BEYOND SUCCESSFUL ROLLOUTS: LEARNING FROM SIMULATOR NEAR-FAILURES

Anonymous authors
Paper under double-blind review

ABSTRACT

A correct answer to a manipulation question does not reveal whether a vision–language model checked the physical constraints that make a plan feasible. We study whether simulator-verified successes and near-failures can supply this missing supervision. Contrastive Embodied Reasoning (CER) scores intermediate reasoning representations against a fixed library of task-matched simulator observations, then combines the averaged contrastive score with a final-answer reward. On CRD-Bench, a 30-class manipulation question-answering diagnostic, CER improves high-complexity accuracy from 33.9% to 44.8% after exposure to verifier-correct model responses is matched (difference 10.9 points; 95% CI [8.3, 13.1]). A separate five-seed factorial with matched libraries and critic capacity finds a 2.53-point average advantage for InfoNCE over independent binary scoring (95% CI [0.5, 4.6]). Automatically mined near-failures approach manually designed ones, while irrelevant or corrupted negatives substantially reduce performance. The advantage over strengthened terminal training persists on 20 additional task classes, unseen layouts, and independently constructed distractors. In executable RLBench planning, success increases from 25.3% to 35.3%; non-contrastive process supervision achieves a comparable gain. These results support simulator state comparisons as supervision for manipulation reasoning and identify a smaller, distinct benefit from contrastive scoring.

1 INTRODUCTION

Consider a model answering a question about placing a bottle in a rack. Selecting the correct plan does not tell us whether it checked that the bottle was grasped before it moved, that its pose allowed insertion, or that the rack supported it afterward. A final-answer verifier rewards the selected option. A simulator can supply more specific evidence: states that satisfy these physical requirements and closely related states that fail them.

We study these comparisons in manipulation question answering with large vision–language models (LVLMs). Outcome-based reinforcement learning improves language-model reasoning (Shao et al., 2024) and embodied reasoning (Kim et al., 2025), but its effectiveness depends in part on sampling correct responses. In our hardest task classes, only 1.8% of responses receive a positive answer reward under the original terminal recipe. Better exploration substantially improves this baseline. The question we address is what simulator supervision adds once the model has access to correct responses.

Contrastive Embodied Reasoning (CER) compares reasoning-token hidden states with a fixed library of successful simulator observations and plausible near-failures. A learned critic scores their compatibility; the average log-contrastive score augments the final-answer reward. This is process supervision through representations, without labels for individual verbal steps or a prescribed reasoning trace. The answer verifier judges generated text; simulator executions supply the separate state library.

Success matching tests whether correct-response exposure accounts for CER's gain; an independent factorial tests joint normalization under matched simulator information. Negative-quality tests, independent task constructions, and representation interventions examine the supervision. Shared-controller planning tests transfer beyond answer selection, with similar execution gains from contrastive and non-contrastive process rewards.

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.

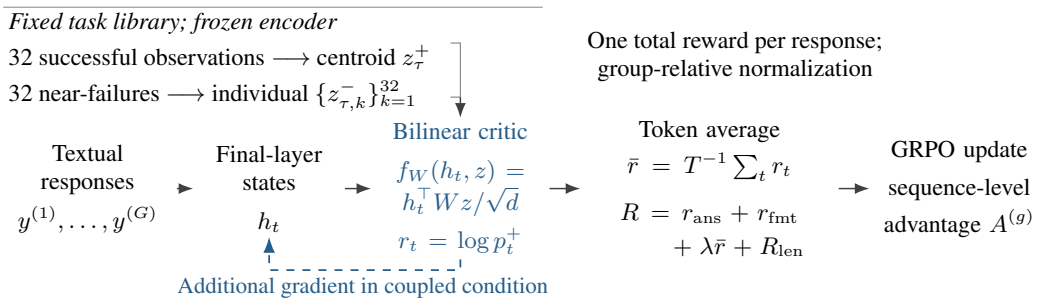

Figure 1: CER training. The critic compares reasoning-token states with a fixed success centroid and individual near-failures. Averaged token scores enter one response reward for GRPO; coupling adds a gradient through the reasoning states. The critic and library are used only during training.

2 Contrastive Embodied Reasoning

Task and supervision. A question q consists of an observation, a manipulation instruction, and four candidate plans or outcomes. The policy generates a textual response y containing reasoning and a final option. A binary verifier checks that option. Separately, for each training task τ, a simulator provides successful observations and physically plausible failures. The process reward compares the response's intermediate representations with these observations (Figure 1).

Fixed simulator library. We execute expert demonstrations in RLBench's PyRep/CoppeliaSim stack (James et al., 2020). A frozen LVLM vision–language encoder mean-pools the final tokens after multimodal fusion. We average 32 successful-observation embeddings into a positive centroid,

$$z_\tau^+ = \frac{1}{32}\sum_{j=1}^{32} \text{Enc}(x_{\tau,j}^+), \tag{1}$$

and retain $K = 32$ negative embeddings $z_{\tau,k}^-$ individually. Perturbations preserve task context while introducing contact offsets, unsupported or uncontained placements, infeasible poses, target occlusions, insufficient force, or swapped subgoal order. Simulator failure verification is followed by a 10% plausibility spot check. We test automatically mined alternatives and report a full-library audit in Appendix B. The encoded library remains fixed throughout training.

