Falsifying Simulator Safety Claims with Action-Cost Baselines

Joy Zheyun Yang*
Department of Computer Science
University of Oxford
Oxford, United Kingdom
Email: joy@robots.ox.ac.uk

Socrates Osorio*
Electrical Engineering and Computer Sciences
University of California, Berkeley
Berkeley, CA, USA
Email: socratesj.osorio@berkeley.edu

***Abstract*—Safety for embodied foundation models is established not only by design-time guarantees but also by *safety by practice*, which means deployment-time evaluation that can falsify a safety claim before the policy is trusted. We argue that simulator success is one such claim, and that embodied safety benchmarks routinely accept it too cheaply. A simulator-trained policy can solve the source task with high-action, transient-contact, or narrow-physics behaviors that would not support a safety case. If a benchmark compares a new safety mechanism only against weak baselines, it will credit the mechanism anyway. We make this concrete and propose a remedy. A benchmark must rule out ordinary action-cost baselines before crediting a learned safety method. We study compact PPO policies on ManiSkill3 PushCube, PullCube, and PokeCube under action, material-friction, contact-restitution, and dynamic-geometry shifts. The central finding is a conservative baseline result. Ordinary action costs are stronger robustness comparators than vanilla PPO, action-delta smoothness, or action-noise domain randomization. Squared action energy reaches 66.2% shifted success on PushCube. Action L1 reaches the highest central estimates on PullCube and PokeCube, although these two tasks carry high seed-to-seed variance that we foreground rather than hide. Under stricter replay, energy wins 5 of 6 PushCube target families and L1 wins 5 of 6 PullCube, while on PokeCube the action-norm cost and L1 each win 3 of 6 and no cost form dominates. A frozen-CLIP visual scorer catches a constructed source-reward exploit and helps in some early-budget selection regimes, but it does not explain the completed ManiSkill gains. Our contribution to embodied-safety benchmarking is a falsification checklist. Source reward, held-out shifts, uncertainty, effort-cost baselines, failure cases, and audit signals must be separated and reported before a method is credited as "safe."**

I. Introduction

Trustworthy embodied foundation models can be pursued along two complementary axes. *Safety by design* reduces risk at the source through training objectives and architectural constraints. *Safety by practice* evaluates and mitigates risk at deployment time. This paper is a safety-by-practice contribution. We argue that a key deployment-time guardrail for embodied policies is the ability to *falsify* a safety claim before trusting it, and that current embodied-safety benchmarking falsifies too little. A household robot that briefly bumps an object into a goal, swings a tool near a person, or uses excessive actuator effort can satisfy a local reward while violating expectations about context, intent, and risk. This problem is acute as robot learning moves toward broad data and generalist policies [4, 3, 11, 10, 5], because simulator reinforcement learning can discover brittle simulator-specific behaviors long before a real robot or a human observer sees them.

We study a specific and common failure mode of safety-by-practice evaluation. A simulator-trained policy can look successful because the source reward is permissive, not because the learned behavior is robust or acceptable. A benchmark that compares a new safety method only against weak baselines will credit it regardless. This is a robotics instance of a broader specification and reward-gaming concern [2, 18]. Our position is that simulator success is a *claim requiring falsification*, not evidence of safety on its own. Before crediting a new plausibility prior, visual reward, alignment mechanism, or safe-learning method, a benchmark must rule out simple control explanations and must evaluate the same checkpoints under held-out shifts that stress different parts of the robot-environment interface. We package this as a falsification checklist that any embodied safety benchmark can adopt.

We instantiate the argument with the Plausibility-Gated Simulator RL (PG-SimRL) result package. Throughout, PG denotes the plausibility-gated method, which is PPO with the originally motivated action-norm cost, and PG+DR denotes that method combined with action-channel domain randomization. The empirical lesson is not that a new learned safety model solves manipulation. It is the opposite. Ordinary action-cost baselines are strong enough to change the mechanism story, so a benchmark that omits them will over-credit. On the same five-seed, 131k-step ManiSkill3 grid, squared action energy is strongest on PushCube, while action L1 is strongest by central estimate on PullCube and PokeCube. The originally motivated action-norm penalty is useful, but it is not the best cost form on any task. A frozen CLIP audit provides useful deployment-time diagnostics, yet the main robustness gains are explained by action costs.

Contribution. We contribute a deployment-time falsification protocol for embodied safety benchmarking. Report source success, shifted success, worst-shift success, paired-seed uncertainty, action-cost families, and audit and failure diagnostics under the same held-out evaluators, and treat action-

*Equal contribution.

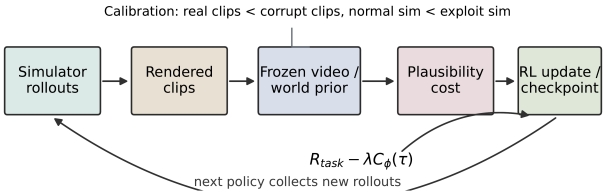

Fig. 1. Training and audit structure. PPO receives task reward minus a cheap action cost. The frozen visual model is used after training to audit or rank checkpoints, so the headline robustness gains are not caused by having CLIP inside the training loop.

cost baselines as a mandatory cheap-control comparator. The point is not to name one universally safe regularizer. It is to prevent weak safety claims from passing a benchmark only because the baseline set omitted strong, cheap controls.

II. The Claim We Test

The safety question here is not whether a policy satisfies formal collision constraints, solves constrained reinforcement learning [1], or transfers to a real robot. It is whether a simulator-RL robustness claim survives a basic falsification test:

> *Could the reported improvement already be explained by ordinary effort or energy costs under the same held-out shifts?*

If the answer is yes, then the result may still be useful, but the mechanism should be described as control-cost regularization rather than as evidence of a new safety concept.

This distinction matters for generalist robots, because open-world safety requires more than one scalar success signal. Source success, visual plausibility, low effort, stable final state, and human-perceived risk can all disagree. We therefore separate three roles that are often conflated: training-time rewards, checkpoint selectors, and post-hoc audit signals. A learned visual prior can be valuable as an audit channel even when it is not the source of closed-loop robustness. Conversely, a policy with high source success can fail a safety-oriented replay target if it relies on transient contacts or on excessive actions.

III. Method and Evaluation

Training with cheap action costs. Let a ManiSkill3 simulator define a Markov decision process with task reward R_{task}. We train PPO policies [17] with shaped returns

$$R_{\text{train}}(\tau) = \sum_t R_{\text{task},t} - \lambda C(\tau), \tag{1}$$

where C is a cheap action cost. We compare the action norm $C_\psi(\tau) = T^{-1}\sum_t \|a_t\|_2$, action L1, squared action energy $C_E(\tau) = T^{-1}\sum_t \|a_t\|_2^2$, and action-delta smoothness. Action-noise domain randomization perturbs the train-time action scale and noise. The norm, L1, and energy penalties use the same nominal coefficient $\lambda = 0.05$, so the comparison is a same-implementation falsification check rather than a global hyperparameter search.

The visual audit, defined precisely. The visual audit uses CLIP ViT-B/32 [14] as a frozen image encoder, following the broader use of pretrained visual, vision-language, and language-model-based representations and rewards [9, 7, 19, 8, 16]. We define the two terms a reader might otherwise have to guess. The *per-task manifold* is a set of points. For each task we take the CLIP embeddings of frames from *successful nominal rollouts*, standardize them using the mean and standard deviation of that fit set, and treat the resulting cloud of points as the task's plausible visual region. The *visual cost* of a candidate rollout is then a k-nearest-neighbor distance score. We embed the candidate, standardize it with the same statistics, and compute the mean Euclidean distance in that standardized space to its five nearest points in the manifold, normalized by the fit-set distance statistics. A larger visual cost means the rollout looks less like a successful nominal rollout. This is a distance to a set of real successful frames, not a cosine similarity to a single prototype. The combined selector ranks checkpoints by source success minus a within-task normalized visual cost. CLIP is never queried inside the headline PPO training runs.

