VISION–LANGUAGE REWARD MODELS CONFUSE NEAR-MISSES WITH SUCCESSES IN ROBOT MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision–language models (VLMs) are increasingly used as reward models for robot manipulation, where they filter and rerank a policy's rollouts. They are typically evaluated on how well they separate successful rollouts from failed ones. Most failures in these evaluations are obvious, such as a dropped object, yet reward models are now often applied to policies that are already competent, where nearly every rollout ends close to success. To measure this setting, we find for each task in simulation how far the object can be moved before the task no longer counts as complete, and we create failures just past that limit (near-misses) and far beyond it (gross failures). We test nine judges on four LIBERO suites. Every judge is worse at telling successes from near-misses than from gross failures. This also holds for failures that a robot policy (OpenVLA) makes on its own, not only for the failures we create. When a judge picks the best of three rollouts, near-misses erase on average 24–62% of its advantage over picking at random, depending on the suite. We also create rollouts in which the object is moved but the task still succeeds. Judges score these lower than untouched successes, which suggests that they track how far the object moved rather than whether the task was done. Across the judges we test, near-miss accuracy varies more from task to task than from judge to judge, so a benchmark's score depends heavily on which tasks it includes. Fine-tuning on our near-misses helps on new tasks from the suites it was trained on but not on an unseen suite. We recommend that reward-model benchmarks report near-miss accuracy separately and show how it varies with task difficulty.

1 INTRODUCTION

Reward models for robot manipulation are increasingly used with policies that are already competent, for example to filter demonstrations or to rerank a pretrained policy's sampled rollouts at test time (Kwok et al., 2025; Dai et al., 2025). In this setting almost every rollout leaves the object close to where it belongs, and the judge must tell a success from a placement a few centimetres outside the goal. Vision–language models are now common in this role, both as trained progress reward models (Lee et al., 2026; Tan et al., 2025; Liang et al., 2026) and as pretrained models that are queried with no further training (Ma et al., 2025; Chen et al., 2026).

Current evaluations do not measure this setting. They report how well a judge separates successful rollouts from failed ones, and most failed rollouts fail in obvious ways, such as a dropped object or a missed grasp, so a judge that detects only these looks accurate. Near-misses are hard to measure for two reasons. We need to be certain that a near-miss really failed, and we need a way to say how near it is that means the same thing for a bowl on a plate as for a can in a basket. Concurrent work defines subtle failures by the time between frames, by the source of the data, or by labels assigned after collection (Sudhakar et al., 2026; Navasardyan et al., 2026), and none of these definitions says how close a failure is to success.

Our key insight is that a simulator already decides exactly when a task is done, using a fixed rule for each task that checks object positions and contacts. For each demonstration, we find how far the object can be moved before the simulator's success check says the task failed, and we call this distance the boundary. We then move the object by fixed multiples of the boundary and let the

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

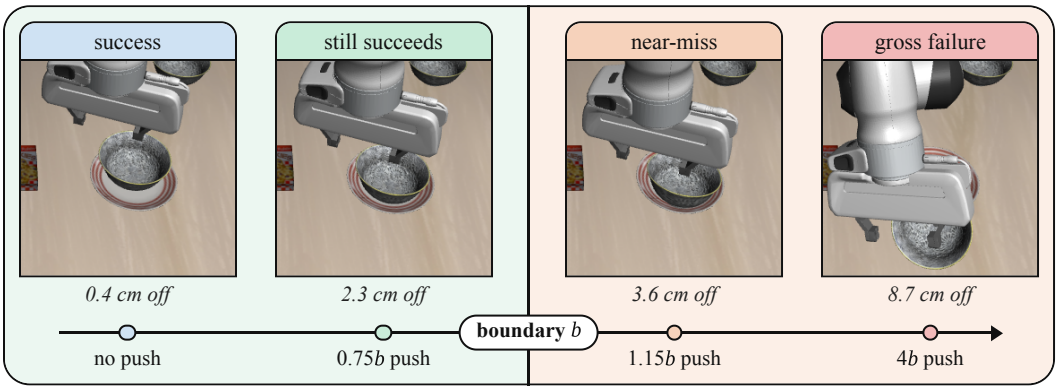


Figure 1: One LIBERO-Spatial instance with the object pushed by multiples of its boundary b, the smallest push at which the simulator's success check fails. Each card shows the final frame and how far the bowl ended up from the centre of the plate.

simulator label every result (Figure 1). A failure at roughly 1.15 times the boundary is just past the limit on any task, however large that limit is in centimetres, so near-misses mean the same thing across tasks.

Moving the object is not the only way a task can nearly succeed, and a benchmark built only from moved objects could reward a judge that simply notices movement. We therefore also create failures in which nothing is moved. Many long-horizon tasks have goals with two parts, such as putting a mug in the microwave and then closing it. We replay the demonstration and stop it after the first part is done, so the scene looks nearly finished but the task has failed. We do this only on two-part tasks, because stopping a one-step task early leaves the object in the robot's gripper, which is not a near-miss.

We find that every judge that can at least spot gross failures is worse at telling successes from near-misses than from gross failures, although the size of the gap varies. This holds on four LIBERO suites (Liu et al., 2023), for both kinds of failure on the long-horizon suite, and for failures that OpenVLA (Kim et al., 2024) makes on its own (§4). When a judge picks the best of three rollouts, near-misses reduce its advantage over picking at random on every suite, by 24–62% on average, and the loss is significant for 24 of the 33 pairs of judge and suite (§5).

To see why judges miss near-misses, we also move the object by less than the boundary, which gives rollouts in which the object was moved but the task still succeeds (§6). A judge that checks whether the task is done should score these like any other success, but judges score them lower, so they respond to how far the object moved, not only to whether the task was done. Near-miss accuracy also varies more between tasks than between judges, with the task explaining several times as much of the variance, and tasks whose near-misses change the image more tend to be easier (§6.1). Fine-tuning on our near-misses helps on new tasks from the suites it was trained on, but not on an unseen suite (§6.2).

We make three contributions.

1. **Boundary bisection**, a way to create failures whose distance past each task's limit is known and whose labels come from the simulator, and *resolution*, the smallest multiple of the boundary at which a judge still reaches an AUROC of 0.75 (§3).
2. **A measurement of how much worse nine judges are on near-misses** than on gross failures, including two published reward models and a frontier API model, and of how much this costs when a judge picks the best of three rollouts (§4, §5).
3. **Evidence that judges respond to how far the object moved**, not only to whether the task was done, and four recommendations for building reward-model benchmarks (§6, §7).

2 Related Work

Learned rewards and success detectors. Learning a reward from examples of success has a long history in robot learning, first as success classifiers trained on goal images (Fu et al., 2018; Singh et al., 2019) and later as vision–language success detectors (Du et al., 2023). Current methods derive the reward from embedding similarity to a goal (Rocamonde et al., 2024; Sontakke et al., 2023; Ma et al., 2023), from preference queries (Wang et al., 2024), from in-context progress estimates (Ma et al., 2025), or from progress models trained on robot rollouts (Lee et al., 2026; Chen et al., 2025; Tan et al., 2025; Liang et al., 2026). RoboReward (Lee et al., 2026) already trains on generated near-misses, TOPReward (Chen et al., 2026) reads a pretrained VLM's probability of answering yes, and Demo2Reward (Gumbsch et al., 2026) and GE-Sim 2.0 (Qiu et al., 2026) improve the reward signal further. These models give useful rewards, but they are evaluated on failures of all sizes mixed together, which cannot show whether a judge separates a near-miss from a success. We evaluate RoboReward and TOPReward as released, and we score our other open VLM judges the way TOPReward does.

Evaluating judges on subtle failures. Concurrent work shares our premise that VLM judges miss small differences and differs in how it defines "small". Sudhakar et al. (2026) find off-the-shelf VLMs near chance at separating visually similar successes and failures and define similarity by time, comparing frames 1–5 s apart with frames 7 s or more apart. FailBench (Navasardyan et al., 2026) collects 2,197 attempts from 14 sources, and its best detector reaches 0.77 balanced accuracy, with errors concentrated where the visual evidence is ambiguous. Other benchmarks test whether a judge can follow a task's progress step by step (Ji et al., 2026; Xia et al., 2026). Tian et al. (2026) argue that embodied reward models over-reward shortcuts, Pacaud et al. (2025) generate failure data at scale, and a recent survey lists telling near-misses from successes as an open problem (Zhang et al., 2026). These works decide in advance what counts as a small difference, for example by time or by visual ambiguity. Boundary bisection instead measures it with each task's own success check, so a near-miss means the same thing on every task, and it also gives rollouts that were moved but still succeed. Table 25 in Appendix G compares the designs.

Reward model evaluation and benchmark composition. In language modelling, reward models also select among samples (Cobbe et al., 2021; Stiennon et al., 2020; Nakano et al., 2021), and the same weakness appears. Aggregate reward-model accuracy saturates and tracks deployment poorly (Lambert et al., 2025; Malik et al., 2026), models above 90% accuracy fall to chance on minimally different pairs (Liu et al., 2025; Wu et al., 2025), accuracy weakly predicts best-of-N and RLHF outcomes (Wen et al., 2025; Frick et al., 2025; Chen et al., 2024), and a policy optimised against the reward finds the errors that aggregate accuracy hides (Gao et al., 2023; Eisenstein et al., 2023; Skalse et al., 2022). Work on benchmark design, including item-response theory, shows that a leaderboard depends on which items a benchmark contains (Lalor et al., 2016; Rodriguez et al., 2021; Dehghani et al., 2021; Madaan et al., 2024; Miller, 2024), and robot success rates move by 30–75 points under scene perturbations (Pumacay et al., 2024; Atreya et al., 2025). These studies document the problem, but in language there is no physical boundary against which to check what a judge responds to. A simulator's success check gives one, and §6.1 uses it to study why scores depend so much on which tasks a benchmark includes.

3 Boundary Bisection

Finding each instance's boundary. We work in LIBERO (Liu et al., 2023). Each demonstration stores the full MuJoCo simulator state, so it replays exactly and any change in the outcome comes from what we change. A task's goal is a logical formula that LIBERO's success check tests on the simulator state. For example, $\mathrm{On}(bowl, plate)$ holds when the bowl touches the plate, sits above it, and has its centre within 3 cm of the plate's centre. To move the object, we push the robot's arm by adding a small offset to its commands over the second half of the episode, so the object is placed slightly off target. Let $s_i(\delta)$ be 1 if demonstration i still succeeds when replayed with a push of size δ, and 0 if it fails. The boundary of demonstration i is the smallest push that makes it fail,

$$b_i = \min\{\, \delta > 0 : s_i(\delta) = 0 \,\}. \tag{1}$$

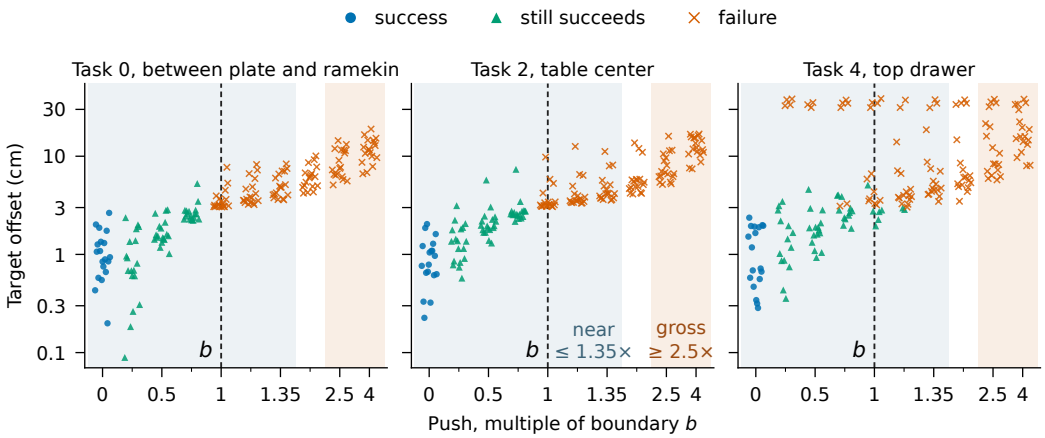


Figure 2: Boundary bisection on LIBERO-Spatial, one panel per task. Each grey line joins one demonstration's rollouts. Push multiples are evenly spaced steps, and target offsets are on a log scale.

We find b_i by binary search over the push size, stopping after six steps and taking the smallest push that failed. We then replay each demonstration with pushes at fixed multiples of its own boundary, $\delta = m\, b_i$ for $m \in M = \{0.25, 0.5, 0.75, 1.0, 1.15, 1.35, 1.7, 2.5, 4.0\}$, which lets us measure how close to the boundary a judge can still tell success from failure. We label each of these rollouts by running the success check again, rather than assuming that pushes below the boundary succeed and pushes above it fail, because a few do not. A rollout that was pushed but still succeeds is a *perturbed success* (P_ok). We group the failures by how far past the boundary they were pushed, into near-misses and gross failures,

$$\mathcal{F}_{\text{near}} = \{\text{failures with } m \le 1.35\}, \qquad \mathcal{F}_{\text{gross}} = \{\text{failures with } m \ge 2.5\}. \tag{2}$$

Figure 2 shows the result on one suite.

We push by multiples of each demonstration's own boundary rather than by fixed distances, because how far an object can miss and still count depends on the scene. A basket is a larger target than a plate, so near-misses are a median 3.1 cm off target on LIBERO-Spatial but 9.2 cm on LIBERO-Object.

Two failure families. We create failures in two ways. *Geometric* failures are the pushes described above, which leave the object in the wrong place. If these were our only failures, the finding in §6 that judges respond to how far the object moved could simply reflect how we built them. The perturbed successes guard against this, since they are moved but still succeed, and so does a second kind of failure that involves no movement at all. Goals on LIBERO-Long have two parts, such as $\mathrm{In}(mug, microwave) \wedge \mathrm{Close}(microwave)$. For these *structural* failures we stop the episode after the object is placed but before the second part is done. We use them only on two-part tasks, because stopping a one-step task early ends mid-motion with the object still in the gripper.

Metrics. A judge gives each trajectory τ a score $f(\tau)$. We measure how well it separates successes from failures with AUROC, the probability that an untouched success scores higher than a failure,

$$A_{\text{band}} = \Pr\left[f(\tau_S) > f(\tau_F)\right], \quad \tau_F \in \mathcal{F}_{\text{band}}, \tag{3}$$

with ties counted as one half. Near AUROC uses failures from the near band and gross AUROC uses failures from the gross band, so 0.5 is chance and 1 is perfect. To see how close to the boundary a judge still works, we also compute $A(m)$ using only the failures at a single multiple m, smoothed with isotonic regression so that it does not decrease as m grows. We define the *resolution* of a judge as the smallest multiple at which $A(m)$ reaches 0.75,

$$r^* = \min\{\, m \in M : A(m) \ge 0.75 \,\}, \tag{4}$$

reported as NR when no $m \le 4$ qualifies. A judge with $r^* \le 1.35$ works inside the near band, and one with $r^* = 4$ detects only gross failures. We also compute $A(m)$ with perturbed successes in place of failures, which we call the *still-successful branch*. A judge that checks whether the task is done should score these like untouched successes and keep this curve at 0.5, whereas a judge that responds to movement pushes it above 0.5 as m grows.

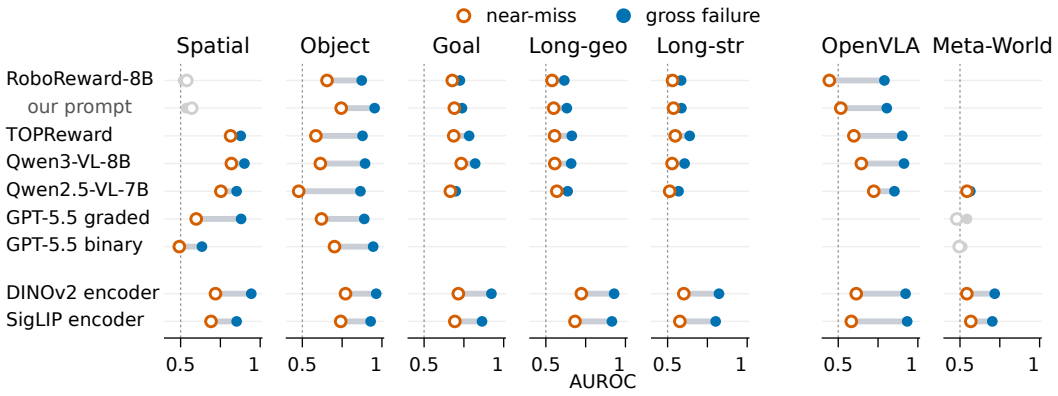


Figure 3: Near-miss and gross-failure AUROC for each judge on each suite. Long-geo and Long-str are LIBERO-Long under geometric and structural failures, OpenVLA is OpenVLA-7B's own failures on LIBERO-Long, and Meta-World repeats the construction in a second simulator. Grey points fail the functionality screen, and the dotted line is chance. Exact values are in Table 24.

Judges and readout. We test nine judges. The first three are published reward models, RoboReward-8B as released and with our prompt (Lee et al., 2026), and TOPReward on its main backbone (Chen et al., 2026). Two are general open-weight VLMs, Qwen3-VL-8B and Qwen2.5-VL-7B. We also test GPT-5.5, a frontier API model from a different model family, with a graded prompt and a binary prompt on LIBERO-Spatial and LIBERO-Object. The last two are pretrained image encoders with no further training, DINOv2 (Oquab et al., 2024) and SigLIP (Zhai et al., 2023). They score a rollout by how close its final frame is to a goal image in embedding space, and the goal image comes from a *different* successful demonstration of the same task, so the encoder cannot simply match the scene. For the open-weight VLMs, we force an answer and read the probabilities the model gives to the allowed answer tokens. If these add up to less than 20% on the first clip, the model is not answering in the expected format and we stop the run (Appendix D). GPT-5.5's score is read from its text answer, since its API returns no token probabilities. The appendix also reports further prompt, frame and resolution variants of these judges and a rubric-prompted Pixtral-12B, and we call each such variant a *configuration*.

Functionality screen. A judge that cannot tell successes from gross failures cannot tell them from near-misses either, so we call a judge *functional* on a suite only if its gross AUROC is at least 0.55, and we leave judges that fail out of the main results (Table 6). Three of the four base model families we tried from outside the Qwen lineage fail this screen.