Contrastive score. Let $h_t \in \mathbb{R}^{d_h}$ be reasoning token t's layer-normalized final-layer state and $z \in \mathbb{R}^{d_z}$ an encoded simulator observation. A bilinear critic $W \in \mathbb{R}^{d_h \times d_z}$ defines

$$f_W(h, z) = h^\top W z/\sqrt{d}, \tag{2}$$

$$r_t = \log p_t^+ = \log \frac{\exp f_W(h_t, z_\tau^+)}{\exp f_W(h_t, z_\tau^+) + \sum_{k=1}^K \exp f_W(h_t, z_{\tau,k}^-)}, \qquad \bar{r} = \frac{1}{T}\sum_{t=1}^T r_t. \tag{3}$$

Here d is the score-scaling dimension and T is the reasoning length. The critic is initialized from scratch and learned during CER training. Joint normalization (van den Oord et al., 2018) rewards compatibility with success relative to the task's near-failures. Every reasoning token is scored against the same library: token t has no temporal correspondence with simulator step t. The target is success–failure compatibility, rather than reconstruction of a physical state or verification of a verbal step.

Training and inference. For each question, the policy samples G textual responses. Each receives one total reward,

$$R(y) = r_{\text{answer}}(y) + r_{\text{format}}(y) + \lambda\bar{r}(y) + R_{\text{len}}(y), \qquad \lambda = 0.6. \tag{4}$$

The answer reward checks the final option; the format reward is shared with terminal training. Under the reported log-score parameterization and threshold, $R_{\text{len}} = 0.2(512 - T)/512$, a linear brevity

reward (Appendix A). We normalize total rewards within each group and update the policy with GRPO and a reference-policy Kullback–Leibler penalty (Shao et al., 2024). The policy advantage is sequence-level, even though its process component is evaluated at each token.

Critic learning and reward coupling are separate choices. In the detached condition, the process score updates the policy solely through the response advantage. In the coupled condition, it also permits gradients through the reasoning states. The factorial in Section 4.2 matches critic training across these conditions. After training, evaluation uses the policy without the critic or library retrieval; no test-task success observations are needed.

Why near-failures matter. The contrastive score assigns greater weight to negatives that remain compatible with the reasoning state. A near-failure can therefore expose a distinction that an unrelated failed scene would not. This weighting also makes confusable mislabeled negatives harmful. Process scores can vary even when all sampled responses have the same binary answer reward, providing a training signal in tied groups. Appendix C gives the score-gradient and reward-centering identities; Sections 4.1 and 4.3 test the corresponding empirical questions.

3 Experimental setup

CRD-Bench. Five training tasks supply 7,500 four-choice question–answer pairs. The original test set contains 30 RLBench-derived task classes with identities disjoint from training. Distractors preserve the task context while violating a physical requirement. For example, a wine-rack question contrasts descend–grasp–lift–align–insert with alternatives that move before grasping or descend beyond a feasible limit.

Three robotics annotators label the required constraints: contact, support, containment, kinematic feasibility, occlusion, force balance, and temporal ordering. Majority labels define $C(\tau)$ and constraint count $\kappa(\tau) = |C(\tau)|$. The test set contains 5, 8, 9, 6, and 2 classes at $\kappa = 2, 3, 4, 5, 6$. We use $\kappa \geq 5$ for high-complexity evaluations. Constraint count correlates with Critical Reasoning Depth, the generation budget needed to cross a fixed above-chance accuracy threshold: Spearman $\rho = 0.82$, 0.77, and 0.72 across the three evaluated backbones. Appendix A defines this diagnostic; Appendix Figure 5 shows accuracy by constraint count.

Training and evaluation. The primary backbone is Qwen2.5-VL-7B-Instruct (Bai et al., 2025). The reference configuration uses five epochs, batch size 128, learning rate 10^{-6}, temperature 1.0, and five responses per prompt. We strengthen terminal training through larger groups, chain-of-thought and expert initialization, curriculum, and verifier search before comparing supervision methods. Development sweeps, success-matched confirmation, and the independent factorial are distinct experiments.

The original results and independent factorial use five seeds. Reported confidence intervals (CIs) are 95% paired hierarchical bootstrap intervals with task class as the top-level resampling unit. The 20-class expansion is selected from 26 candidates using prespecified ambiguity and simulator-stability criteria. Its membership, exclusions, and complexity labels are fixed before evaluation, and robustness sets are excluded from checkpoint selection.

4 Results

4.1 Correct-response exposure

The original terminal recipe produces positive answer rewards on 42.0% of responses at $\kappa = 2$, 6.5% at $\kappa = 5$, and 1.8% at $\kappa = 6$. Increasing group size from 5 to 32 raises high-complexity development accuracy from 16.9% to 26.7%. Curriculum reaches 31.0%, expert initialization 31.8%, and the combined terminal recipe 34.0% (Appendix B). An independent confirmation reaches 32.8%. The weak original terminal result thus reflects a substantial exploration bottleneck.