Evaluators. We evaluate PushCube, PullCube, and PokeCube policies after 131k environment steps with five seeds in ManiSkill3 [20]. The ordinary shifted evaluator averages five off-nominal action-scale and action-noise variants and reports nominal source success separately. The stricter replay evaluator holds the policy weights fixed and changes safety-relevant simulator channels one at a time: action out-of-distribution shifts, a material-friction proxy shift, explicit PhysX material friction through SAPIEN and ManiSkill [22, 20], contact restitution, and dynamic object geometry. This is closer to a simulator stress test than to real-transfer evaluation, and real-world transfer benchmarks such as SIMPLER [6] remain a necessary next step.

Baselines and statistics. The baseline suite includes vanilla PPO, action-delta smoothness, action-channel domain randomization, action norm, action norm plus domain randomization, action L1, and squared action energy. The randomization and robust-RL baselines are intentionally modest relatives of broader sim-to-real and robust-policy training methods [21, 12, 15, 13], and the claim is only that even these cheap controls must be reported before a learned safety mechanism is credited. We then compare methods with a *paired-seed bootstrap*, defined as follows. For a given metric we compute, for each of the five seeds, the difference between two methods, which we call the per-seed delta. We then resample the five seeds with replacement 10,000 times and report the 2.5th and 97.5th percentiles of the mean delta. A *key delta* is one of the headline method-versus-baseline success differences, for example squared energy minus action norm on PushCube. Unless stated otherwise, every tabulated $\pm$ value in this paper is the sample standard deviation across the five seeds. We also keep negative diagnostics visible. A stable-final-state evaluator, which requires success over the final five steps,

TABLE I

MAIN 131K ACTION-SHIFT RESULT. ENTRIES ARE SHIFTED-SUCCESS PERCENTAGES OVER FIVE SEEDS. "BEST ORIGINAL NON-PG" IS THE STRONGEST OF VANILLA PPO, SMOOTHNESS, AND ACTION-NOISE DOMAIN RANDOMIZATION.

Task	Best original non-PG	PG action norm	Action energy	Energy − PG
PushCube	15.1	28.6	66.2	$+37.6$ $[+5.2, +68.6]$
PullCube	9.4	16.6	36.0	$+19.4$ $[-18.8, +58.5]$
PokeCube	11.1	14.0	13.8	-0.2 $[-4.9, +4.2]$

TABLE II

ACTION-COST FAMILY COMPARISON AT $\lambda = 0.05$. ENTRIES ARE SHIFTED-SUCCESS PERCENTAGES OVER FIVE SEEDS.

Task	PG norm	Action L1	Squared energy	Main takeaway
PushCube-v1	28.6 ± 32.8	41.2 ± 20.3	$\mathbf{66.2 \pm 15.4}$	Energy best
PullCube-v1	16.6 ± 16.2	$\mathbf{51.5 \pm 34.4}$	36.0 ± 40.0	L1 best, high variance
PokeCube-v1	14.0 ± 3.9	$\mathbf{22.0 \pm 15.6}$	13.8 ± 4.9	L1 best, high variance

is all-zero for these checkpoints. As we explain in the next section, that zero means the policies do not sustain the goal, not that they take no actions.

IV. RESULTS: ACTION COSTS CHANGE THE STORY

Table I summarizes the main 131k action-shift result. Energy improves PushCube shifted success from 28.6% under the action-norm penalty to 66.2%, with a paired-seed bootstrap interval of $[+5.2, +68.6]$ percentage points. PullCube shows a large central energy improvement but with wide uncertainty. PokeCube is a negative case for energy, roughly tying the action-norm penalty. We state plainly that PullCube and PokeCube carry high seed-to-seed variance (spreads of tens of percentage points over five seeds), so their central estimates should be read as directional evidence rather than as tight claims, and the win-counting under strict replay below is the more stable summary.

Table II shows why the safety claim must be phrased at the family level. Action L1, not squared energy, has the best central estimates on PullCube and PokeCube. The supported conclusion is therefore not that energy is safe. It is that the action-cost form is a first-order robustness variable. A new safety mechanism that beats only vanilla PPO or smoothness has not passed this falsification test.

The stricter replay grid agrees with the ordinary shift grid. Table III counts wins across nominal strict replay, action-OOD, friction proxy, true material friction, contact restitution, and dynamic geometry. Energy is strongest on PushCube and L1 on PullCube, while on PokeCube the action-norm cost and L1 each win 3 of 6 families and no cost form dominates. These replay targets are not a substitute for real deployment, but they make the source-success claim harder to overread. The win-count summary is also more robust to the Pull and Poke variance than the central estimates are, which is why we lead with it for those two tasks.

What the all-zero evaluator means. The stable-final-state evaluator scores a rollout as successful only if the cube stays at the goal for the final five steps. Every checkpoint scores zero on it. This does not mean the policy takes zero actions. It means that these policies achieve transient goal entry, which the source reward already counts as success, but do not hold the goal to the end. The gap between a non-zero source success and a zero stable-final-state success is exactly the kind of over-optimistic scoring our falsification checklist is meant to expose.

TABLE III

STRICT-TARGET SUMMARY FOR ACTION-COST FORMS. COUNTS EXCLUDE THE ALL-ZERO STABLE-FINAL-STATE EVALUATOR AND COUNT WINS ACROSS SIX REPLAY FAMILIES. ALL THREE COST FORMS ARE SHOWN, SO THE POKECUBE SPLIT IS VISIBLE.

Task	PG norm wins	Energy wins	L1 wins	Ties	Main strict-replay takeaway
PushCube	0/6	5/6	0/6	1/6	Energy strongest over strict replay
PullCube	0/6	0/6	5/6	1/6	L1 strongest over strict replay
PokeCube	3/6	0/6	3/6	0/6	PG norm and L1 tie, no form dominates

V. AUDIT SIGNALS AND FAILURE CASES

The CLIP audit is useful but secondary. In a constructed one-dimensional source-reward exploit, source success selects a high-impulse policy with 100% source success and 0% target success, whereas plausibility-audited selection recovers a policy with 96.5% source and 94.5% target success. On ManiSkill, source success and shifted success are already highly correlated at the completed 131k budget, so combined source-plus-CLIP selection mostly ties source-only selection, and CLIP-cost-only selection is worse. At the earlier 65k step budget the combined selector improves source-only regret on four of nine non-stable strict-target cells, ties three, and loses two, for a mean regret improvement of 6.1 percentage points.

The negative cases are part of the safety evidence. PG+DR does not automatically combine the action penalty and randomization, and in this implementation the combination underperforms on all three tasks. PullCube and PokeCube have high variance. Stable-final-state replay is all-zero. These failures keep the result from becoming a deployment claim, and they make the reporting standard more useful. A safety paper should show where the method wins, where it ties cheap costs, and where the simulator reward itself is too weak.

VI. A FALSIFICATION CHECKLIST AND EVIDENCE BUNDLE

The result yields a deployment-time falsification checklist that an embodied safety benchmark can require. First, report source success and held-out success separately, because source reward is not a safety metric by itself. Second, report worst-shift success, not only mean shifted success, because a robot can pass easy perturbations while failing the most safety-relevant condition. Third, include action norm, action L1 or torque-like effort, squared energy, and smoothness as cheap-control baselines before crediting a learned plausibility explanation. Fourth, label visual or language priors by role: training-time reward, checkpoint selector, or post-hoc audit. Evidence gathered in one role should not be reused as evidence for another.

Table IV turns this into an evidence bundle for reviewing simulator-trained generalist robot policies. Safety should be reported as an evidence bundle rather than as a single scalar,

TABLE IV
EVIDENCE BUNDLE FOR SIMULATOR-SIDE ROBOT-SAFETY CLAIMS. THE PROPOSED STANDARD SEPARATES SOURCE SUCCESS FROM ROBUSTNESS, MECHANISM ATTRIBUTION, AND FAILURE DISCLOSURE.

