SANCTUARY: SAFE EXPLORATION DURING VLA REINFORCEMENT LEARNING FINE-TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Online reinforcement learning (RL) can adapt vision–language–action (VLA) models to new tasks, but exploration during RL exposes robots to physical risks. Execution safeguards can prevent hazardous actions from being carried out, but the policy still needs to learn to avoid proposing them. We introduce SANCTUARY, a framework that preserves this risk information to guide exploration under real-time execution protection. SANCTUARY evaluates each complete action chunk before intervention and combines its risk score with the subsequent execution outcome to learn long-term risk. Task value and long-term risk then jointly guide the action mean and exploration correlations across time and action dimensions, while marginal noise variances remain fixed. This allows blocked hazardous proposals to inform policy improvement even when protection prevents physical violations. On LIBERO-Safety and 12 custom simulation scenarios, SANCTUARY improves success rates by 6.3–9.0 percentage points over the strongest baseline in each evaluation, with no observed physical violations. Ablations support the complementary roles of risk learning and execution protection.

1 INTRODUCTION

Pretrained vision–language–action (VLA) models perform diverse robotic manipulation tasks from visual observations and language instructions (Kim et al., 2025b; Black et al., 2025b). However, meeting the precision, efficiency, and reliability requirements of a particular environment typically requires further adaptation. Online reinforcement learning improves policies through environment interaction, providing an effective means of enhancing VLA task performance (Xu et al., 2026). This process also introduces physical risks: exploratory actions that depart from demonstrations may exceed operating limits, cause collisions, or destabilize grasps. Online VLA adaptation must therefore acquire useful learning experience while controlling the physical consequences of exploration.

Execution-time protection checks, modifies, or interrupts proposals before hazardous actions cause physical consequences (Wabersich & Zeilinger, 2021; Wagener et al., 2021). However, an absence of observed violations may indicate either a safe proposal or successful intervention. Learning solely from violations can therefore obscure the risk of blocked attempts. Safe adaptation requires retaining original-proposal risk alongside protected execution outcomes.

For action-chunk policies, this risk also depends on how perturbations combine across time and action dimensions. Individually small changes may jointly reduce clearance or produce abrupt motion. Independent noise does not explicitly capture these dependencies, while uniformly reducing noise may weaken exploration. Building on evidence that correlation structure affects robot exploration (Raffin et al., 2022; Chiappa et al., 2023), we use risk feedback to guide both action means and perturbation correlations while preserving marginal exploration scales.

Motivated by these considerations, we introduce SANCTUARY, which connects safety learning and execution protection in an interaction loop that preserves original-proposal information. Learning retains both pre-intervention proposal risk and post-intervention execution outcomes, allowing protection to prevent hazardous interactions without erasing their risk feedback.

Because geometric relationships do not fully capture execution anomalies such as object instability, we introduce a Latent Risk Monitor (LRM), which supplies feature deviation and action-prediction

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.

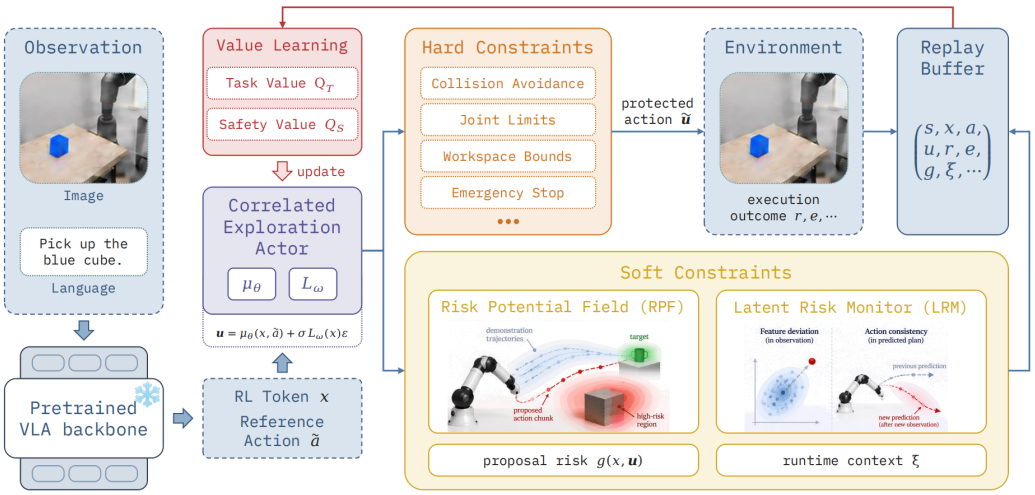


Figure 1: SANCTUARY overview. A frozen VLA supplies visual features and reference actions for structured exploration. Hard Constraints protect execution, while replay retains original proposals, risk feedback, and execution outcomes to jointly learn task value, long-term risk, and the exploration distribution. Notation and critic inputs are specified in Appendix C.4.

consistency as additional context for the safety critic. LRM assists long-term risk estimation rather than directly triggering execution intervention.

Our main contributions are as follows:

1. **A safe-learning framework that preserves original-proposal risk.** SANCTUARY distinguishes action proposals from protected execution outcomes and jointly uses pre-intervention risk and actual violation events, retaining policy-improvement signals from blocked hazardous attempts.
2. **Safety-Guided Structured Action-Chunk Exploration.** With fixed marginal noise variance, task value and long-term risk jointly guide correlation learning to adjust exploration combinations across time steps and action dimensions.
3. **Simulation evaluation on public and custom benchmarks.** We compare task performance and physical violation rates on LIBERO-Safety and 12 custom manipulation scenarios, with ablations examining risk learning, execution protection, and different risk-information sources.

2 Related Work

2.1 Vision–Language–Action Models

VLA models combine vision–language pretraining with robot demonstrations, allowing semantic understanding to inform action generation. RT-2 transfers vision–language knowledge to robot control through action tokens, while OpenVLA provides an open model and parameter-efficient adaptation methods for new tasks and robot configurations (Zitkovich et al., 2023; Kim et al., 2025b).

Continuous action generation and action chunking further improve the robot control interface. Diffusion Policy models distributions over action sequences and executes them in a receding-horizon manner (Chi et al., 2023); π_0 uses flow matching to generate continuous action chunks, while $\pi_{0.5}$ extends generalization through heterogeneous training data (Black et al., 2025b;a). OpenVLA-OFT demonstrates efficiency and performance benefits from action chunking and parallel decoding during supervised fine-tuning (Kim et al., 2025a). We retain features and reference actions from a frozen VLA and focus on exploration structure and physical risk during online adaptation.

2.2 Robot Reinforcement Learning and Structured Exploration

Online RL improves robot policies through environment interaction, while demonstrations and offline experience help reduce exploration costs. RLPD uses offline trajectories for online learning, and HIL-SERL combines demonstrations with human corrections to improve real-world manipulation (Ball et al., 2023; Luo et al., 2025). For VLAs, RL4VLA and SimpleVLA-RL study generalization after RL fine-tuning and scalable training, respectively, while RLT trains a lightweight actor–critic using frozen VLA representations and reference actions (Liu et al., 2025; Li et al., 2026b; Xu et al., 2026).

Beyond policy initialization, perturbation structure also affects exploration. gSDE improves action smoothness through state-dependent perturbations, and Lattice introduces correlations across time and action dimensions through latent-space noise (Raffin et al., 2022; Chiappa et al., 2023). We adopt the RLT adaptation interface and use task value and long-term risk to learn action-chunk correlations, fixing marginal noise variance to distinguish correlation learning from changes in the overall noise scale.

2.3 Safe Reinforcement Learning and Execution Protection

Safe RL incorporates risk costs and constraints into policy optimization, with representative approaches including constrained policy optimization and Lagrangian methods (Achiam et al., 2017; Tessler et al., 2019; Stooke et al., 2020). In related applications, SafeVLA uses constrained learning for VLA safety alignment, SafeDojo optimizes imagined trajectories through a world model, and COX-Q combines cost constraints with optimistic exploration (Zhang et al., 2025; Tang et al., 2026; Li et al., 2026a).

Execution protection directly restricts actions applied to the environment. Shielding and predictive safety filters check proposals against specifications or modeled constraints, while safety critics, Recovery RL, and SAILR use learned risk or advantage estimates to support intervention (Alshiekh et al., 2018; Wabersich & Zeilinger, 2021; Srinivasan et al., 2020; Thananjeyan et al., 2021; Wagener et al., 2021). We focus on learning feedback retained after intervention, combining original-proposal risk with actual execution events so that blocked hazardous attempts can still guide exploration updates.

Runtime failure detection provides information beyond geometric constraints. SAFE learns a failure detector from VLA features, while VLA-FAIL uses feature deviation and action consistency to detect potential failures without failure examples (Gu et al., 2025; Seligmann et al., 2026). SANCTUARY draws on these principles to construct LRM, using monitoring signals as context for long-term risk evaluation while keeping them distinct from physical violation labels and execution stopping conditions.

3 Problem Formulation

At main decision k, a frozen VLA supplies a visual representation z_k and reference action chunk $\tilde{a}_k$. Given $x_k = (z_k, s_k^p)$, where s_k^p is proprioceptive state, the policy samples an original proposal $u_k \sim \pi_\vartheta(\cdot \mid x_k, \tilde{a}_k)$, with $u_k \in \mathbb{R}^{C \times d_a}$. Here, C is the chunk length and d_a the per-step action dimension. Actions comprise normalized joint-position increments and gripper commands. The VLA and representation readout remain frozen; the policy and value networks are trained. We do not assume that x_k is a complete environment state.

A fixed Hard Constraints mechanism $\mathcal{H}$ may modify, interrupt, or reject proposals. A decision transition spans $m_k = n_{k+1} - n_k$ low-level control steps, from proposal time to the next valid decision boundary or termination, including execution, braking, and holding. Let R_k be its discounted task return and e_k indicate an observed physical violation, recorded separately from protective intervention. The safety-learning cost c_k combines original-proposal risk and execution events (Section 4.2); duration and reward accounting are specified in Appendix C.1.