In a separate comparison that matches exposure to verifier-positive responses, terminal training reaches 33.9% and CER 44.8% (Figure 2a). The residual difference is 10.9 points (CI [8.3, 13.1]): correct-response exposure is insufficient to reproduce the full CER result. This comparison tests the

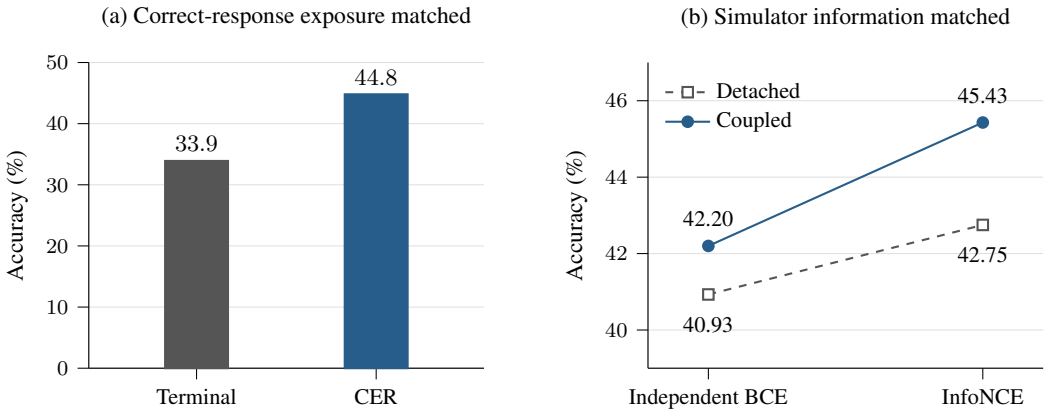

Figure 2: Independent controls. (a) Matching verifier-correct response exposure leaves a CER–terminal difference of 10.9 points (95% CI [8.3, 13.1]). (b) With simulator information matched, the InfoNCE main effect is 2.53 points (95% CI [0.5, 4.6]); its coupled contrast is 3.23 [0.8, 5.6]. Marks show cell means; intervals describe contrasts.

Table 1: High-complexity development accuracy. Each method below the separator receives the same simulator successes and failures. The terminal row is the outcome-only reference. Means are from the development sweep, distinct from both controls in Figure 2.

Supervision	Accuracy (%)
Strengthened terminal	34.0
Hard-negative SFT	31.7
Success/failure DPO	36.8
Constraint auxiliary	39.5
Dense process regression	41.1
Learned process reward	42.3
Dense reward + length term	42.9
CER without length term	44.0
Full CER	45.8

full training recipe, including its process score and length term. It does not isolate those components. Reward-switch experiments give 39.6% for terminal-to-CER and 35.6% for CER-to-terminal (difference 4.0, CI [1.6, 6.4]), supporting the value of maintaining CER supervision later in training.

4.2 Matched simulator information

We compare CER with hard-negative supervised fine-tuning, success/failure direct preference optimization (DPO; Rafailov et al., 2023), constraint classification, dense process regression, and learned process rewards using the same library. Process-supervised alternatives substantially improve on strengthened terminal training (Table 1). Dense reward with the same length term reaches 42.9%, compared with CER's 45.8%; removing CER's length term gives 44.0%. The development sweep supports the usefulness of the simulator evidence across several scoring rules.

An independent 2×2 factorial isolates scoring and coupling (Figure 2b). Independent BCE and jointly normalized InfoNCE share the positive centroid, 32 negatives, bilinear critic, initialization, critic-training schedule, process coefficient, compute, and validation budget. InfoNCE improves accuracy by 1.82 points with detached rewards and 3.23 with coupled rewards. Its main effect is 2.53 points (CI [0.5, 4.6]); the coupled contrast has CI [0.8, 5.6] and is positive on all eight original high-complexity classes. The coupling main effect is 1.98 points (CI [0.0, 3.9]), while the 1.41-point interaction is inconclusive. Contrastive normalization adds a measurable benefit under matched information, smaller than the broader improvement associated with process supervision.

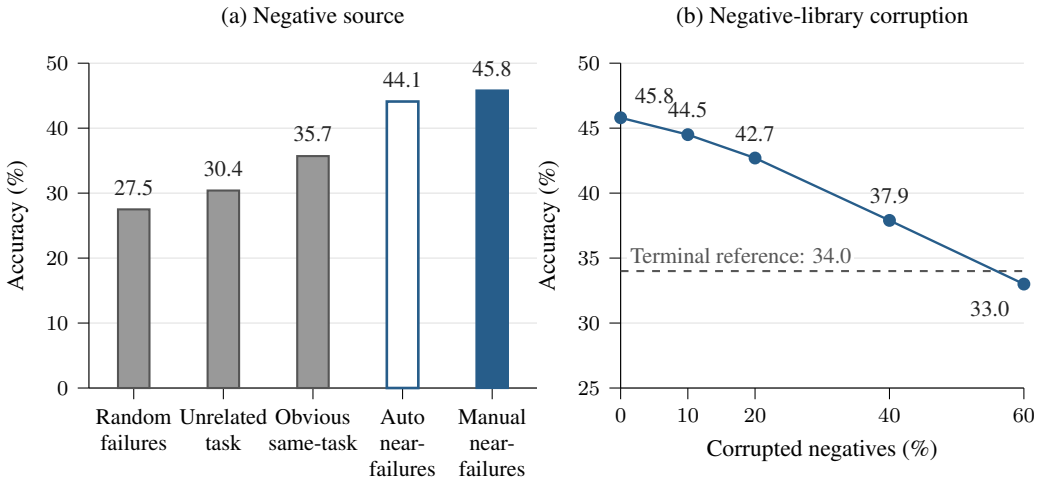

Figure 3: Negative-library quality. (a) Near-failures outperform random, unrelated, and obvious failures. (b) Corruption progressively reduces accuracy; the dashed line is strengthened terminal training (34.0%). These separate sweeps report means without individual uncertainty.

Table 2: Frozen-checkpoint generalization against strengthened terminal training. Each row is a separate evaluation; gains are accuracy percentage points with task-hierarchical 95% CIs.