Data. We use 6,637 rollouts in total, each labelled by the simulator. On LIBERO these are LIBERO-Spatial (541 rollouts from 55 demonstrations), LIBERO-Object (510 from 51), LIBERO-Goal (321 from 51), and LIBERO-Long with geometric and with structural failures (930 each, from 93). The rest are 1,940 further structural failures on LIBERO-Long, 620 rollouts from an earlier, smaller LIBERO-Long run, 360 rollouts in which OpenVLA controls the robot itself, and a Meta-World set (485 from 53) whose successes come from scripted expert policies.

4 Judges Separate Near-Misses Worse Than Gross Failures

Figure 3 shows that on every suite, every functional judge separates successes from near-misses less well than from gross failures. Moving from gross failures to near-misses costs a judge between 0.02 and 0.39 in AUROC. The loss is largest on LIBERO-Object, where the median judge drops by 0.24 and Qwen2.5-VL-7B falls from 0.865 to 0.477, below chance, so an evaluation on gross failures alone can badly overstate how well a judge detects near-misses. The same judge's gap also varies by suite, as with Qwen3-VL-8B at 0.819 on LIBERO-Spatial near-misses but 0.613 on LIBERO-Object, a task effect we return to in §6.1.

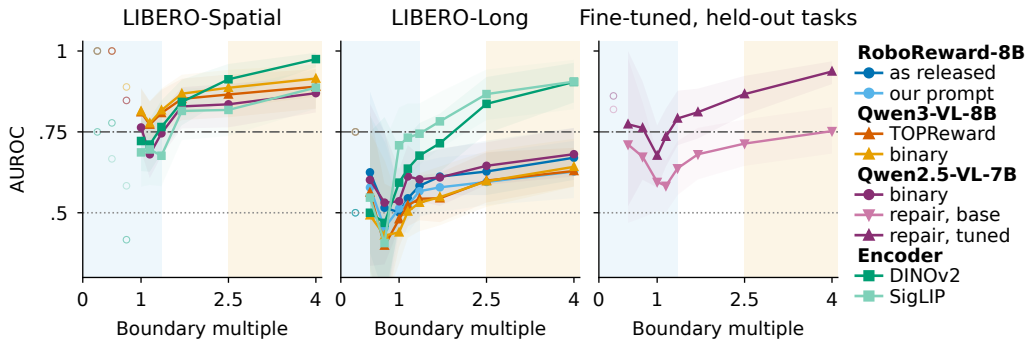


Figure 4: AUROC by failure distance in boundary multiples. Shading marks the near and gross bands and the dash-dot line the 0.75 threshold. LIBERO-Long uses the six-task pilot pools, and the right panel is the judge of §6.2.

How close to the boundary judges work. The two bands group the failures coarsely, so Figure 4 follows each judge's AUROC across the whole range of pushes and Table 21 in Appendix F summarises each curve by its resolution. On the three single-stage LIBERO suites the pretrained encoders reach 0.75 at 1.0 to 1.7 times the boundary, while the VLM judges range from $1.0\times$, reached by Qwen3-VL-8B on LIBERO-Spatial, to $2.5\times$ or never, even though the encoders only measure how far the final frame is from a goal image. The goal image alone does not explain their edge, because giving it to Qwen2.5-VL-7B makes it worse on near-misses (Appendix A).

Long-horizon goals. LIBERO-Long is the one setting where the VLM judges fail outright rather than only on near-misses. Its goals have two parts, and on its ten tasks every VLM judge is near chance on both bands, including RoboReward-8B, a progress model built for multi-stage tasks. The encoders reach up to 0.929 on the same rollouts, and no VLM judge reaches an AUROC of 0.75 at any multiple while both encoders do. More frames do not help, since doubling the frame budget from 8 to 16 changes AUROC by at most $+0.097$ on the six-task pilot pools and leaves every VLM judge near chance (Appendix C). This collapse is partly specific to the failures we build. On the same tasks the VLM judges reach a gross AUROC of 0.79 to 0.91 on OpenVLA-7B's own failures, against at most 0.66 on ours, so they can tell when a long task has gone badly wrong but not when the robot does one part correctly and then stops or is pushed off course.

Failures from a real policy and a second simulator. We check in two ways whether the gap itself comes from how we build failures. First, we ran OpenVLA-7B (Kim et al., 2024) 360 times on LIBERO-Long. It succeeded 226 times, and we sorted 88 of its 134 failures into the near and gross bands by how far along the task it got, measured by aligning each rollout with a demonstration. The other 46 fall between the bands or could not be placed. Every functional judge still separates OpenVLA's near-misses less well than its gross failures (Figure 3). For six of the seven judges scored on these rollouts, including both published reward models and both encoders, near-miss AUROC on OpenVLA's failures is statistically indistinguishable from that on ours. The exception is Qwen2.5-VL-7B, which finds OpenVLA's near-misses easier than ours by 0.21, so its results on our structural failures do not carry over to a real policy. Second, we repeat the whole construction on five Meta-World placement tasks, in a different simulator with a different arm and its own success check, which gives 485 trajectories with 146 near-misses. Both encoders show the same gap there, losing 0.13 to 0.18 in AUROC with 95% intervals that exclude zero. The VLM judges barely work on Meta-World at all, with gross AUROC between 0.51 and 0.57 for Qwen2.5-VL-7B and both GPT-5.5 prompts, so whether a VLM judge works depends on what the scenes look like.

5 The Cost in Best-of-N Selection

An AUROC difference is hard to interpret on its own, so we convert it into the decision a reward model faces when it reranks rollouts. Each pool holds one verified success and two verified failures, and the judge picks the highest-scoring trajectory. The same judge sees two kinds of pool that differ

Table 1: Mean best-of-3 accuracy.

Suite	Gross	Near	Kept
Spatial	0.857	0.730	76%
Object	0.909	0.550	38%
Goal	0.839	0.707	74%
Long-geo	0.770	0.529	45%
Long-str	0.662	0.464	40%

Table 2: Best-of-3 accuracy on LIBERO-Object.

Judge	Gross	Near	Kept
DINOv2 goal-distance	0.961	0.739	65%
SigLIP goal-distance	0.902	0.686	62%
RoboReward-8B (our prompt)	0.961	0.654	51%
TOPReward	0.941	0.461	21%
Qwen3-VL-8B	0.922	0.514	31%
RoboReward-8B (as released)	0.882	0.484	27%
Qwen2.5-VL-7B	0.794	0.315	−4%

only in whether the failures come from the gross or the near band, so random choice succeeds with probability $1/3$ in both. The share of the judge's advantage over random choice that survives on near-misses is

$$\text{kept} = \frac{\text{acc}_{\text{near}} - 1/3}{\text{acc}_{\text{gross}} - 1/3}, \tag{5}$$

where acc is best-of-3 selection accuracy, computed exactly over all distractor subsets.

Table 1 averages over the judges of Figure 3 that pass the functionality screen on each suite, apart from GPT-5.5, which we did not run in selection. Kept is computed from these means, and the LIBERO-Long rows use the earlier six-task pools. Figure 8 in Appendix G shows each judge. On average, near-misses cost judges part of their advantage on every suite, about a quarter on LIBERO-Spatial and LIBERO-Goal and more than half on LIBERO-Object and LIBERO-Long. Table 2 shows that how well a judge selects against gross failures does not carry over to near-misses. Both published reward models lose most of their advantage on LIBERO-Object, and Qwen2.5-VL-7B picks the success about 80% of the time against gross failures but no more often than random choice against near-misses. A control judge with equal skill in both bands shows no significant loss (Appendix G). The encoders also hold up in selection. On ten LIBERO-Long tasks with geometric failures, SigLIP goal-distance picks the success from near-miss pools more often than every VLM judge, significantly more often than Qwen3-VL-8B and Qwen2.5-VL-7B, and no VLM judge selects significantly better than it (Appendix G).

6 Judges Score Displacement, Not Task Completion

A judge can do badly on near-misses for two reasons. It may check the success condition but miss small violations, or it may score something that only tends to go with success, such as how far the object moved. Successes and failures alone cannot tell these apart, because every failure also shows the larger visual change, but the perturbed rollouts that still succeed can. On LIBERO-Spatial, 157 such rollouts leave the object a median of 1.98 cm from its goal and still satisfy the success condition. We call the drop from an unperturbed rollout's score to the score of its perturbed but successful version the penalty. A judge that checks the success condition should show no penalty.

The same control across judges and suites. Across the five VLM judges on the suites where each passes the screen, the penalty is positive for 19 of 23 judge–suite pairs, significant for 9 and never significantly negative. Penalties are largest on LIBERO-Spatial and LIBERO-Goal, reaching $+0.103$ for TOPReward on LIBERO-Spatial, because their goal regions are small enough that a centimetre of displacement is easy to see. On LIBERO-Object, where the displaced object still sits visibly inside its basket, only one judge shows a significant penalty. The resolution curves show the same split. At 0.75 times the boundary, where rollouts still succeed, every functional judge on LIBERO-Spatial tells these rollouts from unperturbed successes with AUROC between 0.60 and 0.69, while the VLM judges on LIBERO-Object stay between 0.45 and 0.59. Judges also prefer some successes to others. When choosing among verified successes, all five VLM judges pick placements 0.21 to 0.31 cm closer to the goal centre than random choice, significantly so for all five, although a judge that checks only the success condition has no reason to prefer one success over another.

Failures and successes equally far from the goal. The penalty shows that judges respond to how far the object ends up from its goal, but a judge could do that and still check the outcome as well. To test this, we compare failures with successes that left the object just as far from the goal. Failures

usually end further away than successes, so comparing all of them would credit a judge that only tracks distance with a drop at the boundary that it does not have. We therefore keep only the range of distances where both occur, for example 3.0 to 7.4 cm on LIBERO-Spatial, and fit each rollout's score drop to its distance from the goal and to whether it failed. The coefficient on failure is the extra score a judge takes away for failing at a given distance. A judge that checks the outcome should take away much more for a failure, while one that reads only distance should take away the same. Across the 18 functional judge–suite pairs where the distance can be measured, which excludes LIBERO-Goal, failing costs significantly more in only one, and in two it costs significantly less. Few successes fall inside the shared range on some suites, 14 against 202 failures on LIBERO-Spatial, so the intervals are wide and a moderate extra penalty for failing cannot be ruled out (Appendix E).

6.1 The Task Dominates Benchmark Scores

If judges respond to how much the image changes, then how hard a task's near-misses are should depend more on the task than on the judge. We scored all 14 single-stage tasks with every judge in Figure 3 except GPT-5.5 and split the variance in near-miss AUROC between tasks and judges. The task accounts for 41% of it and the judge for only 6% (Figure 6). The same object can be easy or hard depending on where it goes. The judges struggle to tell whether the cream cheese reached the basket, with a mean AUROC of 0.456, but not whether it reached the bowl, at 0.893. Task variance stays larger than judge variance in every combination of judges and tasks we tried, although the ratio between them ranges from 1.1 to 19 times (Appendix B). A benchmark's score therefore depends heavily on which tasks it includes.

We also tried to predict which tasks are hard from seven simple measurements of each task, such as how large the object and goal appear in the image and how far a near-miss leaves the object from its goal. None of them predicts a task's mean near-miss AUROC across the 14 tasks after Holm correction. The closest is how much the image changes on a near-miss, the measurement the displacement account points to, and tasks where it changes more are easier ($\rho = 0.446$), but with 14 tasks this is far from significant, so for now difficulty has to be measured on each task.

6.2 Fine-Tuning Helps Only on Seen Suites

Because boundary bisection produces labelled near-misses, it can also test whether judges miss near-misses simply because they never see any in training. We fine-tune Qwen2.5-VL-7B with LoRA on generated failures, following the augmentation recipe of RoboReward (Lee et al., 2026) but with our labels. It trains on some tasks of LIBERO-Spatial, LIBERO-Goal and LIBERO-Long and is tested on the remaining tasks of those suites and on LIBERO-Object, which it never sees. We train one version on binary success labels and one on labels calibrated to each rollout's boundary multiple, and fixed the split, the hyperparameters and the criteria for success before training.

Fine-tuning raises near AUROC on the held-out tasks from 0.563 to 0.736 with binary labels ($+0.173$, 95% CI $[+0.10, +0.25]$) and from 0.589 to 0.745 with calibrated labels ($+0.155$, $[+0.10, +0.20]$; Table 26 and Figure 9). Near-band best-of-3 accuracy rises from 0.512 to 0.726 and from 0.558 to 0.755, roughly doubling the advantage over random choice, and the binary judge's resolution improves from $4.0\times$ to $1.35\times$ the boundary (Figure 4).

The tuned judge still reads displacement. A criterion fixed before training required near AUROC to improve more than gross AUROC, but gross AUROC improves as much with binary labels and more with calibrated ones, and the tuned judge penalises perturbed successes more than the base judge does, even on training tasks where the labels called them successes (Appendix F). On LIBERO-Object, which it never saw, gross AUROC rises by 0.09 to 0.17 while the near-miss gain is $+0.052$ ($[-0.01, +0.12]$) with binary labels and $+0.013$ ($[-0.05, +0.08]$) with calibrated ones, and the untrained DINOv2 goal-distance judge still beats both tuned judges there. Near-miss skill from fine-tuning stays within the suites it was trained on, as the task effect of §6.1 predicts, so a practitioner should bisect and fine-tune on the tasks they deploy on and keep the encoder baseline elsewhere.

7 Recommendations for Benchmark Design

We make four recommendations for reward-model benchmarks.

1. **Report near-miss accuracy separately**, since an aggregate hides the setting where reward models are used. Qwen2.5-VL-7B keeps none of its selection advantage on LIBERO-Object near-misses but still picks correctly 79% of the time against gross failures (§5).
2. **Report accuracy by measured task difficulty**, using one bisection per instance, since a score over a fixed set of tasks depends mostly on which tasks are in it (§6.1).
3. **Screen judges on gross failures** and report their score ranges, since a judge at chance on gross failures is not judging and its near-miss AUROC is noise.
4. **Include a pretrained-encoder baseline**. Goal-image distance under a frozen encoder matches or beats a purpose-trained 8B reward model on the harder suites and is the only kind of judge to reach an AUROC of 0.75 on long-horizon tasks, so without it a benchmark cannot show that the language model helps.

8 Limitations and Future Work

Judge lineage. Most open-weight judges are Qwen-related, because only one of four non-Qwen families, Pixtral-12B, passes the screen, and only on some suites. GPT-5.5 is functional in all four of its LIBERO configurations and shows the gap in each, so the weakness is not specific to Qwen or to open weights, and more families would give a fairer estimate of judge variance.

Mechanism across tasks. Our evidence that judges score displacement comes from comparing rollouts of the same instance. If a judge simply read how many centimetres the object moved, its scores would line up across tasks when plotted against centimetres. An exploratory test finds this for no judge, and each task follows its own curve (Appendix F). This fits the task dominance of §6.1, and suggests that judges respond to how visible a displacement is, which differs from task to task, rather than to its size.

Simulation and scope. Both simulators are built on MuJoCo, so every image the judges see is rendered rather than real. This matters most for judges trained on real robots. RoboReward-8B is near chance on LIBERO-Spatial, which suggests that part of what we measure for such judges is the difference between simulated and real images. Our Meta-World replication rests on the pretrained encoders alone, because the VLM judges barely work there even on gross failures. The task decomposition covers only 14 tasks, and task identity outweighs judge identity in every subset we examined, but by anywhere from $1.1\times$ to $19\times$ depending on which judges and tasks are included. Finally, best-of-N selection draws from a fixed pool of rollouts rather than from a policy that learns from the judge's choices. Finding the success boundaries on a real robot and closing the loop with a policy trained on judge scores are the natural next steps.

9 Conclusion

Reward models for robot manipulation are usually tested on failures that are easy to spot, but a competent policy mostly only fails by small margins. We construct these failures directly for each task by finding how far the object can be moved before the task stops counting as complete. The failures are placed just past that boundary and far beyond it. Every functional judge we test finds the near-misses harder. The reason appears to be that judges measure how far the object is from where it should be rather than whether the task was completed, even marking down rollouts where the object was moved but the task still succeeds. This explains three findings, that near-miss accuracy depends on how visible a task's failures are, partially completed two-part goals fool every VLM judge, and fine-tuning on the misses specifically helps only on task suites the judge has already seen. The effects are seen downstream, since when a judge picks the best of several rollouts, near-misses erase on average 24–62% of its advantage over random choice, depending on the suite. Benchmarks can catch this by reporting near-miss accuracy separately and by measured difficulty, which our released construction makes cheap.

Reproducibility Statement

All experiments are reproducible from the released repository. Judge scores are cached per trajectory and judge, analyses are deterministic given a seed, and every number in this paper is generated

into a machine-readable file rather than transcribed. Each estimator has known-answer tests that plant a known effect, including the null, and require the estimator to recover it. The primary metric, thresholds and exclusions were fixed and committed before the relevant experiments ran, with one amendment recorded before any affected result existed and one gate, also fixed in advance, that failed and is reported as failed.

AI USE STATEMENT

Large language model assistants were used to draft and refactor analysis code, generate tables and figures from committed result files, and edit prose. Every reported number is produced by scripts in the repository from committed data, and the authors verified all claims, citations and numbers.

REFERENCES

Pranav Atreya, Karl Pertsch, et al. Roboarena: Distributed real-world evaluation of generalist robot policies. In *Conference on Robot Learning (CoRL)*, 2025.

Qianzhong Chen, Justin Yu, Mac Schwager, Pieter Abbeel, Yide Shentu, and Philipp Wu. Sarm: Stage-aware reward modeling for long horizon robot manipulation. *arXiv preprint arXiv:2509.25358*, 2025.

Shirui Chen, Cole Harrison, Ying-Chun Lee, Angela Jin Yang, Zhongzheng Ren, Lillian J. Ratliff, Jiafei Duan, Dieter Fox, and Ranjay Krishna. Topreward: Token probabilities as hidden zero-shot rewards for robotics. *arXiv preprint arXiv:2602.19313*, 2026.

Yanjun Chen, Dawei Zhu, Yirong Sun, Xinghao Chen, Wei Zhang, and Xiaoyu Shen. The accuracy paradox in RLHF: When better reward models don't yield better language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Mingtong Dai, Lingbo Liu, Yongjie Bai, Yang Liu, Zhouxia Wang, Rui Su, Chunjie Chen, Liang Lin, and Xinyu Wu. Rover: Robot reward model as test-time verifier for vision-language-action model. *arXiv preprint arXiv:2510.10975*, 2025.

Mostafa Dehghani, Yi Tay, Alexey A. Gritsenko, Zhe Zhao, Neil Houlsby, Fernando Diaz, Donald Metzler, and Oriol Vinyals. The benchmark lottery. *arXiv preprint arXiv:2107.07002*, 2021.