Evidence row	Minimum report	Why it matters for safety
Source task	Nominal success, reward terms, training budget	Shows what the policy was actually optimized to satisfy.
Held-out replay	Mean and worst success under action, contact, material, and geometry shifts	Tests whether success depends on one simulator setting.
Control baselines	Action norm, L1 effort, squared energy, smoothness, and randomization	Rules out cheap effort explanations before crediting a learned safety signal.
Uncertainty	Paired seeds or bootstrap intervals	Prevents high-variance central estimates from becoming mechanism claims.
Failures and audits	Stable-state tests, qualitative rollouts, visual manifold audits, negative baselines	Makes the non-deployment boundary explicit.

and in that bundle simulator success is only the first row. The bundle is intentionally modest. It does not require a real robot, but it does require enough evidence to avoid confusing task reward with safety. It also clarifies how a visual or language prior should enter a safety claim. If the prior is a reward, it is part of the controller and must be evaluated as a closed-loop mechanism. If it is a selector, then source-only and oracle selectors are the relevant comparators. If it is only an audit, its value is in finding failures, not in explaining training-time success.

This framing sharpens the relationship between the two safety axes. Action costs are themselves a safety-by-design mechanism, because a designer who adds an effort penalty is shaping the policy at the source. Our finding is that this cheap design-time mechanism is so effective that it must be a baseline for any deployment-time benchmark. Otherwise the benchmark will credit a more elaborate design-time method, such as a plausibility prior or a visual reward, for robustness it did not uniquely provide. Safety by practice, in other words, is what keeps safety-by-design claims honest. The broader claim is that "safe" should be treated as contextual and evidence-relative. A safe simulator-RL claim is one that survives the obvious cheap falsifiers and then names its remaining blind spots. Appendix A maps the checklist onto the workshop's trustworthiness topics.

VII. LIMITATIONS AND CONCLUSION

This is a simulator-side robustness study, and its scope is narrow in three ways that a reader should keep in mind. First, the benchmark suite is compact and the three tasks are simple, so the findings need to be re-checked on more dexterous and multi-object manipulation before they are generalized. Second, our comparators are cheap control regularizers rather than dedicated safe-RL or constrained-RL methods, so the claim is about what a benchmark must rule out, not about beating the safe-RL literature. Third, PullCube and PokeCube have high seed variance, so their central estimates are directional and the strict-replay win-counts are the load-bearing summary. The study also does not replace real-robot transfer, compliant-actuation tests, camera-domain shifts, deformables, clutter, or human judgments of perceived risk. The right follow-up is a controlled comparison among action norm, L1 effort, squared energy, torque-level costs, robust-RL objectives, and video or language reward baselines under the same material, contact, geometry, rendering, and real-transfer tests, on harder tasks and with more seeds.

Within that scope the conclusion is clear. Simulator-RL safety claims are fragile if a benchmark compares only against vanilla PPO, smoothness, or narrow domain randomization. Simple action costs are strong cheap-control baselines, and the best cost form is task-dependent. Frozen CLIP is useful as a deployment-time audit signal, but the completed ManiSkill gains are action-cost results. Adopting this falsification checklist as a standard step in embodied safety benchmarking would make future safety claims harder to overstate and easier to compare.

REFERENCES

[1] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International Conference on Machine Learning (ICML)*, pages 22–31, 2017.

[2] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016.

[3] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning (CoRL)*, 2023.

[4] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, et al. RT-1: Robotics transformer for real-world control at scale. In *Robotics: Science and Systems (RSS)*, 2023.

[5] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, et al. DROID: A large-scale in-the-wild robot manipulation dataset. In *Robotics: Science and Systems (RSS)*, 2024.

[6] Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, Sergey Levine, Jiajun Wu, Chelsea Finn, Hao Su, Quan Vuong, and Ted Xiao. Evaluating real-world robot manipulation policies in simulation. In *Conference on Robot Learning (CoRL)*, volume 270 of *Proceedings of Machine Learning Research*, 2025.

[7] Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. VIP: Towards universal visual reward and representation via value-implicit pre-training. In *International Conference on Learning Representations (ICLR)*, 2023.

[8] Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu,

Linxi Fan, and Anima Anandkumar. Eureka: Human-level reward design via coding large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

[9] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3M: A universal visual representation for robot manipulation. In *Conference on Robot Learning (CoRL)*, volume 205 of *Proceedings of Machine Learning Research*, 2023.

[10] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. In *Robotics: Science and Systems (RSS)*, 2024.

[11] Open X-Embodiment Collaboration. Open X-Embodiment: Robotic learning datasets and RT-X models. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[12] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3803–3810, 2018.

[13] Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. In *International Conference on Machine Learning (ICML)*, pages 2817–2826, 2017.

[14] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pages 8748–8763, 2021.

[15] Aravind Rajeswaran, Sarvjeet Ghotra, Balaraman Ravindran, and Sergey Levine. EPOpt: Learning robust neural network policies using model ensembles. In *International Conference on Learning Representations (ICLR)*, 2017.

[16] 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.

[17] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[18] 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.

[19] 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.

[20] Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse-kai Chan, Yuan Gao, Xuanlin Li, Tongzhou Mu, Nan Xie, Arnav Chowdhury, Yong Chen, Jiayuan Gu, et al. ManiSkill3: GPU parallelized robotics simulation and rendering for generalizable embodied AI. In *Robotics: Science and Systems (RSS)*, 2025.

[21] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 23–30, 2017.

[22] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, Li Yi, Angel X. Chang, Leonidas J. Guibas, and Hao Su. SAPIEN: A simulated part-based interactive environment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11097–11107, 2020.

APPENDIX A
MAPPING TO WORKSHOP TOPICS

Table V records how the falsification checklist maps onto the workshop's trustworthiness topics across the design and practice axes. This is a benchmarking and runtime-assurance contribution. The guardrail is not a runtime shield but a benchmarking guardrail against overstated mechanism claims, with a runtime-assurance component in the frozen-CLIP audit channel. Failure detection is operationalized as held-out replay plus the all-zero stable-final-state evaluator, which catches transient goal entry being scored as success. Red-teaming appears as the constructed source-reward exploit, a minimal adversarial case that any source-reward metric should survive.

TABLE V
HOW THE FALSIFICATION CHECKLIST MAPS ONTO THE WORKSHOP'S TRUSTWORTHINESS TOPICS ACROSS THE DESIGN AND PRACTICE AXES.

Workshop topic	What this paper contributes
Embodied safety benchmarking	A same-checkpoint replay protocol with source, shifted, worst-shift, and uncertainty reports, plus a mandatory action-cost cheap-control baseline.
Failure detection	Held-out replay and the all-zero stable-final-state evaluator detect transient-success scoring.
Red-teaming	A constructed source-reward exploit any source metric must survive.
Robust training	Action-cost forms are first-order robustness variables, not interchangeable regularizers.
Runtime assurance and uncertainty	A frozen-CLIP audit channel and paired-seed bootstrap intervals on every key delta.
Safety by design versus practice	Cheap design-time costs are strong, so practice-time benchmarks must rule them out before crediting a design-time method.

APPENDIX B
CONSTRUCTED SOURCE-REWARD EXPLOIT

The one-dimensional reference task isolates the safety failure mode. An event-based source success can reward transient goal crossing even when the final state is unacceptable. This is not a robotics benchmark. It is a minimal executable sanity check for why source success needs audits.

TABLE VI
CONSTRUCTED SOURCE-REWARD EXPLOIT. SOURCE SUCCESS USES THE EVENT REWARD, AND TARGET SUCCESS USES THE STRICTER FINAL-STATE EVALUATOR.