Evaluation	CER − terminal	95% CI
20 new high-complexity classes	9.6	[6.2, 13.0]
Unseen layouts	9.3	[5.8, 12.8]
Unseen constraint combinations	10.1	[6.9, 13.3]
Independent distractors	10.7	[7.9, 13.5]

4.3 Negative-library quality

Obvious same-task failures yield 35.7% accuracy, versus 44.1% for automatically mined near-failures and 45.8% for manually designed ones (Figure 3a). Random and unrelated negatives perform worse. Positive alignment also strongly affects performance: shuffled positives give 15.8%, and wrong-task positives matched by constraint count give 19.7%. Physically relevant within-task comparisons perform best.

Corrupting 0%, 10%, 20%, 40%, and 60% of negatives reduces accuracy from 45.8% to 44.5%, 42.7%, 37.9%, and 33.0%, respectively (Figure 3b). At the highest tested corruption, CER falls below the strengthened terminal reference. This is consistent with the score's emphasis on confusable alternatives: incorrectly labeled near-successes can receive substantial weight. The audit of all 160 stored negatives confirms that the supervision is imperfect (Appendix B).

4.4 Generalization

The original constraint taxonomy also informs distractor construction, so evaluations using other task and distractor procedures are essential. CER's advantage persists on the 20-class expansion, unseen layouts, and unseen constraint combinations (Table 2). Across five independently constructed plausible-distractor sets, gains range from 7.6 to 11.1 points; easy random alternatives give 1.1 (Appendix Figure 5b). The contrast remains after normalizing by terminal error: relative error reductions are 11.4–16.0% for plausible failures and 3.2% for random alternatives.

Withholding the target constraint type from training negatives gives 36.9%, compared with terminal training's 29.4% and full CER's 43.5%. Exact exposure to that constraint helps but does not explain the entire gain. Qwen2.5-VL-3B improves from 25.8% to 36.9%, and LLaVA-NeXT-7B from 30.8%

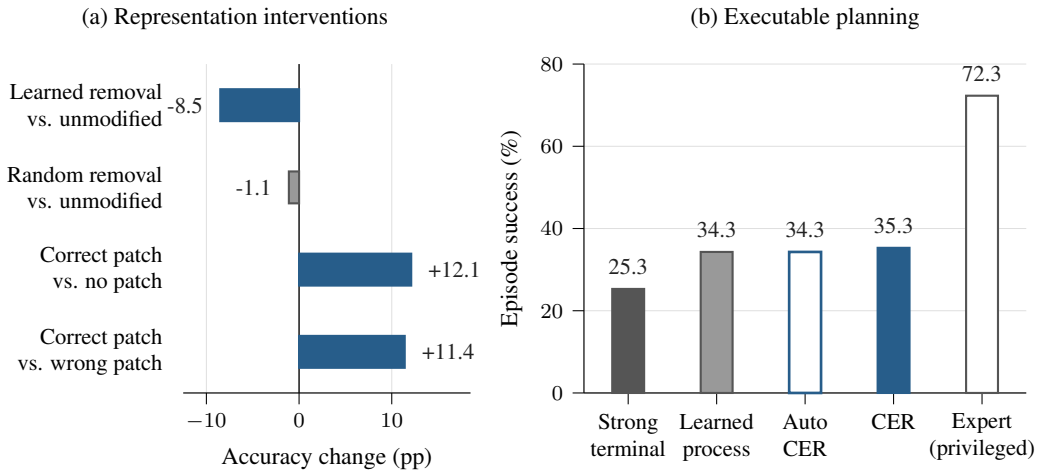

Figure 4: Representations and execution. (a) Learned-constraint and rank-matched random subspace removals are relative to unmodified states. Correct-constraint patching uses the two stated controls. (b) RLBench success on five held-out tasks, with 300 episodes per learned planner and a shared controller. Expert waypoints are privileged. Bars show means; the CER–terminal difference is 10.0 points (95% CI [4.2, 15.8]).

to 40.7%. These tests extend the evidence across the evaluated constructions and backbones, while leaving broader task-family generalization open.

5 Representations and Executed Plans

Constraint representations. A seven-way linear probe on frozen pre-answer final-layer states measures accessible constraint information. Probe training uses held-out generations from the five training-task identities; evaluation uses disjoint CRD-Bench classes. Accuracy is 37.8% for the original terminal checkpoint and 67.3% for CER, against 14.3% chance. On each model's own incorrect-answer subset, the corresponding values are 33.5% and 52.4%. In a separate audit of 300 correct high-complexity predictions per model, 34.2% of terminal rationales and 72.6% of CER rationales name and apply the violated physical relations when rejecting distractors. CER does not consistently articulate these relations, even when its final answer is correct.

Interventions test whether the selected representations affect predictions (Figure 4a). Removing the learned constraint subspace lowers accuracy by 8.5 points; a rank-matched random subspace lowers it by 1.1. Correct-constraint patching improves accuracy by 12.1 points over no patch and 11.4 over wrong-constraint patching. Same-answer/wrong-constraint and same-constraint/wrong-answer controls do not reproduce the effect. These test-time, norm-preserving interventions support functional involvement of the selected representations, without identifying the model's complete causal computation.

Generation budget. CER reaches $36.9 \pm 1.6\%$ high-complexity accuracy at 128 tokens, exceeding the original terminal policy's $33.0 \pm 1.5\%$ at 256 tokens. CER at 256 tokens reaches $43.7 \pm 1.4\%$; filler and physical-checklist controls give $20.6 \pm 1.3\%$ and $31.4 \pm 1.6\%$. These comparisons use the original terminal recipe and standard errors over seeds. More generation alone does not reproduce CER's performance under these conditions.