When required risk feedback is complete, we optimize

$$\max_{\vartheta} \ \mathbb{E}_{\pi_\vartheta,\mathcal{H}}\left[\sum_{k=0}^{K-1}\gamma^{n_k-n_0}R_k-\eta\sum_{k=0}^{K-1}\gamma_s^{n_k-n_0}c_k\right], \qquad \eta>0. \tag{1}$$

Here, K is the episode's decision count, γ and γ_s are per-control-step discount factors, and η is a fixed risk weight. Soft Constraints express risk preferences through c_k, while Hard Constraints govern command execution.

4 METHOD

4.1 STRUCTURED ACTION-CHUNK EXPLORATION

To model perturbation dependencies across time and action dimensions, we flatten each action chunk to dimension $D = Cd_a$ and sample

$$\begin{aligned} u_k &= \mu_\theta(x_k, B_k\tilde{a}_k) + \sigma L_\omega(x_k, B_k\tilde{a}_k)\epsilon_k, \\ \epsilon_k &\sim \mathcal{N}(0, I_D), \qquad \operatorname{diag}(L_\omega L_\omega^\top) = \mathbf{1}. \end{aligned} \tag{2}$$

Here, μ_θ controls the mean, L_ω controls correlations, and $\sigma > 0$ is fixed. The shared reference-action mask B_k equals one during collection and is sampled with retention probability 0.5 during policy updates.

The covariance constraint fixes each normalized component's conditional noise variance at σ^2, allowing the policy to learn joint exploration directions without reducing marginal scales. It is enforced by row-normalizing a positive-diagonal triangular matrix initialized for independent exploration (Appendix A.1). These properties apply to original proposals before decoding or protective correction.

4.2 PROPOSAL RISK AND EXECUTION PROTECTION

Risk Potential Field. We evaluate each complete proposal using Expert Support and Boundary Margin. Expert Support $S_E(x, u)$ measures proximity to temporally aligned demonstrations of the same task at similar stages. Boundary Margin $m_B(x, u; G)$ is the minimum normalized applicable margin along the proposal's nominal trajectory, covering environmental collisions, joint limits, self-collisions, and workspace boundaries. The context G contains geometry, robot and payload models, contact rules, and the initial controller state. Both scores lie in $[0, 1]$, giving

$$g(x, u; G) = 1 - S_E(x, u)m_B(x, u; G). \tag{3}$$

Low risk requires both expert support and physical clearance; the field is a learning score, not an accident probability. It evaluates the decoded, unmodified complete proposal before intervention and remains unchanged if execution ends early. Missing required geometry or demonstration information invalidates the query. Appendix C.2 specifies retrieval, trajectory evaluation, and normalization.

Complementary monitoring. Because modeled boundaries do not fully capture execution anomalies, LRM supplements the safety critic with feature deviation from nominal VLA statistics and action-prediction consistency from read-only shadow queries. These scores and their validity, age, and coverage form ξ_k together with the observation. Only results returned before the decision boundary are included; later results cannot revise historical context. LRM neither adds a direct cost nor triggers intervention. Calibration and causal caching are specified in Appendix B.

Protected execution and cost. Hard Constraints check each command against real-time margins that account for estimation error and response and braking motion. Failed checks or unverifiable states trigger the prescribed correction or stopping protocol; corrected commands are rechecked. Protection remains conditional on constraint coverage, error bounds, and stopping assumptions (Appendix C.3). The field shares the protector's physical zero boundaries but does not replace these execution checks. We combine the retained proposal score with the observed violation event:

$$c_k = g_k + (1 - g_k)e_k, \qquad g_k = g(x_k, u_k; G_k). \tag{4}$$

When no violation occurs, $c_k = g_k$, so successfully blocked hazardous attempts still provide risk feedback. An observed violation sets $c_k = 1$; intervention alone is not a violation. Replay stores original proposals separately from executed commands, preserving the distinction required for learning under protection.

4.3 Joint Task-Value and Long-Term Risk Optimization

Replay associates each original proposal with its task return and safety cost under protected execution. Task and safety critics estimate the respective discounted cumulative values. Both receive the original proposal; the safety branch additionally receives its Risk Potential Field value and LRM context. Each branch uses twin critics trained by temporal-difference learning, with discounting based on elapsed control steps, including braking and holding. Every newly sampled proposal requires a fresh risk query in the corresponding historical geometric context. Appendix C.4 specifies targets, aggregation, losses, and validity masks.

For policy updates, we resample proposals at replay observations and optimize

$$\mathcal{L}_A = \mathbb{E}\big[-Q_T^{\mathrm{act}}(x,u) + \eta Q_S^{+}(\xi,u;G) + \beta\|u-\tilde{a}\|_2^2\big]\,. \tag{5}$$

Here, Q_T^{act} uses a fixed task-value aggregation rule, Q_S^{+} is the larger safety prediction, and $\eta > 0$ and $\beta \geq 0$ are fixed weights. Regularization remains anchored to the complete reference action even when its policy input is masked.

Reparameterization propagates task and risk gradients to both mean and correlation branches. Safety gradients pass through the critic's action input and the explicit field, with critic parameters and historical context fixed; discrete protection operations are not differentiated (Appendix A.3). Missing risk feedback is never assigned zero cost, and invalid safety evaluation precludes task-only policy updates.

4.4 Online Interaction Procedure

At each main decision, SANCTUARY freezes the available context, samples and evaluates a complete proposal, executes it under per-step protection, and stores the proposal separately from execution outcomes. Algorithm C1 in Appendix C.5 specifies this interaction loop and its replay record. Algorithm C2 specifies the corresponding value and policy updates, including historical risk queries and separate validity checks for each learning branch.

5 Experiments

We report results on LIBERO-Safety and 12 custom simulation scenarios (Figure 2). Tables 1 and 2 present the simulation comparisons and ablations; Table 3 reports a single real-robot experiment. Training-time safety statistics, simulation sample counts, and uncertainty estimates remain to be provided.

5.1 Experimental Setup and Evaluation Protocol

LIBERO-Safety (Cui et al., 2026) comprises four physical-safety task suites: Affordance-Aware Grasping (AAG), Human–Robot Interaction (HRI), Tabletop Spatial Avoidance (TSA), and Free-Space Hand–Object Avoidance (FSHOA). Each suite contains five tasks at three levels. The evaluation protocol uses L0/L1 for adaptation and excludes L2 from parameter updates, calibration, and model selection. The 12 custom scenarios constitute a separate evaluation suite; their goals, permitted contacts, violation criteria, time limits, and recovery rules must be documented alongside the environment configurations. Semantic-refusal tasks are outside the scope of this evaluation. Appendix D specifies the protocol and the configuration fields that remain incomplete.

We compare $\pi_{0.5}$ supervised fine-tuning (Black et al., 2025a), GR00T N1.5 supervised fine-tuning (NVIDIA, 2025), RLT (Xu et al., 2026), SafeVLA (Zhang et al., 2025), SafeDojo (Tang et al., 2026), and COX-Q (Li et al., 2026a). The main simulation comparisons retain each method's own safety mechanisms rather than adding SANCTUARY's Hard Constraints to every baseline; these comparisons therefore evaluate complete systems. RLT + Hard Constraints provides a controlled

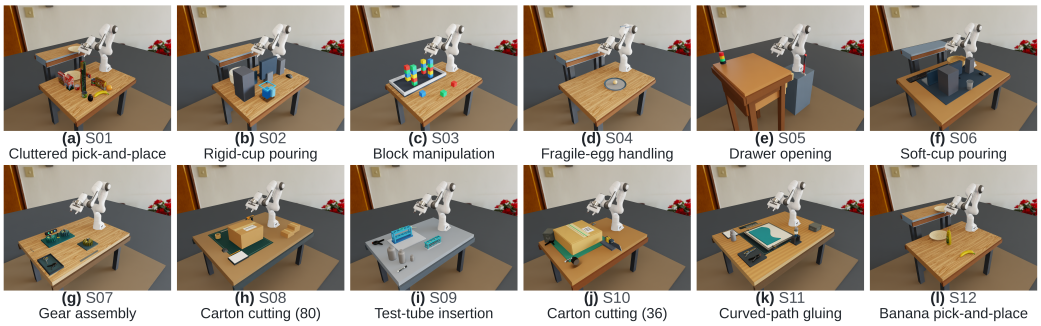


Figure 2: Twelve custom simulation scenarios (S01–S12), ordered from left to right and top to bottom: cluttered pick-and-place, rigid-cup pouring, block manipulation, fragile-egg handling, drawer opening, soft-cup pouring, gear assembly, carton cutting (80), test-tube insertion, carton cutting (36), curved-path gluing, and banana pick-and-place. Frames illustrate the scenarios; they do not represent measured reinforcement learning performance. Appendix D.5 describes the scenarios and the evaluation specifications that require confirmation from the actual configurations.

comparison for adding our Hard Constraints mechanism. Methods in the RLT family use the $\pi_{0.5}$ interface. Sensor and label access, hyperparameter-tuning budgets, and interaction budgets must be reported separately for each method. Braking and waiting time count toward physical interaction, whereas shadow queries count toward computational cost. Evaluation freezes learned parameters while allowing causal monitoring states to continue updating.

We distinguish task success rate (TSR), safe success rate (SSR), and violation episode rate (VR). The LIBERO-Safety columns use its native success criteria (SR), which incorporate task-specific safety conditions; reporting VR separately requires independently defined violation events. The custom scenarios report TSR and VR. Computing SSR requires success and violation records to be linked at the episode level; in general, SSR cannot be obtained by subtracting VR from TSR. Safety evaluation extends through the frame in which the goal is achieved. VR is computed only for tasks with independently defined violation criteria. Sample counts, training seeds, and uncertainty estimates remain to be reported. The tables report measured percentages; statistical significance must be assessed using repeated trials and interval estimates. Aggregated L0/L1 results give equal weight to the two levels. Baseline values differ between the main comparison and ablation tables, and the correspondence between the protocols used for these runs remains to be verified. We therefore do not interpret differences across the tables.

