgment on the stability of a block tower, the GRPO model achieves a mean test accuracy of 0.969 after 10,000 steps (the ceiling here is 1, for all other tasks it is 20). Similarly, the SFT model trained without interaction achieves a mean test accuracy of 0.969.

On the x-only top block task, models are asked to return a single integer to move the top block into a more stable position. Here, the GRPO model achieves a mean test reward of 19.999 after 10,000 steps. The SFT model achieves the same score. For x-only on side block, models are also asked to return an integer to move a block, here one that is misplaced on the floor to the left or right of the tower, to the center of the image. The GRPO model again achieves a mean test reward of 19.998, as does the SFT model. The

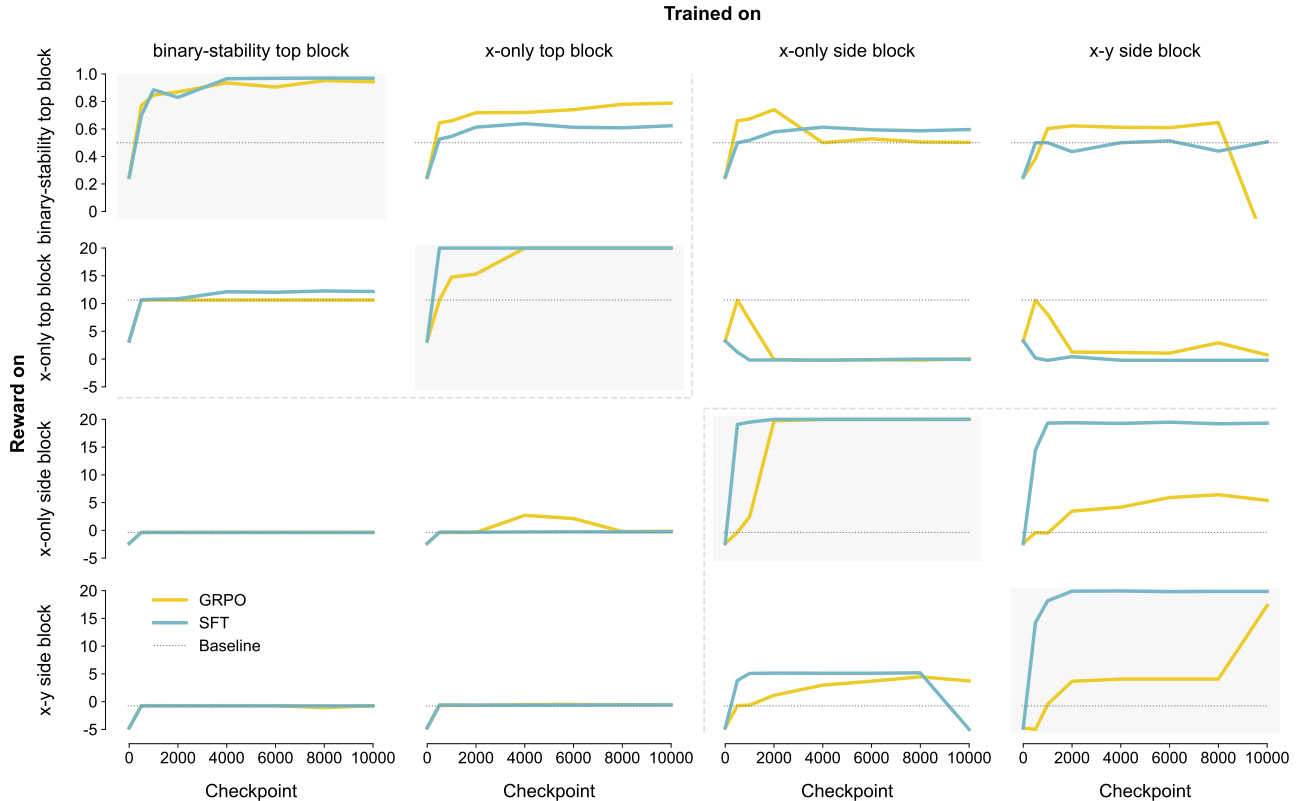


Figure 2. Performance by test task and training task for Qwen3-VL-8B. Rows show models evaluated on a given task. Columns show models trained on a given task. The blue and orange lines show the performance of the models trained with SFT and GRPO, respectively. The grey dotted line shows the baseline for the evaluation task. Plots on the diagonal show within-task performance, meaning models are evaluated on the same task they are trained on. All other subplots represent some degree of generalization.

x-y task on side block is similar to the x-only task, however models here also have to return a second integer to move the block not only to the left or right but also up into the most stable position on top of the tower. Here, the GRPO model gets a mean test reward of 17.313. In contrast, the SFT model gets a mean test reward of 19.858.

To summarize, we find that training with interaction through GRPO allows models to improve performance on their training task. However, the same is true for the SFT models that are trained without interaction — as such, we find no direct benefit of training with interaction when it comes to improving within task performance.

#### 4.2. Generalization to related tasks

We are not just interested in models that perform well on a single physical task, but rather models that robustly generalize from their experience to solve new tasks (Collins et al., 2022; Geirhos et al., 2018; Griffiths & Tenenbaum, 2009). Therefore, we test whether models post-trained on single tasks can generalize to new, related tasks. To test this, we evaluate all post-trained models on all tasks.

We find that no model reliably generalizes to all other tasks,

regardless of if they are trained with interaction (see Fig. 2). However, we find that there is a slight degree of generalization when models are evaluated on different tasks but on the same distribution of data they were trained on (see the upper left and lower right quadrants of Figure 2). For example, the GRPO model trained on the x-y side block task also performs above the baseline on the x-only side block task, achieving a mean reward of 5.396 (the model at 10,000 steps overfits to returning two integers instead of one, filtering only legal answers yields a mean reward of 12.772). This is to be expected because x-only is a subset of the x-y task which requires only the output of the x variable that the model has learned. The SFT model achieves a mean test reward of 19.318 (see Table A.2 in the Appendix for the full set of model results after 10,000 steps).

We also find a slight carry-over between the binary-stability and x-only tasks on the top block data — solving both tasks requires the same task variable, the x-offset of the top block. In the x-only condition, the model is explicitly forced to learn this variable as the amount the top block has to be moved by in order to put it in the most stable position. It is also the single variable needed to solve whether a given block tower is stable or not. Despite this, we only find very

limited generalization between the two conditions, highlighting how constrained generalization from either post-training method is. When evaluated on binary-stability top block, the GRPO models trained for 10,000 steps on x-only top block, x-only side block, and x-y side block get accuracies of 0.624, 0.503, and  $-0.264$  respectively. The SFT models trained on the same conditions perform at 0.624, 0.596, and 0.506 (since this is a binary task the random baseline is 0.5, but illegal answers can pull the reward down).

To conclude, models perform well on their fine-tuning task (see the diagonal in Fig. 2) and they show slight patterns of generalization to other tasks on the same data (from x-only top block to binary stability top block). However, generalization to other data but with the same task is limited (from x-only top block to x-only side block).

### 4.3. Generalization to real images

To test whether models learn physical intuitions that can generalize to real images, we test them on 100 images from [Lerer et al. \(2016\)](#). These images show real wooden block towers consisting of 2 to 4 colored wooden blocks, which are either stable or unstable.

When evaluated on this data, the GRPO models trained for 10,000 steps on binary-stability top block, x-only top block, x-only side block, and x-y side block get accuracies of 0.6, 0.57, 0.52 and 0.31 respectively. The SFT models trained on the same conditions perform at 0.59, 0.53, 0.55 and 0.57

While we can see some transfer, for example from our binary-stability task to these real images, all models perform below the human average<sup>1</sup> (see Fig. 3). Furthermore, we again find no visible benefit of training models with interaction over supervised training when it comes to generalization.

### 4.4. Decodability analysis

To further explore whether the models have learned some task-general features, we analyzed activations during the forward pass to see if the models represent the information necessary to generalize to our set of intuitive physics tasks (see section A.6 in the Appendix for more details). If the activations encode this information, it suggests that the models have the competence to solve the intuitive physics tasks, but fail to convert that into good performance.

We find that the binary stability of a tower is already trivially decodable in the base model, and it does not change in any substantial way through either fine-tuning method (see Fig. 11 in the Appendix). This is likely because there exists an obvious pixel level shortcut where tower stability can be

determined from a small set of pixels along the horizontal center line in the image. However, while this information is decodable from everywhere throughout the model, the base model performs the binary stability task at much lower accuracies than achieved by the linear probes.

The same is true for the offset of the top block. Again, already in the base model, the x-offset is highly decodable and differences between the base, GRPO, and SFT models are small. This analysis suggests that while the models have the competence to solve either task, this does not translate to good performance, hinting at shortcut learning. This invites the question of what information the models use when solving these tasks. To better understand this, we look at what they attend to in an image. We compare the attention maps of post-trained models to those of the base model to see if they learned to focus on specific blocks or positions in the image. However, these attention maps are noisy and do not reliably show any differences in model attention as a result of either post-training method. Some examples are shown in Appendix A.7.

### 4.5. Ablations

#### 4.5.1. OTHER MODELS

To ensure our results transfer from Qwen3-VL-8B to other models, we repeat all experiments with Qwen2.5-VL-7B. Crucially, for this model, we find that generalization is even more limited (see Fig. 15). This model only transfers from the x-y side block to the x-only side block task — this is to be expected, because x-only is a subset of the x-y task. We perform a number of additional ablations on this model to test if generalization can be improved. We outline these ablations in the following paragraphs.

First, it is possible that the models learn some task-general features, which they can't properly make use of due to task-specific properties. To test this, we take the Qwen2.5-VL-7B model fine-tuned on x-only top block and supervise fine-tune it for a limited number of steps on binary-stability top block (see Section A.9.1 in the Appendix). If the model has learned task-general features over the course of the x-only top block training, it should learn the binary stability task more quickly than the base model — meaning that it should require fewer additional fine-tuning steps to reach good performance. We find that later checkpoints of the x-only trained model very quickly reaches high accuracies in the binary-stability task after just a few steps of SFT, and crucially more quickly than the base model, indicating that the model has learned some transferable features (see Fig. 16 and Section A.9.1 in the Appendix for more details).

<sup>1</sup>We calculate the human average based on publicly available, anonymized data taken from [Schulze Buschoff et al. \(2025a\)](#)

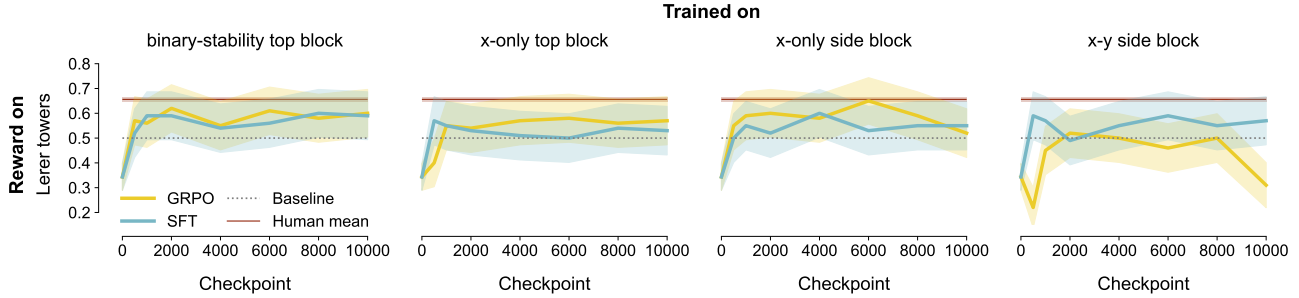


Figure 3. Qwen3-VL-8B trained on all four conditions and evaluated on the real images of block towers from Lerer et al. (2016). Crucially, we find some generalization from our synthetic *binary-stability top block* images to real images of block towers. However, we still find that no model fine-tuned on other tasks generalizes to judging the stability of real block towers. Furthermore, we find that even the models post-trained on the binary-stability task do not perform this task at a human level when presented with real images (the red line shows the mean human performance). Error bars show 95% confidence intervals.