Executed plans. The LVLM generates a hierarchical plan once; a shared fixed low-level controller executes it in closed loop. Native RLBench episode success rises from 25.3% for strengthened terminal training to 35.3% for CER (difference 10.0 points, CI [4.2, 15.8]; Figure 4b). Learned process reward and automatically mined CER each reach 34.3%. Recovery success increases from 28.5% to 48.1%, and infeasible actions fall from 22.1% to 13.5%. The execution result supports a benefit from process supervision; it does not distinguish CER from the strongest process baseline.

A four-task UR5 pilot yields 27/80 successes for CER and 18/80 for strengthened terminal training. This small pilot is exploratory. Against the reference terminal policy, EmbodiedBench manipulation improves from 11.5% to 18.9%, and SpatialRGPT qualitative/quantitative accuracy from 40.5%/15.8% to 52.7%/25.4% (Yang et al., 2025; Cheng et al., 2024). These external evaluations do not include the strongest process baselines.

6 Related work

Reasoning supervision. Human step labels improve mathematical verification (Lightman et al., 2024); Math-Shepherd constructs such labels automatically for verification and reinforcement learning (Wang et al., 2024); learned step rewards also guide inference-time search (Ma et al., 2023). SpatialReasoner-R1 uses segment-level preferences for spatial grounding and logical reasoning in vision–language models (Shen et al., 2025). CER instead scores hidden reasoning states against fixed simulator observations. Its supervision consists of task-matched physical successes and failures, without preferred verbal steps.

Contrastive and embodied learning. Contrastive predictive coding learns relative compatibility (van den Oord et al., 2018); contrastive goal-conditioned RL compares current and future states (Eysenbach et al., 2022). CER compares language-model reasoning states with simulator observations. Robot-R1 studies outcome-based embodied reasoning (Kim et al., 2025); step-aware contrastive navigation uses landmark auditing, valid-prefix supervision, and divergence corrections (Li et al., 2026). We study manipulation-state comparisons while controlling response exposure, simulator information, and coupling.

Reasoning budgets. Chain-of-thought prompting demonstrates the value of intermediate computation (Wei et al., 2022), while R1-Zero analyses show that policy estimators can induce length biases (Liu et al., 2025). CRD records an empirical accuracy threshold under a fixed generation protocol, rather than a number of logical deductions.

7 Limitations

CER requires hidden-state access and a simulator capable of producing credible physical comparisons. Systematic errors in contact, friction, or compliance can teach incorrect distinctions; random corruption does not model those errors. A single success centroid can also obscure multiple valid solutions. The token-level score measures compatibility with the library, not the correctness of each reasoning step.

CRD-Bench is four-choice, and its original $\kappa = 6$ stratum contains only two classes. The expansion and alternative distractors broaden the evaluation within simulator-derived manipulation families, but do not make the constraint taxonomy exhaustive. Planning relies on a fixed low-level controller; the robot pilot is small; and external benchmarks lack the strongest process baselines. The representation interventions do not establish that the measured probe improvement mediates the execution gain.

8 Conclusion

Fixed simulator successes and near-failures improve manipulation question answering beyond matched exposure to correct model responses. A separate controlled factorial supports an advantage for contrastive scoring, while execution tests show a broader benefit shared with non-contrastive process supervision. The quality of the physical comparisons is central: automatic mining retains most of CER's performance, whereas irrelevant or corrupted negatives sharply reduce it.

AI USE STATEMENT

Generative AI tools assisted manuscript organization and editing, literature checking, interpretation of existing results, formulation and verification of the analytical identities, and preparation of figures and analysis scripts. The empirical measurements come from the authors' experiments; the AI-assisted revision did not generate additional training, simulator, or robot results. The authors take responsibility for the manuscript and its AI-assisted content.

REPRODUCIBILITY STATEMENT

Sections 2 and 3 specify the benchmark, reward, and experimental comparisons. Appendix A gives the training hyperparameters and evaluation conventions; Appendix B provides additional results. Appendix C states the assumptions and derivations for the analytical claims. The accompanying manuscript-source package contains the exact figure data and scripts for the descriptive calculations and algebraic checks.

REFERENCES

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL Technical Report. *arXiv preprint arXiv:2502.13923*, 2025. doi: 10.48550/arXiv.2502.13923. URL https://arxiv.org/abs/2502.13923.

An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. SpatialRGPT: Grounded Spatial Reasoning in Vision-Language Models. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/f38cb4cf9a5eaa92b3cfa481832719c6-Abstract-Conference.html.

Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Ruslan Salakhutdinov. Contrastive Learning as Goal-Conditioned Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/e7663e974c4ee7a2b475a4775201ce1f-Abstract-Conference.html.

Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J. Davison. RLBench: The Robot Learning Benchmark & Learning Environment. *IEEE Robotics and Automation Letters*, 2020. URL https://arxiv.org/abs/1909.12271.

Dongyoung Kim, Sumin Park, Huiwon Jang, Jinwoo Shin, Jaehyung Kim, and Younggyo Seo. Robot-R1: Reinforcement Learning for Enhanced Embodied Reasoning in Robotics. *arXiv preprint arXiv:2506.00070*, 2025. doi: 10.48550/arXiv.2506.00070. URL https://arxiv.org/abs/2506.00070.