Yuqing Du, Ksenia Konyushkova, Misha Denil, Akhil Raju, Jessica Landon, Felix Hill, Nando de Freitas, and Serkan Cabi. Vision-language models as success detectors. In *Proceedings of The 2nd Conference on Lifelong Learning Agents (CoLLAs)*, volume 232 of *PMLR*, pp. 120–136, 2023.

Jacob Eisenstein, Chirag Nagpal, Alekh Agarwal, Ahmad Beirami, Alexander D'Amour, DJ Dvijotham, Adam Fisch, Katherine Heller, Stephen Pfohl, Deepak Ramachandran, Peter Shaw, and Jonathan Berant. Helping or herding? reward model ensembles mitigate but do not eliminate reward hacking. *arXiv preprint arXiv:2312.09244*, 2023.

Evan Frick, Tianle Li, Connor Chen, Wei-Lin Chiang, Anastasios N. Angelopoulos, Jiantao Jiao, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. How to evaluate reward models for RLHF. In *International Conference on Learning Representations (ICLR)*, 2025.

Justin Fu, Avi Singh, Dibya Ghosh, Larry Yang, and Sergey Levine. Variational inverse control with events: A general framework for data-driven reward definition. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning (ICML)*, 2023.

Christian Gumbsch, Leonardo Barcellona, Lennard Schünemann, Platon Karageorgis, Andrii Zadaianchuk, Zehao Wang, Sergey Zakharov, Fabien Despinoy, Rahaf Aljundi, and Efstratios Gavves. From demonstrations to rewards: Test-time prompt optimization for VLM reward models. *arXiv preprint arXiv:2606.00083*, 2026.

Yuheng Ji, Yuyang Liu, Huajie Tan, Xuchuan Huang, Fanding Huang, Yijie Xu, Cheng Chi, Yuting Zhao, Huaihai Lyu, Peterson Co, Mingyu Cao, Qiongyu Zhang, Zhe Li, Enshen Zhou, Pengwei Wang, Zhongyuan Wang, Shanghang Zhang, and Xiaolong Zheng. Prm-as-a-judge: A dense evaluation paradigm for fine-grained robotic auditing. *arXiv preprint arXiv:2603.21669*, 2026.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. In *Conference on Robot Learning (CoRL)*, 2024.

Jacky Kwok, Christopher Agia, Rohan Sinha, Matt Foutter, Shulu Li, Ion Stoica, Azalia Mirhoseini, and Marco Pavone. Robomonkey: Scaling test-time sampling and verification for vision-language-action models. *arXiv preprint arXiv:2506.17811*, 2025.

John P. Lalor, Hao Wu, and Hong Yu. Building an evaluation scale using item response theory. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. Rewardbench: Evaluating reward models for language modeling. In *Findings of the Association for Computational Linguistics: NAACL*, 2025.

Tony Lee, Andrew Wagenmaker, Karl Pertsch, Percy Liang, Sergey Levine, and Chelsea Finn. Roboreward: General-purpose vision-language reward models for robotics. *arXiv preprint arXiv:2601.00675*, 2026.

Anthony Liang, Yigit Korkmaz, Jiahui Zhang, Minyoung Hwang, Abrar Anwar, Sidhant Kaushik, Aditya Shah, Alex S. Huang, Luke Zettlemoyer, Dieter Fox, Yu Xiang, Anqi Li, Andreea Bobu, Abhishek Gupta, Stephen Tu, Erdem Biyik, and Jesse Zhang. Robometer: Scaling general-purpose robotic reward models via trajectory comparisons. In *Robotics: Science and Systems (RSS)*, 2026.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2023.

Yantao Liu, Zijun Yao, Rui Min, Yixin Cao, Lei Hou, and Juanzi Li. RM-bench: Benchmarking reward models of language models with subtlety and style. In *International Conference on Learning Representations (ICLR)*, 2025.

Yecheng Jason Ma, William Liang, Vaidehi Som, Vikash Kumar, Amy Zhang, Osbert Bastani, and Dinesh Jayaraman. Liv: Language-image representations and rewards for robotic control. In *International Conference on Machine Learning (ICML)*, 2023.

Yecheng Jason Ma, Joey Hejna, Ayzaan Wahid, Chuyuan Fu, Dhruv Shah, Jacky Liang, Zhuo Xu, Sean Kirmani, Peng Xu, Danny Driess, Ted Xiao, Jonathan Tompson, Osbert Bastani, Dinesh Jayaraman, Wenhao Yu, Tingnan Zhang, Dorsa Sadigh, and Fei Xia. Vision language models are in-context value learners. In *International Conference on Learning Representations (ICLR)*, 2025.

Lovish Madaan, Aaditya K. Singh, Rylan Schaeffer, Andrew Poulton, Sanmi Koyejo, Pontus Stenetorp, Sharan Narang, and Dieuwke Hupkes. Quantifying variance in evaluation benchmarks. *arXiv preprint arXiv:2406.10229*, 2024.

Saumya Malik, Valentina Pyatkin, Sander Land, Jacob Morrison, Noah A. Smith, Hannaneh Hajishirzi, and Nathan Lambert. Rewardbench 2: Advancing reward model evaluation. In *International Conference on Learning Representations (ICLR)*, 2026.

Evan Miller. Adding error bars to evals: A statistical approach to language model evaluations. *arXiv preprint arXiv:2411.00640*, 2024.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. WebGPT: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

Zaruhi Navasardyan, Tatul Danielyan, and Hrant Davtyan. Failbench: How reliable are VLMs at judging robot task success? *arXiv preprint arXiv:2609.03611*, 2026.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research (TMLR)*, 2024.

Paul Pacaud, Ricardo Garcia, Shizhe Chen, and Cordelia Schmid. Scaling cross-environment failure reasoning data for vision-language robotic manipulation. *arXiv preprint arXiv:2512.01946*, 2025.

Wilbert Pumacay, Ishika Singh, Jiafei Duan, Ranjay Krishna, Jesse Thomason, and Dieter Fox. THE COLOSSEUM: A benchmark for evaluating generalization for robotic manipulation. In *Robotics: Science and Systems (RSS)*, 2024.

Boxiang Qiu, Liliang Chen, Yue Liao, Nan Wang, Lintao Wang, Jiayi Luo, Wenzhi Zhao, Shengcong Chen, Di Chen, Ye Li, Chen Gao, Shuicheng Yan, Si Liu, Maoqing Yao, and Guanghui Ren. GE-Sim 2.0: A roadmap towards comprehensive closed-loop video world simulators for robotic manipulation. *arXiv preprint arXiv:2605.27491*, 2026.

Juan Rocamonde, Victoriano Montesinos, Elvis Nava, Ethan Perez, and David Lindner. Vision-language models are zero-shot reward models for reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2024.

Pedro Rodriguez, Joe Barrow, Alexander Miserlis Hoyle, John P. Lalor, Robin Jia, and Jordan Boyd-Graber. Evaluation examples are not equally informative: How should that change NLP leaderboards? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021.

Avi Singh, Larry Yang, Kristian Hartikainen, Chelsea Finn, and Sergey Levine. End-to-end robotic reinforcement learning without reward engineering. In *Robotics: Science and Systems (RSS)*, 2019.

Joar Skalse, Nikolaus H. R. Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward gaming. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Sumedh A Sontakke, Jesse Zhang, Sébastien M. R. Arnold, Karl Pertsch, Erdem Bıyık, Dorsa Sadigh, Chelsea Finn, and Laurent Itti. Roboclip: One demonstration is enough to learn robot policies. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Sruthi Sudhakar, Junbang Liang, Sreehari Rammohan, Pavel Tokmakov, Richard Zemel, and Carl Vondrick. Robot critics that sweat the small stuff. *arXiv preprint arXiv:2606.21572*, 2026.

Huajie Tan, Sixiang Chen, Yijie Xu, Zixiao Wang, Yuheng Ji, Cheng Chi, Yaoxu Lyu, Zhongxia Zhao, Xiansheng Chen, Peterson Co, Shaoxuan Xie, Guocai Yao, Pengwei Wang, Zhongyuan Wang, and Shanghang Zhang. Robo-dopamine: General process reward modeling for high-precision robotic manipulation. *arXiv preprint arXiv:2512.23703*, 2025.

Ran Tian, Yilin Wu, and Andrea Bajcsy. Position: Good embodied reward models need bad behavior data. *arXiv preprint arXiv:2606.01036*, 2026.

Yufei Wang, Zhanyi Sun, Jesse Zhang, Zhou Xian, Erdem Biyik, David Held, and Zackory Erickson. Rl-vlm-f: Reinforcement learning from vision language foundation model feedback. In *International Conference on Machine Learning (ICML)*, 2024.

Xueru Wen, Jie Lou, Yaojie Lu, Hongyu Lin, Xing Yu, Xinyu Lu, Ben He, Xianpei Han, Debing Zhang, and Le Sun. Rethinking reward model evaluation: Are we barking up the wrong tree? In *International Conference on Learning Representations (ICLR)*, 2025.

Zhaofeng Wu, Michihiro Yasunaga, Andrew Cohen, Yoon Kim, Asli Celikyilmaz, and Marjan Ghazvininejad. reWordBench: Benchmarking and improving the robustness of reward models with transformed inputs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2025.

Dayu Xia, Yue Shi, Yao Mu, Huiting Ji, Chaofan Ma, Yingjie Zhou, Hua Chen, Yang Liu, Jiezhang Cao, and Guangtao Zhai. Roboprocessbench: Benchmarking process-aware understanding in vision-language robotic manipulation. *arXiv preprint arXiv:2606.13040*, 2026.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Jianshu Zhang, Keliang Wu, Haoran Lu, Anbang Liu, Ce Zhang, Weijie Yin, Chengxuan Qian, Xiyuan Yang, Zhenyu Pan, Guo Ye, and Han Liu. Progress reward modeling for robotic learning: A comprehensive survey. *arXiv preprint arXiv:2607.21655*, 2026.

A THE CONSTRUCTED JUDGE LADDER

This appendix collects the 33 judge configurations we built on LIBERO-Spatial, which we call the ladder. We built it to test whether a judge's near-miss accuracy predicts how well it selects rollouts beyond what its gross-failure accuracy already predicts, a test that needs judges whose near and gross AUROC vary independently. Table 3 lists every configuration, including deliberately degraded ones and ones at chance. The published reward models and the later judges from other model families are separate experiments and are not part of the ladder.

Construction and estimator. The baseline configuration samples eight frames uniformly, uses a graded prompt and keeps the native resolution. The other configurations change the model size, the prompt, the number or order of frames, the resolution, whether a goal image is given, or a crop around the goal region, and one sees only the final frame. The encoders compare the final frame with a goal image taken from another successful instance of the same task, and the goal-conditioned VLMs receive the same image. The goal image never comes from the instance being scored, which fixes an earlier leak described in Appendix D. As in the main text, failures at most $1.35\times$ the boundary form the near band and those at least $2.5\times$ form the gross band, and each failure is compared with the unperturbed success of its own instance. A success can therefore enter the AUROC several times, so n counts distinct instances rather than independent rollouts, and the estimates are descriptive. The table also gives each configuration's score range and standard deviation.

Table 3: Constructed ladder on LIBERO-Spatial. AUROC pools each failure with its own baseline success. Degraded configurations and those that fail the screen are kept to document the construction. All configurations share $n = 55$ instances and 167 near and 108 gross failures.

Configuration	Near	Gross	Score range	Score sd
Qwen2.5-VL-7B; graded; 8 frames	0.707	0.790	[0.405, 0.921]	0.1020
Qwen2.5-VL-7B; binary; 8 frames	0.753	0.852	[0.016, 0.407]	0.0671
Qwen2.5-VL-7B; graded; 2 frames	0.695	0.848	[0.202, 0.764]	0.0902
Qwen2.5-VL-7B; graded; 8 frames; shuffled	0.612	0.643	[0.404, 0.876]	0.0855
Qwen2.5-VL-7B; graded; 8 frames; resolution 56	0.630	0.536	[0.419, 0.706]	0.0515
Qwen2.5-VL-3B; graded; 8 frames	0.497	0.483	[0.316, 0.752]	0.1025
Qwen2.5-VL-3B; binary; 8 frames	0.506	0.495	[0.182, 0.378]	0.0345
Qwen2.5-VL-3B; graded; 2 frames	0.458	0.425	[0.310, 0.645]	0.0635
Qwen2.5-VL-3B; graded; 8 frames; shuffled	0.493	0.463	[0.291, 0.715]	0.0885
Qwen2.5-VL-3B; graded; 8 frames; resolution 56	0.529	0.514	[0.346, 0.529]	0.0294
Qwen2.5-VL-7B; rubric; 8 frames	0.749	0.872	[0.576, 0.886]	0.0531
Qwen2.5-VL-7B; graded; 1 frame	0.563	0.587	[0.101, 0.657]	0.0828
Qwen2.5-VL-7B; graded; 4 frames	0.707	0.799	[0.369, 0.834]	0.0862
Qwen2.5-VL-7B; graded; 16 frames	0.645	0.703	[0.417, 0.967]	0.1316
Qwen2.5-VL-7B; graded; 8 frames; resolution 112	0.677	0.708	[0.445, 0.869]	0.0918
Frozen DINOv2; goal image	0.666	0.901	[-0.697, -0.229]	0.0854
Frozen SigLIP; goal image	0.691	0.852	[-0.484, -0.170]	0.0556
Frozen DINOv2; goal image; crop 0.7	0.696	0.896	[-0.775, -0.244]	0.0887
Frozen DINOv2; goal image; crop 0.5	0.719	0.944	[-0.963, -0.272]	0.1473
Frozen DINOv2; goal image; crop 0.35	0.687	0.884	[-1.129, -0.291]	0.1595
Frozen DINOv2; goal image; crop 0.25	0.588	0.846	[-1.233, -0.326]	0.1964
Frozen DINOv2; goal image; crop 0.15	0.562	0.751	[-1.301, -0.000]	0.2195
Frozen SigLIP; goal image; crop 0.7	0.547	0.757	[-0.508, -0.173]	0.0609
Frozen SigLIP; goal image; crop 0.5	0.584	0.855	[-0.661, -0.215]	0.0819
Frozen SigLIP; goal image; crop 0.35	0.641	0.840	[-0.669, -0.220]	0.0928
Frozen SigLIP; goal image; crop 0.25	0.573	0.799	[-0.715, -0.200]	0.1042
Frozen SigLIP; goal image; crop 0.15	0.508	0.666	[-0.797, -0.121]	0.1299
Qwen2.5-VL-7B; goal image, binary; 8 frames	0.620	0.679	[0.011, 0.068]	0.0086
Qwen2.5-VL-7B; goal image, graded; 8 frames	0.636	0.756	[0.490, 0.751]	0.0416
Qwen2.5-VL-7B; goal image, binary; 8 frames; crop 0.35	0.532	0.484	[0.009, 0.068]	0.0101
Qwen2.5-VL-7B; binary; 8 frames; crop 0.35	0.553	0.495	[0.010, 0.119]	0.0204
Qwen2.5-VL-7B; binary; 8 frames; crop 0.15	0.541	0.602	[0.037, 0.133]	0.0195
Qwen2.5-VL-3B; goal image, binary; 8 frames	0.497	0.495	[0.165, 0.294]	0.0252

The goal-region crop. A crop is a square window around the goal whose side is a fraction of the frame's shorter side. We find the goal once per instance by comparing the successful rollout with the other rollouts of the same instance. The changes they all share mark where the object was picked up, and removing those from the success's changes leaves the goal region, or, if too little is left, the window falls back to all of the success's changes. The window is the same for every rollout of an instance and is applied to both the rollout and the goal image, so it tells the judge where the goal is, and we report it as such. We expected a tight crop to make a judge better at near-misses than at gross failures, since a synthetic test showed that a pixel distance saturates once the object leaves a tight window. The encoders did not behave this way. A moderate DINOv2 crop improves both bands and tighter crops lower both (Table 4). The synthetic test checks the estimator, not how an encoder represents an enlarged crop from a new visual domain, so this is a failed prediction about the encoders rather than a failed test.

Table 4: Goal-region crop axis. Fraction is window side divided by the frame's shorter side; full means no crop. Goal image and goal location are privileged inputs.

Judge	Goal image	Fraction	Near	Gross
DINOv2	yes	full	0.666	0.901
SigLIP	yes	full	0.691	0.852
DINOv2	yes	0.7	0.696	0.896
DINOv2	yes	0.5	0.719	0.944
DINOv2	yes	0.35	0.687	0.884
DINOv2	yes	0.25	0.588	0.846
DINOv2	yes	0.15	0.562	0.751
SigLIP	yes	0.7	0.547	0.757
SigLIP	yes	0.5	0.584	0.855
SigLIP	yes	0.35	0.641	0.840
SigLIP	yes	0.25	0.573	0.799
SigLIP	yes	0.15	0.508	0.666
Qwen2.5-VL-7B, binary	yes	0.35	0.532	0.484
Qwen2.5-VL-7B, binary	no	0.35	0.553	0.495
Qwen2.5-VL-7B, binary	no	0.15	0.541	0.602

What the ladder allows. Across all 33 configurations, near and gross AUROC correlate at $\rho = 0.819$, above the cutoff of 0.8 fixed in advance, so the ladder cannot support the planned test (Table 5). The 15 configurations with neither a goal image nor a crop correlate even more strongly, at $\rho = 0.975$, and an earlier subset chosen by configuration name, which also kept the VLM crops without a goal image, gives $\rho = 0.953$ over 17. None of the configurations we built is better at near-misses than its gross accuracy predicts, although such a judge might exist outside this construction.

Table 5: Correlation between near and gross AUROC across ladder configurations.

Pool	Configurations	ρ(near, gross)
All configurations	33	0.819
Earlier subset chosen by name (includes crops)	17	0.953
Neither crop nor reference image	15	0.975

B Per-Task Discrimination Across Suites

The tables below break the results of §4 down by task for LIBERO-Spatial, LIBERO-Object, LIBERO-Goal and the geometric and structural LIBERO-Long failures, the last two on the full ten-task replication.

Coverage and functionality. The judges are those of the main text, the rubric-prompted configurations, SmolVLM2, and GPT-5.5 on the suites where it was run. Table 6 applies the main text's screen within each suite, which requires a gross AUROC of at least 0.55 pooled over the suite's tasks. A

judge that fails the screen is limited by transfer or by how its score is read out, which says nothing about its near-miss ability, and because the screen is applied per suite rather than per task, no task is dropped for its AUROC. AUROC depends only on the order of the scores, so a judge whose scores barely vary can still reach an ordinary-looking AUROC, and the tables therefore give each judge's score range next to its AUROC.