Second, the model might need to be trained for longer to unlock generalization. To test whether generalizable physical intuitions could emerge in GRPO models over time, we trained Qwen2.5-VL-7B for up to 48,000 steps. We find that as we exceed 10,000 steps, the model tends to overfit too strongly to the specific reward function of the training task to generalize to other tasks (see Fig. 17 and Section A.9.2 in the Appendix for more details).

Third, as outlined by previous work (Schulze Buschoff et al., 2025a), it is possible that models trained on a single task fail to generalize because they are not exposed to enough variance in their post-training. To check if generalization can be improved by incorporating multiple tasks, we post-train Qwen2.5-VL-7B with GRPO, first on *x-only side block* and then on *binary-stability top block*. Since the model has been trained on the *x-only* task and also on the top block data set (albeit not at the same time) we would expect this model to generalize well to the *x-only top block* task. We find that this model is still able to perform both tasks it was trained on — however, it does not generalize to *x-only top block* (see Fig. 19 in the Appendix). We also train an SFT model in the same blocked manner, first training it on *x-only side block* and then on *binary-stability top block*. The performance of this model quickly degrades on the task it was trained on first, suggesting that there might be for training with interaction when it comes to learning multiple tasks sequentially. In contrast, a SFT variant with interleaved data, trained on *x-only side block* and *binary-stability top block* at the same time, performs reasonably well on both of its post-training tasks (see Fig. 19 and Section A.9.4 in the Appendix for more details).

Finally, we perform a number additional ablations to ensure our results are not simply artifacts of the training duration, the generation length, or the adapter rank. We report these results in Section A.9 of the Appendix. We also evaluate newer and bigger model variants, as well as models trained with another RL algorithm in Section 4.5.2 below. Ad-

ditionally, we also test the post-trained models ability to generalize to real images (see Sections 4.3 and A.10 below).

#### 4.5.2. OTHER RL IMPLEMENTATIONS

To ensure that our results transfer to other RL algorithms, we post-train Qwen3-VL-8B and Qwen3-VL-32B with Group Sequence Policy Optimization (GSPO) on the *x-only top block* task. GSPO replaces the token-level optimization in GRPO with sequence-level optimization (see Zheng et al. (2025) for more information). We find that the 8B model performs well on their post-training task, and that it shows a similar pattern of generalization to the related binary-stability task as the GRPO model (see Fig. 13). However, we still find that it does not generalize to the other tasks, such as for example *x-only side block* (see also Section A.8 in the Appendix) — this task is the same as the models’ post-training task, only with larger block displacements. If the models learned the mapping between block distance to center and the integer action space, they should in principle also be able to solve this task — but we do not find this, neither in the models trained with interaction nor the models trained with SFT.

The 32B model trained with either GRPO or GSPO does not show meaningful generalization to any condition (see Figs 14 in the Appendix). In contrast, the 32B model post-trained with SFT generalizes somewhat from the *x-only top block* post-training task to the *binary-stability top block* task. This again indicates that there is no reliable benefit of training models with interaction when it comes to learning generalizable physical intuitions.

## 5. Discussion

Recent evidence has suggested that vision language models (VLMs) do not have robust human-like intuitions about the physical world. For instance, they struggle to reason about the stability of block towers or about cause and effect (Schulze Buschoff et al., 2025a), even when they are fine-tuned on related tasks (Schulze Buschoff et al., 2025b). Humans, on the other hand, have robust intuitions about the physical world, which they learn in part from interacting with their environment.

To capture this aspect of human learning, we trained VLMs on intuitive physics tasks that require building block towers through interaction with an environment. We trained these models using the online reinforcement learning algorithm Group-Relative Policy Optimization (GRPO), and compared it to Supervised Fine-Tuning (SFT), an analogue of offline reinforcement learning (Levine et al., 2020). We then tested these trained models on held-out tower building tasks and on judging the stability of block towers.

Given the relevance of interaction for learning intuitive physics, we defined three hypotheses: (1) that GRPO-trained models would outperform SFT-trained models on held-out instances of the task they were trained on, and (2) that GRPO-trained models would generalize better than SFT-trained models to new tasks, such as judging tower stability.

Our experiments found no evidence in favor of (1). Across all four tasks, both GRPO and SFT post-training led to models performing close to ceiling on held-out test instances. This supports recent results showing that task-specific post-training can make VLMs perform well within the visual intuitive physics problems they are trained on (Balazadeh et al., 2024; Schulze Buschoff et al., 2025b).

In contrast, for (2), we found that interaction did not confer a clear advantage for generalizing to new tasks. We found that both GRPO and SFT allowed slight generalization within tasks that share the same data distribution (for example from x-only top block to binary-stability top block; note that this is not the case for the older Qwen2.5-VL-7B model). However, we find no generalization to other tasks in a zero-shot setting.

To summarize, while we find some traces of generalization, it is very limited and the post-trained models do not transfer to all related tasks. We hypothesized that interaction would be helpful for learning generalizable physical intuitions. However, we again do not find clear evidence that training these models with interaction gives them generalizable physical intuitions, nor that it is better than SFT when it comes to generalization to related tasks.

While our decodability analysis showed that the properties of interest for solving the physics tasks, such as both binary stability and x-offset, are highly decodable from activations at all intermediate layers in the base, SFT-trained and GRPO-trained models, neither post-training method was able to make the model use this information on out-of-distribution tasks.

To conclude, interaction, in the sense of one-step reinforcement learning with GRPO, does not appear to confer a general advantage for solving a family of related intuitive physics problems. Indeed, GRPO appears to perform similarly within-distribution to SFT, which is also unable to facilitate reliable generalization. These results, along with our decodability analysis, indicate that models trained with either post-training method learn non-general shortcuts rather than robust physical intuitions.

There are several avenues of research that would strengthen the conclusions of this study. We investigate only three models of sizes 7B, 8B, and 32B, using relatively small quantities of data. Future work will examine whether our conclusions apply to larger models trained on larger volumes of data. We also only investigated 1-step interactions with the environment. It remains possible that advantages of interaction only surface when models are able to interact with their environment over long state-action sequences. Future work will investigate practical methods for testing this with modern vision-language models.

## 6. Conclusion

We hypothesized that through interaction with an environment, vision language models would be able to learn generalizable physical intuitions. However, we find little evidence of this — neither models trained with GRPO nor SFT were able to reliably generalize from their training task to other related tasks. This suggests that these models are not learning true physical intuitions, but rather task-specific shortcuts.

Together, our results suggest that prominent post-training methods are constrained in the ways that they can improve models when it comes to intuitive physics. It remains unclear whether post-training models on specific cognitive tasks is sufficient for developing models that reason about the world in a human-like manner. Developing machine learning models with these abilities may require different pre- and post-training paradigms that go beyond parameter-efficient adaptation.



## References

- Baillargeon, R., Kotovsky, L., and Needham, A. The acquisition of physical knowledge in infancy. *Clarendon Press/Oxford University Press*, 1995.
- Balazadeh, V., Ataei, M., Cheong, H., Khasahmadi, A. H., and Krishnan, R. G. Synthetic vision: Training vision-language models to understand physics. *arXiv preprint arXiv:2412.08619*, 2024.
- Barsalou, L. W. Perceptual symbol systems. *Behavioral and brain sciences*, 22(4):577–660, 1999.
- Battaglia, P., Ullman, T., Tenenbaum, J., Sanborn, A., Forbus, K., Gerstenberg, T., and Lagnado, D. Computational models of intuitive physics. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, 2012.
- Binz, M., Akata, E., Bethge, M., Brändle, F., Callaway, F., Coda-Forno, J., Dayan, P., Demircan, C., Eckstein, M. K., Éltető, N., et al. Centaur: a foundation model of human cognition. *arXiv preprint arXiv:2410.20268*, 2024.
- Campbell, D., Rane, S., Giallanza, T., De Sabbata, C. N., Ghods, K., Joshi, A., Ku, A., Frankland, S., Griffiths, T., Cohen, J. D., et al. Understanding the limits of vision language models through the lens of the binding problem. *Advances in Neural Information Processing Systems*, 37: 113436–113460, 2024.
- Chomsky, N. *Aspects of the Theory of Syntax*. MIT Press, 1965.
- Chu, J. and Schulz, L. E. Play, curiosity, and cognition. *Annual Review of Developmental Psychology*, 2(1):317–343, 2020.
- Chu, T., Zhai, Y., Yang, J., Tong, S., Xie, S., Schuurmans, D., Le, Q. V., Levine, S., and Ma, Y. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. *arXiv preprint arXiv:2501.17161*, 2025.
- Clark, A. *Being there: Putting brain, body, and world together again*. MIT press, 1998.
- Collins, K. M., Wong, C., Feng, J., Wei, M., and Tenenbaum, J. B. Structured, flexible, and robust: benchmarking and improving large language models towards more human-like behavior in out-of-distribution reasoning tasks. *arXiv preprint arXiv:2205.05718*, 2022.
- Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36, 2024.
- Frankland, S. M., Webb, T., Lewis, R., and Cohen, J. D. No coincidence, george: Processing limits in cognitive function reflect the curse of generalization. 2021.
- Gan, C., Schwartz, J., Alter, S., Mrowca, D., Schrimpf, M., Traer, J., De Freitas, J., Kubilius, J., Bhandwaldar, A., Haber, N., et al. Threedworld: A platform for interactive multi-modal physical simulation. *arXiv preprint arXiv:2007.04954*, 2020.
- Geirhos, R., Temme, C. R., Rauber, J., Schütt, H. H., Bethge, M., and Wichmann, F. A. Generalisation in humans and deep neural networks. *Advances in neural information processing systems*, 31, 2018.
- Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., and Wichmann, F. A. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.
- Gibson, J. J. *The ecological approach to visual perception: classic edition*. Psychology press, 1979.
- Gopnik, A., Meltzoff, A. N., and Kuhl, P. K. *The scientist in the crib: Minds, brains, and how children learn*. William Morrow & Co., 1999.
- Griffiths, T. L. and Tenenbaum, J. B. Theory-based causal induction. *Psychological review*, 116(4):661, 2009.
- Han, D., Han, M., and Unsloth team. Unsloth. *Unsloth*, 2023. URL <http://github.com/unslothai/unsloth>.
- Han, Z., Gao, C., Liu, J., Zhang, J., and Zhang, S. Q. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*, 2024.
- Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.
- Hu, J., Zhang, Y., Han, Q., Jiang, D., Zhang, X., and Shum, H.-Y. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *arXiv preprint arXiv:2503.24290*, 2025.
- Hussain, Z., Binz, M., Mata, R., and Wulff, D. U. A tutorial on open-source large language models for behavioral science. *Behavior Research Methods*, 56(8):8214–8237, 2024.
- Jin, Z., Chen, Y., Leeb, F., Gresele, L., Kamal, O., Zhiheng, L., Blin, K., Adauto, F. G., Kleiman-Weiner, M., Sachan, M., et al. Cladder: Assessing causal reasoning in language models. In *Thirty-seventh conference on neural information processing systems*, 2023.
- Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. Building machines that learn and think like people. *Behavioral and brain sciences*, 40:e253, 2017.