Haoyuan Li, Rui Liu, Hehe Fan, and Yi Yang. Let's Reward Step-by-Step: Step-Aware Contrastive Alignment for Vision-Language Navigation in Continuous Environments. *arXiv preprint arXiv:2603.09740*, 2026. doi: 10.48550/arXiv.2603.09740. URL https://arxiv.org/abs/2603.09740.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's Verify Step by Step. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/aca97732e30bcf1303bc22ac3924fd16-Abstract-Conference.html.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding R1-Zero-Like Training: A Critical Perspective. *arXiv preprint arXiv:2503.20783*, 2025. doi: 10.48550/arXiv.2503.20783. URL https://arxiv.org/abs/2503.20783.

Qianli Ma, Haotian Zhou, Tingkai Liu, Jianbo Yuan, Pengfei Liu, Yang You, and Hongxia Yang. Let's reward step by step: Step-Level reward model as the Navigators for Reasoning. *arXiv preprint arXiv:2310.10080*, 2023. doi: 10.48550/arXiv.2310.10080. URL https://arxiv.org/abs/2310.10080.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/a85b405ed65c6477a4fe8302b5e06ce7-Abstract-Conference.html.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint arXiv:2402.03300*, 2024. doi: 10.48550/arXiv.2402.03300. URL https://arxiv.org/abs/2402.03300.

Yifan Shen, Yuanzhe Liu, Jingyuan Zhu, Xu Cao, Xiaofeng Zhang, Yixiao He, Wenming Ye, James M. Rehg, and Ismini Lourentzou. Fine-Grained Preference Optimization Improves Spatial Reasoning in VLMs. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/1a17a06de88cf77f25cda0da91615a54-Abstract-Conference.html.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding. *arXiv preprint arXiv:1807.03748*, 2018. doi: 10.48550/arXiv.1807.03748. URL https://arxiv.org/abs/1807.03748.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-Shepherd: Verify and Reinforce LLMs Step-by-step without Human Annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9426–9439, 2024. doi: 10.18653/v1/2024.acl-long.510. URL https://aclanthology.org/2024.acl-long.510/.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

Rui Yang, Hanyang Chen, Junyu Zhang, Mark Zhao, Cheng Qian, Kangrui Wang, Qineng Wang, Teja Venkat Koripella, Marziyeh Movahedi, Manling Li, Heng Ji, Huan Zhang, and Tong Zhang. EmbodiedBench: Comprehensive Benchmarking Multi-modal Large Language Models for Vision-Driven Embodied Agents. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 70576–70631, 2025. URL https://proceedings.mlr.press/v267/yang25f.html.

A METHOD SPECIFICATION AND DIAGNOSTIC CONVENTIONS

The primary setup uses Qwen2.5-VL-7B-Instruct, five training tasks, 7,500 question–answer pairs, five epochs, batch size 128, learning rate 10^{-6}, temperature 1.0, and five rollouts per query. Larger-group controls use 16 or 32 rollouts. Success centroids average 32 encoded observations per task; each task has 32 individual negative embeddings. Libraries remain fixed. The full CER coefficient is $\lambda = 0.6$. State-library generation takes approximately three hours. The original CER training run uses four A100 80GB GPUs for approximately 18 hours, compared with approximately 12 hours for terminal GRPO.

Length regularization. The full configuration uses the length regularizer

$$R_{\text{len}} = \gamma \, \mathbf{1}[r_T < \vartheta] \frac{L_{\text{target}} - T}{L_{\text{target}}}, \qquad \gamma = 0.2, \quad \vartheta = 0.1, \quad L_{\text{target}} = 512. \tag{5}$$

Because $r_T \leq 0$, the indicator is active under this log-probability parameterization, and Equation 5 is equivalent to a linear brevity reward. The no-length-term condition reaches 44.0%; dense process reward with the same length term reaches 42.9%, compared with 45.8% for full CER. These controls separate the length term from the process score.

Gradient conditions. The detached condition uses process scores in the group-relative rollout reward without a direct score gradient into policy states. The coupled condition permits an additional representation-dependent gradient. The factorial holds the library, critic capacity, initialization, process coefficient, compute, and validation budget fixed. Both conditions use sequence-level group-relative advantages. Coupling adds the representation gradient without changing the simulator evidence.

Critical Reasoning Depth. Let $a_{m,D}(L, \tau)$ denote the accuracy of checkpoint m under decoding protocol D with token cap L. We define the above-chance budget threshold

$$L^*_{m,D}(\tau) = \inf\{L : a_{m,D}(L, \tau) \geq 0.25 + \delta\}, \qquad \delta = 0.10. \tag{6}$$

CRD is specific to the checkpoint and decoding protocol. We estimate first crossings on a fixed budget grid, using log-budget interpolation and right-censoring at 256 tokens. The budget protocol is given below. CRD measures an empirical accuracy threshold rather than a number of logical deductions.

Budget measurement. The primary truncation control imposes tokenizer-level `max_new_tokens` during generation at budgets $\{2, 4, 8, 16, 32, 64, 128, 256\}$. Missing or malformed answers count as incorrect. The threshold is $0.25 + \delta = 0.35$, with $\delta = 0.10$ above four-choice chance. The CRD definition uses the first threshold crossing and the convention $\inf \varnothing = +\infty$. Between tested budgets, first crossings are interpolated in log-budget space; tasks that do not cross are reported as right-censored at > 256. A separate format control reserves an answer head outside the scratchpad budget and preserves the qualitative dependence on budget.