Estimators. Per-task near AUROC uses the estimator of §6.1 and needs at least 8 near-miss failures, and gross AUROC pairs each failure with the success of its own instance, as in Appendix A. A dash marks a task with too few near-misses or no gross failures, and every task in the pool is kept. The per-task values are descriptive, with no claim of significance for any single task, and the score ranges cover all scored rollouts of the task. GPT-5.5 scores are parsed from its text answers, and rollouts whose answer could not be parsed are dropped. Its AUROCs compare all distinct successes with the failures in each band, as in its original analysis, rather than pairing each failure with its own success, so they share the band definitions but weight successes differently.

Sensitivity of the task decomposition (§6.1). Over the 14 single-stage tasks and the seven judges of §6.1, task variance is 7.1 times judge variance, and adding the long-horizon suites raises this to 14 to 19 times. The grid needs a score from every judge on every task, so it keeps RoboReward-8B on LIBERO-Spatial, where both of its prompts fail the screen. Leaving it out gives 48% task and 7% judge variance. Adding the rubric-prompted Qwen2.5-VL-7B and Pixtral-12B spreads the judges further apart and lowers the ratio to 2.0 times, or 1.1 times when a judge that fails the screen is also included. Most of the VLM judges descend from Qwen, which may understate judge variance. GPT-5.5, from a different family, is scored only on LIBERO-Spatial and LIBERO-Object, so it cannot enter the grid, but its near-miss AUROC falls below every functional judge on LIBERO-Spatial and within their range on LIBERO-Object.

Table 6: Functionality screen for each suite. A judge passes with a gross AUROC of at least 0.55.

Suite	Judge	Gross	Pass	Score range
LIBERO-Spatial	RoboReward-8B (as released)	0.515	no	[0.133, 0.991]
LIBERO-Spatial	RoboReward-8B (our prompt)	0.534	no	[0.128, 0.988]
LIBERO-Spatial	TOPReward	0.878	yes	[0.004, 0.963]
LIBERO-Spatial	Qwen3-VL-8B	0.901	yes	[0.000, 0.777]
LIBERO-Spatial	Qwen2.5-VL-7B	0.852	yes	[0.016, 0.407]
LIBERO-Spatial	Qwen2.5-VL-7B (rubric)	0.872	yes	[0.576, 0.886]
LIBERO-Spatial	DINOv2 goal-distance	0.944	yes	[-0.963, -0.272]
LIBERO-Spatial	SigLIP goal-distance	0.852	yes	[-0.484, -0.170]
LIBERO-Spatial	Pixtral-12B (rubric)	0.571	yes	[0.785, 0.972]
LIBERO-Spatial	SmolVLM2 (graded)	0.479	no	[0.593, 0.745]
LIBERO-Spatial	GPT-5.5 (graded)	0.880	yes	[0.000, 1.000]
LIBERO-Spatial	GPT-5.5 (binary)	0.634	yes	[0.000, 1.000]
LIBERO-Object	RoboReward-8B (as released)	0.873	yes	[0.359, 1.000]
LIBERO-Object	RoboReward-8B (our prompt)	0.954	yes	[0.497, 1.000]
LIBERO-Object	TOPReward	0.878	yes	[0.016, 0.993]
LIBERO-Object	Qwen3-VL-8B	0.894	yes	[0.003, 0.977]
LIBERO-Object	Qwen2.5-VL-7B	0.865	yes	[0.023, 0.593]
LIBERO-Object	Qwen2.5-VL-7B (rubric)	0.787	yes	[0.576, 0.927]
LIBERO-Object	DINOv2 goal-distance	0.964	yes	[-0.928, -0.242]
LIBERO-Object	SigLIP goal-distance	0.929	yes	[-0.512, -0.156]
LIBERO-Object	Pixtral-12B (rubric)	0.738	yes	[0.857, 0.977]
LIBERO-Object	SmolVLM2 (graded)	0.402	no	[0.614, 0.818]
LIBERO-Object	GPT-5.5 (graded)	0.889	yes	[0.000, 1.000]
LIBERO-Object	GPT-5.5 (binary)	0.945	yes	[0.000, 1.000]
LIBERO-Goal	RoboReward-8B (as released)	0.725	yes	[0.158, 1.000]
LIBERO-Goal	RoboReward-8B (our prompt)	0.738	yes	[0.204, 0.999]
LIBERO-Goal	TOPReward	0.783	yes	[0.004, 0.974]
LIBERO-Goal	Qwen3-VL-8B	0.821	yes	[0.001, 0.867]
LIBERO-Goal	Qwen2.5-VL-7B	0.699	yes	[0.013, 0.438]
LIBERO-Goal	Qwen2.5-VL-7B (rubric)	0.734	yes	[0.621, 0.859]

Continued on next page

Table 6 (continued)

Suite	Judge	Gross	Pass	Score range
LIBERO-Goal	DINOv2 goal-distance	0.923	yes	[-1.046, -0.253]
LIBERO-Goal	SigLIP goal-distance	0.864	yes	[-0.512, -0.169]
LIBERO-Goal	Pixtral-12B (rubric)	0.541	no	[0.810, 0.963]
LIBERO-Goal	SmolVLM2 (graded)	0.550	no	[0.511, 0.736]
LIBERO-Long, geometric	RoboReward-8B (as released)	0.616	yes	[0.249, 0.985]
LIBERO-Long, geometric	RoboReward-8B (our prompt)	0.632	yes	[0.250, 0.999]
LIBERO-Long, geometric	TOPReward	0.663	yes	[0.009, 0.992]
LIBERO-Long, geometric	Qwen3-VL-8B	0.659	yes	[0.001, 0.967]
LIBERO-Long, geometric	Qwen2.5-VL-7B	0.637	yes	[0.009, 0.469]
LIBERO-Long, geometric	Qwen2.5-VL-7B (rubric)	0.592	yes	[0.508, 0.898]
LIBERO-Long, geometric	DINOv2 goal-distance	0.929	yes	[-1.100, -0.123]
LIBERO-Long, geometric	SigLIP goal-distance	0.915	yes	[-0.581, -0.151]
LIBERO-Long, geometric	Pixtral-12B (rubric)	0.605	yes	[0.754, 0.947]
LIBERO-Long, structural	RoboReward-8B (as released)	0.585	yes	[0.245, 0.996]
LIBERO-Long, structural	RoboReward-8B (our prompt)	0.587	yes	[0.258, 0.999]
LIBERO-Long, structural	TOPReward	0.639	yes	[0.007, 0.992]
LIBERO-Long, structural	Qwen3-VL-8B	0.607	yes	[0.001, 0.963]
LIBERO-Long, structural	Qwen2.5-VL-7B	0.569	yes	[0.016, 0.562]
LIBERO-Long, structural	Qwen2.5-VL-7B (rubric)	0.541	no	[0.524, 0.909]
LIBERO-Long, structural	DINOv2 goal-distance	0.823	yes	[-1.046, -0.104]
LIBERO-Long, structural	SigLIP goal-distance	0.802	yes	[-0.544, -0.149]
LIBERO-Long, structural	Pixtral-12B (rubric)	0.405	no	[0.718, 0.953]

Table 7: LIBERO-Spatial, per-task AUROC for the judges that pass the screen on this suite. n is baseline instances and N/G near and gross failures. A dash marks too few near-misses or no gross failures.

Judge	Near	Gross	n	N/G	Score range
Task 0: pick up the black bowl between the plate and the ramekin and place it on the plate					
TOPReward	0.931	0.996	19	54/36	[0.026, 0.777]
Qwen3-VL-8B	0.948	0.998	19	54/36	[0.002, 0.321]
Qwen2.5-VL-7B	0.816	0.931	19	54/36	[0.016, 0.223]
Qwen2.5-VL-7B (rubric)	0.763	0.942	19	54/36	[0.576, 0.825]
DINOv2 goal-distance	0.793	0.985	19	54/36	[-0.942, -0.272]
SigLIP goal-distance	0.867	0.931	19	54/36	[-0.484, -0.193]
Pixtral-12B (rubric)	0.390	0.646	19	54/36	[0.785, 0.908]
GPT-5.5 (graded)	0.557	0.906	19	54/35	[0.000, 1.000]
GPT-5.5 (binary)	0.543	0.694	18	53/36	[0.000, 1.000]

Continued on next page

Table 7 (continued)

Judge	Near	Gross	n	N/G	Score range
Task 2: pick up the black bowl from table center and place it on the plate					
TOPReward	0.820	0.814	18	54/36	[0.014, 0.881]
Qwen3-VL-8B	0.818	0.856	18	54/36	[0.001, 0.500]
Qwen2.5-VL-7B	0.741	0.860	18	54/36	[0.026, 0.321]
Qwen2.5-VL-7B (rubric)	0.755	0.847	18	54/36	[0.619, 0.848]
DINOv2 goal-distance	0.736	0.951	18	54/36	[-0.963, -0.302]
SigLIP goal-distance	0.608	0.826	18	54/36	[-0.399, -0.183]
Pixtral-12B (rubric)	0.547	0.691	18	54/36	[0.906, 0.972]
GPT-5.5 (graded)	0.556	0.889	18	54/36	[0.500, 1.000]
GPT-5.5 (binary)	0.444	0.625	18	54/36	[0.000, 1.000]
Task 4: pick up the black bowl in the top drawer of the wooden cabinet and place it on the plate					
TOPReward	0.823	0.878	18	59/36	[0.004, 0.963]
Qwen3-VL-8B	0.821	0.890	18	59/36	[0.000, 0.777]
Qwen2.5-VL-7B	0.768	0.869	18	59/36	[0.017, 0.407]
Qwen2.5-VL-7B (rubric)	0.768	0.864	18	59/36	[0.622, 0.886]
DINOv2 goal-distance	0.647	0.889	18	59/36	[-0.926, -0.321]
SigLIP goal-distance	0.647	0.843	18	59/36	[-0.417, -0.170]
Pixtral-12B (rubric)	0.609	0.728	18	59/36	[0.888, 0.958]
GPT-5.5 (graded)	0.683	0.853	18	59/36	[0.000, 1.000]
GPT-5.5 (binary)	0.484	0.583	18	59/36	[0.000, 1.000]

Table 8: LIBERO-Object, per-task AUROC for the judges that pass the screen on this suite. n is baseline instances and N/G near and gross failures. A dash marks too few near-misses or no gross failures.

Judge	Near	Gross	n	N/G	Score range
Task 0: pick up the alphabet soup and place it in the basket					
RoboReward-8B (as released)	0.386	0.843	11	33/22	[0.470, 0.995]
RoboReward-8B (our prompt)	0.683	0.975	11	33/22	[0.570, 0.999]
TOPReward	0.580	0.938	11	33/22	[0.020, 0.967]
Qwen3-VL-8B	0.643	0.934	11	33/22	[0.003, 0.924]
Qwen2.5-VL-7B	0.592	0.878	11	33/22	[0.023, 0.531]
Qwen2.5-VL-7B (rubric)	0.599	0.899	11	33/22	[0.640, 0.919]
DINOv2 goal-distance	0.829	0.975	11	33/22	[-0.875, -0.282]
SigLIP goal-distance	0.945	1.000	11	33/22	[-0.497, -0.191]
Pixtral-12B (rubric)	0.751	0.831	11	33/22	[0.919, 0.976]
GPT-5.5 (graded)	0.649	0.809	11	32/19	[0.000, 1.000]
GPT-5.5 (binary)	0.824	0.900	10	33/22	[0.000, 1.000]
Task 1: pick up the cream cheese and place it in the basket					
RoboReward-8B (as released)	0.417	0.702	11	34/22	[0.597, 0.999]
RoboReward-8B (our prompt)	0.527	0.897	11	34/22	[0.497, 1.000]
TOPReward	0.382	0.913	11	34/22	[0.068, 0.967]
Qwen3-VL-8B	0.289	0.905	11	34/22	[0.011, 0.915]
Qwen2.5-VL-7B	0.111	0.740	11	34/22	[0.076, 0.407]
Qwen2.5-VL-7B (rubric)	0.278	0.775	11	34/22	[0.585, 0.828]
DINOv2 goal-distance	0.770	1.000	11	34/22	[-0.928, -0.308]
SigLIP goal-distance	0.694	0.963	11	34/22	[-0.452, -0.162]
Pixtral-12B (rubric)	0.494	0.793	11	34/22	[0.880, 0.967]
GPT-5.5 (graded)	0.500	0.977	11	34/22	[0.500, 1.000]
GPT-5.5 (binary)	0.500	1.000	11	34/22	[0.000, 1.000]

Continued on next page

Table 8 (continued)

Judge	Near	Gross	n	N/G	Score range
Task 2: pick up the salad dressing and place it in the basket					
RoboReward-8B (as released)	0.730	0.961	8	23/16	[0.359, 1.000]
RoboReward-8B (our prompt)	0.781	0.984	8	23/16	[0.530, 1.000]
TOPReward	0.648	0.965	8	23/16	[0.060, 0.993]
Qwen3-VL-8B	0.716	0.984	8	23/16	[0.016, 0.977]
Qwen2.5-VL-7B	0.521	0.922	8	23/16	[0.047, 0.531]
Qwen2.5-VL-7B (rubric)	0.526	0.891	8	23/16	[0.588, 0.927]
DINOv2 goal-distance	0.722	0.914	8	23/16	[-0.831, -0.336]
SigLIP goal-distance	0.497	0.766	8	23/16	[-0.491, -0.202]
Pixtral-12B (rubric)	0.684	0.867	8	23/16	[0.925, 0.977]
GPT-5.5 (graded)	0.609	0.824	8	23/16	[0.250, 1.000]
GPT-5.5 (binary)	0.785	0.906	8	23/16	[0.000, 1.000]
Task 3: pick up the bbq sauce and place it in the basket					
RoboReward-8B (as released)	0.846	0.917	11	33/22	[0.544, 1.000]
RoboReward-8B (our prompt)	0.857	0.963	11	33/22	[0.533, 1.000]
TOPReward	0.646	0.899	11	33/22	[0.016, 0.986]
Qwen3-VL-8B	0.737	0.917	11	33/22	[0.003, 0.958]
Qwen2.5-VL-7B	0.636	0.913	11	33/22	[0.026, 0.593]
Qwen2.5-VL-7B (rubric)	0.640	0.812	11	33/22	[0.576, 0.913]
DINOv2 goal-distance	0.835	0.992	11	33/22	[-0.803, -0.242]
SigLIP goal-distance	0.882	0.950	11	33/22	[-0.458, -0.156]
Pixtral-12B (rubric)	0.579	0.804	11	33/22	[0.857, 0.965]
GPT-5.5 (graded)	0.727	0.955	11	33/22	[0.000, 1.000]
GPT-5.5 (binary)	0.758	0.977	11	33/22	[0.000, 1.000]
Task 4: pick up the ketchup and place it in the basket					
RoboReward-8B (as released)	0.845	1.000	10	32/20	[0.750, 1.000]
RoboReward-8B (our prompt)	0.880	1.000	10	32/20	[0.749, 1.000]
TOPReward	0.621	0.965	10	32/20	[0.037, 0.974]
Qwen3-VL-8B	0.711	0.985	10	32/20	[0.004, 0.932]
Qwen2.5-VL-7B	0.435	0.932	10	32/20	[0.037, 0.469]
Qwen2.5-VL-7B (rubric)	0.497	0.843	10	32/20	[0.586, 0.903]
DINOv2 goal-distance	0.701	0.900	10	32/20	[-0.800, -0.271]
SigLIP goal-distance	0.747	0.890	10	32/20	[-0.512, -0.160]
Pixtral-12B (rubric)	0.570	0.755	10	32/20	[0.880, 0.970]
GPT-5.5 (graded)	0.609	0.875	10	32/20	[0.500, 1.000]
GPT-5.5 (binary)	0.672	0.925	10	32/20	[0.000, 1.000]

Table 9: LIBERO-Goal, per-task AUROC for the judges that pass the screen on this suite. n is baseline instances and N/G near and gross failures. A dash marks too few near-misses or no gross failures.

Judge	Near	Gross	n	N/G	Score range
Task 0: open the middle drawer of the cabinet					
RoboReward-8B (as released)	0.880	0.980	6	10/7	[0.215, 0.728]
RoboReward-8B (our prompt)	0.910	1.000	6	10/7	[0.204, 0.731]
TOPReward	0.910	0.959	6	10/7	[0.009, 0.852]
Qwen3-VL-8B	0.985	1.000	6	10/7	[0.001, 0.148]
Qwen2.5-VL-7B	1.000	1.000	6	10/7	[0.018, 0.202]
Qwen2.5-VL-7B (rubric)	0.990	1.000	6	10/7	[0.637, 0.835]
DINOv2 goal-distance	1.000	1.000	6	10/7	[-1.046, -0.530]
SigLIP goal-distance	1.000	1.000	6	10/7	[-0.468, -0.217]

Continued on next page

Table 9 (continued)