- Lerer, A., Gross, S., and Fergus, R. Learning physical intuition of block towers by example. In *International conference on machine learning*, pp. 430–438. PMLR, 2016.
- Levine, S., Kumar, A., Tucker, G., and Fu, J. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.
- Liu, Z., Chen, C., Li, W., Qi, P., Pang, T., Du, C., Lee, W. S., and Lin, M. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.
- Mecattaf, M. G., Slater, B., Tešić, M., Prunty, J., Voudouris, K., and Cheke, L. G. A little less conversation, a little more action, please: Investigating the physical common-sense of llms in a 3d embodied environment. *arXiv preprint arXiv:2410.23242*, 2024.
- Merleau-Ponty, M. *Phenomenology of Perception*. Routledge & Kegan Paul, 1945.
- Motamed, S., Culp, L., Swersky, K., Jaini, P., and Geirhos, R. Do generative video models learn physical principles from watching videos? *arXiv preprint arXiv:2501.09038*, 2025.
- Nicolopoulou, A. Play, cognitive development, and the social world: Piaget, vygotsky, and beyond. *Human Development*, 36(1):1–23, 1993.
- Ostrovski, G., Castro, P. S., and Dabney, W. The difficulty of passive learning in deep reinforcement learning. In *Advances in Neural Information Processing Systems*, 34, 2021.
- Piaget, J. *The Origins of Intelligence in Children*. International University Press, 1952.
- Piloto, L. S., Weinstein, A., Battaglia, P., and Botvinick, M. Intuitive physics learning in a deep-learning model inspired by developmental psychology. *Nature human behaviour*, 6(9):1257–1267, 2022.
- Rahmanzadehgervi, P., Bolton, L., Taesiri, M. R., and Nguyen, A. T. Vision language models are blind. In *Proceedings of the Asian Conference on Computer Vision*, pp. 18–34, 2024.
- Schulz, L. E. and Bonawitz, E. B. Serious fun: preschoolers engage in more exploratory play when evidence is confounded. *Developmental Psychology*, 43(4):1045, 2007.
- Schulze Buschoff, L. M., Akata, E., Bethge, M., and Schulz, E. Visual cognition in multimodal large language models. *Nature Machine Intelligence*, pp. 1–11, 2025a.
- Schulze Buschoff, L. M., Voudouris, K., Akata, E., Bethge, M., Tenenbaum, J. B., and Schulz, E. Testing the limits of fine-tuning for improving visual cognition in vision language models. In *Forty-second International Conference on Machine Learning*, 2025b.
- Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.
- Shapiro, L. and Spaulding, S. Embodied Cognition. In Zalta, E. N. and Nodelman, U. (eds.), *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Summer 2025 edition, 2025.
- Silver, D. and Sutton, R. S. Welcome to the era of experience. *Google AI*, 1, 2025.
- Smith, L. and Gasser, M. The development of embodied cognition: Six lessons from babies. *Artificial Life*, 11(1-2):13–29, 2005.
- Smith, P. K. Does play matter? functional and evolutionary aspects of animal and human play. *Behavioral and Brain Sciences*, 5(1):139–155, 1982.
- Spelke, E. S. Principles of object perception. *Cognitive science*, 14(1):29–56, 1990.
- Spelke, E. S. and Kinzler, K. D. Core knowledge. *Developmental science*, 10(1):89–96, 2007.
- Sutton, R. S., Barto, A. G., et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.
- Varela, F. J., Thompson, E., and Rosch, E. *The embodied mind, revised edition: Cognitive science and human experience*. MIT press, 1991.
- Wu, Y., Zhou, Y., Ziheng, Z., Peng, Y., Ye, X., Hu, X., Zhu, W., Qi, L., Yang, M.-H., and Yang, X. On the generalization of sft: A reinforcement learning perspective with reward rectification. *arXiv preprint arXiv:2508.05629*, 2025.
- Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.
- Yu, Q., Zhang, Z., Zhu, R., Yuan, Y., Zuo, X., Yue, Y., Dai, W., Fan, T., Liu, G., Liu, L., et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.
- Zheng, C., Liu, S., Li, M., Chen, X.-H., Yu, B., Gao, C., Dang, K., Liu, Y., Men, R., Yang, A., et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

## A. Appendix

### A.1. Data examples

#### A.1.1. TOP BLOCK DATASET

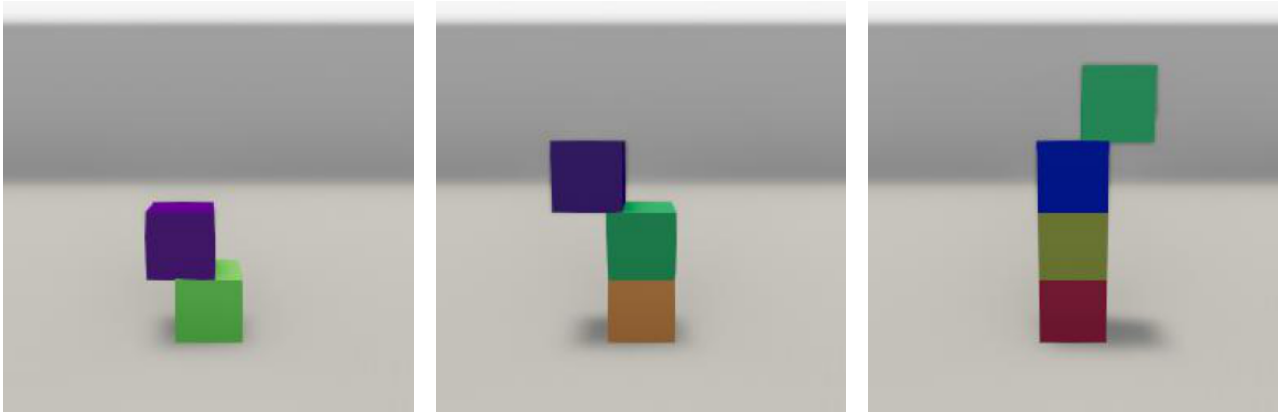


Figure 4. Example images for the *top block* dataset. Images feature towers with 2 to 4 blocks with the top block displaced.

#### A.1.2. SIDE BLOCK DATASET

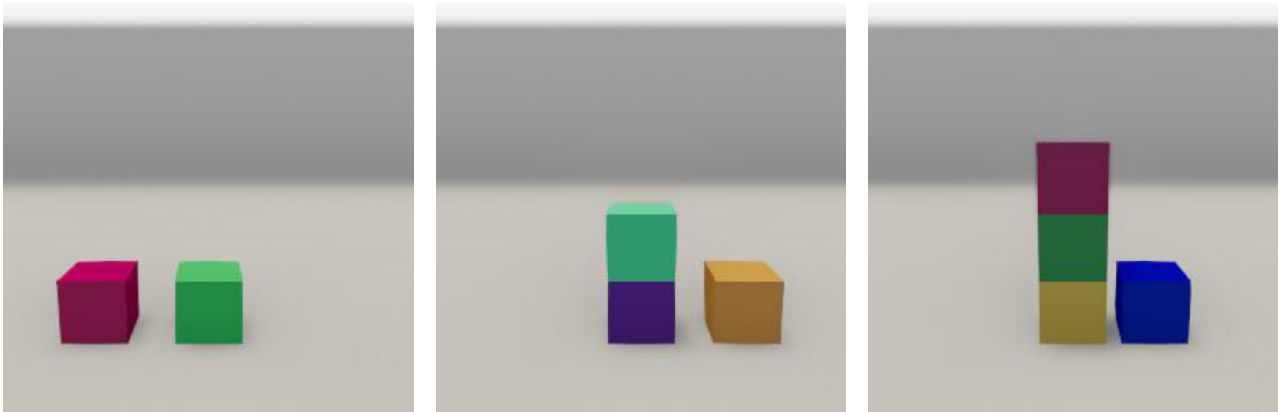


Figure 5. Example images for the *side block* dataset. Images feature towers with 1 to 3 blocks with a misplaced block to the side.

#### A.1.3. LERER DATASET

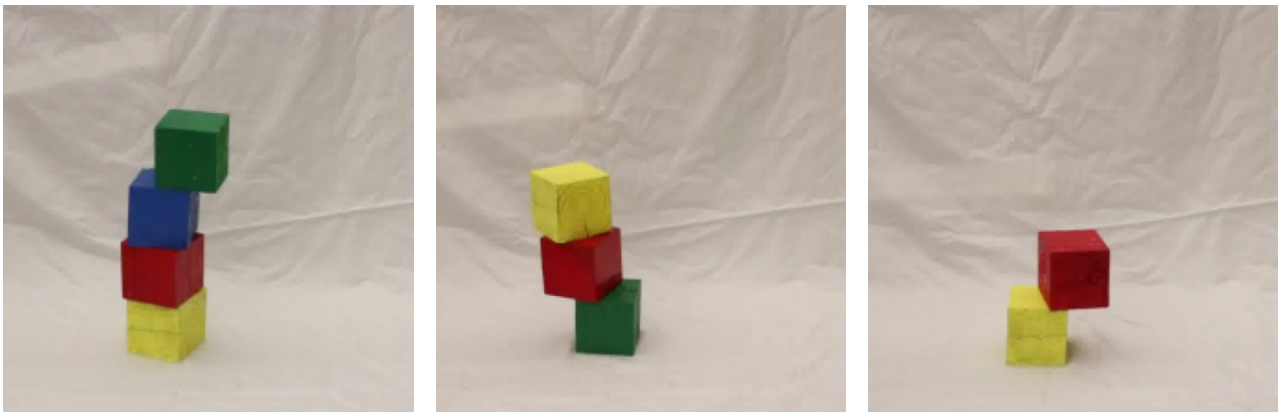


Figure 6. Example images for the *Lerer* evaluation dataset. Images are real pictures of block towers with 2 to 4 blocks.

### A.2. Main result tables

The following table shows the results in Fig. 2 for all Qwen3-VL-8B models after 10.000 steps of GRPO and SFT:

Evaluated on	Trained for 10.000 steps with GRPO on			
	x-only binary stability	x-only top block	x-only side block	x-y side block
x-only binary stability	0.943	0.788	0.503	-0.264
x-only top block	10.626	19.999	0.042	0.75
x-only side block	-0.373	-0.152	19.998	5.396
x-y side block	-0.723	-0.535	3.738	19.868

Evaluated on	Trained for 10.000 steps with SFT on			
	x-only binary stability	x-only top block	x-only side block	x-y side block
x-only binary stability	0.969	0.624	0.596	0.506
x-only top block	12.170	20	-0.06	-0.214
x-only side block	-0.373	-0.256	20	19.316
x-y side block	-0.759	-0.582	-5	15.839

### A.3. Reward function visualisation

Below, we visualize the reward functions for the *x-only* and *x-y* tasks. For *x-only*, the reward for answers that result in an unstable tower is calculated as  $2 \cdot e^{-d^2} - 2$ , where  $d$  is the distance on the  $x$ -dimension. For answers that result in a stable tower, we compute the reward as  $20 \cdot e^{-d^2}$  (see Fig. 7 below).