Method	Source	Target	Exploit	Plausible
Vanilla RL	100.0	0.0	100.0	0.0
RL + smoothness	45.1	15.8	11.9	0.0
Domain randomization	62.7	43.4	8.4	30.7
PG-SimRL	96.5	94.5	0.0	91.4
PG-SimRL + domain randomization	96.5	94.5	0.0	91.4

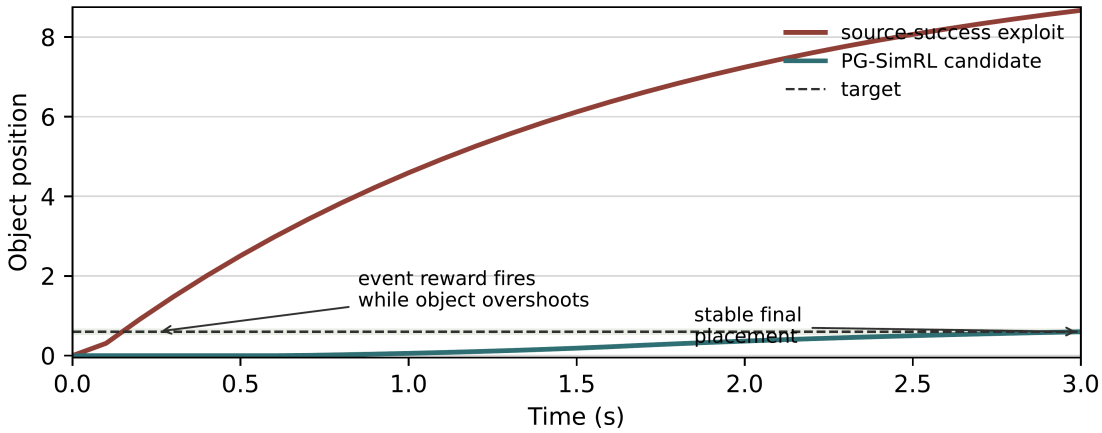

Fig. 2. Toy rollout trajectories showing how a high-impulse source-reward shortcut crosses the success line but fails the stricter final-state target.

TABLE VII
FULL 131K ACTION-SHIFT GRID. THE SHIFT AVERAGE EXCLUDES THE NOMINAL SOURCE VARIANT AND AVERAGES ACTION-SCALE AND ACTION-NOISE SHIFTS.

Task	Method	Seeds	Source	Shift avg.	Worst shift
PokeCube-v1	PG-SimRL	0,1,2,3,4	16.2 ± 6.8	14.0 ± 3.9	6.2 ± 3.8
PokeCube-v1	Action L1	0,1,2,3,4	18.8 ± 11.7	22.0 ± 15.6	12.5 ± 17.1
PokeCube-v1	PG-SimRL+DR	0,1,2,3,4	9.4 ± 5.4	9.5 ± 4.8	5.0 ± 4.7
PokeCube-v1	Action energy	0,1,2,3,4	10.0 ± 9.5	13.8 ± 4.9	3.8 ± 3.4
PokeCube-v1	Smoothness	0,1,2,3,4	9.4 ± 7.0	11.1 ± 6.6	5.0 ± 3.6
PokeCube-v1	Domain rand.	0,1,2,3,4	8.8 ± 3.4	10.9 ± 4.1	5.6 ± 4.1
PokeCube-v1	Vanilla PPO	0,1,2,3,4	5.0 ± 4.7	8.9 ± 3.5	4.4 ± 3.6
PullCube-v1	PG-SimRL	0,1,2,3,4	20.0 ± 21.4	16.6 ± 16.2	10.0 ± 14.0
PullCube-v1	Action L1	0,1,2,3,4	48.8 ± 33.5	51.5 ± 34.4	41.2 ± 38.2
PullCube-v1	PG-SimRL+DR	0,1,2,3,4	0.6 ± 1.4	0.5 ± 1.1	0.0 ± 0.0
PullCube-v1	Action energy	0,1,2,3,4	35.0 ± 45.6	36.0 ± 40.0	25.0 ± 34.5
PullCube-v1	Smoothness	0,1,2,3,4	11.2 ± 21.8	9.4 ± 13.9	5.6 ± 12.6
PullCube-v1	Domain rand.	0,1,2,3,4	5.6 ± 7.8	8.0 ± 7.6	2.5 ± 5.6
PullCube-v1	Vanilla PPO	0,1,2,3,4	2.5 ± 5.6	5.5 ± 9.6	2.5 ± 5.6
PushCube-v1	PG-SimRL	0,1,2,3,4	27.5 ± 35.5	28.6 ± 32.8	16.2 ± 27.1
PushCube-v1	Action L1	0,1,2,3,4	40.0 ± 19.6	41.2 ± 20.3	26.2 ± 16.8
PushCube-v1	PG-SimRL+DR	0,1,2,3,4	4.4 ± 3.6	5.1 ± 4.1	0.6 ± 1.4
PushCube-v1	Action energy	0,1,2,3,4	67.5 ± 19.5	66.2 ± 15.4	48.8 ± 22.7
PushCube-v1	Smoothness	0,1,2,3,4	21.2 ± 27.0	15.1 ± 24.3	8.1 ± 18.2
PushCube-v1	Domain rand.	0,1,2,3,4	8.8 ± 5.6	7.6 ± 6.5	1.2 ± 1.7
PushCube-v1	Vanilla PPO	0,1,2,3,4	7.5 ± 4.7	5.5 ± 3.6	0.6 ± 1.4

TABLE VIII
ACTION-COST COEFFICIENT AND FAMILY CHECK ON THE 131K ACTION-SHIFT GRID. THE MAIN COMPARISON USES THE SHARED $\lambda = 0.05$ SETTING, AND THE EXTRA ENERGY ROWS SHOW THAT THE COEFFICIENT CHOICE MATERIALLY CHANGES THE RANKING.

Task	PG norm	Action L1	Energy 0.02	Energy 0.05	Energy 0.10
PokeCube-v1	14.0 ± 3.9	22.0 ± 15.6	9.0 ± 6.1	13.8 ± 4.9	20.0 ± 11.2
PullCube-v1	16.6 ± 16.2	51.5 ± 34.4	12.8 ± 12.3	36.0 ± 40.0	8.8 ± 14.9
PushCube-v1	28.6 ± 32.8	41.2 ± 20.3	4.5 ± 5.2	66.2 ± 15.4	59.5 ± 11.1

APPENDIX C
FULL 131K ACTION-SHIFT GRID

APPENDIX D
STRICT REPLAY AND TRUE-DYNAMICS STRESS TESTS

APPENDIX E
PER-TASK EFFORT DIAGNOSTICS AND FINAL-STATE AUDIT

This appendix reports the three manipulation tasks separately so the family-level conclusion can be inspected per task. For each task we list source success together with effort diagnostics (action L1, squared energy, action delta Δa, and torque-like proxies $|\hat{\tau}|$ and $|\hat{\tau}\dot{q}|$) for the action-norm penalty, action L1, and squared action energy. These tables show that the cost form that wins on robustness is not the one that minimizes every effort metric. The safest-looking effort profile and the most robust policy can be different rows.

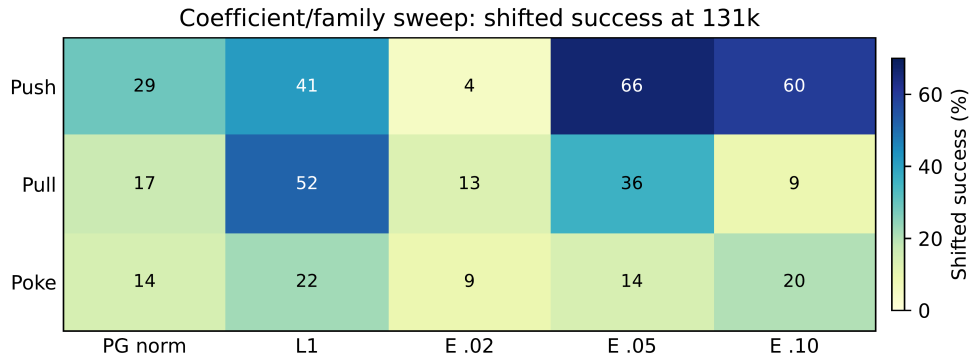

Fig. 3. Energy coefficient sweep. The sweep supports the paper's procedural claim that the action-cost form and coefficient are experimental variables, not fixed safety guarantees.