Judge	Near	Gross	n	N/G	Score range
Task 1: put the bowl on the stove					
RoboReward-8B (as released)	0.887	0.903	6	16/12	[0.751, 1.000]
RoboReward-8B (our prompt)	0.922	0.875	6	16/12	[0.747, 0.999]
TOPReward	0.680	0.882	6	16/12	[0.060, 0.967]
Qwen3-VL-8B	0.674	0.861	6	16/12	[0.004, 0.852]
Qwen2.5-VL-7B	0.838	0.764	6	16/12	[0.042, 0.245]
Qwen2.5-VL-7B (rubric)	0.809	0.743	6	16/12	[0.621, 0.818]
DINOv2 goal-distance	0.723	0.986	6	16/12	[-0.813, -0.294]
SigLIP goal-distance	0.812	0.958	6	16/12	[-0.385, -0.177]
Task 2: put the wine bottle on top of the cabinet					
RoboReward-8B (as released)	—	—	6	0/0	[0.540, 0.852]
RoboReward-8B (our prompt)	—	—	6	0/0	[0.535, 0.761]
TOPReward	—	—	6	0/0	[0.076, 0.971]
Qwen3-VL-8B	—	—	6	0/0	[0.008, 0.867]
Qwen2.5-VL-7B	—	—	6	0/0	[0.076, 0.148]
Qwen2.5-VL-7B (rubric)	—	—	6	0/0	[0.730, 0.776]
DINOv2 goal-distance	—	—	6	0/0	[-0.674, -0.473]
SigLIP goal-distance	—	—	6	0/0	[-0.348, -0.229]
Task 3: open the top drawer and put the bowl inside					
RoboReward-8B (as released)	—	1.000	2	6/4	[0.259, 0.735]
RoboReward-8B (our prompt)	—	1.000	2	6/4	[0.330, 0.747]
TOPReward	—	1.000	2	6/4	[0.004, 0.294]
Qwen3-VL-8B	—	1.000	2	6/4	[0.001, 0.148]
Qwen2.5-VL-7B	—	1.000	2	6/4	[0.023, 0.182]
Qwen2.5-VL-7B (rubric)	—	1.000	2	6/4	[0.637, 0.818]
DINOv2 goal-distance	—	1.000	2	6/4	[-1.001, -0.524]
SigLIP goal-distance	—	1.000	2	6/4	[-0.492, -0.322]
Task 4: put the bowl on top of the cabinet					
RoboReward-8B (as released)	—	—	5	0/0	[0.753, 0.859]
RoboReward-8B (our prompt)	—	—	5	0/0	[0.745, 0.751]
TOPReward	—	—	5	0/0	[0.119, 0.852]
Qwen3-VL-8B	—	—	5	0/0	[0.026, 0.469]
Qwen2.5-VL-7B	—	—	5	0/0	[0.033, 0.133]
Qwen2.5-VL-7B (rubric)	—	—	5	0/0	[0.638, 0.759]
DINOv2 goal-distance	—	—	5	0/0	[-0.559, -0.301]
SigLIP goal-distance	—	—	5	0/0	[-0.361, -0.238]
Task 5: push the plate to the front of the stove					
RoboReward-8B (as released)	0.578	0.611	3	8/6	[0.757, 0.995]
RoboReward-8B (our prompt)	0.484	0.667	3	8/6	[0.629, 0.969]
TOPReward	0.445	0.500	3	8/6	[0.060, 0.731]
Qwen3-VL-8B	0.492	0.611	3	8/6	[0.008, 0.269]
Qwen2.5-VL-7B	0.414	0.500	3	8/6	[0.076, 0.133]
Qwen2.5-VL-7B (rubric)	0.586	0.611	3	8/6	[0.648, 0.726]
DINOv2 goal-distance	1.000	0.944	3	8/6	[-0.831, -0.253]
SigLIP goal-distance	0.812	0.889	3	8/6	[-0.329, -0.183]
Task 6: put the cream cheese in the bowl					
RoboReward-8B (as released)	0.958	1.000	6	19/12	[0.158, 0.990]
RoboReward-8B (our prompt)	0.967	0.986	6	19/12	[0.263, 0.954]
TOPReward	0.958	1.000	6	19/12	[0.007, 0.294]
Qwen3-VL-8B	0.983	1.000	6	19/12	[0.001, 0.107]
Qwen2.5-VL-7B	0.655	0.646	6	19/12	[0.013, 0.107]
Qwen2.5-VL-7B (rubric)	0.640	0.778	6	19/12	[0.647, 0.799]
DINOv2 goal-distance	0.831	1.000	6	19/12	[-0.824, -0.290]
SigLIP goal-distance	0.895	1.000	6	19/12	[-0.512, -0.183]

Continued on next page

Table 9 (continued)

Judge	Near	Gross	n	N/G	Score range
Task 7: turn on the stove					
RoboReward-8B (as released)	—	—	6	0/0	[0.271, 0.875]
RoboReward-8B (our prompt)	—	—	6	0/0	[0.409, 0.981]
TOPReward	—	—	6	0/0	[0.245, 0.974]
Qwen3-VL-8B	—	—	6	0/0	[0.011, 0.818]
Qwen2.5-VL-7B	—	—	6	0/0	[0.026, 0.294]
Qwen2.5-VL-7B (rubric)	—	—	6	0/0	[0.729, 0.859]
DINOv2 goal-distance	—	—	6	0/0	[-0.377, -0.286]
SigLIP goal-distance	—	—	6	0/0	[-0.216, -0.169]
Task 8: put the bowl on the plate					
RoboReward-8B (as released)	0.707	0.760	5	15/10	[0.524, 0.777]
RoboReward-8B (our prompt)	0.787	0.880	5	15/10	[0.513, 0.776]
TOPReward	0.847	1.000	5	15/10	[0.018, 0.852]
Qwen3-VL-8B	0.753	1.000	5	15/10	[0.002, 0.469]
Qwen2.5-VL-7B	0.633	0.630	5	15/10	[0.047, 0.294]
Qwen2.5-VL-7B (rubric)	0.620	0.620	5	15/10	[0.633, 0.812]
DINOv2 goal-distance	0.613	0.940	5	15/10	[-0.644, -0.285]
SigLIP goal-distance	0.413	0.900	5	15/10	[-0.494, -0.215]
Task 9: put the wine bottle on the rack					
RoboReward-8B (as released)	0.802	0.938	6	9/8	[0.518, 0.995]
RoboReward-8B (our prompt)	0.741	0.906	6	9/8	[0.575, 0.937]
TOPReward	0.778	0.781	6	9/8	[0.007, 0.963]
Qwen3-VL-8B	0.802	0.797	6	9/8	[0.001, 0.755]
Qwen2.5-VL-7B	0.481	0.891	6	9/8	[0.060, 0.438]
Qwen2.5-VL-7B (rubric)	0.543	0.812	6	9/8	[0.655, 0.834]
DINOv2 goal-distance	0.444	0.938	6	9/8	[-0.883, -0.387]
SigLIP goal-distance	0.481	0.750	6	9/8	[-0.457, -0.223]

Table 10: LIBERO-Long, geometric, per-task AUROC for the judges that pass the screen on this suite. n is baseline instances and N/G near and gross failures. A dash marks too few near-misses or no gross failures.

Judge	Near	Gross	n	N/G	Score range
Task 0: put both the alphabet soup and the tomato sauce in the basket					
RoboReward-8B (as released)	0.662	0.645	11	40/22	[0.249, 0.337]
RoboReward-8B (our prompt)	0.760	0.727	11	40/22	[0.251, 0.358]
TOPReward	0.882	0.895	11	40/22	[0.009, 0.133]
Qwen3-VL-8B	0.934	0.942	11	40/22	[0.001, 0.047]
Qwen2.5-VL-7B	0.811	0.921	11	40/22	[0.009, 0.245]
Qwen2.5-VL-7B (rubric)	0.787	0.843	11	40/22	[0.532, 0.758]
DINOv2 goal-distance	0.811	0.979	11	40/22	[-1.092, -0.367]
SigLIP goal-distance	0.681	0.983	11	40/22	[-0.427, -0.151]
Pixtral-12B (rubric)	0.461	0.643	11	40/22	[0.846, 0.947]
Task 1: put both the cream cheese box and the butter in the basket					
RoboReward-8B (as released)	0.439	0.615	12	34/24	[0.249, 0.496]
RoboReward-8B (our prompt)	0.250	0.517	12	34/24	[0.250, 0.543]
TOPReward	0.364	0.793	12	34/24	[0.009, 0.223]
Qwen3-VL-8B	0.323	0.703	12	34/24	[0.001, 0.095]
Qwen2.5-VL-7B	0.287	0.646	12	34/24	[0.015, 0.223]
Qwen2.5-VL-7B (rubric)	0.274	0.477	12	34/24	[0.508, 0.742]
DINOv2 goal-distance	0.838	1.000	12	34/24	[-1.034, -0.369]
SigLIP goal-distance	0.800	1.000	12	34/24	[-0.465, -0.172]
Pixtral-12B (rubric)	0.694	0.736	12	34/24	[0.816, 0.933]

Continued on next page

Table 10 (continued)

Judge	Near	Gross	n	N/G	Score range
Task 2: turn on the stove and put the moka pot on it					
RoboReward-8B (as released)	0.765	1.000	9	25/18	[0.261, 0.751]
RoboReward-8B (our prompt)	0.810	1.000	9	25/18	[0.471, 0.857]
TOPReward	0.881	1.000	9	25/18	[0.068, 0.982]
Qwen3-VL-8B	0.907	1.000	9	25/18	[0.011, 0.777]
Qwen2.5-VL-7B	0.746	0.861	9	25/18	[0.018, 0.469]
Qwen2.5-VL-7B (rubric)	0.787	0.889	9	25/18	[0.663, 0.892]
DINOv2 goal-distance	0.662	0.846	9	25/18	[-0.862, -0.379]
SigLIP goal-distance	0.792	0.815	9	25/18	[-0.574, -0.252]
Pixtral-12B (rubric)	0.638	0.790	9	25/18	[0.869, 0.938]
Task 3: put the black bowl in the bottom drawer of the cabinet and close it					
RoboReward-8B (as released)	0.709	0.695	8	27/16	[0.508, 0.866]
RoboReward-8B (our prompt)	0.761	0.672	8	27/16	[0.713, 0.983]
TOPReward	0.631	0.520	8	27/16	[0.060, 0.706]
Qwen3-VL-8B	0.571	0.418	8	27/16	[0.006, 0.202]
Qwen2.5-VL-7B	0.508	0.617	8	27/16	[0.068, 0.202]
Qwen2.5-VL-7B (rubric)	0.406	0.508	8	27/16	[0.639, 0.788]
DINOv2 goal-distance	0.974	0.992	8	27/16	[-0.990, -0.333]
SigLIP goal-distance	0.904	0.961	8	27/16	[-0.514, -0.214]
Pixtral-12B (rubric)	0.723	0.602	8	27/16	[0.878, 0.929]
Task 4: put the white mug on the left plate and put the yellow and white mug on the right plate					
RoboReward-8B (as released)	0.622	0.967	11	46/22	[0.283, 0.751]
RoboReward-8B (our prompt)	0.488	0.950	11	46/22	[0.319, 0.775]
TOPReward	0.485	0.829	11	46/22	[0.010, 0.924]
Qwen3-VL-8B	0.488	0.882	11	46/22	[0.001, 0.706]
Qwen2.5-VL-7B	0.653	0.868	11	46/22	[0.107, 0.294]
Qwen2.5-VL-7B (rubric)	0.560	0.740	11	46/22	[0.676, 0.768]
DINOv2 goal-distance	0.597	0.880	11	46/22	[-1.035, -0.123]
SigLIP goal-distance	0.654	0.913	11	46/22	[-0.379, -0.178]
Pixtral-12B (rubric)	0.611	0.312	11	46/22	[0.828, 0.923]
Task 5: pick up the book and place it in the back compartment of the caddy					
RoboReward-8B (as released)	0.312	0.735	10	31/20	[0.530, 0.816]
RoboReward-8B (our prompt)	0.328	0.595	10	31/20	[0.647, 0.940]
TOPReward	0.172	0.415	10	31/20	[0.798, 0.992]
Qwen3-VL-8B	0.212	0.403	10	31/20	[0.469, 0.967]
Qwen2.5-VL-7B	0.418	0.458	10	31/20	[0.165, 0.438]
Qwen2.5-VL-7B (rubric)	0.472	0.525	10	31/20	[0.756, 0.898]
DINOv2 goal-distance	0.581	0.920	10	31/20	[-0.825, -0.292]
SigLIP goal-distance	0.457	0.910	10	31/20	[-0.567, -0.164]
Pixtral-12B (rubric)	0.477	0.565	10	31/20	[0.805, 0.924]
Task 6: put the white mug on the plate and put the chocolate pudding to the right of the plate					
RoboReward-8B (as released)	0.868	0.944	9	27/18	[0.479, 0.617]
RoboReward-8B (our prompt)	0.975	0.988	9	27/18	[0.485, 0.747]
TOPReward	0.842	0.994	9	27/18	[0.009, 0.202]
Qwen3-VL-8B	0.784	0.929	9	27/18	[0.001, 0.148]
Qwen2.5-VL-7B	0.726	0.840	9	27/18	[0.095, 0.269]
Qwen2.5-VL-7B (rubric)	0.737	0.926	9	27/18	[0.660, 0.804]
DINOv2 goal-distance	0.967	1.000	9	27/18	[-1.077, -0.357]
SigLIP goal-distance	0.988	1.000	9	27/18	[-0.571, -0.214]
Pixtral-12B (rubric)	0.877	0.944	9	27/18	[0.857, 0.932]

Continued on next page

Table 10 (continued)

Judge	Near	Gross	n	N/G	Score range
Task 7: put both the alphabet soup and the cream cheese box in the basket					
RoboReward-8B (as released)	0.541	0.704	9	29/18	[0.258, 0.493]
RoboReward-8B (our prompt)	0.591	0.815	9	29/18	[0.297, 0.783]
TOPReward	0.715	0.864	9	29/18	[0.026, 0.407]
Qwen3-VL-8B	0.661	0.809	9	29/18	[0.005, 0.321]
Qwen2.5-VL-7B	0.604	0.772	9	29/18	[0.033, 0.378]
Qwen2.5-VL-7B (rubric)	0.600	0.688	9	29/18	[0.550, 0.826]
DINOv2 goal-distance	0.643	0.944	9	29/18	[-0.998, -0.410]
SigLIP goal-distance	0.517	0.889	9	29/18	[-0.499, -0.179]
Pixtral-12B (rubric)	0.553	0.481	9	29/18	[0.754, 0.892]
Task 8: put both moka pots on the stove					
RoboReward-8B (as released)	0.610	0.778	6	27/12	[0.259, 0.502]
RoboReward-8B (our prompt)	0.701	0.917	6	27/12	[0.255, 0.738]
TOPReward	0.631	0.868	6	27/12	[0.014, 0.622]
Qwen3-VL-8B	0.586	0.771	6	27/12	[0.003, 0.165]
Qwen2.5-VL-7B	0.748	0.875	6	27/12	[0.076, 0.349]
Qwen2.5-VL-7B (rubric)	0.643	0.708	6	27/12	[0.547, 0.782]
DINOv2 goal-distance	0.413	0.847	6	27/12	[-1.100, -0.320]
SigLIP goal-distance	0.665	0.986	6	27/12	[-0.581, -0.213]
Pixtral-12B (rubric)	0.480	0.764	6	27/12	[0.868, 0.937]
Task 9: put the yellow and white mug in the microwave and close it					
RoboReward-8B (as released)	0.490	0.664	8	21/16	[0.516, 0.985]
RoboReward-8B (our prompt)	0.649	0.727	8	21/16	[0.618, 0.999]
TOPReward	0.364	0.617	8	21/16	[0.014, 0.867]
Qwen3-VL-8B	0.446	0.605	8	21/16	[0.002, 0.438]
Qwen2.5-VL-7B	0.257	0.461	8	21/16	[0.020, 0.133]
Qwen2.5-VL-7B (rubric)	0.416	0.586	8	21/16	[0.541, 0.723]
DINOv2 goal-distance	0.864	0.969	8	21/16	[-0.943, -0.399]
SigLIP goal-distance	0.952	1.000	8	21/16	[-0.433, -0.189]
Pixtral-12B (rubric)	0.601	0.680	8	21/16	[0.894, 0.941]

Table 11: LIBERO-Long, structural, per-task AUROC for the judges that pass the screen on this suite. n is baseline instances and N/G near and gross failures. A dash marks too few near-misses or no gross failures.

Judge	Near	Gross	n	N/G	Score range
Task 0: put both the alphabet soup and the tomato sauce in the basket					
RoboReward-8B (as released)	0.457	0.455	11	33/22	[0.245, 0.420]
RoboReward-8B (our prompt)	0.405	0.467	11	33/22	[0.258, 0.402]
TOPReward	0.770	0.924	11	33/22	[0.007, 0.148]
Qwen3-VL-8B	0.731	0.909	11	33/22	[0.001, 0.068]
Qwen2.5-VL-7B	0.595	0.831	11	33/22	[0.017, 0.245]
DINOv2 goal-distance	0.780	0.835	11	33/22	[-0.999, -0.408]
SigLIP goal-distance	0.658	0.884	11	33/22	[-0.370, -0.155]
Task 1: put both the cream cheese box and the butter in the basket					
RoboReward-8B (as released)	0.431	0.399	12	36/24	[0.250, 0.487]
RoboReward-8B (our prompt)	0.215	0.253	12	36/24	[0.259, 0.493]
TOPReward	0.252	0.530	12	36/24	[0.010, 0.269]
Qwen3-VL-8B	0.174	0.413	12	36/24	[0.001, 0.148]
Qwen2.5-VL-7B	0.262	0.443	12	36/24	[0.016, 0.182]
DINOv2 goal-distance	0.731	0.917	12	36/24	[-1.023, -0.350]
SigLIP goal-distance	0.551	0.865	12	36/24	[-0.481, -0.172]

Continued on next page

Table 11 (continued)

Judge	Near	Gross	n	N/G	Score range
Task 2: turn on the stove and put the moka pot on it					
RoboReward-8B (as released)	0.780	0.889	9	26/18	[0.250, 0.755]
RoboReward-8B (our prompt)	0.803	0.889	9	26/18	[0.265, 0.929]
TOPReward	0.761	0.904	9	26/18	[0.033, 0.982]
Qwen3-VL-8B	0.765	0.883	9	26/18	[0.006, 0.777]
Qwen2.5-VL-7B	0.530	0.725	9	26/18	[0.037, 0.562]
DINOv2 goal-distance	0.586	0.870	9	26/18	[-0.948, -0.318]
SigLIP goal-distance	0.689	0.809	9	26/18	[-0.539, -0.221]
Task 3: put the black bowl in the bottom drawer of the cabinet and close it					
RoboReward-8B (as released)	0.565	0.804	8	23/15	[0.508, 0.908]
RoboReward-8B (our prompt)	0.412	0.644	8	23/15	[0.598, 0.984]
TOPReward	0.724	0.720	8	23/15	[0.023, 0.777]
Qwen3-VL-8B	0.653	0.582	8	23/15	[0.002, 0.245]
Qwen2.5-VL-7B	0.484	0.151	8	23/15	[0.076, 0.269]
DINOv2 goal-distance	0.578	0.902	8	23/15	[-1.002, -0.354]
SigLIP goal-distance	0.662	0.964	8	23/15	[-0.421, -0.187]
Task 4: put the white mug on the left plate and put the yellow and white mug on the right plate					
RoboReward-8B (as released)	0.684	0.905	11	36/22	[0.301, 0.750]
RoboReward-8B (our prompt)	0.665	0.872	11	36/22	[0.312, 0.752]
TOPReward	0.679	0.847	11	36/22	[0.010, 0.924]
Qwen3-VL-8B	0.698	0.812	11	36/22	[0.002, 0.706]
Qwen2.5-VL-7B	0.598	0.612	11	36/22	[0.095, 0.349]
DINOv2 goal-distance	0.607	0.872	11	36/22	[-0.944, -0.104]
SigLIP goal-distance	0.593	0.843	11	36/22	[-0.408, -0.157]
Task 5: pick up the book and place it in the back compartment of the caddy					
RoboReward-8B (as released)	0.392	0.545	10	34/20	[0.704, 0.752]
RoboReward-8B (our prompt)	0.529	0.690	10	34/20	[0.726, 0.774]
TOPReward	0.547	0.705	10	34/20	[0.818, 0.992]
Qwen3-VL-8B	0.572	0.642	10	34/20	[0.469, 0.963]
Qwen2.5-VL-7B	0.540	0.420	10	34/20	[0.133, 0.438]
DINOv2 goal-distance	0.452	0.735	10	34/20	[-0.640, -0.296]
SigLIP goal-distance	0.515	0.775	10	34/20	[-0.407, -0.149]
Task 6: put the white mug on the plate and put the chocolate pudding to the right of the plate					
RoboReward-8B (as released)	0.786	0.858	9	27/18	[0.503, 0.617]
RoboReward-8B (our prompt)	0.872	0.914	9	27/18	[0.565, 0.747]
TOPReward	0.607	0.586	9	27/18	[0.033, 0.269]
Qwen3-VL-8B	0.553	0.509	9	27/18	[0.007, 0.165]
Qwen2.5-VL-7B	0.471	0.679	9	27/18	[0.133, 0.321]
DINOv2 goal-distance	0.790	1.000	9	27/18	[-0.859, -0.279]
SigLIP goal-distance	0.490	0.784	9	27/18	[-0.393, -0.192]
Task 7: put both the alphabet soup and the cream cheese box in the basket					
RoboReward-8B (as released)	0.494	0.642	9	27/18	[0.263, 0.528]
RoboReward-8B (our prompt)	0.568	0.790	9	27/18	[0.321, 0.783]
TOPReward	0.708	0.870	9	27/18	[0.011, 0.407]
Qwen3-VL-8B	0.665	0.827	9	27/18	[0.002, 0.321]
Qwen2.5-VL-7B	0.547	0.796	9	27/18	[0.016, 0.378]
DINOv2 goal-distance	0.346	0.586	9	27/18	[-0.848, -0.339]
SigLIP goal-distance	0.444	0.494	9	27/18	[-0.390, -0.187]
Task 8: put both moka pots on the stove					
RoboReward-8B (as released)	0.594	0.556	6	21/12	[0.296, 0.501]
RoboReward-8B (our prompt)	0.544	0.514	6	21/12	[0.358, 0.733]
TOPReward	0.594	0.542	6	21/12	[0.033, 0.407]
Qwen3-VL-8B	0.458	0.479	6	21/12	[0.004, 0.119]
Qwen2.5-VL-7B	0.567	0.764	6	21/12	[0.148, 0.349]
DINOv2 goal-distance	0.379	0.694	6	21/12	[-1.026, -0.363]
SigLIP goal-distance	0.399	0.861	6	21/12	[-0.544, -0.197]