For *x-y*, we compute the euclidean distance between the final position of the moved block and the optimal position on top of the tower. For answers that are above ground but do not result in a stable bigger tower, we calculate the reward as  $2 \cdot e^{-d^2} - 2$ . For answers that are within the tower, we compute  $2 \cdot e^{-d^2} - 4$ . And for answers that result in a stable bigger tower, we compute the reward as  $20 \cdot e^{-d^2}$  (see Fig. 8 below).

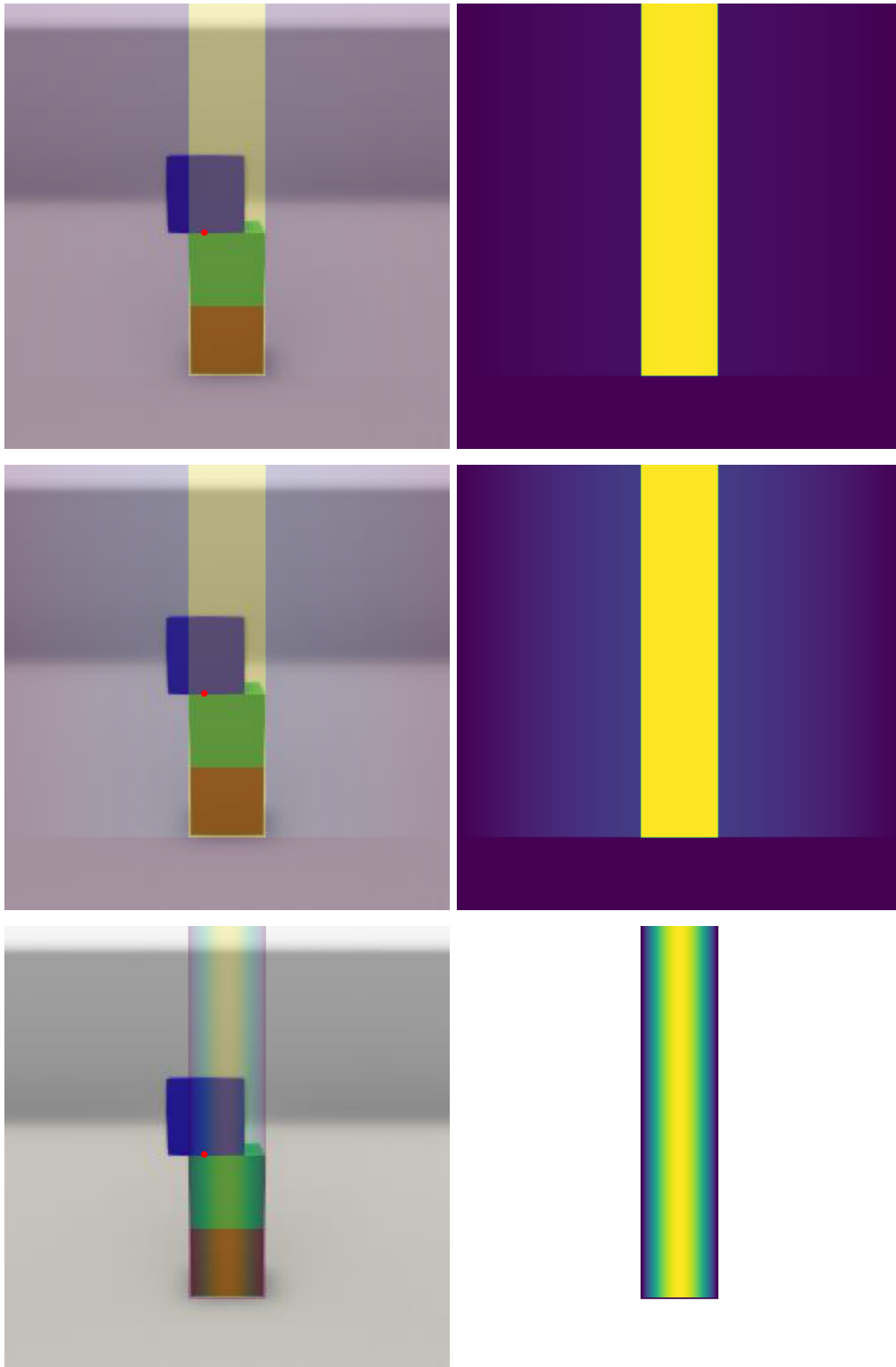


Figure 7. Reward function for the x-only task on the top block dataset. The red dot sits at the lower center of the block from which the reward is calculated. The first row shows the non normalized reward values. The second row shows the symmetric-log transformed reward values. The third row shows the log normalized reward values.

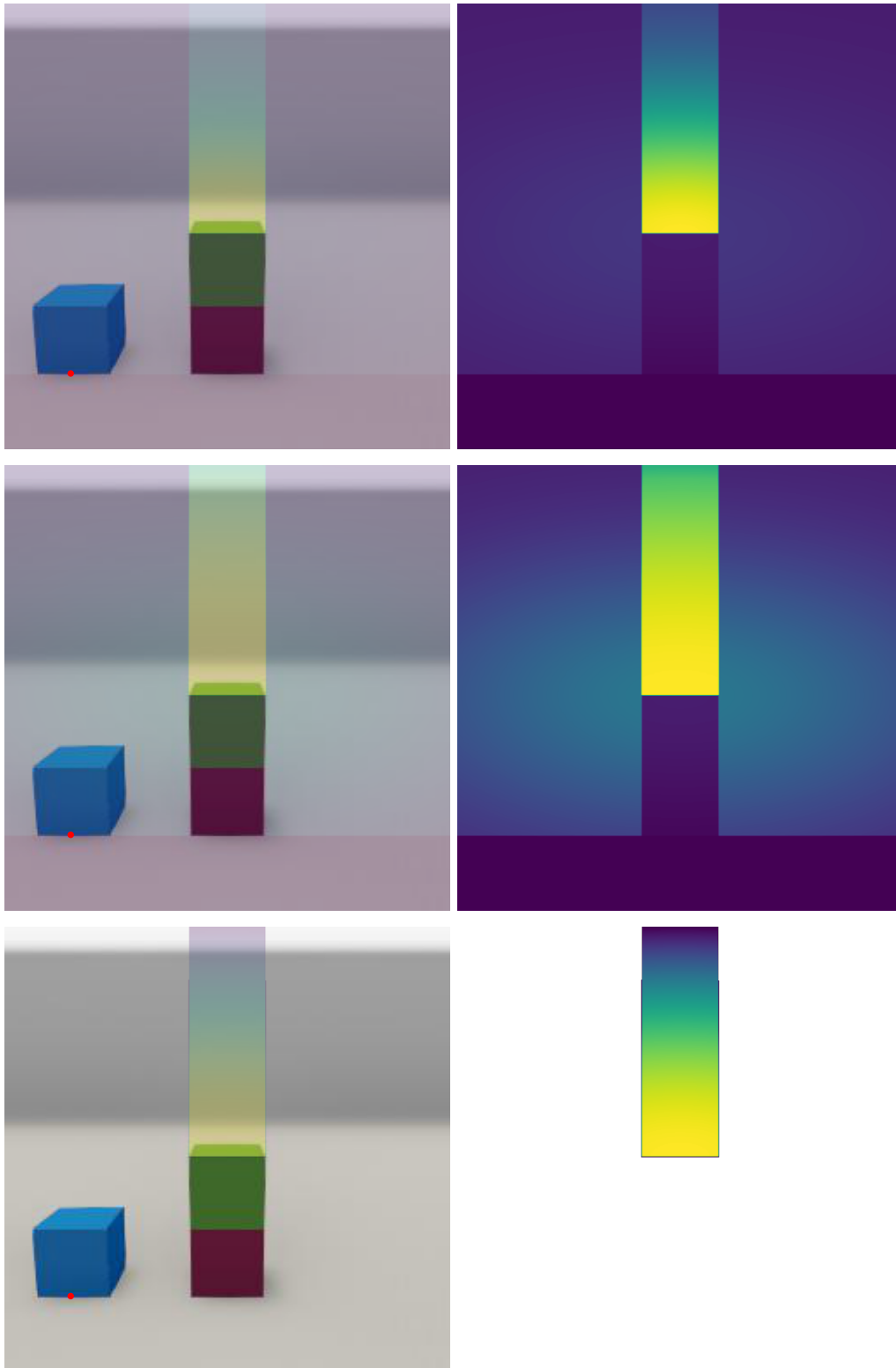


Figure 8. Reward function for the x-y task on the side block dataset. The red dot sits at the lower center of the block from which the reward is calculated. The first row shows the non normalized reward values. The second row shows the symlog transformed reward values. The third row shows the log normalized reward values.

#### A.4. Task Prompts

We use the following prompt variations for the four tasks depending on action type and dataset (see Fig. 1 for an overview):

For *binary judgment* on the *top block* dataset:

In the image you see a block tower with a single misplaced block on top. Your task is to determine if this is a stable tower. Respond with Yes if the tower is stable or No if it is not stable. Return your final answer between `<answer>` `</answer>`.

For *x-only* on the *top block* dataset:

In the image you see a block tower with a single misplaced block on top. Your task is to build a stable tower by moving the top block to the most stable position. You can move the top block to the left or right, by responding with an integer between -600 and 600. Return your final answer between `<answer>` `</answer>`.

For *x-only* on the *side block* dataset:

In the image you see a block tower in the centre with a single misplaced block to the side. Your task is to build a stable tower by moving the misplaced block to the most stable position on the top of the tower. You can move the top block to the left or right, by responding with an integer between -600 and 600. Return your final answer between `<answer>` `</answer>`.

For *x-y* on the *side block* dataset:

In the image you see a block tower in the centre with a single misplaced block to the side. Your task is to build a stable tower by moving the misplaced block to the most stable position on the top of the tower. You can move the misplaced block to the left or right and up, by responding with two integers. The first integer should be between -600 and +600 and moves the block left or right. The second integer should be between 0 and +1000 and moves the block up. Return your final answer between `<answer>` `</answer>`.

For *binary-stability* on the *Lerer* dataset:

In the image you see a block tower. Your task is to determine if this is a stable tower. Respond with Yes if the tower is stable or No if it is not stable. Return your final answer between `<answer>` `</answer>`.

For *x-only* on the *top block* dataset with reasoning (long generation):

In the image you see a block tower with a single misplaced block on top. Your task is to build a stable tower by moving the top block to the most stable position. You can move the top block to the left or right, by responding with an integer between -600 and 600. Provide your reasoning between `<think>` and `</think>`. You can think about the problem for as long as you'd like. While thinking, you should robustly verify your solution. Return your final answer between `<answer>` `</answer>`.