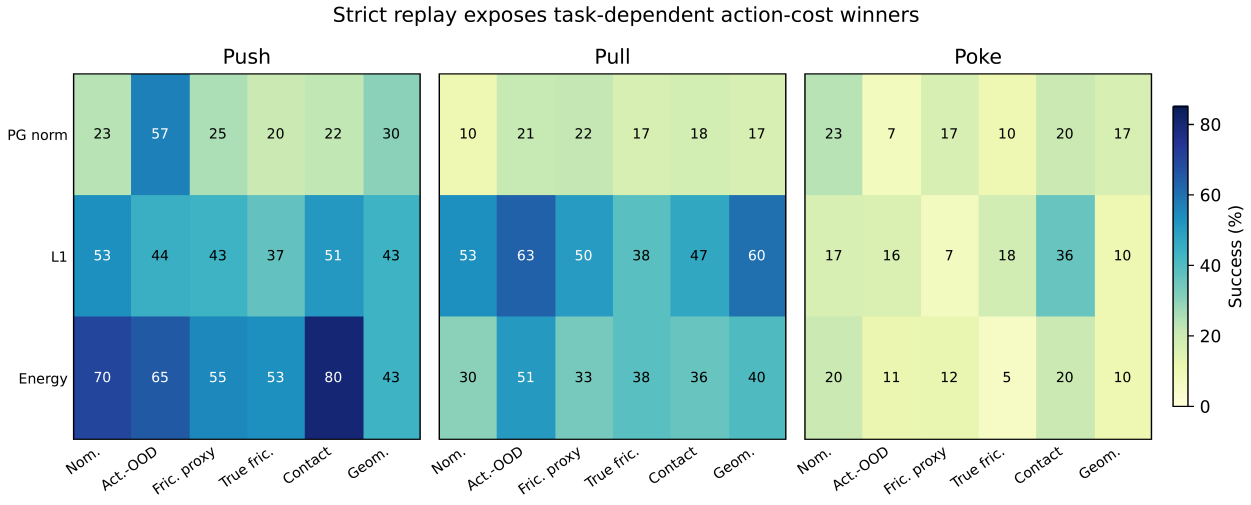

Fig. 4. Strict-replay success heatmap across action-cost forms and replay families.

Appendix F
Higher-Budget and Cross-Surface Diagnostics

Appendix G
Visual Audit and Selector Diagnostics

Appendix H
Domain Randomization, VLM Reward, and Negative Baselines

TABLE IX
PER-FAMILY STRICT-TARGET PERCENTAGES FOR THE ACTION NORM, ACTION L1, AND SQUARED ACTION ENERGY. VALUES ARE MEANS OVER SEEDS 0 TO 4, EXCLUDING THE ALL-ZERO STABLE-FINAL-STATE EVALUATOR.

Task	Target	PG norm	Action L1	Energy
PushCube-v1	Nominal replay	23.3	53.3	70.0
PushCube-v1	Action-OOD	56.7	44.4	65.0
PushCube-v1	Friction proxy	25.0	43.3	55.0
PushCube-v1	True friction	20.0	36.7	53.3
PushCube-v1	Contact restitution	22.2	51.1	80.0
PushCube-v1	Dynamic geometry	30.0	43.3	43.3
PullCube-v1	Nominal replay	10.0	53.3	30.0
PullCube-v1	Action-OOD	21.1	62.8	50.6
PullCube-v1	Friction proxy	21.7	50.0	33.3
PullCube-v1	True friction	16.7	38.3	38.3
PullCube-v1	Contact restitution	17.8	46.7	35.6
PullCube-v1	Dynamic geometry	16.7	60.0	40.0
PokeCube-v1	Nominal replay	23.3	16.7	20.0
PokeCube-v1	Action-OOD	7.2	16.1	11.1
PokeCube-v1	Friction proxy	16.7	6.7	11.7
PokeCube-v1	True friction	10.0	18.3	5.0
PokeCube-v1	Contact restitution	20.0	35.6	20.0
PokeCube-v1	Dynamic geometry	16.7	10.0	10.0

TABLE X
PAIRED-SEED BOOTSTRAP FOR SQUARED ACTION ENERGY VERSUS ACTION NORM ON THE ORDINARY 131K ACTION-SHIFT GRID. PUSH IS STATISTICALLY CLEAR, PULL IS DIRECTIONALLY LARGE BUT HIGH VARIANCE, AND POKE IS TIED.

Task	Baseline	Seeds	Action energy shift	Base shift	Δ [95% CI]	$P(\Delta > 0)$
PokeCube-v1	PG-SimRL	0–4	13.8	14.0	-0.3 [-4.9, 4.2]	41.3
PullCube-v1	PG-SimRL	0–4	36.0	16.6	19.4 [-18.8, 58.5]	82.1
PushCube-v1	PG-SimRL	0–4	66.2	28.6	37.6 [5.2, 68.6]	98.4

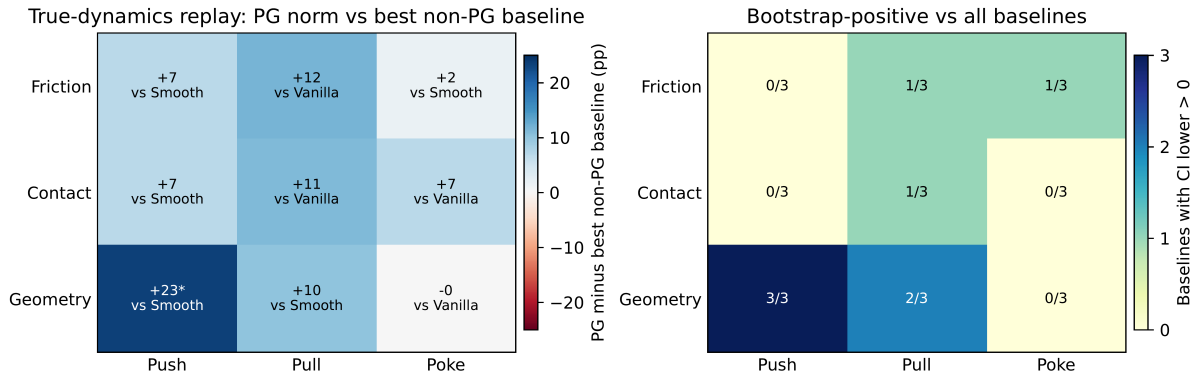

Fig. 5. Best-baseline comparison under true-dynamics stress tests that alter explicit physics properties.

TABLE XI
TRUE-DYNAMICS BEST-BASELINE TABLE. THESE ARE SIMULATION STRESS TESTS, NOT REAL-TRANSFER EVIDENCE.

Target	Task	Best non-PG	PG %	Base %	Δ pp [95% CI]
True fric.	Push	Smooth	20.0	13.3	+6.7 [-13.3, +33.3]
True fric.	Pull	Vanilla	16.7	5.0	+11.7 [-5.0, +35.0]
True fric.	Poke	Smooth	10.0	8.3	+1.7 [-3.3, +6.7]
Contact	Push	Smooth	22.2	15.6	+6.7 [-20.0, +48.9]
Contact	Pull	Vanilla	17.8	6.7	+11.1 [-4.4, +28.9]
Contact	Poke	Vanilla	20.0	13.3	+6.7 [-4.4, +15.6]
Geom.	Push	Smooth	30.0	6.7	+23.3 [+10.0, +40.0]
Geom.	Pull	Smooth	16.7	6.7	+10.0 [+0.0, +23.3]
Geom.	Poke	Vanilla	16.7	16.7	-0.0 [-20.0, +20.0]

TABLE XII
PUSHCUBE-V1 PER-TASK EFFORT DIAGNOSTICS OVER SEEDS 0 TO 4. SQUARED ENERGY REACHES THE HIGHEST SOURCE SUCCESS HERE, CONSISTENT WITH ITS MAIN-TABLE ROBUSTNESS LEAD.

Task	Method	Seeds	Source	L1	Energy	Δa	$\lvert\hat{\tau}\rvert$	$\lvert\hat{\tau}\dot{q}\rvert$
PushCube-v1	PG-SimRL	0,1,2,3,4	18.8 ± 28.6	3.75 ± 0.40	2.64 ± 0.47	0.79 ± 0.46	37.82 ± 6.97	28.47 ± 9.11
PushCube-v1	Action L1	0,1,2,3,4	52.5 ± 22.8	5.31 ± 0.29	4.64 ± 0.41	1.49 ± 0.84	35.66 ± 8.56	40.74 ± 10.95
PushCube-v1	Action energy	0,1,2,3,4	63.7 ± 10.3	5.01 ± 0.81	4.22 ± 1.04	0.98 ± 0.23	32.47 ± 3.40	36.90 ± 5.48

TABLE XIII
PULLCUBE-V1 PER-TASK EFFORT DIAGNOSTICS OVER SEEDS 0 TO 4. ACTION L1 REACHES THE HIGHEST CENTRAL SOURCE SUCCESS, WITH HIGH SEED VARIANCE.