B ADDITIONAL NUMERICAL RESULTS

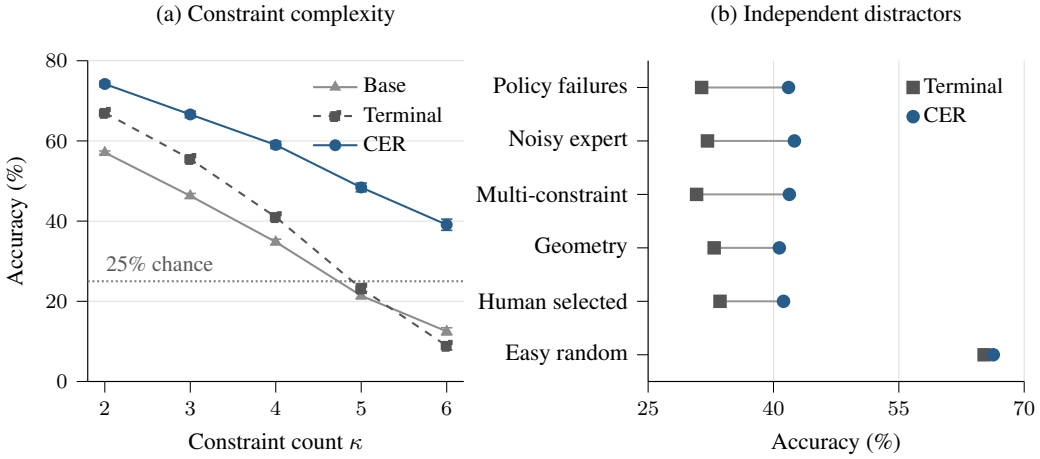

Figure 5: Diagnostic complexity and independently constructed distractors. (a) Original training configurations, with standard errors over five seeds. The two highest-complexity strata contain eight task classes. (b) Strengthened terminal and CER means for independent distractor constructions; lines connect the two means for each construction. These are separate evaluations; individual construction means have no reported uncertainty.

Table 3: Original per-complexity accuracy, mean ± reported standard error over five seeds. These are the original training configurations, not the strengthened or success-matched conditions. Results are stratified by physical constraint count.

Model	$\kappa = 2$	$\kappa = 3$	$\kappa = 4$	$\kappa = 5$	$\kappa = 6$
Base	57.1 ± 0.4	46.3 ± 0.6	34.8 ± 0.7	21.4 ± 0.8	12.5 ± 0.9
Direct SFT	41.5 ± 1.1	29.0 ± 1.3	12.7 ± 1.0	4.5 ± 0.8	0.8 ± 0.5
CoT SFT	53.7 ± 0.9	42.5 ± 1.0	29.1 ± 1.2	14.9 ± 1.0	6.7 ± 0.8
Terminal	67.0 ± 0.8	55.5 ± 0.9	41.0 ± 1.0	23.2 ± 1.2	8.9 ± 1.0
CER	74.2 ± 0.7	66.6 ± 0.8	59.0 ± 0.9	48.4 ± 1.1	39.1 ± 1.4

Table 4: Outcome-only exploration controls on original high-complexity tasks. These development conditions change rollout generation, initialization, or search. Best-of-32 also uses additional inference-time verifier access.

Condition	Accuracy (%)
Original terminal, $G = 5$	16.9
Larger groups, $G = 16$ / $G = 32$	24.0 / 26.7
CoT initialization, $G = 16$	29.1
Curriculum, $G = 16$	31.0
Expert initialization, $G = 16$	31.8
Best-of-32 verifier search	32.8
Combined strengthened terminal	34.0
CER development reference	45.8

Table 5: Independent five-seed factorial. The displayed cell means reproduce the reported main effects after rounding.

Objective	Detached	Coupled
Independent BCE	40.93	42.20
InfoNCE	42.75	45.43

The contrastive main effect from these means is $(1.82 + 3.23)/2 = 2.525$ points; the coupling main effect is $(1.27 + 2.68)/2 = 1.975$. The interaction is $3.23 - 1.82 = 1.41$ points and is reported as inconclusive. Confidence intervals for the main effects are given in Section 4.2.

Table 6: Independent test-distractor constructions. Relative error reduction is computed from each row's mean accuracies using Equation 11.

Construction	Terminal	CER	Gain (pp)	Error reduction
Policy-generated failure	31.4	41.8	10.4	15.2%
Noisy expert failure	32.1	42.5	10.4	15.3%
Multi-constraint failure	30.8	41.9	11.1	16.0%
Geometry perturbation	32.9	40.7	7.8	11.6%
Human-selected alternative	33.6	41.2	7.6	11.4%
Easy random alternative	65.2	66.3	1.1	3.2%

The aggregate gain across independently constructed plausible distractors is 10.7 points (CI [7.9, 13.5]); Table 6 gives the results by construction type.

Negative-library audit. The audit covers all 160 stored negatives and reports 98.8% simulator-valid failures, 95.6% intended constraint violations, 92.4% plausible near-successes, 7.2% major unintended secondary violations, and 4.1% ambiguous states; reported agreement is 0.81–0.87. These summaries indicate imperfect supervision and motivate the corruption test.

Additional execution evidence. Under post-plan perturbations across the seven constraint categories, macro-average frozen-plan success is 20.3% for terminal training, 30.9% for CER, and 32.5% for process supervision. These results further support the broader process-supervision benefit. The UR5 pilot uses four tasks, randomized object positions, and a fixed perception/control stack; its small size limits conclusions about physical generalization.

SFT sensitivity. The original direct-SFT configuration is weak, particularly at $\kappa = 6$. One epoch, lower learning rate, KL regularization, and base-data replay improve the reported overall/$\kappa = 6$ scores to 26.4/4.1, 28.7/5.9, 31.2/7.8, and 32.5/9.1, respectively. These results indicate substantial sensitivity to fine-tuning choices on the narrow training distribution.