### A.5. Training logs

#### A.5.1. QWEN3-VL-8B

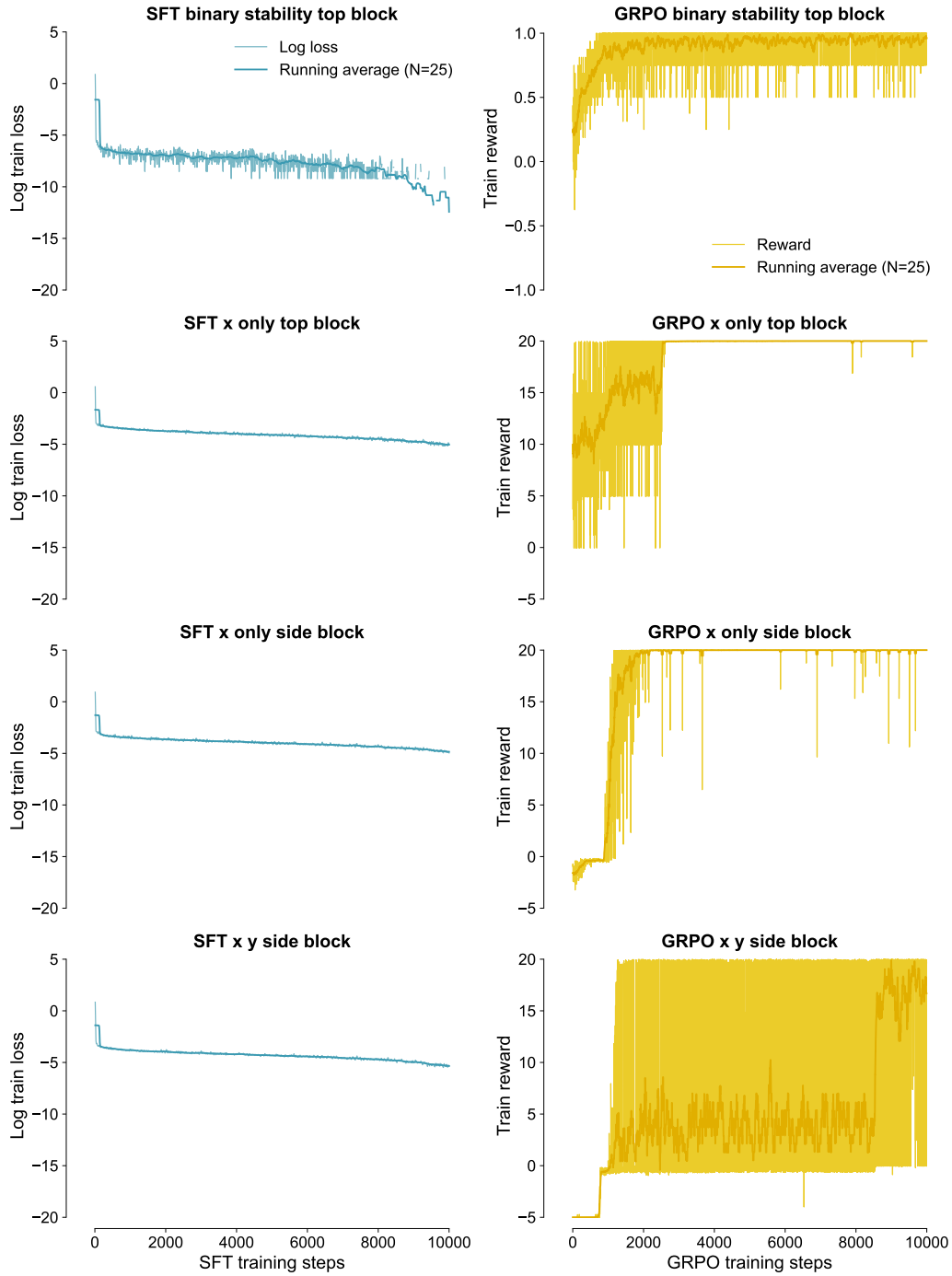


Figure 9. Training logs for the SFT (left) and GRPO (right) Qwen3-VL-8B models trained on all datasets. For the SFT models, we show the log loss and a running average with a window of 25. For the GRPO models, we show the mean reward and a running average with a window of 25.



A.5.2. QWEN2.5-VL-7B

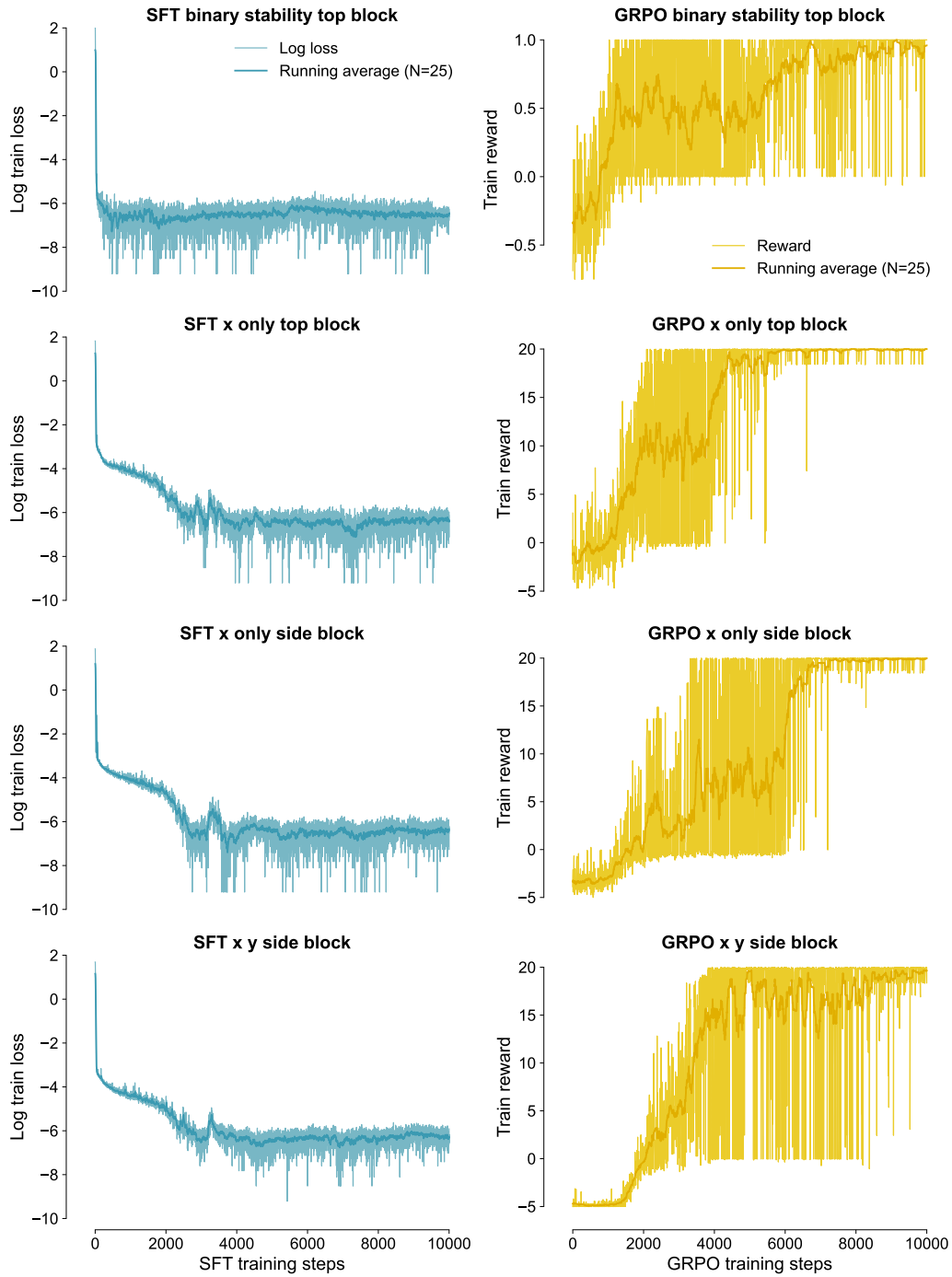


Figure 10. Training logs for the SFT (left) and GRPO (right) Qwen2.5-VL-7B models trained on all datasets. For the SFT models, we show the log loss and a running average with a window of 25. For the GRPO models, we show the mean reward and a running average with a window of 25.

### A.6. Decoding analysis

For the top block dataset, we train linear probes on the representation of the model at each layer to predict the binary stability of a tower and the x-offset of the top block from those representations. Since the image tokens appear before the text tokens, the linear probes only have access to the visual information. We run this process with 10-fold cross validation using 600 images in total. For the binary stability analysis, we train L2 regularized logistic regression models on the representations. For the x-offset analysis, we train linear regression models with spherical Gaussian priors.

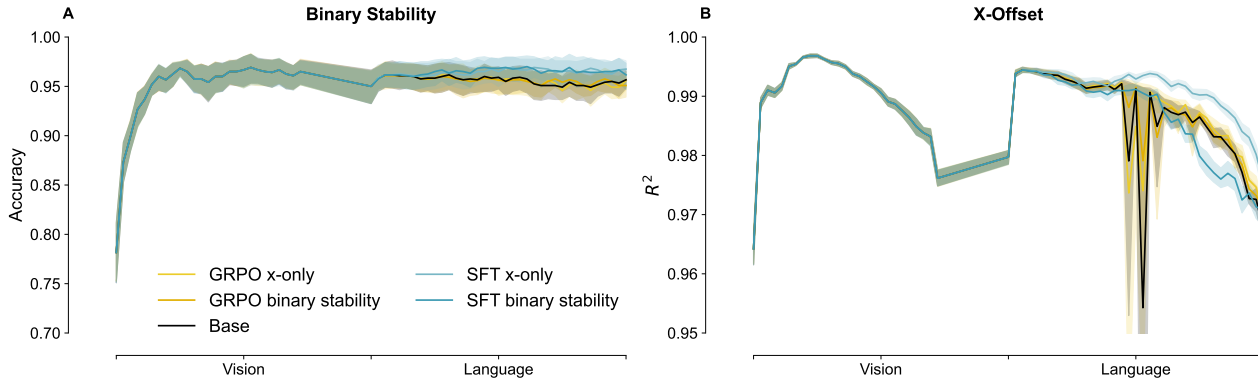


Figure 11. Physical property decodability analysis. (A) shows the decodability of the ground truth binary stability for the base model, as well as models post-trained with GRPO and SFT on x-only top block and binary stability top block. (B) shows the decodability of the x-offset of the uppermost block in the top block dataset for the same set of models.

### A.7. Attention maps

To better understand how finetuned models learn to solve our tasks, we compared the attention maps of the fine-tuned models and the base model. More specifically, our goal was to provide a qualitative comparison of how much the last token in the question prompt attends to the different image tokens, throughout the layers of the language model. In Fig. 12, we show the attention maps averaged across heads for each layer for both the base model and a GRPO model post-trained on x-only top block. As seen in the example below, there is no clear change as a result of the post-training method that would give us a better understanding of the strategy used by the post-trained models.