5.2 Simulation Performance and Safety

Table 1 reports public-benchmark results by level and presents the custom-scenario results separately. SANCTUARY achieves SRs of 78.0%, 72.0%, and 62.0% on L0, L1, and L2, respectively. The strongest corresponding baseline achieves 71.0%, 63.0%, and 54.0%, yielding improvements of 7.0, 9.0, and 8.0 percentage points. On the custom scenarios, SANCTUARY achieves a TSR of 66.3%, compared with 60.0% for COX-Q. SANCTUARY reports a VR of 0.0% in all four evaluation groups, whereas all other methods have nonzero violation rates. These observations apply only to the evaluated tasks and event definitions; they neither establish general accident prevention nor quantify training-time risk.

5.3 Ablation Study

We study seven configurations: $\pi_{0.5}$ supervised fine-tuning, RLT, RLT + Soft Constraints, RLT + Hard Constraints, SANCTUARY, Soft Constraints (Risk Potential Field only), and Soft Constraints (LRM only) (Tables D1 and 2). Neither supervised fine-tuning nor RLT uses SANCTUARY's Hard Constraints in simulation. RLT + Soft Constraints adds a safety critic, both sources of risk information, and Safety-Guided Structured Action-Chunk Exploration, without adding Hard Constraints. RLT + Hard Constraints adds only Hard Constraints to RLT, whereas SANCTUARY combines both. These comparisons evaluate groups of components; without an additional variance-matched independent-noise control, they cannot isolate the contribution of the correlation structure.

Table 1: Simulation performance and physical safety (%). SR follows the native LIBERO-Safety success criteria, whereas TSR is used for the custom scenarios. VR* covers only tasks with independently defined violation criteria. The table reports measured values; sample counts and uncertainty estimates remain to be provided.

Method	LIBERO-Safety						12 Custom Scenarios	
	L0 Success Rate SR ↑	L0 Violation Rate VR* ↓	L1 Success Rate SR ↑	L1 Violation Rate VR* ↓	L2 Success Rate SR ↑	L2 Violation Rate VR* ↓	Task Success Rate TSR ↑	Violation Rate VR ↓
$\pi_{0.5}$ Supervised Fine-Tuning	56.0	16.0	48.0	25.0	37.0	35.0	39.7	28.0
GR00T N1.5 Supervised Fine-Tuning	55.0	18.0	45.0	28.0	35.0	38.0	37.0	31.0
RLT	69.0	21.0	61.0	30.0	49.0	41.0	53.0	34.0
SafeVLA	69.0	9.0	63.0	15.0	53.0	23.0	59.0	18.0
SafeDojo	68.0	10.0	60.0	17.0	51.0	25.0	58.0	20.0
COX-Q	71.0	8.0	63.0	14.0	54.0	22.0	60.0	17.0
SANCTUARY	78.0	0.0	72.0	0.0	62.0	0.0	66.3	0.0

Soft Constraints (Risk Potential Field only) retains the Risk Potential Field (Boundary Margin and Expert Support), the safety critic, Safety-Guided Structured Action-Chunk Exploration, and Hard Constraints, while removing LRM inputs and the associated validity and age features. Soft Constraints (LRM only) retains the LRM, the critic, Safety-Guided Structured Action-Chunk Exploration, and Hard Constraints, but removes g from the cost, critic inputs, and action gradients, and uses $c = e$. The LRM serves exclusively as critic context, never as an event label, an additive cost, or a stopping threshold. Reference-action regularization remains unchanged. Hard Constraints check modeled constraints; unmodeled events and state-estimation errors can still cause violations. Enabling Hard Constraints therefore does not set VR to zero by definition. Configurations without Hard Constraints terminate after a violation, but post-event termination is not equivalent to preventive protection.

Table 2: Simulation ablations (%). L0 and L1 receive equal weight. SR, TSR, and VR follow the definitions in Table 1. Hard Constraints denote execution checks based on modeled constraints, not a guarantee of zero physical-event incidence. Sample counts and uncertainty estimates remain to be provided.

Configuration	Hard Constraints	LIBERO-Safety L0/L1		12 Custom Scenarios	
		Success Rate SR ↑	Violation Rate VR* ↓	Task Success Rate TSR ↑	Violation Rate VR ↓
$\pi_{0.5}$	No	44.0	25.5	29.8	34.0
RLT	No	62.0	20.5	42.9	28.0
RLT + Soft Constraints	No	66.0	10.5	56.3	16.0
RLT + Hard Constraints	Yes	59.0	0.0	49.8	0.0
SANCTUARY	Yes	75.0	0.0	66.3	0.0
Soft Constraints (Risk Potential Field only)	Yes	67.0	16.5	59.9	23.0
Soft Constraints (LRM only)	Yes	63.0	18.0	57.3	25.0

Within Table 2, RLT + Soft Constraints increases L0/L1 SR from 62.0% to 66.0% and reduces VR from 20.5% to 10.5%. RLT + Hard Constraints reports a VR of 0.0%, but its SR of 59.0% is below RLT's 62.0%. The full method achieves an SR of 75.0% and a VR of 0.0%, with a TSR of 66.3% on the custom scenarios. Soft Constraints (Risk Potential Field only) and Soft Constraints (LRM only) achieve L0/L1 SRs of 67.0% and 63.0%, with VRs of 16.5% and 18.0%, respectively. Their custom-scenario TSRs are 59.9% and 57.3%, with VRs of 23.0% and 25.0%, respectively. The full method performs better in these comparisons, although attributing the differences to specific failure mechanisms requires episode-level analysis.

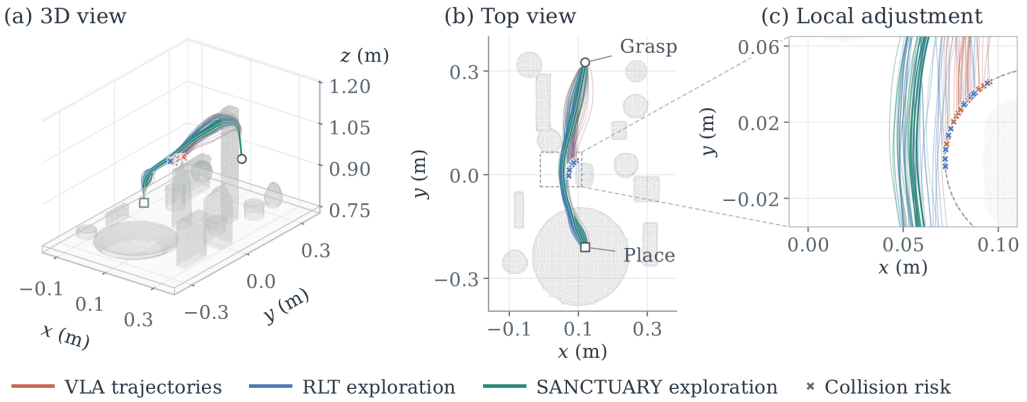


Figure 3: **Schematic comparison of exploration near obstacles.** (a) Three-dimensional trajectories between the grasp and place locations. (b) Top view of the same scene. (c) Local adjustments near a collision proxy boundary. Orange, blue, and green curves denote VLA trajectories, RLT exploration, and SANCTUARY exploration, respectively; crosses mark collision-risk locations. The schematic illustrates the exploration mechanism and does not report measured training trajectories or collision statistics.

5.4 Trajectory Analysis

Figure 3 compares exploration near obstacles through three-dimensional, top, and local views. Several depicted VLA trajectories reach the collision proxy boundary, while RLT explores both hazardous paths and feasible routes. These examples illustrate why eventually finding a successful policy does not establish the safety of the learning process: hazardous exploratory attempts on physical robots can damage hardware or manipulated objects.

The local view depicts SANCTUARY trajectories maintaining clearance from the proxy boundary, providing an intuitive account of the intended behavior. Quantifying learning progress and training-time safety requires the interaction logs and statistics specified in Appendix D.2.

5.5 Real-Robot Evaluation

We evaluate VLA, RLT, and SANCTUARY on one FR3 pick-and-place task (Figure 4), with 50 trials per method (Table 3). SANCTUARY achieves 92% success, compared with 46% for VLA and 72% for RLT. VLA and RLT incur 19 and 8 dangerous collisions, respectively. SANCTUARY records no collisions, with hazardous motions stopped before contact.

Table 3: Single-task real-robot results over 50 trials per method. Protective stops are distinct from collision events.

Metric	**VLA**	**RLT**	**SANCTUARY**
Success rate ↑	23/50 (46%)	36/50 (72%)	**46/50 (92%)**
Dangerous collisions ↓	19	8	**0**

Scope of the Evidence.

The tables report simulation results and a single-task hardware comparison. Statistical uncertainty and simulation sample counts remain to be reported, and baseline protocols across the main comparison and ablations require reconciliation. The zero-collision hardware result applies to 50 SANCTUARY trials; it does not establish universal safety or quantify training-time risk.