Continued on next page

Table 11 (continued)

Judge	Near	Gross	n	N/G	Score range
Task 9: put the yellow and white mug in the microwave and close it					
RoboReward-8B (as released)	0.355	0.492	8	26/16	[0.519, 0.996]
RoboReward-8B (our prompt)	0.439	0.484	8	26/16	[0.630, 0.999]
TOPReward	0.528	0.543	8	26/16	[0.016, 0.651]
Qwen3-VL-8B	0.442	0.348	8	26/16	[0.003, 0.182]
Qwen2.5-VL-7B	0.484	0.500	8	26/16	[0.018, 0.119]
DINOv2 goal-distance	0.746	0.922	8	26/16	[-1.046, -0.468]
SigLIP goal-distance	0.964	0.961	8	26/16	[-0.493, -0.198]

C THE FRAME-BUDGET CONTROL

Table 12 compares 8 and 16 frames for four VLM judges on both LIBERO-Long failure families. Each pair scores the same rollouts with the same model and prompt and changes only the number of frames. These are the six-task pilot pools behind the control in §4, not the ten-task replication of Appendix B. The changes go both ways, from -0.051 to +0.097, depending on the judge and the failure family, and no judge moves far from chance. The largest gain is on gross failures for Qwen2.5-VL-7B with the binary prompt on structural failures. The differences are descriptive rather than significance tests. Sixteen frames is the most we stored for each rollout, so a larger budget would need new rollouts, and we cannot rule out that one would help.

Table 12: Frame-budget control on the six-task LIBERO-Long pilot pools. Each pair scores the same rollouts with the same model and prompt. Every judge has 31 instances, with 102 near and 62 gross failures on geometric failures and 92 and 62 on structural ones.

Frames	Near	Gross	Score range
LIBERO-Long, geometric, RoboReward-8B (as released)			
8	0.549	0.649	[0.250, 0.866]
16	0.523	0.598	[0.252, 0.988]
Δ	-0.026	-0.051	
LIBERO-Long, geometric, TOPReward			
8	0.514	0.614	[0.009, 0.992]
16	0.549	0.646	[0.014, 0.989]
Δ	+0.035	+0.032	
LIBERO-Long, geometric, Qwen3-VL-8B			
8	0.493	0.620	[0.001, 0.963]
16	0.539	0.668	[0.002, 0.967]
Δ	+0.047	+0.048	
LIBERO-Long, geometric, Qwen2.5-VL-7B			
8	0.587	0.663	[0.020, 0.438]
16	0.571	0.684	[0.033, 0.531]
Δ	-0.016	+0.021	
LIBERO-Long, structural, RoboReward-8B (as released)			
8	0.561	0.632	[0.251, 0.908]
16	0.532	0.585	[0.251, 0.922]
Δ	-0.029	-0.047	
LIBERO-Long, structural, TOPReward			
8	0.542	0.631	[0.009, 0.989]
16	0.590	0.652	[0.012, 0.988]
Δ	+0.048	+0.021	
LIBERO-Long, structural, Qwen3-VL-8B			
8	0.515	0.600	[0.001, 0.947]
16	0.592	0.645	[0.002, 0.953]

Continued on next page

Table 12 (continued)

Frames	Near	Gross	Score range
Δ	+0.077	+0.046	
LIBERO-Long, structural, Qwen2.5-VL-7B			
8	0.546	0.566	[0.016, 0.562]
16	0.585	0.663	[0.020, 0.531]
Δ	+0.039	+0.097	

D Measurement Failures and Their Detection

The answer-mass guard of §3 matters. Two earlier versions of our pipeline produced plausible curves while putting under 1% of the probability on the answer tokens, and two models we tried answer a yes-or-no question with digits, so a pipeline that computes $P(\text{Yes})/(P(\text{Yes}) + P(\text{No}))$ would have reported a confident score read from about 11% of the probability. Table 13 lists the 11 failures of our measurement or estimators that we found during the project and how each was detected. Digit answers from different model families count as one failure, and changes to our conclusions after adding models or tasks are not counted. Not every failure was caught by a known-answer test. Some were found by checking the probability on the answer tokens, by auditing task identity or by looking at the frames, and for the early failures the tests now in the repository were written afterwards.

Table 13: Failures of our measurement and estimators found during the project, how each was detected, and what it would have produced if missed.

Failure	How it was detected	What it would produce if missed
Answer position: preamble	Inspecting the generated tokens and the probability on the answer tokens; the answer-mass guard now rejects an unreadable answer.	Confident near/gross curves from tokens that were not the answer.
Answer position: whitespace and token form	Checking which token follows the answer prefix, with and without a trailing space, and scoring whichever form carries the probability.	A normalised score read from a negligible share of the probability, with a plausible-looking curve.
A fixed grid of pushes left too few near-misses	Counting failures in the near band. Tests now plant a known success threshold and require bisection to find it.	A finished sweep with too few near-misses to measure the gap.
The push appeared to move only the arm	Comparing object positions in the simulator with the intended ones; a test requires the object to move before release.	Arm motion reported as moving the object, without checking how far the object moved.
Regression over the whole displacement range	A simulated judge that reads only appearance, with successes and failures at different displacements; the estimator must refuse to extrapolate past the shared range.	A trend with displacement presented as a drop in score when the task fails.
Raw mean gap within the shared range	A simulated judge that reads only appearance has no extra penalty for failing, and adjusting for displacement within the shared range must recover that.	Leftover differences in displacement presented as an extra penalty for failing.
Errors silently skipped for each configuration	An integration test requires output records with the expected fields and treats an empty configuration as an error.	A message that the ladder finished, with no scores written.
Goal image excluded by rollout instead of instance	Synthetic instances with distinct layouts show whether a failure receives its own instance's goal image; the fix excludes the whole instance.	Successes and failures getting different access to the goal image, which shifts the encoder AUROCs.
Numeric answers to a binary prompt	The answer-mass guard and inspection of the most likely tokens show digits instead of yes or no. This recurs across model families.	A confident binary score from renormalising a small share of the probability.
Duplicated tasks treated as independent	Auditing task identity across failure families and averaging the repeated task before the held-out permutation test.	Inflated support for a predictor of task difficulty.

Continued on next page

Table 13 (continued)

Failure	How it was detected	What it would produce if missed
Inverted Meta-World frames	Looking at saved rollouts showed that the renderer flipped the frames. The readout was healthy, so its guard could not catch this.	Chance-level encoder scores blamed on the judges rather than on upside-down scenes.

The goal-image leak is the one failure whose scores we kept from before and after the fix. Table 14 shows that fixing it lowered the encoders' near AUROC by 0.03 to 0.04 and barely changed their gross AUROC, and every ladder table uses the fixed scores. Two other results are not failures in this sense. The crop prediction of Appendix A failed although its synthetic test passed, and fluctuations in the selection null are expected as long as their intervals cover the planted null.

Table 14: Instance-exclusion correction. Scores from the first version of the ladder are shown only for the audit.

Configuration	Old near	Corrected near	Old gross	Corrected gross
DINOv2 goal-distance	0.697	0.666	0.903	0.901
SigLIP goal-distance	0.726	0.691	0.859	0.852

E Controls and Regressions for Section 6

These tables expand §6. They cover every judge we ran the control on, seven VLM judges including the rubric-prompted Qwen2.5-VL-7B and Pixtral-12B, on all five suites, with the long-horizon suites taken from the six-task pilot pools. At the time we admitted a judge by its mean AUROC for successes against all failures over the single-stage suites, not by the per-suite gross-failure screen of the main text, so a cell here does not mean that its judge passes the screen on that suite. Over these 35 cells the penalty is positive in 27 and significantly positive in 12 (Table 15). The main text counts only the five core VLM judges on the suites where each passes the screen, which gives 19 positive and 9 significant out of 23.

Still-successful control. For each perturbed rollout that still succeeds, the penalty is the score of its unperturbed rollout minus its own score, and each cell averages these penalties. A positive value means the judge marks down a perturbation that leaves the task solved. The bootstrap resamples whole instances rather than single rollouts, since the controls from one instance are not independent. Table 16 lists every cell, including the negative estimates and those whose intervals cross zero.

Regression at matched distance from the goal. This regression asks whether a failure costs more than a success that left the object equally far from its goal. Successes and failures mostly occupy different distance ranges, so a fit over all rollouts would show a drop at the boundary for any judge that only tracks distance. We therefore keep the range where both occur and fit the score drop jointly to distance from the goal and a failure indicator, with an intercept. The failure indicator is not standardised, so its coefficient is in score units. Simply comparing mean drops within the range is not enough, because on a simulated judge that reads only appearance it still produces an apparent failure effect of $+0.026$. The range is fixed from the full sample during the bootstrap. The distance can be measured only for tasks that place an object at a target. This covers every rollout on LIBERO-Spatial and LIBERO-Object but only some on LIBERO-Goal and the long-horizon suites, and on LIBERO-Goal no range contains both successes and failures. Table 17 gives the coverage, the range and the number of successes and failures inside it. An interval covering zero does not show that a judge ignores the outcome, especially where few successes fall inside the range.

All intervals use 2000 bootstrap resamples and the 2.5th and 97.5th percentiles. Cells share instances across judges, and the two long-horizon suites share instances with each other, so the intervals are not simultaneous and their signs are not independent tests.

Table 15: Signs of the still-successful penalty over all judge–suite cells.

Quantity	Cells
All cells	35
Positive estimate	27
Negative estimate	8
Significantly positive	12
Significantly negative	0
Interval covers zero	23

Table 16: Still-successful penalty for every judge and suite. A positive penalty marks down a perturbation that leaves the task solved. Intervals resample whole instances.

Judge	Instances	Controls	Mean score	Penalty [CI]
LIBERO-Spatial				
RoboReward-8B (as released)	55	157	0.5438	+0.0220 [-0.0110, +0.0546]
RoboReward-8B (our prompt)	55	157	0.5196	+0.0191 [-0.0201, +0.0559]
TOPReward	55	157	0.4374	+0.1027 [+0.0622, +0.1448]
Qwen3-VL-8B	55	157	0.1557	+0.0521 [+0.0289, +0.0788]
Qwen2.5-VL-7B	55	157	0.1694	+0.0205 [+0.0070, +0.0336]
Qwen2.5-VL-7B (rubric)	55	157	0.7684	+0.0110 [+0.0009, +0.0207]
Pixtral-12B (rubric)	55	157	0.9189	-0.0036 [-0.0074, +0.0000]
LIBERO-Object				
RoboReward-8B (as released)	51	151	0.9292	+0.0050 [-0.0121, +0.0226]
RoboReward-8B (our prompt)	51	151	0.9637	+0.0163 [+0.0025, +0.0315]
TOPReward	51	151	0.8237	-0.0265 [-0.0616, +0.0054]
Qwen3-VL-8B	51	151	0.7278	-0.0174 [-0.0621, +0.0234]
Qwen2.5-VL-7B	51	151	0.3248	-0.0125 [-0.0298, +0.0034]
Qwen2.5-VL-7B (rubric)	51	151	0.8141	-0.0018 [-0.0083, +0.0045]
Pixtral-12B (rubric)	51	151	0.9540	-0.0016 [-0.0033, +0.0003]
LIBERO-Goal				
RoboReward-8B (as released)	51	98	0.7672	+0.0476 [+0.0116, +0.0862]
RoboReward-8B (our prompt)	51	98	0.7441	+0.0396 [+0.0124, +0.0698]
TOPReward	51	98	0.5038	+0.0549 [-0.0206, +0.1305]
Qwen3-VL-8B	51	98	0.2167	+0.0336 [-0.0157, +0.0809]
Qwen2.5-VL-7B	51	98	0.1406	+0.0253 [+0.0122, +0.0397]
Qwen2.5-VL-7B (rubric)	51	98	0.7553	+0.0149 [+0.0065, +0.0234]
Pixtral-12B (rubric)	51	98	0.9059	-0.0032 [-0.0079, +0.0019]
LIBERO-Long, geometric				
RoboReward-8B (as released)	31	84	0.5930	+0.0040 [-0.0126, +0.0215]
RoboReward-8B (our prompt)	31	84	0.6015	-0.0105 [-0.0280, +0.0060]
TOPReward	31	84	0.4478	+0.0069 [-0.0433, +0.0582]
Qwen3-VL-8B	31	84	0.2713	+0.0180 [-0.0257, +0.0700]
Qwen2.5-VL-7B	31	84	0.2120	+0.0144 [-0.0053, +0.0350]
Qwen2.5-VL-7B (rubric)	31	84	0.7332	+0.0048 [-0.0067, +0.0152]
Pixtral-12B (rubric)	31	84	0.8960	+0.0070 [+0.0018, +0.0126]
LIBERO-Long, structural				
RoboReward-8B (as released)	31	96	0.5930	+0.0169 [+0.0041, +0.0320]
RoboReward-8B (our prompt)	31	96	0.6015	+0.0050 [-0.0049, +0.0173]
TOPReward	31	96	0.4478	+0.0427 [-0.0056, +0.0960]
Qwen3-VL-8B	31	96	0.2713	+0.0235 [-0.0111, +0.0576]
Qwen2.5-VL-7B	31	96	0.2120	+0.0133 [+0.0022, +0.0259]
Qwen2.5-VL-7B (rubric)	31	96	0.7332	+0.0052 [-0.0012, +0.0121]
Pixtral-12B (rubric)	31	96	0.8960	+0.0013 [-0.0039, +0.0067]

Table 17: Regression at matched distance from the goal. Coverage counts perturbed rollouts with a measurable distance, the range is where both outcomes occur, and P/F counts successes and failures inside it. The coefficient is the extra score drop for failing at a given distance.

Judge	Coverage	Range (cm)	P/F	Failure coefficient [CI]
LIBERO-Spatial				
RoboReward-8B (as released)	486/486	3.03–7.36	14/202	-0.0271 [-0.2103, +0.1283]
RoboReward-8B (our prompt)	486/486	3.03–7.36	14/202	-0.0141 [-0.2341, +0.1634]
TOPReward	486/486	3.03–7.36	14/202	-0.0818 [-0.2152, +0.0456]
Qwen3-VL-8B	486/486	3.03–7.36	14/202	-0.0809 [-0.2253, +0.0192]
Qwen2.5-VL-7B	486/486	3.03–7.36	14/202	+0.0103 [-0.0289, +0.0442]
Qwen2.5-VL-7B (rubric)	486/486	3.03–7.36	14/202	+0.0021 [-0.0355, +0.0390]
Pixtral-12B (rubric)	486/486	3.03–7.36	14/202	+0.0013 [-0.0130, +0.0126]
LIBERO-Object				
RoboReward-8B (as released)	459/459	1.30–4.76	125/10	+0.0105 [-0.0545, +0.0764]
RoboReward-8B (our prompt)	459/459	1.30–4.76	125/10	+0.0275 [-0.0315, +0.1012]
TOPReward	459/459	1.30–4.76	125/10	-0.0833 [-0.2378, +0.0040]
Qwen3-VL-8B	459/459	1.30–4.76	125/10	-0.1147 [-0.2956, -0.0021]
Qwen2.5-VL-7B	459/459	1.30–4.76	125/10	-0.0650 [-0.1120, -0.0232]
Qwen2.5-VL-7B (rubric)	459/459	1.30–4.76	125/10	-0.0118 [-0.0309, +0.0125]
Pixtral-12B (rubric)	459/459	1.30–4.76	125/10	+0.0044 [-0.0040, +0.0099]
LIBERO-Goal				
RoboReward-8B (as released)	99/270	–	–	No shared range
RoboReward-8B (our prompt)	99/270	–	–	No shared range
TOPReward	99/270	–	–	No shared range
Qwen3-VL-8B	99/270	–	–	No shared range
Qwen2.5-VL-7B	99/270	–	–	No shared range
Qwen2.5-VL-7B (rubric)	99/270	–	–	No shared range
Pixtral-12B (rubric)	99/270	–	–	No shared range
LIBERO-Long, geometric				
RoboReward-8B (as released)	126/279	0.39–22.41	41/53	+0.0195 [-0.0052, +0.0492]
RoboReward-8B (our prompt)	126/279	0.39–22.41	41/53	+0.0414 [+0.0125, +0.0703]
TOPReward	126/279	0.39–22.41	41/53	+0.0196 [-0.0718, +0.1308]
Qwen3-VL-8B	126/279	0.39–22.41	41/53	-0.0314 [-0.0905, +0.0333]
Qwen2.5-VL-7B	126/279	0.39–22.41	41/53	-0.0115 [-0.0284, +0.0071]
Qwen2.5-VL-7B (rubric)	126/279	0.39–22.41	41/53	-0.0122 [-0.0222, -0.0020]
Pixtral-12B (rubric)	126/279	0.39–22.41	41/53	-0.0028 [-0.0086, +0.0043]
LIBERO-Long, structural				
RoboReward-8B (as released)	126/279	1.24–20.18	39/69	+0.0114 [-0.0207, +0.0389]
RoboReward-8B (our prompt)	126/279	1.24–20.18	39/69	-0.0015 [-0.0245, +0.0161]
TOPReward	126/279	1.24–20.18	39/69	+0.0183 [-0.0311, +0.0619]
Qwen3-VL-8B	126/279	1.24–20.18	39/69	+0.0116 [-0.0545, +0.0739]
Qwen2.5-VL-7B	126/279	1.24–20.18	39/69	-0.0166 [-0.0383, +0.0025]
Qwen2.5-VL-7B (rubric)	126/279	1.24–20.18	39/69	-0.0051 [-0.0153, +0.0037]
Pixtral-12B (rubric)	126/279	1.24–20.18	39/69	-0.0022 [-0.0093, +0.0056]

First analysis of Qwen2.5-VL-7B on LIBERO-Spatial. Our first version of this analysis used Qwen2.5-VL-7B with a graded and a binary prompt on LIBERO-Spatial. It fitted each rollout's score to distance and failure over the whole range, with both predictors standardised (Table 18). This fit is confounded in the way described above, and its coefficients are not on the same scale as the matched fit, so we report it only for completeness. Table 19 reruns the matched fit on the same scores. The same analysis also regressed the score drop on the visual change measured by DINOv2 and SigLIP embeddings and on the outcome (Table 20). Visual change explains more of the variance than the outcome in three of the four fits, the exception being the binary prompt with DINOv2. These encoder fits exist only for LIBERO-Spatial.