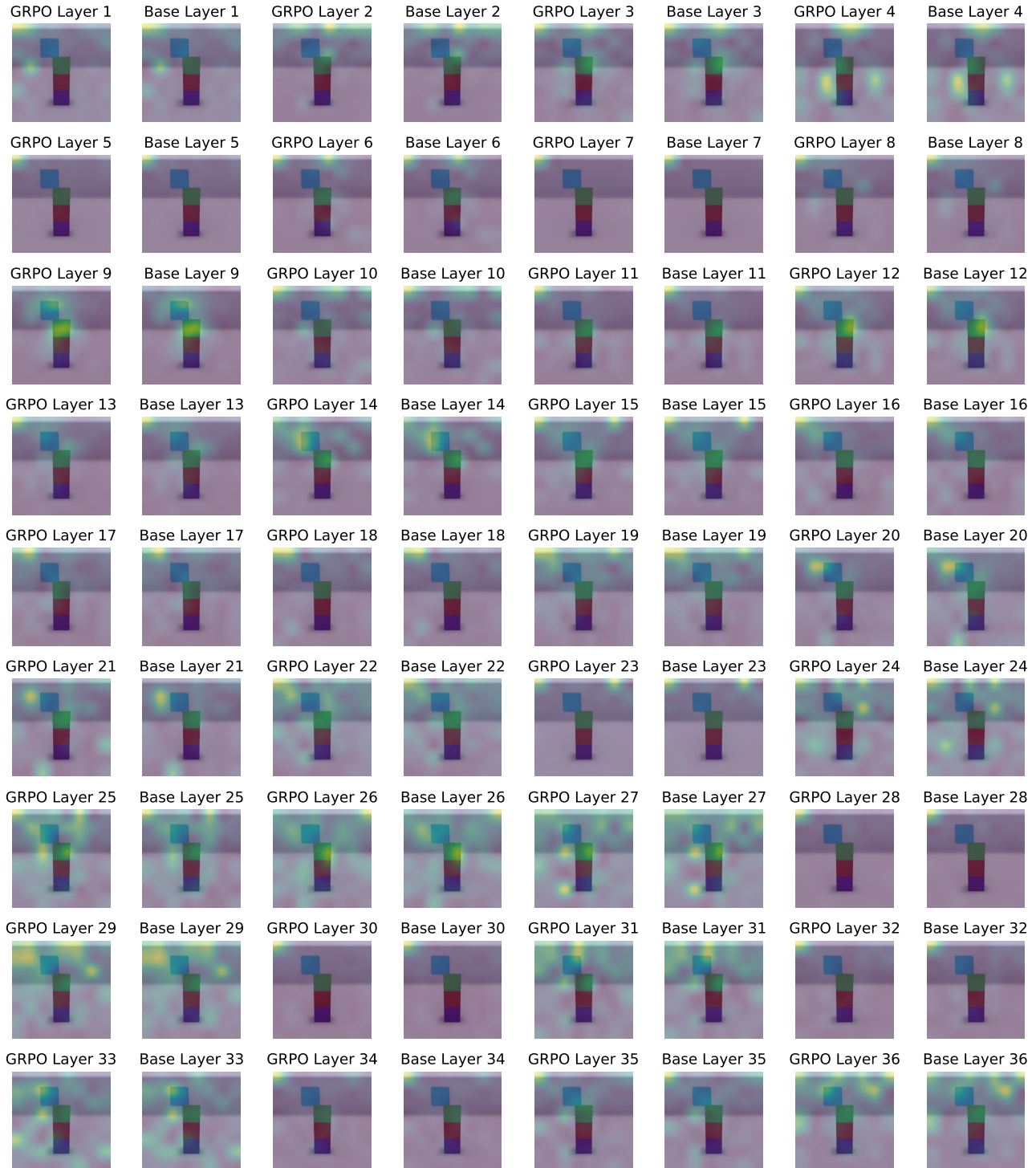


Figure 12. Attention maps for the base model and the model post-trained with GRPO on x-only top block. The model is asked if a given tower is stable or not. Attention maps over the different heads are averaged in each layer.

### A.8. Bigger model and another RL implementation

To test whether our results generalize to other models, we train Qwen3-VL-32B with GRPO and SFT on the *x-only top block* task (see Fig. 14 below). To test whether our results generalize to other RL implementations, we also train Qwen3-VL-8B and Qwen3-VL-32B with GSPO on the *x-only top block* task (see Fig. 13 and Fig. 14 below).

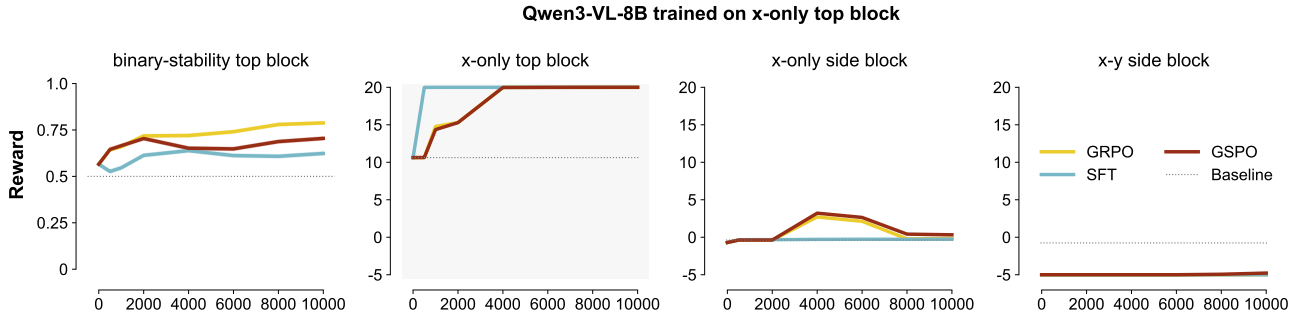


Figure 13. Qwen3-VL-8B model trained with GRPO, GSPO, and SFT on the *x-only top block* task. The model also does not generalize from its’ fine-tuning task to other related tasks. Noticeably, the model is above chance for the *binary-stability top block* from the get-go and improves slightly over the course of training on the related *x-only top block* task under all post-training regimes.

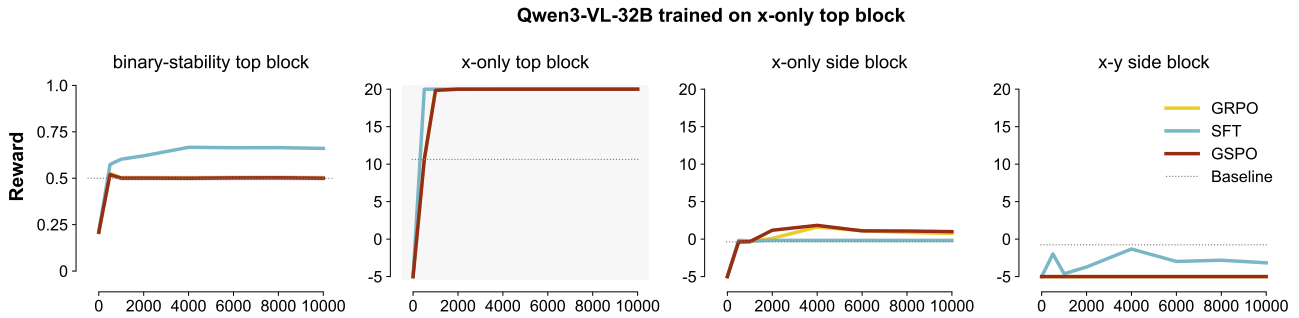


Figure 14. Qwen3-VL-32B model trained with GRPO, GSPO, and SFT. The model is trained on the *x-only top block* task. Noticeably, the SFT post-trained model improves slightly over the course of training on the related *binary-stability top block*. In contrast, the GRPO/GSPO post-trained models do not generalize from the *x-only top block* task to the *binary-stability top block* task.

### A.9. Additional checks on Qwen2.5-VL-7B

We also fine-tuned Qwen2.5-VL-7B on the full set of task and action combinations (see Fig. 15 for the full set of results). We find that this model, in contrast to Qwen3-VL-8B (see Fig. 2) shows no trace of generalization. We first test whether the model has learned some generalizable understanding after all, that would lead to faster supervised fine-tuning (see Section A.9.1). We then test a number of ablations to test whether other parameter combinations could lead to a better generalizing model (see Section A.9.3).

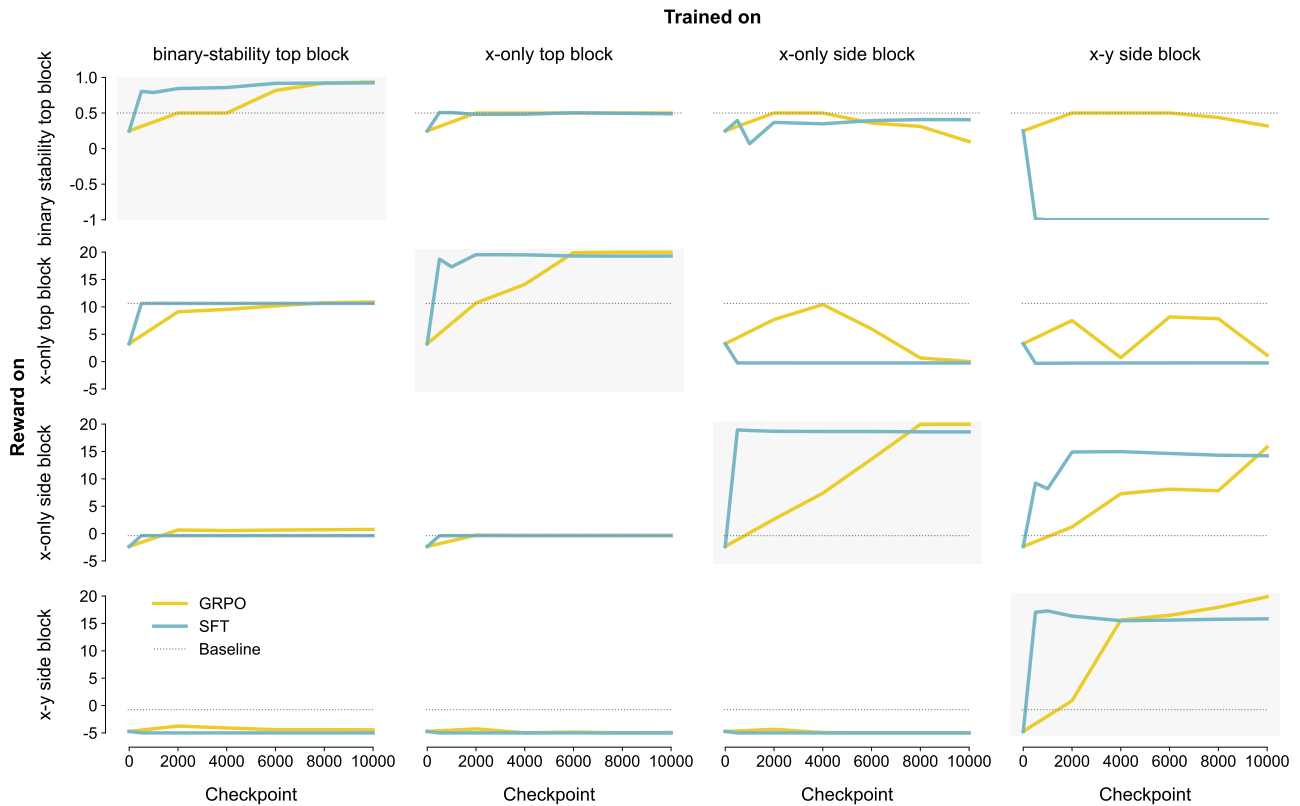


Figure 15. Qwen2.5-VL-7B performance by test task and training task. Rows show models evaluated on a given task. Columns show models trained on a given task. The blue and orange lines show the performance of the models trained with SFT and GRPO, respectively. The grey dotted line shows the baseline for the evaluation task. Plots on the diagonal show within-task performance, meaning models are evaluated on the same task they are trained on. All other subplots represent some degree of generalization.

A.9.1. ADDITIONAL SUPERVISED FINE-TUNING

It is possible that the models have learned some task-general features, but that they fail to perform well on new tasks due to some task-specific properties. To test this, we take checkpoints of the GRPO and SFT x-only top block models, and fine-tune them on the binary stability top block task with some additional steps of SFT. If the models have learned some task-general features, they should learn the binary stability task more quickly than the base model — this means that they should require fewer additional fine-tuning steps to reach good performance.