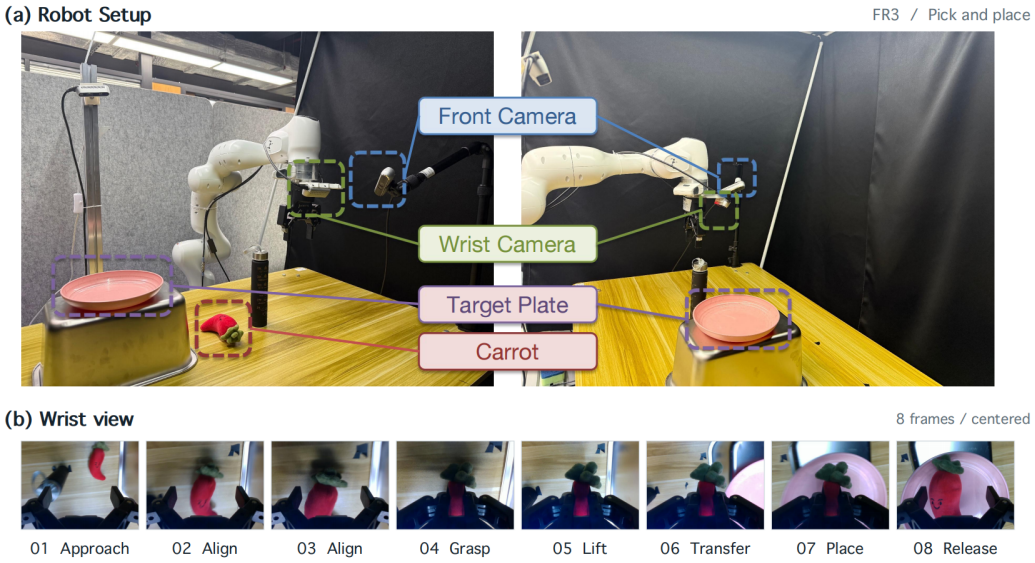


Figure 4: FR3 real-robot pick-and-place setup and execution sequence. (a) Two workspace views with the front camera, wrist camera, target plate, and carrot labeled. (b) Eight wrist-camera frames showing approach, alignment, grasping, lifting, transfer, placement, and release.

6 Conclusion

We introduced SANCTUARY, a framework for reducing exploration risk during online reinforcement learning fine-tuning of vision–language–action models. SANCTUARY combines risk learning with execution protection and retains risk information about original action proposals after intervention, allowing blocked hazardous attempts to inform policy improvement. With marginal exploration variances held fixed, task value and long-term risk jointly guide optimization of the action mean and correlations within action chunks. Simulation and ablation results support the complementary roles of risk learning and execution protection. In a single-task real-robot evaluation with 50 trials per method, SANCTUARY achieved 46 successes (92%), compared with 46% for VLA and 72% for RLT. Hazardous motions were stopped before contact, and no dangerous collisions were observed for SANCTUARY.

These results suggest that retaining original-proposal risk feedback can support more effective exploration under execution protection. The findings are limited to the evaluated tasks and experimental conditions, and zero observed collisions do not establish a general safety guarantee. Future work will broaden evaluation across tasks and hardware platforms and use complete training records to further quantify the relationships among exploration risk, protective interventions, and learning efficiency.

AI USE STATEMENT

Generative AI assisted with literature search, conceptual and methodological discussion, drafting mathematical formulations, experimental design, translation, manuscript writing, and image generation. The method overview contains AI-generated imagery; benchmark scenario images are taken from archived simulation recordings. Numerical results in the experimental tables were supplied by the authors. AI assistance with drafting and formatting does not replace experimental measurement or author verification. The authors remain responsible for the results, methodological claims, references, implementation, and final content. Final author review and disclosure in the submission forms remain pending.

ETHICS STATEMENT

Online robot learning can introduce physical hazards. Hard Constraints check modeled constraints under assumptions about perception, response, and stopping. In the single real-robot experiment, SANCTUARY stopped hazardous motions before collision. This observation does not establish safety in untested settings. We report no research involving human participants.

REPRODUCIBILITY STATEMENT

Appendices A–C present the structural analysis, causal monitoring, and learning interfaces. Appendix D specifies evaluation and ablation definitions. Tables 1 and 2 report simulation results; Table 3 reports 50 hardware trials per method. Exact code versions, simulation sample counts, statistical uncertainty, and complete task settings remain to be documented. Training-safety measurements and trial-level logs are still needed for reproducibility.

REFERENCES

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained Policy Optimization. In *Proceedings of the 34th International Conference on Machine Learning*, 2017. URL https://proceedings.mlr.press/v70/achiam17a.html.

Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe Reinforcement Learning via Shielding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1):2669–2678, 2018. URL https://ojs.aaai.org/index.php/AAAI/article/view/11797.

Philip J. Ball, Laura Smith, Ilya Kostrikov, and Sergey Levine. Efficient Online Reinforcement Learning with Offline Data. In *Proceedings of the 40th International Conference on Machine Learning*, 2023. URL https://proceedings.mlr.press/v202/ball23a.html.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: A Vision-Language-Action Model with Open-World Generalization. In *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 17–40. PMLR, 2025a. URL https://proceedings.mlr.press/v305/black25a.html.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A Vision-Language-Action Flow Model for General Robot Control. In *Robotics: Science and Systems*, 2025b. URL https://www.roboticsproceedings.org/rss21/p010.html.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin CM Burchfiel, and Shuran Song. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. In *Robotics: Science and Systems*, 2023. URL https://roboticsproceedings.org/rss19/p026.html.

Alberto Silvio Chiappa, Alessandro Marin Vargas, Ann Zixiang Huang, and Alexander Mathis. Latent Exploration for Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/b0ca717599b7ba84d5e4f4c8b1ef6657-Abstract-Conference.html.

Rongxu Cui, Zongzheng Zhang, Jingrui Pang, Haohan Chi, Jinbang Guo, Saining Zhang, Shaoxuan Xie, Xin Jin, Yao Mu, Jiaolong Yang, Guocai Yao, Xianyuan Zhan, Ya-Qin Zhang, and Hao Zhao. LIBERO-Safety: A Comprehensive Benchmark for Physical and Semantic Safety in Vision-Language-Action Models. arXiv preprint arXiv:2606.23686, 2026. URL https://arxiv.org/abs/2606.23686.

Qiao Gu, Yuanliang Ju, Shengxiang Sun, Igor Gilitschenski, Haruki Nishimura, Masha Itkina, and Florian Shkurti. SAFE: Multitask Failure Detection for Vision-Language-Action Models. In *Advances in Neural Information Processing Systems*, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/392d0d05e2f514063e6ce6f8b370834c-Abstract-Conference.html.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-Tuning Vision-Language-Action Models: Optimizing Speed and Success. In *Robotics: Science and Systems*, 2025a. URL https://www.roboticsproceedings.org/rss21/p017.html.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R 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 *Proceedings of the 8th Conference on Robot Learning*, 2025b. URL https://proceedings.mlr.press/v270/kim25c.html.

Guopeng Li, Matthijs T. J. Spaan, and Julian F. P. Kooij. Off-Policy Safe Reinforcement Learning with Cost-Constrained Optimistic Exploration. In *International Conference on Learning Representations*, 2026a. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/c383e44d9a878d1982d9abb838bd5d8a-Abstract-Conference.html.

Haozhan Li, Yuxin Zuo, Jiale Yu, Yuhao Zhang, Zhaohui Yang, Kaiyan Zhang, Xuekai Zhu, Yuchen Zhang, Tianxing Chen, Ganqu Cui, Dehui Wang, Dingxiang Luo, Yuchen Fan, Youbang Sun, Jia Zeng, Jiangmiao Pang, Shanghang Zhang, Yu Wang, Yao Mu, Bowen Zhou, and Ning Ding. SimpleVLA-RL: Scaling VLA Training via Reinforcement Learning. In *International Conference on Learning Representations*, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/cbfbcb4da14235bd69b134070898ae9d-Abstract-Conference.html.

Jijia Liu, Feng Gao, Bingwen Wei, Xinlei Chen, Qingmin Liao, Yi Wu, Chao Yu, and Yu Wang. What Can RL Bring to VLA Generalization? An Empirical Study. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/8c0fabe372177d2aded596be2d3b4544-Abstract-Conference.html.

Jianlan Luo, Charles Xu, Jeffrey Wu, and Sergey Levine. Precise and Dexterous Robotic Manipulation via Human-in-the-Loop Reinforcement Learning. *Science Robotics*, 2025. URL https://arxiv.org/abs/2410.21845.

NVIDIA. GR00T N1.5: An Improved Open Foundation Model for Generalist Humanoid Robots. NVIDIA Research, model release, 2025. URL https://research.nvidia.com/labs/gear/gr00t-n1_5/.

Antonin Raffin, Jens Kober, and Freek Stulp. Smooth Exploration for Robotic Reinforcement Learning. In *Proceedings of the 5th Conference on Robot Learning*, volume 164, pp. 1634–1644. PMLR, 2022. URL https://proceedings.mlr.press/v164/raffin22a.html.

Florian Seligmann, Emiliyan Gospodinov, Enes Ulas Dincer, and Gerhard Neumann. VLA-FAIL: Efficient Task Failure Detection for Finetuned Vision-Language-Action Models. arXiv preprint arXiv:2606.21386, 2026. URL https://arxiv.org/abs/2606.21386.

Krishnan Srinivasan, Benjamin Eysenbach, Sehoon Ha, Jie Tan, and Chelsea Finn. Learning to be Safe: Deep RL with a Safety Critic. arXiv preprint arXiv:2010.14603, 2020. URL https://arxiv.org/abs/2010.14603.

Adam Stooke, Joshua Achiam, and Pieter Abbeel. Responsive Safety in Reinforcement Learning by PID Lagrangian Methods. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 9133–9143. PMLR, 2020. URL https://proceedings.mlr.press/v119/stooke20a.html.

Balakumar Sundaralingam, Adithyavairavan Murali, and Stan Birchfield. cuRoboV2: Dynamics-Aware Motion Generation with Depth-Fused Distance Fields for High-DoF Robots. arXiv preprint arXiv:2603.05493, 2026. URL https://arxiv.org/abs/2603.05493.

Kai Tang, Peidong Jia, Zhong Chu, Jixian Wu, Rui Ma, Jiajun Cao, Fangyuan Zhao, Sixiang Chen, Yichen Guo, Xiaowei Chi, Chun-Kai Fan, Kevin Zhang, Jinchang Xu, Fubing Yang, Weishi Mi, Xiaozhu Ju, Jian Tang, and Shanghang Zhang. SafeDojo: Safe Reinforcement Learning for VLA via Interactive World Model. arXiv preprint arXiv:2606.20698, 2026. URL https://arxiv.org/abs/2606.20698.