Task	Method	Seeds	Source	L1	Energy	Δa	$\lvert\hat{\tau}\rvert$	$\lvert\hat{\tau}\dot{q}\rvert$
PullCube-v1	PG-SimRL	0,1,2,3,4	17.5 ± 19.0	3.77 ± 0.24	2.71 ± 0.26	1.28 ± 0.64	28.44 ± 4.36	24.22 ± 7.28
PullCube-v1	Action L1	0,1,2,3,4	48.8 ± 36.8	4.28 ± 0.53	3.23 ± 0.68	1.07 ± 0.44	29.50 ± 11.88	25.28 ± 12.67
PullCube-v1	Action energy	0,1,2,3,4	32.5 ± 44.5	1.69 ± 0.33	0.67 ± 0.16	0.28 ± 0.22	15.48 ± 7.87	8.06 ± 6.96

TABLE XIV

POKECUBE-V1 PER-TASK EFFORT DIAGNOSTICS OVER SEEDS 0 TO 4. ACTION L1 IS AGAIN THE STRONGEST CENTRAL ESTIMATE, WHILE SQUARED ENERGY TIES THE ACTION-NORM PENALTY.

Task	Method	Seeds	Source	L1	Energy	Δa	$\lvert\hat{\tau}\rvert$	$\lvert\hat{\tau}\dot{q}\rvert$
PokeCube-v1	PG-SimRL	0,1,2,3,4	16.2 ± 3.4	2.82 ± 0.53	1.81 ± 0.61	0.31 ± 0.08	15.29 ± 7.84	7.84 ± 1.33
PokeCube-v1	Action L1	0,1,2,3,4	27.5 ± 15.1	5.15 ± 0.91	4.42 ± 1.19	0.96 ± 0.49	14.02 ± 1.05	15.96 ± 4.54
PokeCube-v1	Action energy	0,1,2,3,4	16.2 ± 8.4	3.68 ± 0.34	2.69 ± 0.39	0.44 ± 0.12	14.55 ± 4.68	10.59 ± 2.74

TABLE XV

STABLE-FINAL-STATE AUDIT. A SUCCESS THAT REQUIRES HOLDING THE GOAL OVER THE FINAL FIVE STEPS IS ALL-ZERO FOR EVERY TASK AND COST FORM, WHICH SHOWS THAT TRANSIENT GOAL ENTRY IS EASIER THAN SUSTAINED PLACEMENT. THIS NEGATIVE RESULT IS PART OF THE SAFETY CASE, BECAUSE SOURCE SUCCESS OVERSTATES WHAT THESE CHECKPOINTS ACTUALLY ACHIEVE.

Task	Stable rows	Stable episodes	Stable successes	Stable %	Best strict event row
Push	15	90	0	0.0	PG / Fric. proxy (100.0%)
Pull	15	90	0	0.0	L1 / Nom. (100.0%)
Poke	15	90	0	0.0	L1 / Contact (55.6%)

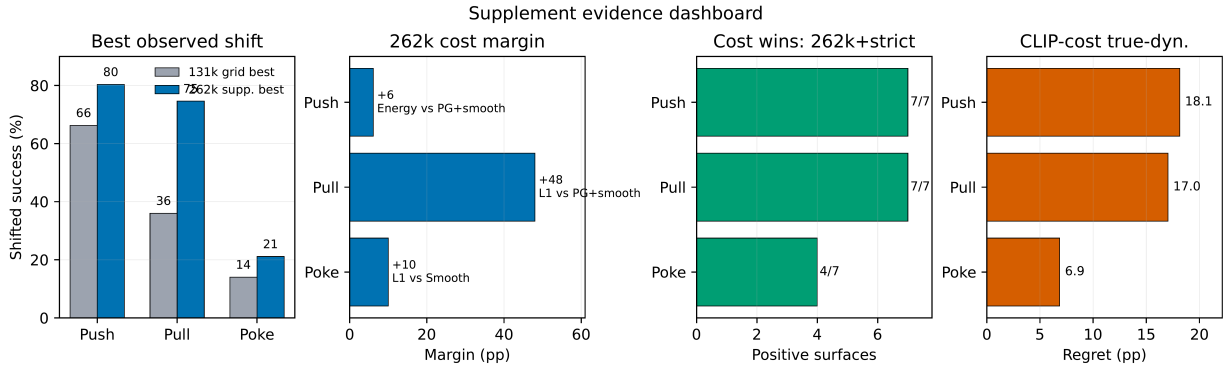

Fig. 6. Supplementary evidence dashboard combining same-budget, strict-replay, and cross-surface diagnostics.

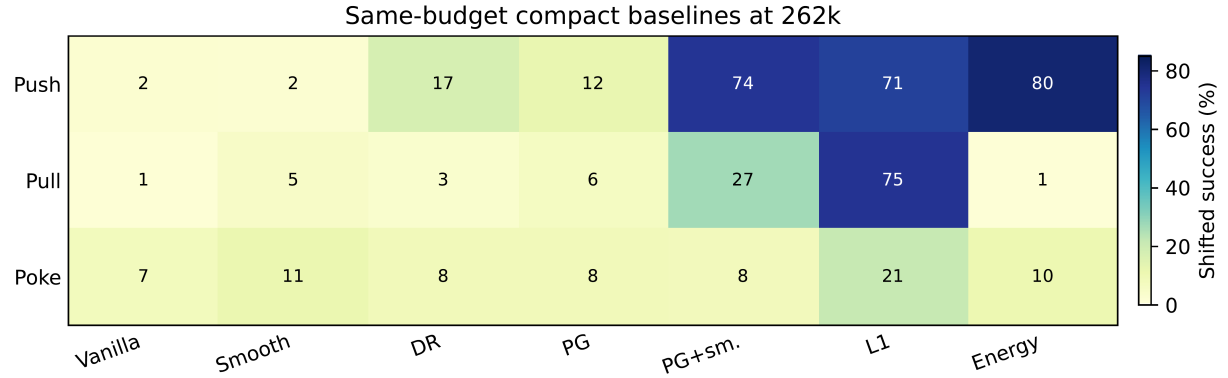

Fig. 7. Same-budget 262k comparison. A higher budget helps interpret whether the 131k result is a short-horizon artifact.

TABLE XVI

262K RANK AGREEMENT AMONG SOURCE, SHIFTED AVERAGE, AND WORST-SHIFT SUCCESS.

Task	Source–shift ρ	Source–worst ρ	Shift–worst ρ	Source top	Shift top	Worst top
Push	0.93	0.92	0.95	Energy	Energy	Energy
Pull	0.93	0.96	0.93	L1	L1	L1
Poke	0.31	0.12	0.95	L1	L1	L1

TABLE XVII
262K SOURCE-SELECTOR ALIGNMENT. THIS TABLE HELPS IDENTIFY WHEN SOURCE SUCCESS IS ALREADY PREDICTIVE AND WHEN A SELECTOR HAS ROOM TO IMPROVE.