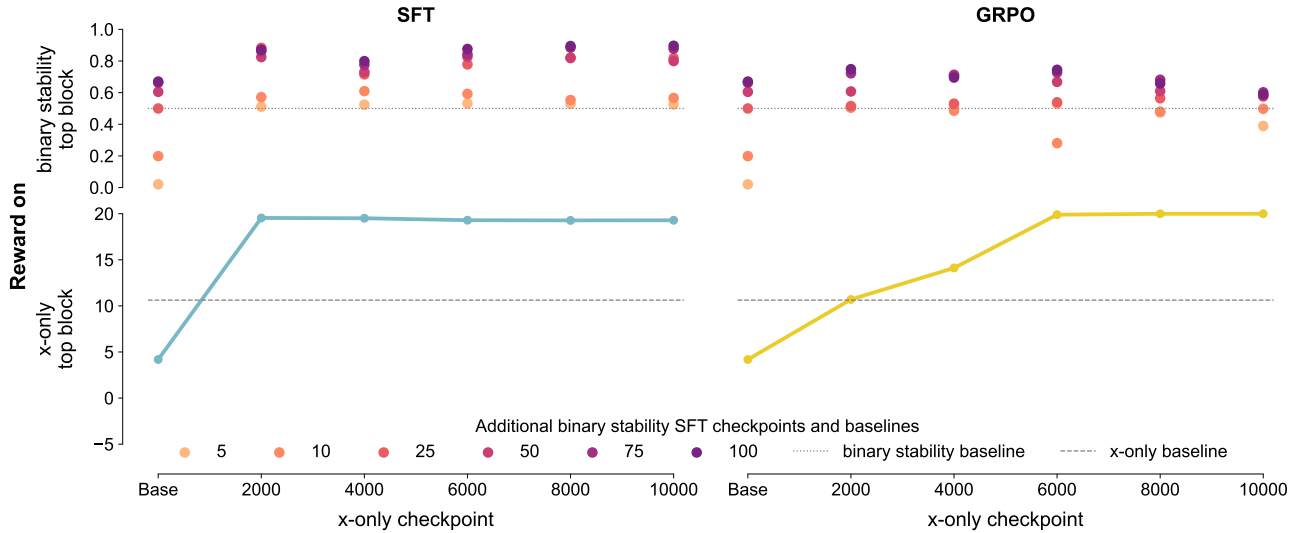


Figure 16. Generalization after additional post-training for Qwen2.5-VL-7B. The blue line on the bottom left shows the test performance of the SFT-trained model and the orange line on the bottom right shows that of the GRPO-trained model. Both models were trained on the x-only top block task. For each checkpoint, we take the model trained on this task up to that checkpoint and train it with SFT on the binary stability top block task for up to 100 additional steps. The stacked dots on the top show different checkpoints of the additional binary stability trained models, based on the x-only model at that checkpoint, and the reward they achieve on the binary stability task.

We see that the models very quickly reach high accuracies in the binary stability task after just a few steps of supervised fine-tuning (see Fig. 16). In comparison, the base model fine-tuned with the same number of steps performs less well — while it takes 25 steps for the base model to reach the random baseline for this binary task, all SFT x-only top block checkpoints are already over the baseline after 5 additional SFT steps on binary stability. This is likely in part because the base model has not yet learned to format its answers correctly, whereas all post-trained models have experience with the correct answer format. However, after 100 steps of SFT on binary stability, the base model only returns legal answers but still achieves a lower accuracy than the post-trained models with the same number of additional SFT steps, meaning formatting can not explain the whole performance gap.

We see that SFT post-trained models in general achieve slightly higher accuracies than their GRPO counterparts after 100 additional steps of SFT on the binary stability task: the models trained with SFT on x-only top block up to 2000, 4000, 6000, 8000, and 10000 steps achieve accuracies of 0.778, 0.844, 0.829, 0.806, and 0.800. In contrast, the same checkpoints for the GRPO trained models get accuracies of 0.749, 0.695, 0.745, 0.659, and 0.588. We also see that GRPO models from earlier x-only checkpoints are able to achieve higher accuracies on the binary stability task than later checkpoints.

A.9.2. LONGER TRAINING HORIZON

We find that training models with GRPO for longer only leads to overfitting to the specific training task (see Fig. 17). As training exceeds 10.000 steps, models tend to overfit too strongly to the specific reward function of the training task to generalize to other tasks — while we still saw some generalization for the *x-y side block* trained model to the *x-only side block* task, this disappears as the models are trained for longer. The results reported above all use a restricted generation length due to resource constraints.

To test whether generalizable physical intuitions could emerge in GRPO models over time, we trained Qwen2.5-VL-7B for up to 48.000 steps. We find that as we exceed 10.000 steps, the model tends to overfit too strongly to the specific reward function of the training task to generalize to other tasks — while we still saw some generalization for the *x-y side block* trained model to the *x-only side block* task, this disappears as the models are trained for longer.

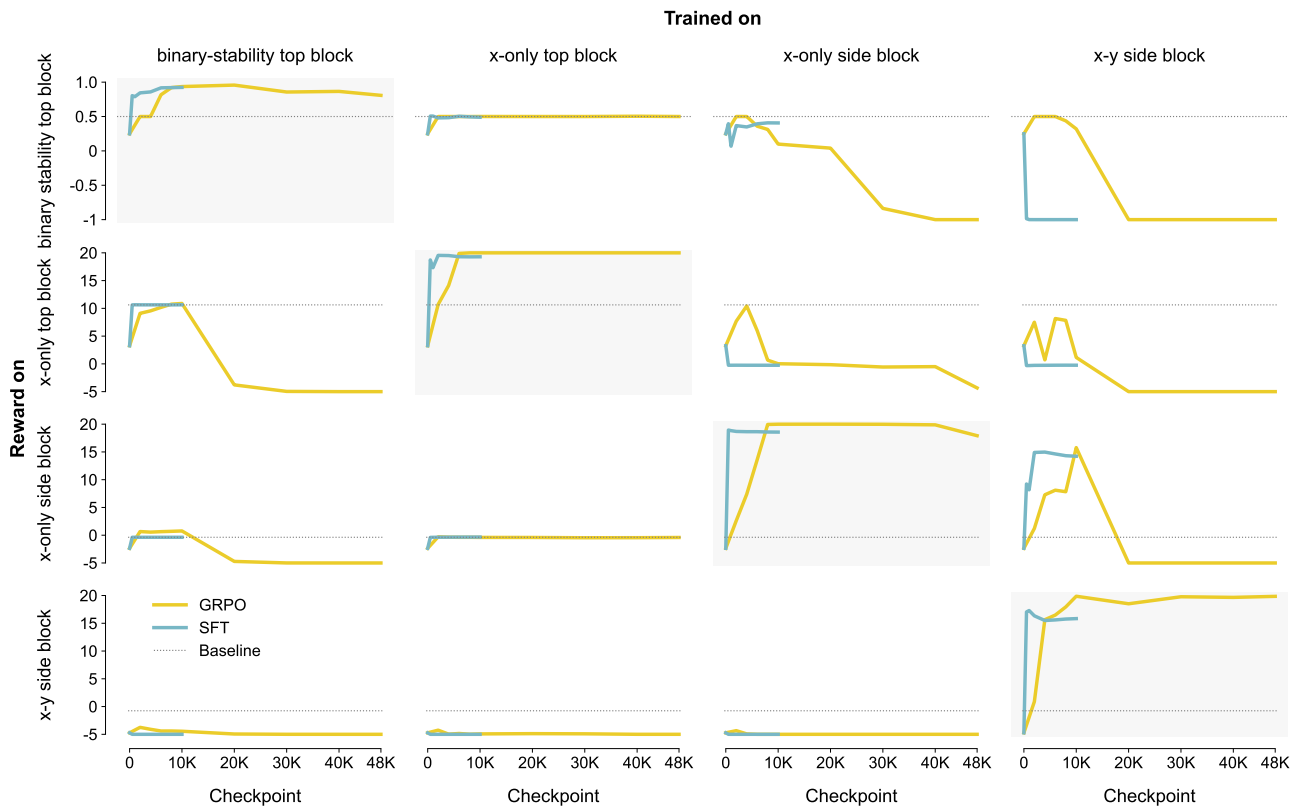


Figure 17. Performance for a longer horizon. We show the same plot as 15 but with results for up to 48K training steps for the Qwen2.5-VL-7B GRPO models. Performance is shown for each combination of test task and training task. Rows show models evaluated on a given task. Columns show models trained on a given task. The blue and orange lines show the performance of the models trained with SFT and GRPO, respectively. The grey dotted line shows the baseline for the evaluation task. Plots on the diagonal show within-task performance, meaning models are evaluated on the same task they are trained on. All other subplots represent some degree of generalization.

A.9.3. TRAINING ABLATIONS

To test if allowing the model to reason about the task for longer improves generalization, we train a model on the *x-only top block* task with a longer generation length (see A.4 for the reasoning prompt). However, as shown in Fig. 18, we find that this model also does not generalize to the other tasks.

The results we report use a default rank of 16 for all models. To make sure that this does not cause the models to overfit to our task specifically, we train models on the *x-only top block* task using ranks of 1 and 8 instead. We find that these models show the same failure to generalize. While models of all ranks learn to perform well on their training task, they do not generalize to any other task (see Fig. 18 in the Appendix).

Additionally, to ensure that the models do not suffer from overfitting the vision encoder, we train a model with the standard rank of 16 but without fine-tuning the vision encoder. We find that this model shows a similar performance over all tasks as the model with vision fine-tuning, again not generalizing to other related tasks from the training task (see Fig. 18 in the Appendix).