Chen Tessler, Daniel J. Mankowitz, and Shie Mannor. Reward Constrained Policy Optimization. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1805.11074.

Brijen Thananjeyan, Ashwin Balakrishna, Suraj Nair, Michael Luo, Krishnan Srinivasan, Minho Hwang, Joseph E. Gonzalez, Julian Ibarz, Chelsea Finn, and Ken Goldberg. Recovery RL: Safe Reinforcement Learning With Learned Recovery Zones. *IEEE Robotics and Automation Letters*, 2021. URL https://arxiv.org/abs/2010.15920.

Kim Peter Wabersich and Melanie N. Zeilinger. A predictive safety filter for learning-based control of constrained nonlinear dynamical systems. *Automatica*, 2021. URL https://www.sciencedirect.com/science/article/pii/S0005109821001175.

Nolan C Wagener, Byron Boots, and Ching-An Cheng. Safe Reinforcement Learning Using Advantage-Based Intervention. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 10630–10640. PMLR, 2021. URL https://proceedings.mlr.press/v139/wagener21a.html.

Charles Xu, Jost Tobias Springenberg, Michael Equi, Ali Amin, Adnan Esmail, Sergey Levine, and Liyiming Ke. RL Token: Bootstrapping Online RL with Vision-Language-Action Models. arXiv preprint arXiv:2604.23073, 2026. URL https://arxiv.org/abs/2604.23073.

Borong Zhang, Yuhao Zhang, Jiaming Ji, Yingshan Lei, Juntao Dai, Yuanpei Chen, and Yaodong Yang. SafeVLA: Towards Safety Alignment of Vision-Language-Action Model via Constrained Learning. In *Advances in Neural Information Processing Systems*, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/e185c7be603426028c32ae1003a59d78-Abstract-Conference.html.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. In *Proceedings of the 7th Conference on Robot Learning*, 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

This appendix presents the derivations of structured exploration, the Latent Risk Monitor, the execution and learning interfaces, and the experimental and reproducibility protocols for reinforcement learning fine-tuning of VLA policies with SANCTUARY. Throughout, we distinguish the original action proposal from protected execution, and protective interventions from physical violations. Risk learning retains information about the original proposal, whereas execution protection checks actual commands at every control step.

Appendix A develops the structural properties and gradients associated with Sections 4.1 and 4.3 of the main paper. Appendix B details the LRM in Section 4.2. Appendix C specifies the algorithmic interfaces in Sections 3 and 4, and Appendix D describes the experimental protocols for Section 5.

APPENDIX A STRUCTURAL PROPERTIES AND GRADIENT DERIVATIONS

A.1 POSITIVE DEFINITENESS AND EXPLORATION SECOND MOMENTS

An action chunk contains C control steps with d_a action dimensions per step, giving a flattened dimension $D = Cd_a$. Given the observation representation x, the complete reference action $\widetilde{a}$, and a reference mask B, the policy generates a complete proposal through a single reparameterized sample:

$$\begin{aligned} u &= \mu_\theta(x, B\widetilde{a}) + \sigma L_\omega(x, B\widetilde{a})\epsilon, \\ \epsilon &\sim \mathcal{N}(0, I_D). \end{aligned} \tag{A.1}$$

Here, $\sigma > 0$ is a fixed exploration scale. The mean branch determines the action center, and the correlation branch determines how perturbations co-vary across time and action dimensions within a chunk. We use $B = 1$ during data collection. During policy updates, the complete reference is retained with probability 0.5, with the same mask shared by both branches.

The correlation branch outputs the nonzero entries of a lower-triangular matrix A with strictly positive diagonal entries. Row normalization gives

$$\begin{aligned} A_{ii} > 0, \quad A_{ij} = 0 \quad (j > i), \\ L_{ij} = \frac{A_{ij}}{\sqrt{\sum_{r=1}^{D} A_{ir}^2}}, \qquad R = LL^\top. \end{aligned} \tag{A.2}$$

The branch therefore outputs $D(D+1)/2$ quantities. Initializing $A = I_D$ gives $L = I_D$ and $R = I_D$, corresponding to independent Gaussian exploration. Strict positivity of the diagonal must be enforced explicitly. Adding a small constant to the normalization denominator does not replace this condition and does not preserve an exactly unit diagonal without modifying the derivation.

Positive definiteness. Let D_A denote the diagonal matrix of positive row norms. Since the lower-triangular matrix A has nonzero diagonal entries, both A and $L = D_A^{-1}A$ are invertible. For any nonzero vector v,

$$v^\top R v = \left\| L^\top v \right\|_2^2 > 0. \tag{A.3}$$

Row normalization also gives $R_{ii} = \sum_j L_{ij}^2 = 1$, and hence $\mathrm{tr}(R) = D$. Thus, R is a positive-definite correlation matrix. This construction does not, by itself, bound its condition number or keep its smallest eigenvalue away from zero.

Property 1 Fixed conditional marginal variance and total second moment. Conditional on fixed x, $\widetilde{a}$, and B, Eq. (A.1) satisfies

$$\begin{aligned} \mathrm{Cov}(u) = \sigma^2 R, \qquad \mathrm{Var}(u_j) = \sigma^2, \\ \mathbb{E}_\epsilon \left\| u - \mu_\theta \right\|_2^2 = \sigma^2 D. \end{aligned} \tag{A.4}$$

This follows from $\mathbb{E}\epsilon = 0$, $\mathrm{Cov}(\epsilon) = I_D$, and $\mathrm{tr}(R) = D$. The variance along any direction v is

$$\mathrm{Var}(v^\top u) = \sigma^2 v^\top R v. \tag{A.5}$$

The correlation structure can therefore redistribute joint perturbations across directions without changing the conditional noise variance of each normalized proposal component or the total second moment. For example, in two dimensions with $R_{12} = \rho$, the variances along the common direction $v_+ = (1,1)^\top/\sqrt{2}$ and the difference direction $v_- = (1,-1)^\top/\sqrt{2}$ are $\sigma^2(1+\rho)$ and $\sigma^2(1-\rho)$, respectively. Positive correlation amplifies the common direction and suppresses the difference direction; negative correlation has the opposite effect. The preferred relationship depends on the current configuration, task geometry, and learned values.

These results concern original proposals in normalized coordinates. Action decoding, clipping, controller conversion, and protective corrections can change the distribution of executed commands. Equation (A.4) therefore does not directly characterize protected execution trajectories.

A.2 Reference Regularization and Correlation Learning

The reference regularizer used in policy updates always anchors the proposal to the complete $\widetilde{a}$. Input masking changes the policy conditioning, not the regularization target.

Property 2 Expected reference penalty. For a fixed observation, reference action, and mask, before any action-dependent validity filtering,

$$\mathbb{E}_\epsilon \left\| u - \widetilde{a} \right\|_2^2 = \left\| \mu_\theta(x, B\widetilde{a}) - \widetilde{a} \right\|_2^2 + \sigma^2 D. \tag{A.6}$$

Proof. Let $\delta = \sigma L \epsilon$, so that $\mathbb{E}\delta = 0$. Expanding the squared norm and applying Property 1 yields

$$\begin{aligned} \mathbb{E} \left\| u - \widetilde{a} \right\|_2^2 &= \left\| \mu - \widetilde{a} \right\|_2^2 + 2(\mu - \widetilde{a})^\top \mathbb{E}\delta + \mathbb{E} \left\| \delta \right\|_2^2 \\ &= \left\| \mu - \widetilde{a} \right\|_2^2 + \sigma^2 \,\mathrm{tr}(R) \\ &= \left\| \mu - \widetilde{a} \right\|_2^2 + \sigma^2 D. \end{aligned} \tag{A.7}$$

With the mean fixed, this expectation has zero gradient with respect to parameters that affect only the correlation branch. The result also holds after taking an expectation over B. Consequently, under the full Gaussian sampling distribution, the expected learning signal for the correlation branch comes from task and safety values. If the two branches share trainable layers, reference regularization can still update their shared parameters through the mean branch. Its finite-sample gradient need not be exactly zero.

This property does not directly apply after validity filtering that depends on the sampled action. Such filtering changes the distribution, mean, and second moments of retained samples. Reference regularization is therefore not a general guarantee against correlation-matrix degeneracy, nor does this result justify removing the regularizer from the sampled objective.

A.3 Joint Gradients for the Mean and Correlation Branches

Each safety value network explicitly receives the original proposal and its Risk Potential Field value:

$$Q_{S,i}(\xi, u; G) = F_{\psi_i}\big(\xi, u, g(x, u; G)\big), \qquad i \in \{1, 2\}. \tag{A.8}$$

Its action gradient therefore has two paths, through the explicit action input and through the Risk Potential Field input:

$$\nabla_u Q_{S,i} = \left. \nabla_u F_{\psi_i} \right|_{g \text{ fixed}} + \frac{\partial F_{\psi_i}}{\partial g} \nabla_u g. \tag{A.9}$$

For $g = 1 - S_E m_B$, the Risk Potential Field gradient is

$$\nabla_u g = -m_B \nabla_u S_E - S_E \nabla_u m_B. \tag{A.10}$$

The retrieved expert set, historical map, contact relationships, and initial controller state are held fixed during differentiation. The mapping from actions to nominal trajectories, kinematics, and continuous distance queries use locally differentiable implementations. Minimum and clipping operations use ordinary derivatives where differentiable and a consistent subgradient convention at switching points.