C Analysis of the contrastive reward and aggregate results

Joint relative margins. Write $\Delta_{t,k} = f_W(h_t, z_\tau^+) - f_W(h_t, z_{\tau,k}^-)$. Then

$$r_t = -\log\left(1 + \sum_k e^{-\Delta_{t,k}}\right), \qquad \nabla_{h_t} r_t = \frac{1}{\sqrt{d}} \sum_k p_{t,k}^- W(z_\tau^+ - z_{\tau,k}^-), \tag{7}$$

where $p_{t,k}^-$ is negative k's probability in the same $(K+1)$-way softmax. The derivative holds W and the library fixed.

Writing $f^+ = f_W(h_t, z^+)$ and $f_k^- = f_W(h_t, z_k^-)$, division by e^{f^+} in Equation 3 gives the first identity in Equation 7. Differentiating yields

$$\frac{\partial r_t}{\partial f_k^-} = -p_{t,k}^-, \qquad \frac{\partial r_t}{\partial f^+} = 1 - p_t^+ = \sum_k p_{t,k}^-.$$

The chain rule with a fixed bilinear critic gives the hidden-state gradient in Equation 7. Adding a common scalar to every compatibility score leaves the reward unchanged. These properties explain the score's emphasis on relative, confusable alternatives. They do not imply that all high-scoring failures are physically informative or that a detached policy receives the direct gradient.

A finite-library discrimination bound. Let e_0 be the fraction of reasoning states for which some library negative scores at least as highly as the positive. Then

$$e_0 = \frac{1}{T} \sum_{t=1}^{T} \mathbf{1}\left[\min_k \Delta_{t,k} \le 0\right] \;\le\; \min\left\{1, \frac{-\bar{r}}{\log 2}\right\}. \tag{8}$$

Each misranked state incurs contrastive loss at least $\log 2$; averaging gives the bound. More generally, the loss controls violations of any fixed margin.

Let $\ell_t = -r_t = \log(1 + \sum_k e^{-\Delta_{t,k}})$ and $\bar{\ell} = T^{-1} \sum_t \ell_t = -\bar{r}$. For any finite margin m, define

$$e_m = \frac{1}{T} \sum_{t=1}^{T} \mathbf{1}\left[\min_k \Delta_{t,k} \le m\right].$$

If token t is counted in e_m, at least one margin satisfies $\Delta_{t,k} \le m$, so $\ell_t \ge \log(1 + e^{-m})$. All other token losses are nonnegative. Summing yields

$$e_m \le \min\left\{1, \frac{\bar{\ell}}{\log(1 + e^{-m})}\right\}.$$

Setting $m = 0$ gives Equation 8, including ties as errors. For $m > 0$, the bound controls the fraction of reasoning states that fail to separate every stored negative by more than m. It applies to the raw averaged log-contrastive score, not a group-normalized advantage. It does not certify negative-label correctness, physical validity outside the library, or downstream task success. We do not instantiate an empirical error bound because token-level score measurements are not reported. The reproduction script checks the inequality on explicitly illustrative margin arrays.

Tied outcome rewards. If answer, format, and length terms are constant within a group of G responses, centering Equation 4 gives

$$R_i - \overline{R} = \lambda(\bar{r}_i - \bar{\overline{r}}), \qquad \overline{R} = G^{-1} \sum_i R_i, \quad \bar{\overline{r}} = G^{-1} \sum_i \bar{r}_i. \tag{9}$$

Distinct process scores can order responses even if all have an incorrect final answer. This identity concerns sequence-level advantages; it does not require token-specific policy advantages.

Informative outcome groups. Under independent Bernoulli sampling with fixed correctness probability p, a group contains both successful and failed answers with probability

$$P_{\text{mixed}}(p, G) = 1 - (1-p)^G - p^G. \tag{10}$$

Using the original-policy rate $p = 0.018$ gives 8.68%, 25.22%, and 44.08% for $G = 5, 16, 32$. These are analytic fixed-policy illustrations, not measured group statistics for the trained larger-group policies. A group with identical binary outcomes supplies no centered answer-reward contrast; other reward terms can still vary. Equation 9 shows how a process score supplies a relative ordering in this case.

If success probabilities vary by query and samples are conditionally independent, the relevant quantity is $\mathbb{E}_q[P_{\text{mixed}}(p_q, G)]$. For $G \geq 2$,

$$\frac{\partial^2 P_{\text{mixed}}}{\partial p^2} = -G(G-1)\big[(1-p)^{G-2} + p^{G-2}\big] \leq 0.$$

Jensen's inequality gives $\mathbb{E}_q[P_{\text{mixed}}(p_q, G)] \leq P_{\text{mixed}}(\mathbb{E}_q[p_q], G)$. Under these assumptions and consistent query weighting, heterogeneity makes the class-average plug-in value an upper bound. Sampling dependence can invalidate this illustration.

A descriptive headroom check. For two accuracies measured in percent on the same distractor construction, define

$$\text{RER} = \frac{a_{\text{CER}} - a_{\text{terminal}}}{100 - a_{\text{terminal}}}. \tag{11}$$

Table 6 reports this arithmetic normalization. The smaller gain for easy random alternatives persists after normalizing by the terminal model's remaining error. This check does not match task difficulty, establish paired significance, or permit aggregation across construction types without weights.

Interpretation. The margin identities describe the score geometry. The mixed-group calculation assumes conditionally independent sampling at fixed success probability, and the relative error reductions are descriptive transformations of the displayed accuracies. These calculations do not replace empirical measurements of training dynamics.