Task	Metric	Source pick	Oracle	Source %	Selected %	Regret pp
Push	Shifted avg.	Energy	Energy	82.5	80.4	0.0
Push	Worst shift	Energy	Energy	82.5	73.1	0.0
Pull	Shifted avg.	L1	L1	72.5	74.6	0.0
Pull	Worst shift	L1	L1	72.5	65.0	0.0
Poke	Shifted avg.	L1	L1	17.5	21.1	0.0
Poke	Worst shift	L1	L1	17.5	10.6	0.0

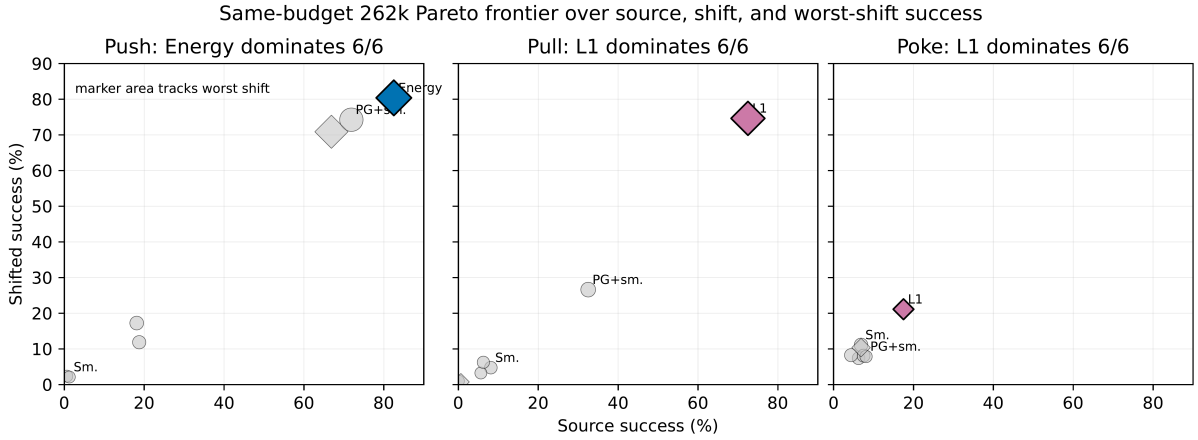

Fig. 8. 262k Pareto frontier over success and effort-like diagnostics.

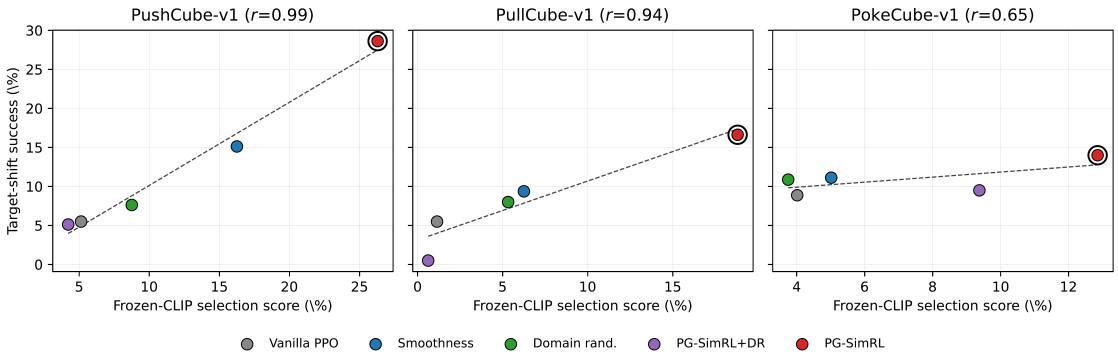

Fig. 9. Frozen-CLIP selection score versus shifted success. The high correlation at 131k partly reflects source success already being predictive.

TABLE XVIII
65K EARLY-BUDGET STRICT-TARGET SELECTION. COMBINED SOURCE-PLUS-CLIP IMPROVES SOURCE-ONLY REGRET ON FOUR OF NINE NON-STABLE CELLS, TIES THREE, AND LOSES TWO.

Task	Target	Source success Pick	Source success Reg.	Combined (α=0.05) Pick	Combined (α=0.05) Reg.	CLIP cost only Pick	CLIP cost only Reg.	Oracle %
PokeCube-v1	action_ood	Smooth	+20.4	PG	+0.0	PG	+0.0	23.1
PokeCube-v1	friction_proxy	Smooth	+0.0	PG	+2.8	PG	+2.8	13.9
PokeCube-v1	nominal	Smooth	+22.2	PG	+0.0	PG	+0.0	33.3
PokeCube-v1	stable	Smooth	+0.0	PG	+0.0	PG	+0.0	0.0
PullCube-v1	action_ood	Smooth	+13.9	Smooth	+13.9	PG	+0.0	17.6
PullCube-v1	friction_proxy	Smooth	+8.3	Smooth	+8.3	PG	+0.0	13.9
PullCube-v1	nominal	Smooth	+11.1	Smooth	+11.1	PG	+0.0	16.7
PullCube-v1	stable	Smooth	+0.0	Smooth	+0.0	PG	+0.0	0.0
PushCube-v1	action_ood	PG	+5.6	Smooth	+1.9	Vanilla	+0.0	30.6
PushCube-v1	friction_proxy	PG	+0.0	Smooth	+5.6	Vanilla	+11.1	16.7
PushCube-v1	nominal	PG	+16.7	Smooth	+0.0	Vanilla	+22.2	22.2
PushCube-v1	stable	PG	+0.0	Smooth	+0.0	Vanilla	+0.0	0.0

TABLE XIX
PAIRED-SEED BOOTSTRAP OF SOURCE-ONLY VERSUS COMBINED-CLIP AT THE MATCHED 65K BUDGET.

Task	Target	P(src=or.)	P(comb=or.)	Δ regret (pp) [95% CI]	P(comb>src)
PokeCube-v1	action_ood	0.44	0.53	+1.7 [-25.0, +21.3]	0.15
PokeCube-v1	friction_proxy	0.47	0.53	+0.2 [-11.1, +11.1]	0.10
PokeCube-v1	nominal	0.02	0.05	+3.3 [-5.6, +27.8]	0.19
PullCube-v1	action_ood	0.38	0.46	+1.0 [+0.0, +14.8]	0.07
PullCube-v1	friction_proxy	0.43	0.50	+0.7 [+0.0, +16.7]	0.06
PullCube-v1	nominal	0.39	0.46	+1.2 [+0.0, +22.2]	0.07
PushCube-v1	action_ood	0.43	0.55	+1.6 [+0.0, +19.4]	0.23
PushCube-v1	friction_proxy	0.78	0.57	-1.3 [-5.6, +0.0]	0.00
PushCube-v1	nominal	0.33	0.40	+4.7 [+0.0, +33.3]	0.25

Fig. 10. Visual selection bootstrap diagnostic. The visual audit is useful in tied or misleading source-success regimes, but it is not a universal selector.

TABLE XX
SCORER ABLATION FOR THE VISUAL CHANNEL ON SYNTHETIC CLIPS. TOP-10 PRECISION IS TIE-AWARE.

Encoder	Exploit cost	Non-exploit cost	AUC	Top-10 precision
Frame-difference	1.000	0.785	0.762	0.547
Frozen CLIP	0.999	0.775	0.939	0.904

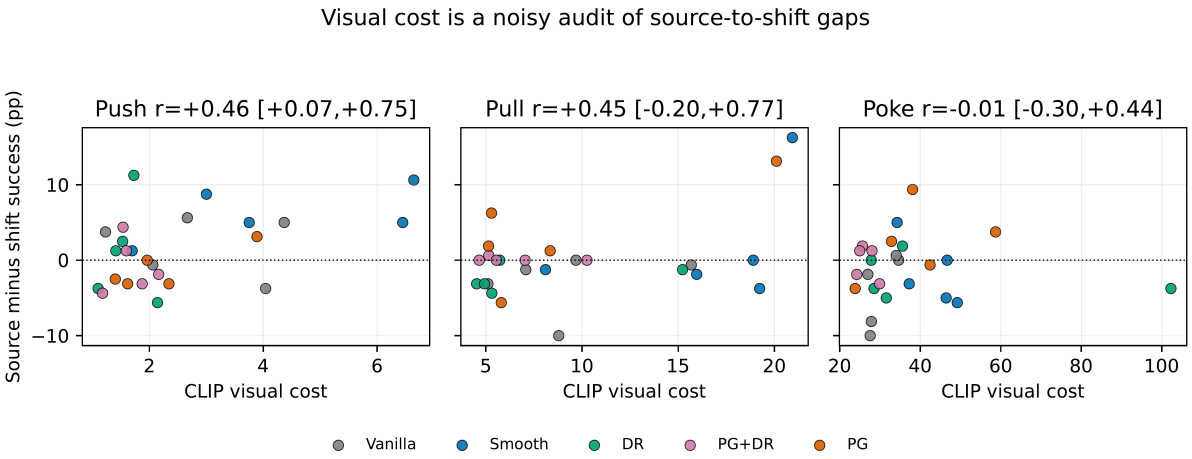

Fig. 11. Audit gap scatter. Visual costs can expose some source-shift gaps, but the signal is task-dependent.