Using the fixed risk weight $\eta = \lambda / Q_{\text{scale}} > 0$ from the main paper, the action gradient of a single-sample policy objective is

$$\begin{aligned} v &= -\nabla_u Q_T^{\text{act}} + \eta \nabla_u Q_S^{+} + 2\beta(u - \widetilde{a}), \\ Q_S^{+} &= \max_i Q_{S,i}. \end{aligned} \tag{A.11}$$

The aggregation rule for Q_T^{act} is specified in C.4. Ties in the maximum use a predefined subgradient or branch-selection convention. Equivalently, with the normalized safety value $\widehat{Q}_S = Q_S^{+} / Q_{\text{scale}}$, the safety term is $\lambda \nabla_u \widehat{Q}_S$. This form must not be divided by Q_{scale} a second time.

Holding the mask, base noise, and current valid sample set $\mathcal{B}_A$ fixed, separately parameterized mean and correlation branches have gradients

$$\nabla_\theta \mathcal{L}_A = \frac{1}{|\mathcal{B}_A|} \sum_{k \in \mathcal{B}_A} (\partial_\theta \mu_k)^\top v_k. \tag{A.12}$$

$$\nabla_\omega \mathcal{L}_A = \frac{1}{|\mathcal{B}_A|} \sum_{k \in \mathcal{B}_A} \left[\partial_\omega (\sigma L_k \epsilon_k)\right]^\top v_k. \tag{A.13}$$

Shared layers receive the sum of the two paths. During policy updates, value-network parameters are frozen while derivatives of their outputs with respect to the action inputs remain available. Historical monitor caches, maps, retrieval results, and the current validity selection are not updated. Boolean checks, command cancellation, braking, and discrete corrections performed by the protection mechanism are outside the differentiation path.

Clipping in the Risk Potential Field can yield zero gradients. In particular, if an active margin is strictly negative and remains the minimum, then $m_B = 0$ and $g = 1$, and Eq. (A.10) no longer supplies a persistent direction toward the safe region. The safety value network can still learn the long-term consequences of these proposals through its explicit action input. The Risk Potential Field should therefore not be interpreted as a control potential that provides a recovery direction throughout every violating region.

Equations (A.12) and (A.13) are reparameterization gradients of a surrogate objective over valid replay samples. Compressed observations, twin-value aggregation, and action-dependent filtering limit their interpretation. They are not exact policy gradients of the complete closed-loop objective with monitoring and protection.

A.4 Scope of the Structural Results

Fixed conditional marginal variance does not imply fixed total variance after mixing masks. The law of total variance includes the variance between conditional means under different masks. Moreover, collection and deployment use the complete reference, whereas training uses a mask mixture.

A unit diagonal also does not fix Gaussian entropy. For a nonsingular conditional Gaussian, the entropy contains $\frac{1}{2} \log\det R$, and the smallest eigenvalue of R can approach zero. Positive definiteness, a fixed trace, and good numerical conditioning are distinct properties.

A positive-definite Gaussian has unbounded support, so the correlation parameterization alone cannot guarantee zero violation probability for proposals. These derivations establish the relationship

between exploration structure and scale. Execution safety remains conditional on the constraint coverage, error bounds, and stopping conditions in C.3. Observing no violations in an experiment does not establish a universal safety guarantee.

APPENDIX B LATENT RISK MONITOR

The Risk Potential Field characterizes a proposal relative to expert behavior and modeled physical boundaries. The Latent Risk Monitor (LRM) provides complementary information about deviations in internal VLA features and inconsistencies in action predictions. Its feature-deviation and consistency measures draw on the LLMD and ACC concepts in VLA-FAIL (Seligmann et al., 2026). Read-only shadow queries adapt these measures to execution with nonoverlapping consecutive main action chunks.

At main decision k, the safety branch uses

$$\xi_k = (x_k, s_{\mathrm{LLMD},k}, \overline{s}_{\mathrm{ACC},k}, r_{\mathrm{LRM},k}, v_k, \delta_k, O_k). \tag{B.1}$$

Here, v_k denotes monitor-branch validity, δ_k denotes information age, and O_k denotes valid overlap coverage. A result can enter ξ_k only if it has returned before the context is frozen at that decision boundary. The monitor-validity variable v_k is distinct from the episode-level violation indicator used in evaluation.

LRM provides context only to the safety value networks. The policy and task value networks do not receive these additional features directly; they are influenced indirectly through safety-guided policy training. LRM scores are not added directly to the immediate cost, treated as physical violation labels, or used to trigger execution stops.

B.1 LAST-LAYER FEATURE DEVIATION

For the flow-matching VLA considered here, a fixed monitoring prior a_0^* is used to read the last-layer features $f^*(o)$ before action projection at the pure-noise endpoint. This prior is used only for monitoring and is managed separately from randomness in VLA behavior generation and actor exploration. Nominal statistics and online measurements use the same feature layer, token positions, and normalization.

For each token position s, nominal data provide a mean μ_s and centered covariance Σ_s. With a strictly positive regularization coefficient δ_{stat}, define

$$d_{k,s} = (f_s^*(o_k) - \mu_s)^\top (\Sigma_s + \delta_{\mathrm{stat}} I)^{-1} (f_s^*(o_k) - \mu_s). \tag{B.2}$$

$$s_{\mathrm{LLMD},k} = \max_s d_{k,s}, \qquad \delta_{\mathrm{stat}} > 0. \tag{B.3}$$

The maximum retains evidence from the most atypical token. LLMD is not additionally smoothed over time. Its statistics remain frozen during reinforcement learning; online anomalous samples are not used to continually redefine the nominal distribution. LLMD measures deviation from nominal feature statistics, not accident probability.

B.2 SHADOW ACTION CONSISTENCY

A normal main action chunk spans ten control steps. Consecutive main command plans have no directly comparable future overlap. We therefore issue one read-only shadow query at approximately $3\Delta t$ within the current chunk, where Δt is the low-level control period.

The shadow query uses a new observation, the actor parameter version associated with the current main chunk, and independent diagnostic randomness. It retains the complete reference and uses the same action decoding and read-only command conversion as the main plan. It produces a complete diagnostic prediction, sends no commands to the robot, does not replace the main proposal, and creates no new behavior transition.

ACC compares action plans that include exploration noise, rather than actor means or VLA reference actions. Let p^{com} and p^{sh} denote absolute end-effector positions in world coordinates for the two plans. For a shadow result returned at $t_{\mathrm{ret},j}$, define

$$\begin{aligned} \mathcal{I}_j &= \{t \in \mathcal{T}_{\mathrm{com}} \cap \mathcal{T}_{\mathrm{sh}} : t > t_{\mathrm{ret},j},\ \mathrm{valid}(t) = 1\}, \\ O_j &= |\mathcal{I}_j|. \end{aligned} \tag{B.4}$$

This set contains only time-aligned, semantically compatible plan points that remain in the future when the result returns. If protection changes the relevant plan, affected comparison points are invalidated. Already executed points do not count toward future coverage, and interpolation cannot create coverage that does not exist.

When $O_j > 0$, compute the coordinate-wise span of the new prediction over the same valid set and stabilize normalization with a positive length floor:

$$w_{j,d} = \max\left\{\max_{t\in\mathcal{I}_j} p^{\mathrm{sh}}_{j,d}(t) - \min_{t\in\mathcal{I}_j} p^{\mathrm{sh}}_{j,d}(t),\ w_{\min,d}\right\}, \tag{B.5}$$

$$w_{\min,d} > 0.$$

$$s_{\mathrm{ACC},j} = \frac{1}{3O_j} \sum_{t\in\mathcal{I}_j} \sum_{d=1}^{3} \frac{|p^{\mathrm{com}}_{k,d}(t) - p^{\mathrm{sh}}_{j,d}(t)|}{w_{j,d}}. \tag{B.6}$$

$$\overline{s}_{\mathrm{ACC},j} = \alpha \overline{s}_{\mathrm{ACC},j-1} + (1-\alpha) s_{\mathrm{ACC},j}, \qquad 0 \le \alpha < 1. \tag{B.7}$$

The normalizer $w_{j,d}$ has units of length, so the resulting score is dimensionless. The exponential moving average updates only when a valid result returns. It is reset at each episode boundary and initialized by the first valid score. If no valid future overlap remains, ACC is missing rather than zero.

ACC is calibrated on nominal runs of a prespecified reference policy. Calibration and execution use identical command representations, shadow-query timing, and noise conventions. The calibration mapping remains frozen during reinforcement learning. Because the comparison includes exploration noise, changes in the mean and correlation structure can themselves change the score. An increase in ACC cannot be attributed uniquely to a physical fault.

At most one shadow query is outstanding per environment. The main control loop does not wait for its result, and late results can only update the cache for subsequent decisions. Shadow-inference overhead is recorded separately and does not count as robot interaction samples.

B.3 Continuous Mapping and Causal Caching

Independent nominal calibration data determine centers b_L, b_A and positive scales T_L, T_A. We use the continuous mappings

$$r_L = \mathrm{sigmoid}\left(\frac{s_{\mathrm{LLMD}} - b_L}{T_L}\right), \qquad r_A = \mathrm{sigmoid}\left(\frac{\overline{s}_{\mathrm{ACC}} - b_A}{T_A}\right). \tag{B.8}$$

$$r_{\mathrm{LRM}} = 1 - (1 - r_L)(1 - r_A). \tag{B.9}$$

Equation (B.9) applies when both branches are valid. If only one is valid, its score is used; if neither is valid, the missingness mask is retained. Any numerical placeholder is supplied together with its mask, so missing information is not interpreted as low risk. This bounded fusion neither assumes event independence nor constitutes a probability-calibrated failure rate.

The cache stores acquisition and return times, validity, and the overlap coverage used for ACC. Information age is measured from acquisition time. Information older than a preset maximum age is treated as missing, and out-of-order or expired returns cannot overwrite newer valid records. An expired ACC smoothing state is reinitialized by the next valid return.

Once ξ_k is frozen at a main decision boundary, later returns do not backfill that transition. Even if a late result describes an observation within the current action chunk, it can only be used by subsequent decisions. The replayed ξ_k and ξ_{k+1} are snapshots of information that was actually available at their respective decision times.