Table 18: First analysis of Qwen2.5-VL-7B on LIBERO-Spatial, with the penalty and the whole-range coefficients on distance and failure. The whole-range fit is confounded.

Prompt	Penalty [CI]	β_d [CI]	β_f [CI]
graded	+0.0267 [+0.0113, +0.0423]	-0.0074 [-0.0193, +0.0042]	-0.0211 [-0.0297, -0.0125]
binary	+0.0205 [+0.0070, +0.0336]	-0.0154 [-0.0225, -0.0087]	-0.0140 [-0.0215, -0.0073]

Table 19: The matched fit rerun on the scores of Table 18.

Prompt	Range (cm)	P/F	Failure coefficient [CI]
graded	3.03–7.36	14/202	+0.0137 [-0.0311, +0.0579]
binary	3.03–7.36	14/202	+0.0103 [-0.0289, +0.0442]

Table 20: Score drop of Qwen2.5-VL-7B on LIBERO-Spatial regressed on visual change, measured by each encoder, and on the outcome.

Prompt	Encoder	β_v	β_f	Partial R^2_v	Partial R^2_f	Full R^2	n
graded	DINOv2	0.0199	0.0157	0.0345	0.0216	0.1342	486
graded	SigLIP	0.0341	0.0099	0.1136	0.0096	0.2133	486
binary	DINOv2	0.0075	0.0177	0.0076	0.0419	0.1074	486
binary	SigLIP	0.0161	0.0139	0.0387	0.0289	0.1385	486

F Resolution Curves and the Cross-Task Collapse Test

Estimator. For each judge on each suite, the failures at a given boundary multiple are compared with the unperturbed successes of the same instances. A multiple with fewer than 8 such instances is drawn in the curves but not used for the threshold. The AUROC is smoothed across multiples by isotonic regression, and r^*_θ is the smallest multiple at which the smoothed curve reaches θ, or NR if it does not by $4\times$. Intervals are the 2.5th and 97.5th percentiles over 1,000 bootstraps of whole instances. The still-successful branch is built the same way from perturbed rollouts that still succeed and is read at $0.75\times$. VLM judges appear only on the suites where they pass the screen, while the two encoders are always shown. The long-horizon suites use the six-task pilot pools.

Repair on held-out tasks. The fine-tuned judge of §6.2 was trained on LIBERO-Spatial tasks 0 and 2 and on tasks 0, 2, 3, 5, 6, 8 and 9 of LIBERO-Goal and LIBERO-Long. Table 22 compares the base and tuned judges on the remaining tasks of those suites, pooled and per suite, and on LIBERO-Object, which was never used in training. The last rows give the still-successful AUROC on the training tasks themselves. There the binary labels marked perturbed successes as successes, so a penalty on them means the judge learned to read how far the object moved rather than the label. On the held-out LIBERO-Goal tasks each failure multiple has at most six instances, too few for a threshold, and the control has five.

Cross-task collapse test. A judge that reads only how far the object moved should score every task the same way at the same distance in centimetres, so its per-task curves should line up when plotted against centimetres and spread out when plotted against boundary multiples. A judge that checks the success condition should do the reverse. Within LIBERO-Spatial the two axes are nearly proportional, because objects at the boundary end 3.2 to 3.6 cm from their goal on every task. We therefore pool the placement suites LIBERO-Spatial, LIBERO-Object and Meta-World, where that distance ranges from 3.2 to 7.9 cm. Each task's curve of mean score is rescaled to 0 at its successes and 1 at $4\times$. On each axis we fit a monotone curve to all tasks but one and use it to predict the held-out task. The statistic D is the prediction error on the multiple axis minus that on the centimetre axis, so $D > 0$ favours the distance reader. We bootstrap instances 1,000 times and compare against 999 random reassignments of tasks to their centimetre scales. Synthetic judges that read only distance or only the success condition, on tasks with boundaries of 4, 7 and 9 cm, give the expected answers. On the real

Table 21: Resolution at two thresholds and the still-successful AUROC at $0.75\times$ for every judge that passes the screen on each suite.

Suite	Judge	$r^*_{0.75}$ [95% CI]	$r^*_{0.80}$ [95% CI]	Still-successful AUROC [95% CI]
LIBERO-Spatial	TOPReward	1.00 [1.00, 1.35]	1.35 [1.00, 2.50]	0.69 [0.62, 0.77]
	Qwen3-VL-8B	1.00 [1.00, 1.35]	1.35 [1.00, 1.70]	0.68 [0.62, 0.76]
	Qwen2.5-VL-7B	1.70 [1.00, 1.70]	1.70 [1.35, 4.00]	0.69 [0.62, 0.76]
	Qwen2.5-VL-7B (rubric)	1.35 [1.00, 1.70]	1.70 [1.35, 4.00]	0.68 [0.59, 0.76]
	Pixtral-12B (rubric)	NR [NR, NR]	NR [NR, NR]	0.49 [0.45, 0.54]
	DINOv2 goal-distance	1.35 [1.00, 1.70]	1.70 [1.35, 2.50]	0.66 [0.54, 0.77]
	SigLIP goal-distance	1.70 [1.00, 4.00]	1.70 [1.70, 4.00]	0.60 [0.48, 0.71]
LIBERO-Object	RoboReward-8B (as released)	2.50 [1.35, 2.50]	2.50 [2.50, 4.00]	0.54 [0.48, 0.60]
	RoboReward-8B (our prompt)	1.35 [1.00, 1.70]	1.35 [1.15, 2.50]	0.59 [0.53, 0.64]
	TOPReward	2.50 [1.70, 2.50]	2.50 [2.50, 4.00]	0.53 [0.47, 0.59]
	Qwen3-VL-8B	1.70 [1.35, 2.50]	2.50 [1.70, 2.50]	0.55 [0.49, 0.62]
	Qwen2.5-VL-7B	2.50 [2.50, 4.00]	2.50 [2.50, 4.00]	0.45 [0.38, 0.53]
	Qwen2.5-VL-7B (rubric)	4.00 [2.50, 4.00]	4.00 [4.00, 4.00]	0.49 [0.44, 0.53]
	Pixtral-12B (rubric)	4.00 [1.70, NR]	NR [4.00, NR]	0.46 [0.40, 0.52]
	DINOv2 goal-distance	1.00 [1.00, 1.70]	1.35 [1.00, 1.70]	0.67 [0.59, 0.75]
	SigLIP goal-distance	1.35 [1.00, 1.70]	1.70 [1.00, 2.50]	0.61 [0.52, 0.70]
LIBERO-Goal	RoboReward-8B (as released)	NR [1.15, NR]	NR [4.00, NR]	0.55 [0.47, 0.63]
	RoboReward-8B (our prompt)	NR [1.00, NR]	NR [2.50, NR]	0.57 [0.49, 0.66]
	TOPReward	2.50 [1.15, NR]	4.00 [1.35, NR]	0.55 [0.48, 0.63]
	Qwen3-VL-8B	1.70 [1.00, 4.00]	4.00 [1.00, NR]	0.60 [0.52, 0.69]
	Qwen2.5-VL-7B	NR [1.35, NR]	NR [4.00, NR]	0.61 [0.51, 0.70]
	Qwen2.5-VL-7B (rubric)	4.00 [1.15, NR]	NR [1.70, NR]	0.59 [0.49, 0.69]
	DINOv2 goal-distance	1.35 [1.00, 2.50]	1.70 [1.00, 2.50]	0.66 [0.56, 0.77]
	SigLIP goal-distance	1.70 [1.00, 2.50]	2.50 [1.00, 4.00]	0.60 [0.48, 0.74]
LIBERO-Long, geometric	RoboReward-8B (as released)	NR [4.00, NR]	NR [NR, NR]	0.56 [0.49, 0.65]
	RoboReward-8B (our prompt)	NR [NR, NR]	NR [NR, NR]	0.51 [0.41, 0.61]
	TOPReward	NR [NR, NR]	NR [NR, NR]	0.51 [0.43, 0.60]
	Qwen3-VL-8B	NR [NR, NR]	NR [NR, NR]	0.49 [0.41, 0.57]
	Qwen2.5-VL-7B	NR [4.00, NR]	NR [NR, NR]	0.49 [0.42, 0.56]
	Qwen2.5-VL-7B (rubric)	NR [NR, NR]	NR [NR, NR]	0.50 [0.43, 0.58]
	Pixtral-12B (rubric)	NR [NR, NR]	NR [NR, NR]	0.62 [0.51, 0.73]
	DINOv2 goal-distance	2.50 [1.35, 4.00]	2.50 [1.70, 4.00]	0.58 [0.47, 0.69]
	SigLIP goal-distance	1.70 [1.00, 2.50]	2.50 [1.00, 4.00]	0.66 [0.53, 0.77]
LIBERO-Long, structural	RoboReward-8B (as released)	NR [NR, NR]	NR [NR, NR]	0.58 [0.52, 0.65]
	RoboReward-8B (our prompt)	NR [NR, NR]	NR [NR, NR]	0.51 [0.45, 0.56]
	TOPReward	NR [NR, NR]	NR [NR, NR]	0.57 [0.51, 0.64]
	Qwen3-VL-8B	NR [NR, NR]	NR [NR, NR]	0.55 [0.48, 0.64]
	Qwen2.5-VL-7B	NR [NR, NR]	NR [NR, NR]	0.53 [0.48, 0.58]
	DINOv2 goal-distance	2.50 [1.70, 4.00]	4.00 [1.70, NR]	0.51 [0.40, 0.61]
	SigLIP goal-distance	2.50 [1.70, 4.00]	2.50 [1.70, 4.00]	0.55 [0.43, 0.67]
Meta-World	Qwen2.5-VL-7B	NR [NR, NR]	NR [NR, NR]	0.52 [0.47, 0.57]
	DINOv2 goal-distance	4.00 [2.50, NR]	NR [4.00, NR]	0.49 [0.41, 0.57]
	SigLIP goal-distance	NR [2.50, NR]	NR [4.00, NR]	0.53 [0.42, 0.65]

Table 22: Resolution and still-successful AUROC of the repaired judge before and after fine-tuning.

Evaluation tasks	Labels	n	Base $r^*_{0.75}$	Tuned $r^*_{0.75}$	Still-successful AUROC, base $\rightarrow$ tuned
Held-out, pooled	binary	67	4.00 [1.70, NR]	1.35 [0.50, 2.50]	0.57 $\rightarrow$ 0.60
	calibrated	67	NR [4.00, NR]	1.35 [0.50, 1.35]	0.54 $\rightarrow$ 0.60
LIBERO-Spatial, held-out	binary	18	1.70 [1.00, 4.00]	1.00 [1.00, 1.00]	0.66 $\rightarrow$ 0.86
	calibrated	18	1.35 [1.00, NR]	1.00 [1.00, 1.00]	0.60 $\rightarrow$ 0.86
LIBERO-Goal, held-out	binary	17	insuff.	insuff.	0.84 $\rightarrow$ 0.68
	calibrated	17	insuff.	insuff.	0.88 $\rightarrow$ 0.96
LIBERO-Long, held-out	binary	32	NR [4.00, NR]	2.50 [1.15, 4.00]	0.47 $\rightarrow$ 0.55
	calibrated	32	NR [NR, NR]	1.35 [1.15, 2.50]	0.46 $\rightarrow$ 0.52
LIBERO-Object, unseen	binary	51	2.50 [2.50, 4.00]	1.70 [1.35, 1.70]	0.45 $\rightarrow$ 0.49
	calibrated	51	2.50 [2.50, 4.00]	1.70 [1.35, 1.70]	0.49 $\rightarrow$ 0.44
Training tasks	binary	132	—	—	0.58 $\rightarrow$ 0.74
	calibrated	132	—	—	0.57 $\rightarrow$ 0.85

judges (Table 23) the distance prediction holds for none, fails for three and is inconclusive for the rest, including all four repaired judges.

The test has two limits, so we treat it as exploratory. Rescaling uses the held-out task's own endpoints and can remove its physical scale even for a judge that reads distance linearly. When a judge scores successes and $4\times$ failures almost the same, the rescaling also inflates the errors, which is why some judges have very large errors in Table 23. Without rescaling, each task keeps its own baseline and the curves do not line up on either axis. The result agrees with §6.1, where judges do not share one response to distance across tasks. The claim of §6 rests on the within-instance controls, not on this test.

Table 23: Leave-one-task-out collapse test on the pooled placement suites, with $D = \mathrm{MSE}_{\text{multiple}} - \mathrm{MSE}_{\text{cm}}$. The prediction fails when the bootstrap interval lies below zero and holds when it lies above. All judges with scores on these suites are included, whether or not they pass the screen.

Judge	Tasks	MSE (multiple)	MSE (cm)	D [95% CI]	Permutation p	Prediction
Qwen2.5-VL-7B	12	51.401	52.421	-1.019 [-2.295, 0.059]	0.285	inconclusive
Qwen2.5-VL-7B (rubric)	12	1.629	1.655	-0.026 [-2.875, 0.071]	0.934	inconclusive
DINOv2 goal-distance	12	0.022	0.054	-0.031 [-0.107, -0.018]	0.967	fails
Pixtral-12B (rubric)	8	0.127	0.080	0.047 [-0.108, 0.163]	0.001	inconclusive
Qwen3-VL-8B	8	0.127	0.169	-0.042 [-0.069, -0.030]	0.680	fails
Repair base (binary)	6	0.158	0.160	-0.002 [-0.051, 0.080]	0.216	inconclusive
Repair base (calibrated)	6	0.055	0.065	-0.010 [-0.042, 0.040]	0.509	inconclusive
Repair tuned (binary)	6	0.040	0.045	-0.005 [-0.040, 0.036]	0.378	inconclusive
Repair tuned (calibrated)	6	0.020	0.031	-0.010 [-0.032, 0.010]	0.401	inconclusive
RoboReward-8B (as released)	8	0.913	0.951	-0.038 [-5.354, 0.234]	0.808	inconclusive
RoboReward-8B (our prompt)	8	10.240	9.676	0.564 [-8.724, 0.446]	0.009	inconclusive
SigLIP goal-distance	12	0.353	0.427	-0.074 [-1.115, 0.001]	0.226	inconclusive
SmolVLM2 (graded)	8	570.480	597.126	-26.646 [-17.314, 0.555]	0.091	inconclusive
TOPReward	8	0.115	0.160	-0.045 [-0.070, -0.035]	0.694	fails

G Additional Figures and Tables

Table 24: The AUROC values of Figure 3 for successes against near-miss and gross failures. Long-geo and Long-str are LIBERO-Long with geometric and structural failures, on the ten-task replication. †At chance on both bands, which we attribute to its real-robot training data. ‡Parsed from text, because the API returns no log probabilities.

	Spatial		Object		Goal		Long-geo		Long-str	
Judge	Near	Gross	Near	Gross	Near	Gross	Near	Gross	Near	Gross
RoboReward-8B (as released)	0.539†	0.515†	0.655	0.873	0.677	0.725	0.539	0.616	0.531	0.585
RoboReward-8B (our prompt)	0.570†	0.534†	0.744	0.954	0.690	0.738	0.549	0.632	0.537	0.587
TOPReward	0.814	0.878	0.585	0.878	0.686	0.783	0.555	0.663	0.548	0.639
Qwen3-VL-8B	0.819	0.901	0.613	0.894	0.733	0.821	0.556	0.659	0.528	0.607
Qwen2.5-VL-7B	0.753	0.852	**0.477**	0.865	0.665	0.699	0.569	0.637	0.513	0.569
GPT-5.5 (graded)‡	0.598	0.880	0.620	0.889	—	—	—	—	—	—
GPT-5.5 (binary)‡	0.492	0.634	0.702	0.945	—	—	—	—	—	—
DINOv2 goal-distance	0.719	0.944	0.771	0.964	0.715	0.923	**0.723**	**0.929**	0.602	0.823
SigLIP goal-distance	0.691	0.852	0.741	0.929	0.693	0.864	0.683	0.915	0.577	0.802

Selection estimator (§5). The probability of picking the success is computed exactly over every pair of distractors, with ties split so that a constant judge scores $1/3$. The bootstrap resamples instances and uses the same resample for both bands. A simulated judge with the same skill on both bands still shows a difference of -0.103 at 40 instances, although its interval covers zero, so point estimates alone would overstate the loss.