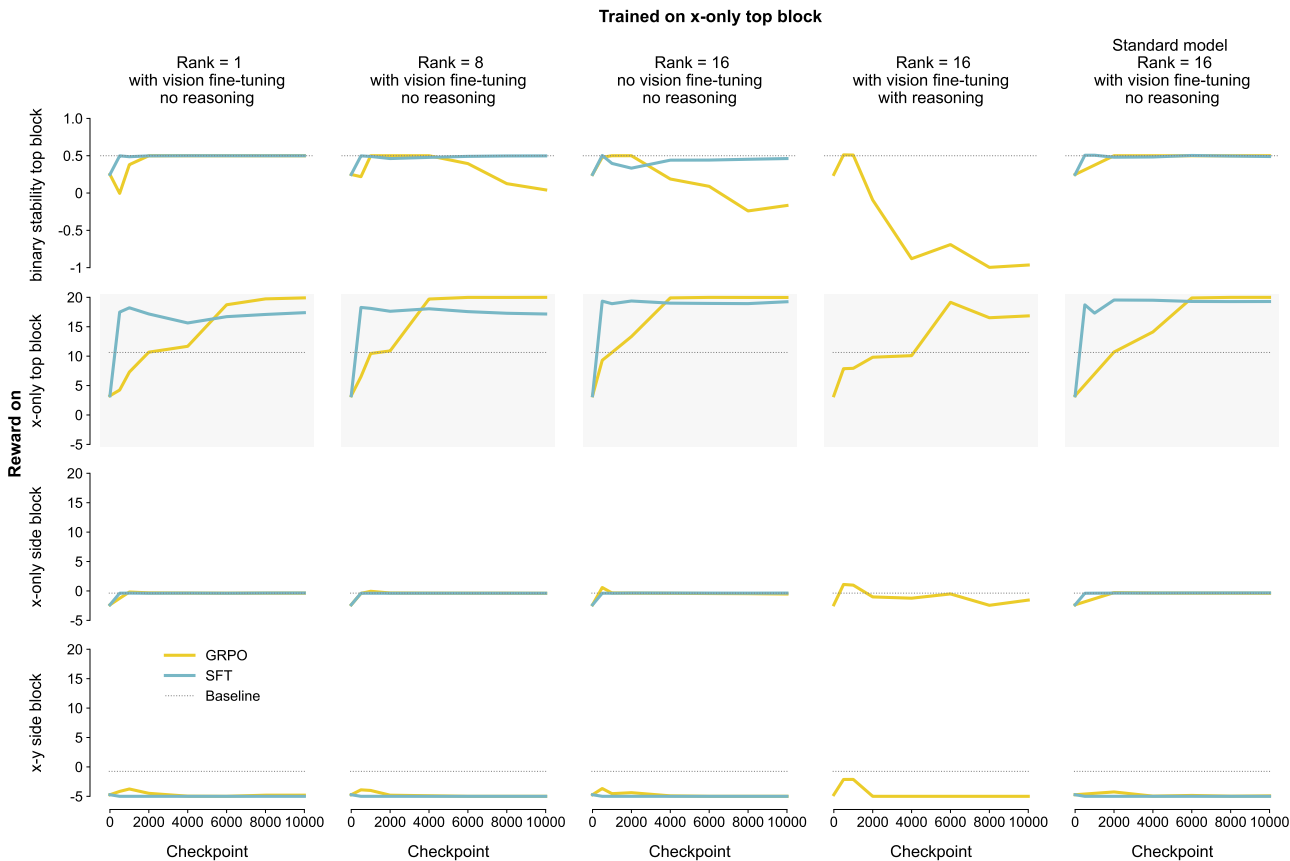


Figure 18. Ablation checks. All models are trained on x-only top block but they have lower ranks (1 and 8 compared to 16) or are trained without fine-tuning of the vision encoder or they are trained with reasoning (a larger generation length). All ablations learn to perform well on the task they are trained on (shown in the second row). However, all models fail to generalize to other related tasks — just as the standard model we used throughout our experiments (last column).



A.9.4. BLOCKED AND INTERLEAVED JOINT TRAINING

To test whether models could generalize if they are exposed to multiple tasks at the same time, we trained models on two tasks: *x-only side block* and *binary-stability top block*. Since this model has seen the *x-only* task and also the *top block* data set (albeit not at the same time), it should be able to generalize to the *x-only top block* data set. We show results for this in the figure below.

We find that the GRPO model that has been trained on both tasks can still perform both tasks. The model has some trouble keeping the correct formatting for the first task block it was trained on, but filtering only legal answers reveals that it still retains the capacity to solve it. The SFT model on the other hand quickly degrades in performance on the task it was trained on first. This indicates some benefit of GRPO when training models on multiple tasks successively. However, the joint SFT model that is trained on both tasks at the same time, in contrast to in a blocked manner, can overcome this shortcoming, performing reasonably well on both of its post-training tasks.

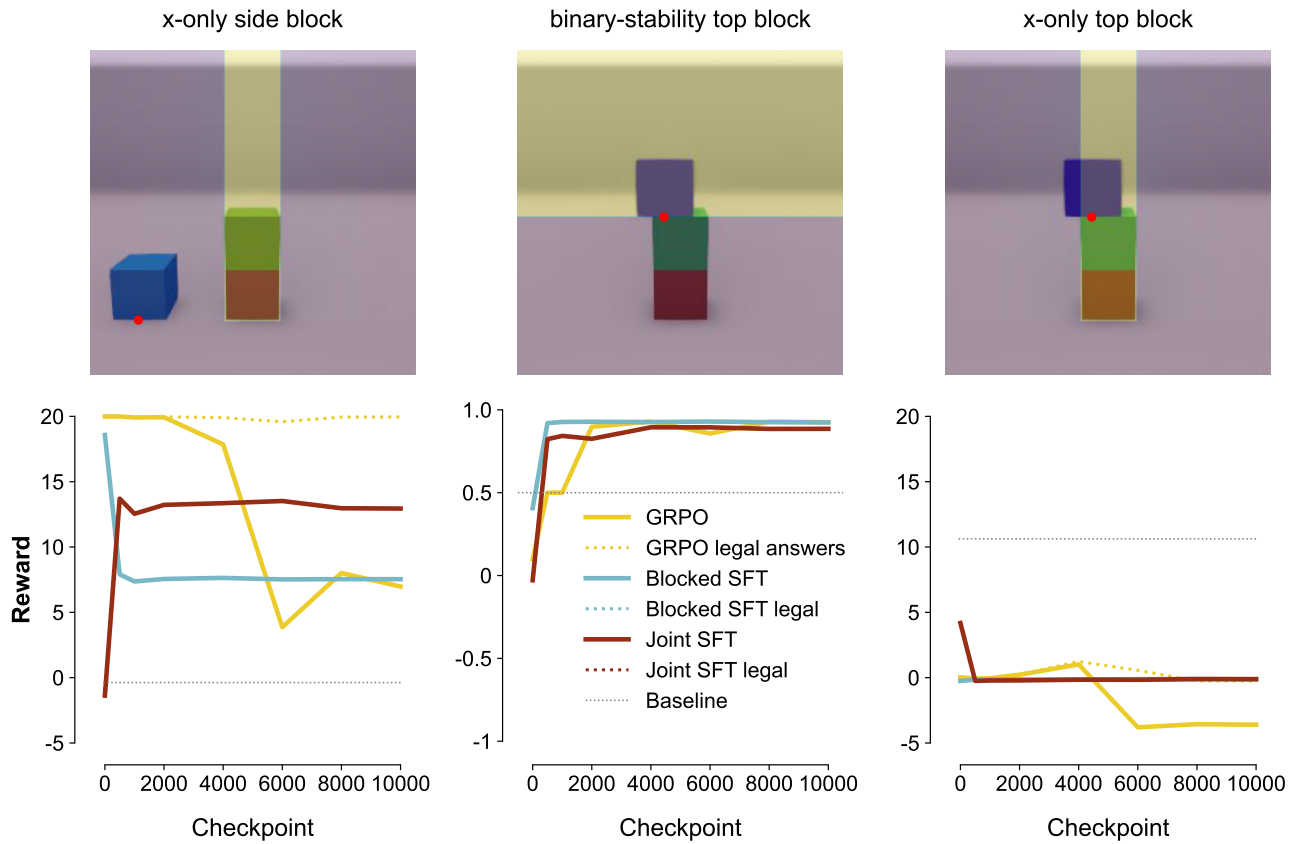


Figure 19. Blocked joint training. The model was first trained for 10,000 steps on *x-only side block*. It is then trained for 10,000 steps on *binary-stability top block* and performance is shown below over these 10,000 steps. The model forgets the proper formatting of responses for the initial *x-only side block* task (see continuous line on the left), but legal answers still perfectly solve the task (see dotted line on the left). The model can perform both tasks it was trained on, however it does not generalize to the *x-only top block* task, even though it has seen both the *x-only* task and also the *top block* data set (albeit not at the same time).

### A.10. Generalization to real images

To test whether models could generalize to the same task but presented in natural images, we first test whether any model (Qwen3-VL-8B, Qwen2.5-VL-7B, Qwen3-VL-32B) post-trained with any method (GRPO, SFT, GSPO) on the x-only top block task can transfer to predicting binary stability for real images of block towers from [Lerer et al. \(2016\)](#). Both tasks require models to infer the offset of the top block. However, we find that no model generalizes to this task.

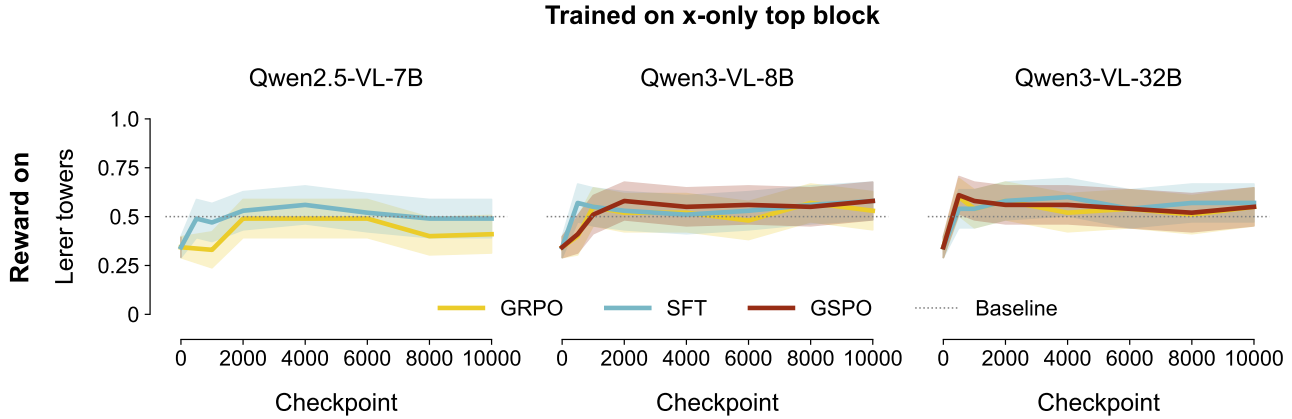


Figure 20. All models trained with GRPO, GSPO, and SFT on the *x-only top block* task, evaluated on the real block towers from [Lerer et al. \(2016\)](#). While we still found some generalization from post-training on our *x-only top block* task to our *binary-stability top block* task, the models do not generalize to this external task. Error bars show 95% confidence intervals.

We also test whether any model Qwen2.5-VL-7B model post-trained with either GRPO or SFT can transfer to predicting binary stability for real images of block towers from [Lerer et al. \(2016\)](#). Here, the *binary-stability top block* model is trained on exactly the same task, only with artificial block towers instead of real images of block towers. We again find that no model generalizes to this task, even the model trained on the similar *binary-stability* task

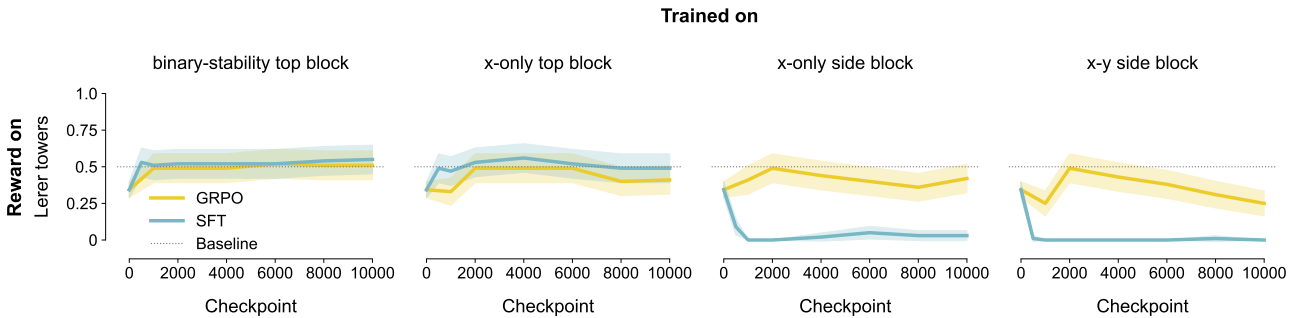


Figure 21. Qwen2.5-VL-7B models trained on all four tasks, evaluated on the real block towers from [Lerer et al. \(2016\)](#). We again find that no model performs well on this task — even the model trained on our similar *binary-stability top block* task. Error bars show 95% confidence intervals.