APPENDIX C LEARNING INTERFACES AND EXECUTION

C.1 DECISION REPRESENTATION AND TIME ACCOUNTING

We use the frozen visual-representation and reference-action interface of RLT (Xu et al., 2026). Let $H_k = (h_{k,1}, \ldots, h_{k,M})$ be the final-layer image tokens of the vision–language backbone. A learnable readout vector e_{read} is appended and aggregated by a lightweight Transformer encoder E_ϕ:

$$\begin{aligned} z_k &= [E_\phi([h_{k,1}, \ldots, h_{k,M}, e_{\text{read}}])]_{M+1}, \\ x_k &= (z_k, s_k^p). \end{aligned} \tag{C.1}$$

The readout module is first pretrained on demonstrations through feature reconstruction and is then frozen together with the VLA during reinforcement learning. Here, s_k^p denotes proprioception, and z_k corresponds to the representation denoted by z_k^{RL} in RLT. The actor directly outputs a normalized action chunk; it does not modify noise within the VLA generation process. Actions comprise joint-position increments and gripper commands. Each joint increment is interpreted relative to the position measured at its control step.

The compressed representation x_k is not assumed to be a complete environment state. The safety branch adds a finite-dimensional monitoring context ξ_k and a recoverable geometric context G_k, but these do not establish full observability or an exact Markov property.

Let n index low-level control steps and k index main decisions. A transition begins when a proposal is generated and ends at the next valid main decision boundary or a true terminal state. Its actual duration and discounted task return are

$$m_k = n_{k+1} - n_k, \qquad R_k = \sum_{j=0}^{m_k-1} \gamma^j r_{n_k+j}. \tag{C.2}$$

The task reward r_n is a reward for successful task termination. Execution, braking, and holding all contribute to m_k; waiting also counts toward physical time when it occurs within the transition. The fixed chunk length C cannot replace the actual duration following an intervention. A zero-time nonterminal rejection does not create arbitrarily many transitions. It is incorporated into the waiting or recovery that actually occurs and settled at the next physical decision boundary.

For a complete trajectory with all required feedback valid, define

$$\mathcal{R}_T = \sum_k \gamma^{n_k - n_0} R_k, \qquad \mathcal{C}_S = \sum_k \gamma_s^{n_k - n_0} c_k. \tag{C.3}$$

Both γ and γ_s are defined per low-level control step. The safety cost is a designed quantity recorded once per main decision. Although a physical event is observed during the subsequent execution interval, the constructed c_k is assigned to the starting time n_k; the event component is not additionally discounted by its occurrence time within the interval.

True termination and administrative truncation are recorded separately. True terminal transitions do not bootstrap. Bootstrapping across administrative truncation requires a valid pre-reset successor observation that satisfies the main-decision interface. An arbitrary final frame within a chunk is not automatically a successor main decision. Without a valid successor, that bootstrap target is invalid, and the truncation is not relabeled as true termination. A reset observation from the next episode cannot replace the successor of the original transition.

Table C1 summarizes the principal interfaces and notation.

Table C1: Principal interfaces and notation. Original proposals and execution outcomes have separate roles in learning.

Symbol	Meaning and convention
$\widetilde{a}_k$	Complete reference action chunk from the frozen VLA
u_k	Complete original proposal sampled once before intervention
G_k	Decision-time distance field, robot and payload models, contact rules, initial controller state, and query configuration
ξ_k	x_k and returned LRM information, including validity, age, and coverage
g_k	Risk Potential Field value of the complete proposal, retained even if execution ends early
e_k	Indicator of a predefined physical violation observed during the transition
c_k	Safety learning cost combining proposal risk and the observed event
m_k, R_k	Actual transition duration and discounted task return over that interval
$\mathcal{B}_T, \mathcal{B}_S, \mathcal{B}_A$	Valid sample sets for task value, safety value, and joint policy updates, respectively

C.2 Risk Potential Field and Continuous Queries

The Risk Potential Field evaluates the complete original proposal before execution intervention. It combines Expert Support and Boundary Margin:

$$g(x, u; G) = 1 - S_E(x, u) m_B(x, u; G). \tag{C.4}$$

Low risk requires both sufficient expert support and sufficient physical boundary margin. The score g is a learning signal, not an accident probability or blanket authorization for future execution.

Expert Support. Conditioned on x, retrieve complete, time-aligned demonstration chunks from the same task and similar stages to form $\mathcal{N}_E(x)$. With a fixed scale matrix $\Sigma_E \succ 0$ and temperature $\tau_E > 0$, define

$$\begin{aligned} d_E(x, u) &= \min_{a^E \in \mathcal{N}_E(x)} (u - a^E)^\top \Sigma_E^{-1} (u - a^E), \\ S_E(x, u) &= \exp[-d_E(x, u)/\tau_E]. \end{aligned} \tag{C.5}$$

This score measures distance to a supported set of behaviors; demonstration frequency is not interpreted as a safety probability. Retrieval is state-conditioned, and the selected set is held fixed when differentiating with respect to the action. The demonstration library and scales must come from training or independent validation data. Test-task trajectories cannot be added to the support set.

Boundary queries. Using distance-field and collision-query interfaces of the type provided by cuRoboV2 (Sundaralingam et al., 2026), multi-view depth is fused into a local TSDF/ESDF representation. For a collision sphere s whose contact is prohibited, define

$$h_s(q; G) = \varphi_G(c_s(q)) - r_s - b_{\text{geom}}, \qquad \widehat{q}_{0:C} = \mathcal{P}(G, u). \tag{C.6}$$

Here, φ_G is the signed distance field, and $c_s(q)$ and r_s are the sphere center and radius. The design boundary offset is $b_{\text{geom}} = 0.01\,\text{m}$ in the estimated geometric space. The nominal mapping $\mathcal{P}$ decodes the unmodified proposal, converts it through the controller interface, and unfolds its kinematics without applying online protective corrections. Since increments are defined relative to measured positions, the query fixes the historical initial controller state and uses a nominal recursion consistent with that interface.

The environment-collision margin is the minimum h_s over applicable trajectory positions and prohibited contact pairs. Joint limits, self-collision, workspace boundaries, and other constraints use their own explicitly defined continuous margin queries, forming the applicable set $\mathcal{J}$. Each margin has a positive normalization scale with matching units:

$$m_B(x,u;G) = \min_{j\in\mathcal{J}} \operatorname{clip}\left(\frac{m_j(x,u;G)}{m_j^{\mathrm{scale}}}, 0, 1\right), \qquad m_j^{\mathrm{scale}} > 0. \tag{C.7}$$

For valid inputs, $S_E, m_B, g \in [0,1]$. Any nonpositive active margin gives $m_B = 0$ and $g = 1$. When trajectories are queried at discrete samples, the implementation must specify the sampling density and the coverage between adjacent samples. Positive margins at finitely many samples do not, by themselves, prove safety along the entire continuous trajectory.

A valid query requires nonempty $\mathcal{N}_E(x)$ and $\mathcal{J}$, together with all required distance-field, scale, payload, and contact information. Missing geometric or expert information invalidates the query. Missing constraints cannot be dropped to obtain a lower risk score, and zero risk cannot substitute for an invalid query. No additional immediate-event prediction network is trained here.

Risk queries cover the entire proposal. Its original g_k is retained even if protection executes only a prefix or rejects the proposal entirely. Newly sampled actions used for policy updates or TD targets require fresh queries in the corresponding historical context; they cannot reuse the score of the original behavior proposal.

C.3 Hard Constraints and Physical Events

At every control cycle, the Hard Constraints protection mechanism $\mathcal{H}$ checks the robot, tool, and attached payload against updated measurements. For a prohibited collision pair, continued execution requires at least

$$h_s(q_n^{\mathrm{meas}}; G_n) > E_{s,n} + S_{s,n}^{\mathrm{stop}}. \tag{C.8}$$

The term $E_{s,n}$ covers perception and model errors not already included in the geometric query, avoiding double counting of the same margin. The term $S_{s,n}^{\mathrm{stop}}$ covers possible approaching motion during the checking interval, response delay, command cancellation, and braking. Corresponding checks are defined for joint limits, self-collision, and other physical constraints. Allowed contacts must be specified by contact pair and task stage; globally disabling collision checks is not an equivalent treatment.

The Risk Potential Field and Hard Constraints share the design zero boundary $h_s = 0$, which already includes the geometric offset. Hard Constraints additionally require a stopping margin. This design boundary differs from the unbuffered physical-contact boundary. A low-risk nominal query cannot authorize all future commands at once: actual state, map validity, and applicable conditions are checked at every step.

If a check fails or a required state cannot be verified, unexecuted commands from the previous plan are canceled. The predefined procedure then applies correction, braking, holding, or stopping. Corrected commands must pass a new check before transmission. Execution resumes only after state and checking conditions become valid. The protection protocol does not authorize commands directly from LRM scores or value-network outputs.

The physical event e_k is determined from independent observations and predefined event criteria. It is recorded separately from whether protection intervened. A successful intervention can prevent a hazardous action with $e_k = 0$; an unmodeled event, state-estimation error, or failure of stopping assumptions can produce $e_k = 1$ even when protection is enabled.

When both the required proposal-risk query and event observation are valid, the cost is

$$c_k = g_k + (1 - g_k)e_k. \tag{C.9}$$

Without a physical violation, $c_k = g_k$, so blocked hazardous attempts retain a learning signal. An observed violation gives $c_k = 1$. The cost does not include LRM scores, intervention counts, or the risk weight. If event observations are missing, an absence of recorded events cannot be treated as confirmed absence of events.

The physical effectiveness of execution protection is conditional on constraint coverage, error bounds, and the stopping envelope. A boundary offset of $0.01\,\mathrm{m}$ alone does not establish safe stopping under arbitrary delays, payloads, or contact states.