Encoder against VLM judges in selection (§5). On ten LIBERO-Long tasks with geometric failures, we compare near-band best-of-3 accuracy of SigLIP goal-distance with that of each of the five VLM judges over 88 instances. SigLIP picks the success 68% of the time and RoboReward-8B, the best VLM judge, 58%, a difference of $+0.098$ with 95% interval $[-0.030, +0.235]$. After Holm correction for the five comparisons, SigLIP beats Qwen3-VL-8B by $+0.184$ (adjusted $p = 0.030$)

Table 25: Where each benchmark's negatives come from and what it can measure. *Sim. label* means a simulator predicate assigns the label, and *Per-inst. boundary* means failures are placed relative to each instance's own boundary. *Still-succ. control* means perturbed rollouts that still succeed are kept as controls, *Pretrained encoder* means a goal-distance baseline is reported, *BoN cost* means selection is measured separately for near and gross failures, and *Task var.* means task variance is compared with judge variance. A question mark marks a property the paper does not state.

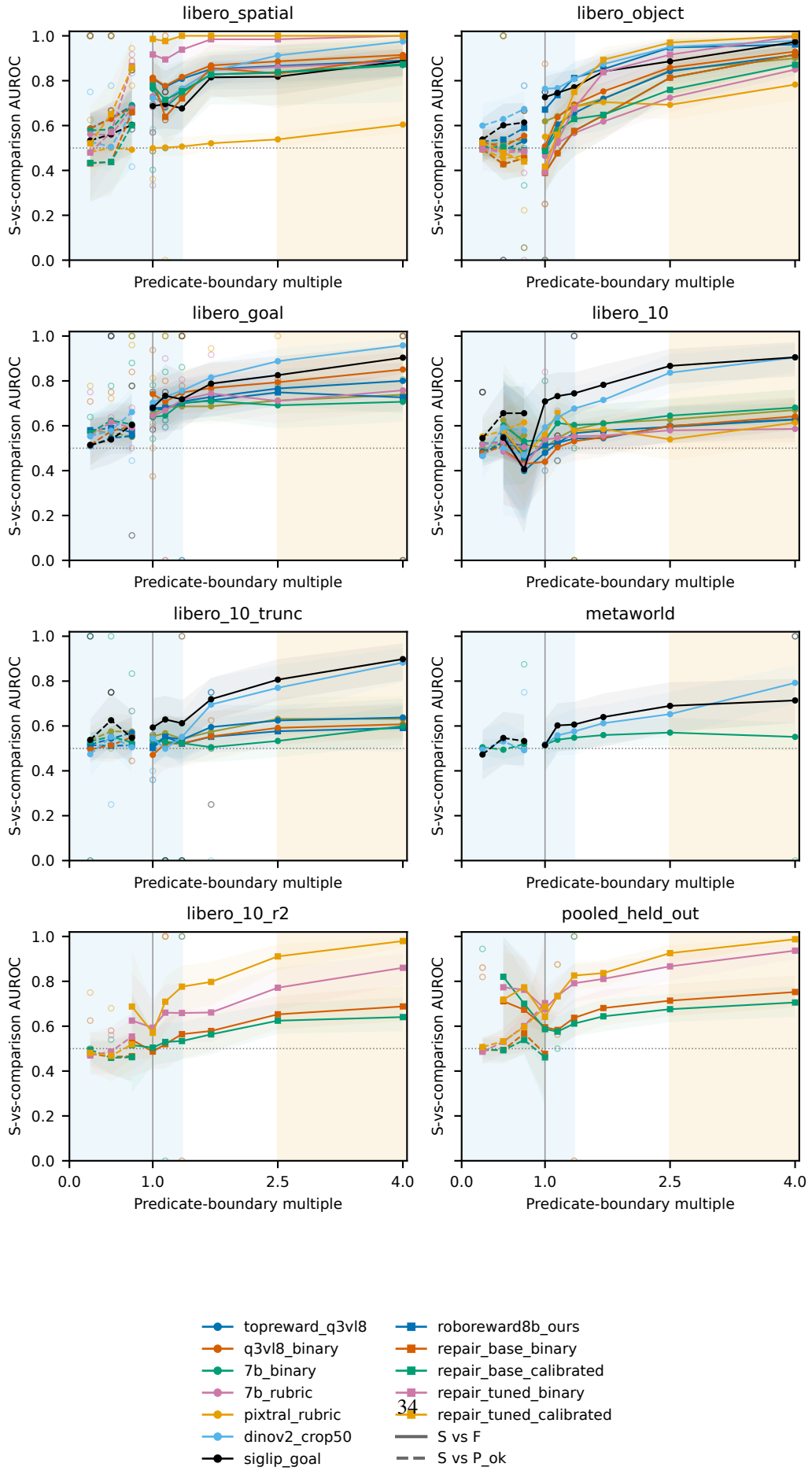
Work	Negatives from	Sim. label	Per-inst. boundary	Still-succ. control	Pretrained encoder	BoN cost	Task var.
RoboReward (2026)	relabel + clip	n	n	n	n	n	n
Robo-Dopamine (2025)	not stated	?	n	n	n	n	n
Sudhakar et al. (2026)	policy rollouts	part.[†]	n	n	n	n	n
FailBench (2026)	public corpora	?	n	n	n	n	n[‡]
GE-Sim 2.0 (2026)	generated video	y[*]	n	n	n	n	n
This paper	predicate bisection	y	y	y	y	y	y

[*] A learned judge on generated video, not a physics predicate. [†] RoboCasa rollouts are scored by the simulator, but the LBM and real-arm evaluations are not. [‡] Groups tasks by the visual evidence they need, but does not separate task from judge variance.

and Qwen2.5-VL-7B by $+0.179$ ($p = 0.022$), and the other contrasts are not significant. Correcting over seven comparisons instead, adding the rubric-prompted Qwen2.5-VL-7B and Pixtral-12B as we originally planned, gives the same two significant contrasts ($p = 0.046$ and 0.031).

Table 26: The repair and its comparisons. All rows train on some tasks of LIBERO-Spatial, LIBERO-Goal and LIBERO-Long and are tested on the remaining tasks and on LIBERO-Object, which is never used in training. Calibrated labels give each rollout's position relative to its instance's boundary, which only our construction can supply.

Judge	Held-out near	Held-out gross	LIBERO-Object near
Qwen2.5-VL-7B (base)	0.563	0.699	0.472
DINOv2 probe (trained head)	0.561	0.769	0.653
DINOv2 goal-distance (untrained)	0.700	0.945	**0.784**
Hybrid (λ fit on train)	0.702	0.945	0.749
LoRA, binary labels	0.736	0.905	0.524
LoRA, calibrated labels	**0.745**	**0.958**	0.573


libero_spatial
libero_object
libero_goal
libero_10
libero_10_trunc
metaworld
libero_10_r2
pooled_held_out
S-vs-comparison AUROC
Predicate-boundary multiple
0.0
0.2
0.4
0.6
0.8
1.0
0.0
1.0
2.5
4.0
topreward_q3vl8
q3vl8_binary
7b_binary
7b_rubric
pixtral_rubric
dinov2_crop50
siglip_goal
roboreward8b_ours
repair_base_binary
repair_base_calibrated
repair_tuned_binary
repair_tuned_calibrated
S vs F
S vs P_ok

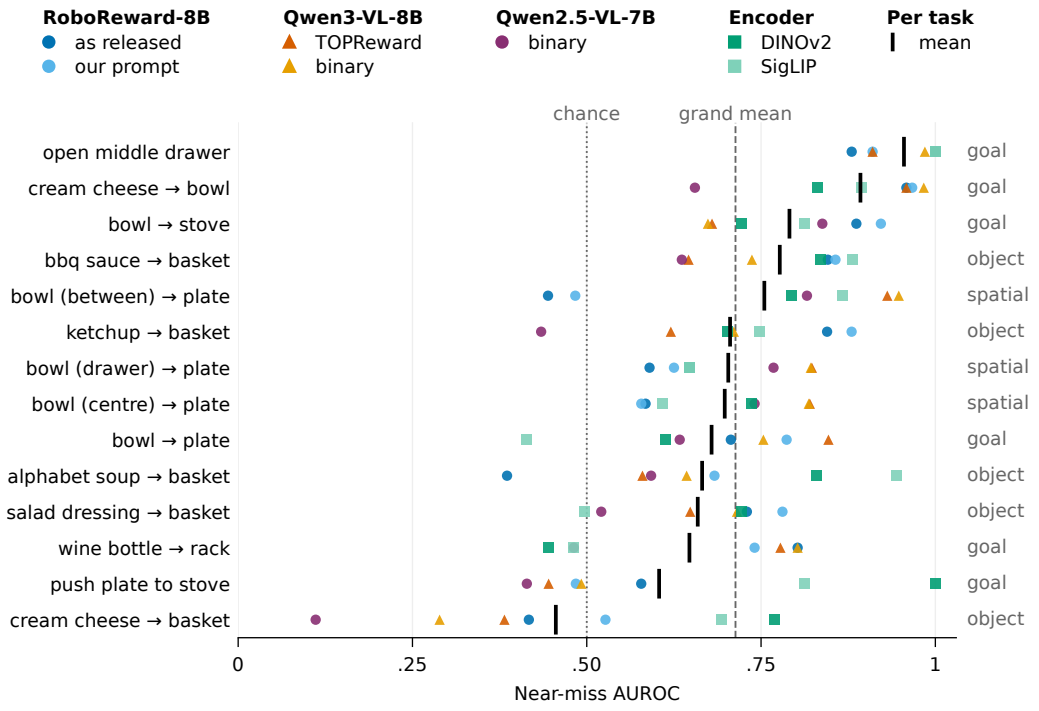


Figure 6: Near-miss AUROC of the seven judges of §6.1 on the 14 single-stage tasks with at least eight near-miss failures, ordered by task mean. Grey bars span the judges on each task, and the right column names the suite. The dashed line is the single number an aggregate benchmark reports.

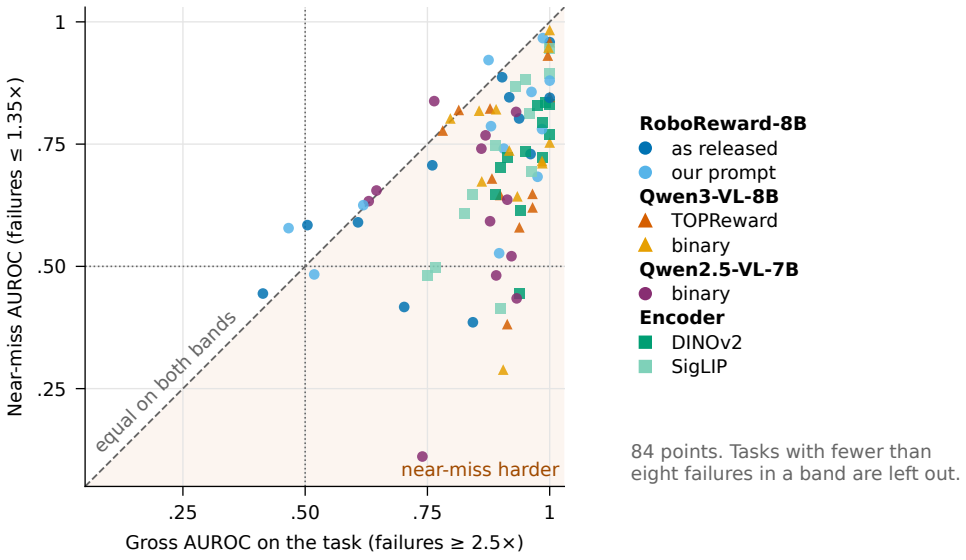


Figure 7: The grid of Figure 6 as gross against near-miss AUROC, one point per task and judge. Points below the diagonal are tasks where near-misses are harder than gross failures. Tasks with fewer than eight failures in either band are left out.

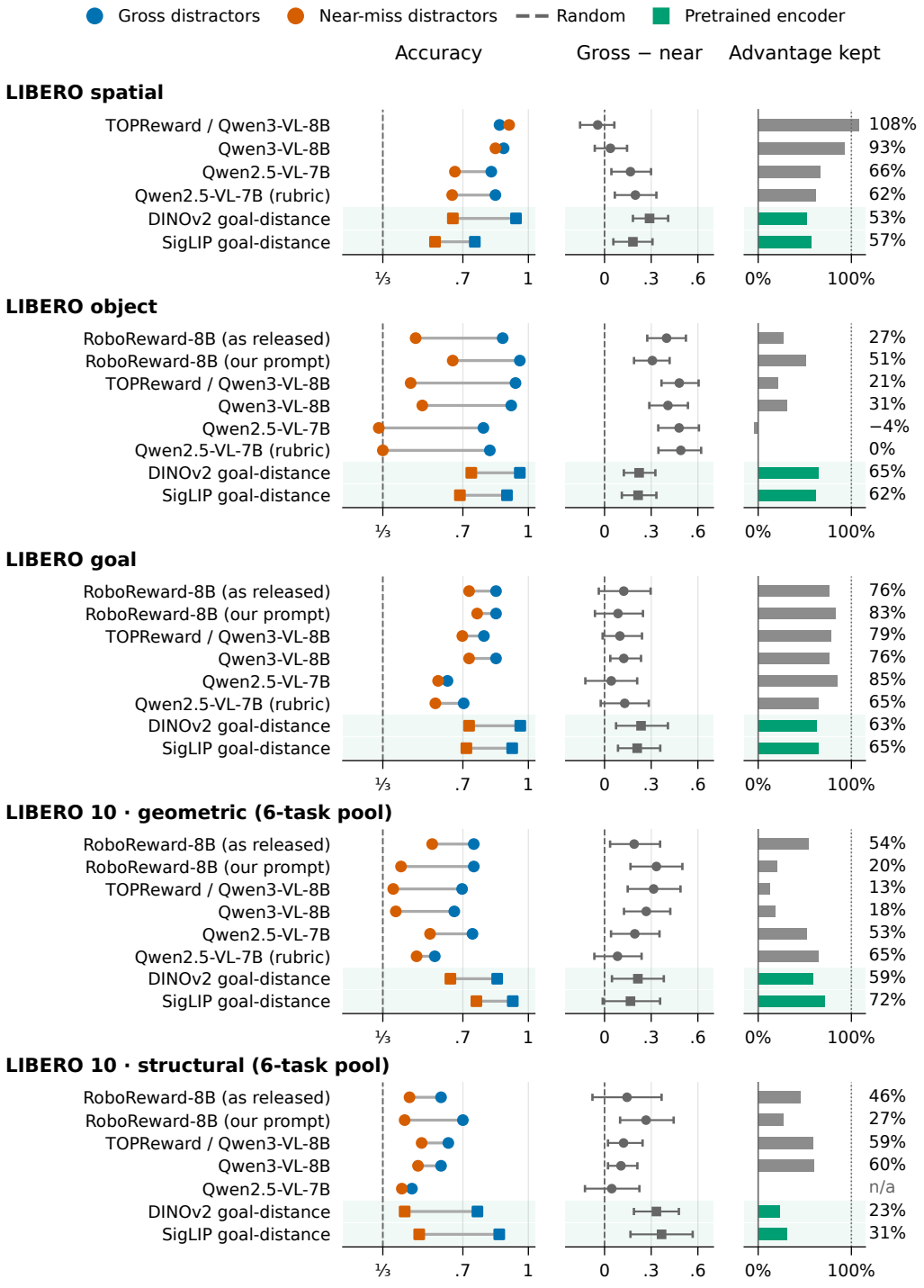
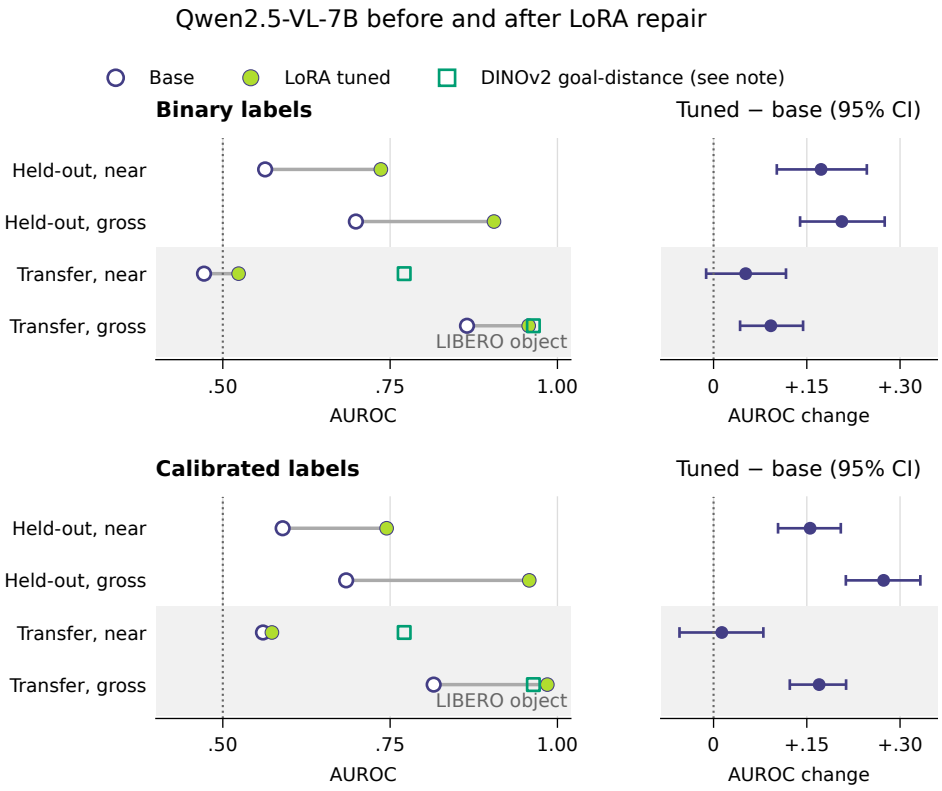


Figure 8: Best-of-3 selection with one verified success and two verified failures on every suite (§5). Gross distractors are failures at least $2.5\times$ the instance boundary and near-miss distractors are failures at most $1.35\times$; random selection is 0.333 with either. The middle column shows the gross-minus-near difference with paired bootstrap intervals, and the right column the fraction of the advantage over random that is kept. Squares mark pretrained encoders. Every configuration that passes the screen is shown, and the long-horizon panels use the six-task pilot pools.



Held-out pools the spatial, goal and long-horizon subsets. Shaded rows are the transfer suite.

Note. DINOv2 is a full-suite reference with different AUROC weighting, not a matched baseline.

Figure 9: The repair (§6.2) under binary and calibrated labels. Held-out tasks pool LIBERO-Spatial, LIBERO-Goal and LIBERO-Long, and the transfer suite is LIBERO-Object. Intervals are paired bootstrap intervals on the change from base to tuned. The DINOv2 goal-distance reference is computed on the full suite with different AUROC weighting, so it is not a matched held-out baseline.