TABLE XXI
TUNED ACTION-CHANNEL DOMAIN-RANDOMIZATION CHECKS AT THE SAME 131K BUDGET. MILD AND WIDE ACTION-SCALE AND NOISE VARIANTS DO NOT EXPLAIN THE BEST ACTION-COST GAINS.

Task	Method	Seeds	Source	Shift avg.	Worst shift
PokeCube-v1	Domain rand.(0.9-1.1,n=0.02)	0,1,2,3,4	5.0 ± 5.2	8.8 ± 5.6	1.2 ± 2.8
PokeCube-v1	Domain rand.(0.6-1.4,n=0.1)	0,1,2,3,4	5.0 ± 5.2	10.8 ± 6.7	2.5 ± 3.4
PullCube-v1	Domain rand.(0.9-1.1,n=0.02)	0,1,2,3,4	1.2 ± 2.8	4.2 ± 4.3	0.0 ± 0.0
PullCube-v1	Domain rand.(0.6-1.4,n=0.1)	0,1,2,3,4	1.2 ± 2.8	3.2 ± 6.0	1.2 ± 2.8
PushCube-v1	Domain rand.(0.9-1.1,n=0.02)	0,1,2,3,4	12.5 ± 4.4	13.5 ± 3.5	1.2 ± 2.8
PushCube-v1	Domain rand.(0.6-1.4,n=0.1)	0,1,2,3,4	11.2 ± 10.3	5.2 ± 4.1	0.0 ± 0.0

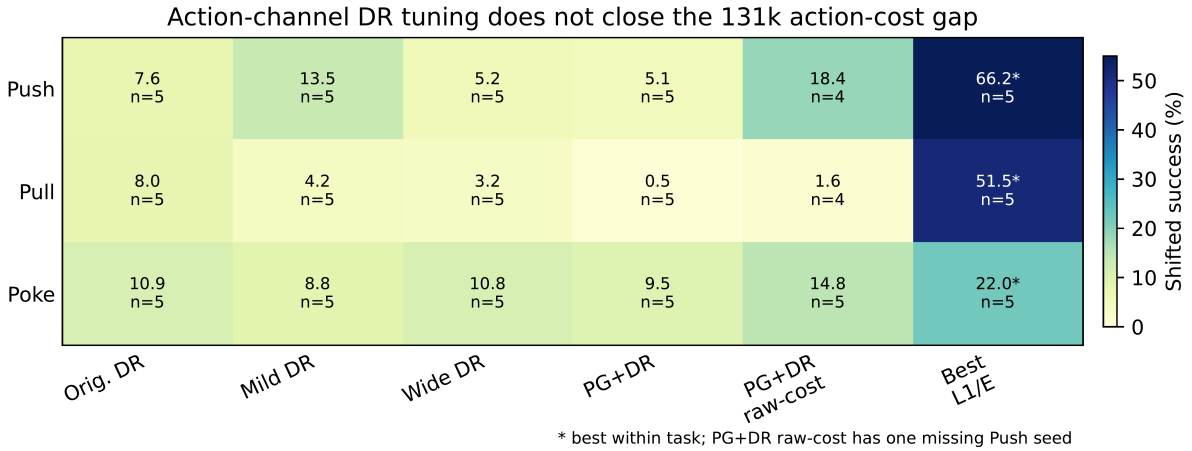

Fig. 12. Action-channel domain-randomization tuning comparison.

TABLE XXII
LIMITED TRAINED CLIP-REWARD BASELINE ON PUSHCUBE. THIS STRESS BASELINE TESTS NAIVELY MOVING THE FROZEN PRIOR INTO THE PPO REWARD LOOP. IT IS NOT A FULL REPRODUCTION OF VIDEO OR LANGUAGE REWARD METHODS.

Method	Seeds	Source (%)	Shift avg. (%)	Shift worst (%)	vs. PG-SimRL (pp)	$P(\Delta>0)$
PG-SimRL	5	27.5±31.8	28.6±29.4	16.2±24.2	,	,
CLIP-reward (alpha=0.05)	3	0.0±0.0	1.7±2.4	0.0±0.0	-30.6 [-76.2, -6.2]	0%
CLIP-reward (alpha=0.1)	3	0.0±0.0	0.0±0.0	0.0±0.0	-32.3 [-81.2, -6.2]	0%
CLIP-reward (alpha=0.5)	3	0.0±0.0	0.0±0.0	0.0±0.0	-32.3 [-81.2, -6.2]	0%
Domain rand.	5	8.8±5.0	7.6±5.8	1.2±1.5	,	,
Smoothness	5	21.2±24.2	15.1±21.8	8.1±16.2	,	,
Vanilla PPO	5	7.5±4.2	5.5±3.2	0.6±1.2	,	,

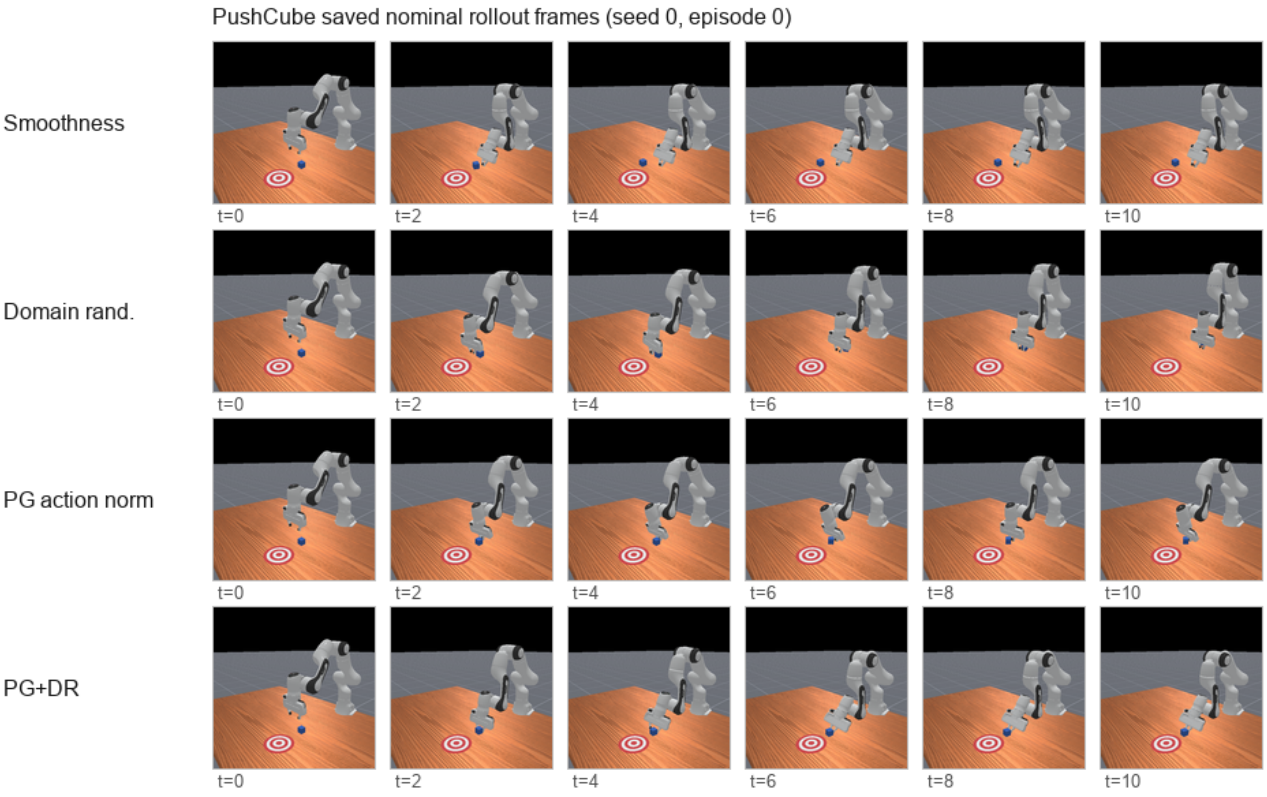

Fig. 13. Saved PushCube nominal rollout frames for qualitative audit. These screenshots are qualitative only and not additional quantitative evidence.