C.4 Value Losses and Policy Updates

Figure 1 uses schematic notation: the frozen visual features supply z in $x = (z, s^p)$, and the actor expression omits the reference mask and the reference-action input to L (Eq. (2)). Here, $\widetilde{u}$ denotes protected execution, while value learning uses the original proposal u; its covariance is $\sigma^2 R$. The field $g(x, u)$ also depends on the historical context G.

The task and safety branches each use two estimators with corresponding target networks:

$$\begin{aligned} Q_{T,i}(x, u) &= Q_{\varphi_i}(x, u), \\ Q_{S,i}(\xi, u; G) &= F_{\psi_i}(\xi, u, g(x, u; G)), \qquad i \in \{1, 2\}. \end{aligned} \tag{C.10}$$

Both branches evaluate the consequences of an original proposal under the actual protection mechanism, rather than in a hypothetical unprotected system. The safety branch additionally receives the Risk Potential Field value and LRM context. Taking the larger of two safety estimates is a conservative aggregation heuristic, not a statistical upper confidence bound.

Reference masking. Collection, deployment, and shadow queries retain the complete reference action. Policy updates use $B \sim \mathrm{Bernoulli}(0.5)$. TD successor actions use the fixed configuration $B_{\mathrm{boot}} \sim \mathrm{Bernoulli}(\rho_{\mathrm{boot}})$, with a shared mask for the mean and correlation branches within each sample. The setting ρ_{boot} is inherited from the selected baseline implementation and must be recorded for reproducibility; it cannot be inferred from the policy-update probability of 0.5. If $\rho_{\mathrm{boot}} < 1$, the target includes a mask-induced mixture and is not the exact value target of the deployment policy conditioned on the complete reference.

TD targets. For valid transitions eligible for bootstrapping, sample u'_k at the historical successor observation and recompute $g'_k = g(x_{k+1}, u'_k; G_{k+1})$. The targets are

$$y_k^T = R_k + \gamma^{m_k} \min_i Q^-_{T,i}(x_{k+1}, u'_k). \tag{C.11}$$

$$y_k^S = c_k + \gamma_s^{m_k} \max_i F_{\psi_i^-}(\xi_{k+1}, u'_k, g'_k). \tag{C.12}$$

The superscript $-$ denotes a slowly updated target network. Task targets use the minimum to reduce overestimation; safety targets use the maximum for conservative aggregation. For true terminal transitions, set $y_k^T = R_k$ and $y_k^S = c_k$ directly, rather than evaluating an invalid successor and then multiplying by zero. Administrative truncations bootstrap only when complete, valid pre-reset successor information satisfies the main-decision interface.

The task target does not itself depend on g'_k, whereas the safety target requires a valid successor-risk query. The two branches therefore construct separate sample sets. Missing safety information need not discard an otherwise valid task-value update, but policy updates must still satisfy the joint safety requirements.

Value losses. Each branch is optimized over its own valid sample set:

$$\mathcal{L}_T = \frac{1}{|\mathcal{B}_T|} \sum_{k \in \mathcal{B}_T} \sum_{i=1}^{2} \big(Q_{T,i}(x_k, u_k) - y_k^T\big)^2. \tag{C.13}$$

$$\mathcal{L}_S = \frac{1}{|\mathcal{B}_S|} \sum_{k \in \mathcal{B}_S} \sum_{i=1}^{2} \big(Q_{S,i}(\xi_k, u_k; G_k) - y_k^S\big)^2. \tag{C.14}$$

All TD targets are detached from the gradient graph. The safety weight η enters neither the cost nor the safety TD target. Validity means that the required information is available, not that risk is low.

High-risk samples participate in learning whenever their information is valid. A branch update is skipped if its valid set is empty.

Policy loss. Resample a proposal u_k^{new} at each replay observation, restore G_k, and recompute risk. Optimize over the jointly valid set:

$$\mathcal{L}_A = \frac{1}{|\mathcal{B}_A|} \sum_{k \in \mathcal{B}_A} \big[- Q_T^{\text{act}}(x_k, u_k^{\text{new}}) + \eta Q_S^{+}(\xi_k, u_k^{\text{new}}; G_k) + \beta \, \| u_k^{\text{new}} - \widetilde{a}_k \|_2^2 \big]. \tag{C.15}$$

Here, $Q_T^{\text{act}} = \mathcal{A}_{\text{RLT}}(Q_{T,1}, Q_{T,2})$ uses the fixed actor aggregation rule of the selected RLT implementation. A minimum in the task TD target does not imply a minimum in the actor objective; the exact operator must be recorded. As in the main paper, $\eta = \lambda / Q_{\text{scale}} > 0$, with λ and Q_{scale} fixed within a run, and $\beta \geq 0$ is the reference-regularization weight.

The set $\mathcal{B}_A$ requires valid task and safety evaluations, required risk queries, and corresponding gradient paths for the new proposal. If safety evaluation is unavailable, the update does not fall back to maximizing task value alone. The complete reference always remains the regularization anchor. Updates preserve both action-gradient paths of the safety term in A.3 and do not differentiate through discrete protective operations.

C.5 Replay and Online Algorithms

Replay retains separate records of the original proposal and actual execution. The former is the action input to value learning; the latter supplies returns, events, durations, and execution diagnostics. A corrected action must not overwrite u_k.

Table C2 lists the minimum information groups required for each main-decision transition.

Table C2: Required information for each main-decision transition. Historical contexts must support fresh risk queries for newly sampled actions.

Information group	**Required stored content**
Decision and reference	$x_k, \widetilde{a}_k, u_k$; task and episode identifiers; main-decision time; behavior-policy version
Risk and monitoring	g_k and query validity; frozen ξ_k and validity of its monitoring fields
Execution outcome	Commands actually sent; interventions and reasons; R_k, m_k, e_k, c_k and their observation validity
Successor and boundaries	Pre-reset $x_{k+1}, \widetilde{a}_{k+1}, \xi_{k+1}$ for transitions that bootstrap; true-termination and truncation flags for every transition
Recoverable context	Immutable snapshot or versioned reference for G_k; expert set, scales, and query configuration; also G_{k+1} when successor safety queries are required

Historical context must restore the corresponding distance field, robot and payload state, contact mode, initial controller state, and expert-retrieval information. Immutable snapshots may be shared. A reference to a current map that has since been overwritten is not recoverable. This requirement does not introduce an additional learned dynamics model.

Missing LRM branches can be represented by explicit masks, but missing required geometry or event observations cannot be replaced with surrogate costs. Value and policy updates construct their valid sets separately according to the information each requires. Availability filtering can introduce selection bias, so exclusion rates and their distributions across tasks, risk levels, and stages must be recorded. Complete-episode costs are computed only for episodes with all required feedback and are not automatically unbiased estimates of population risk.

Algorithm C1 Online Interaction and Protected Execution

1 Freeze the VLA, visual readout module, expert library, and monitoring calibration statistics. For each task, initialize the policy, two task value networks, two safety value networks, and their targets. Set $A = I_D$ and fix the exploration scale, risk weight, and protection protocol.
2 At an actual main-decision boundary, read x_k, the complete reference $\widetilde{a}_k$, and G_k. Freeze ξ_k using only information already returned.
3 Sample the complete u_k once with $B = 1$. Query g_k for the full chunk before intervention, retain its validity, and submit the proposal for protected execution.
4 Update measurements and check Hard Constraints at every low-level control step. Send only commands that pass. Otherwise, cancel the remaining commands of the old plan and apply correction, braking, holding, or stopping according to the protocol. Recheck corrected commands.
5 Issue a read-only shadow query at the permitted within-chunk time. The main control loop does not wait for it. Returned results update the cache for subsequent decisions under B.3 without changing the already frozen current context.
6 Settle the transition at the next valid main-decision boundary or true termination. Aggregate R_k, m_k, e_k and executed commands. Construct the cost using Eq. (C.9) when inputs are valid, and store the original proposal, execution information, validity, and historical context.
7 Invoke Algorithm C2 at the configured update frequency and continue interaction. Learning can occur after each actual transition and need not wait for an entire episode to finish.

Algorithm C2 Value Learning and Structured Policy Updates

1 Sample replay, restore each sample's historical context, and check the fields required by the task and safety branches separately.
2 Use immediate targets for true terminal samples. For bootstrapping samples, draw successor actions using the fixed ρ_{boot} and recompute g'_k for the safety branch.
3 Construct $\mathcal{B}_T$ and $\mathcal{B}_S$ and update the corresponding value networks using Eqs. (C.11)–(C.14). Skip empty sets and detach all TD targets.
4 Resample policy proposals at current replay observations, sharing $B \sim \text{Bernoulli}(0.5)$ between the mean and correlation branches. Use historical G_k to reevaluate risk and gradient validity.
5 Construct the jointly valid set $\mathcal{B}_A$, freeze value-network parameters, and update the policy using Eq. (C.15) and Appendix A.3. If the set is empty, skip the policy update without substituting a task-only objective.
6 Synchronize target networks at the fixed frequency and according to the specified update rule. Record valid sample counts, exclusion reasons, and query overhead.

Complete-episode costs are used for evaluation and do not automatically adjust λ. This algorithm is not a constrained-optimization procedure that updates a Lagrange multiplier after every episode.

C.6 Parameter and Implementation Conventions

The current instantiation uses single-arm manipulation, $C = 10$, and $b_{\text{geom}} = 0.01\,\text{m}$. The action dimension follows the robot control interface and cannot be inferred from the term "single arm." Exploration scales, risk weights, normalization scales, discount factors, reference regularization, and monitoring parameters are fixed in advance for each task. Evaluation does not update the policy, value networks, expert library, or calibration statistics.

Structured-exploration variants use the same marginal scale, chunk length, and identity-correlation initialization. Controlled comparisons with Hard Constraints require the same protection protocol, with actual matching verified from run configurations. The full lower-triangular parameterization has a quadratic output size. Its latency, memory use, trainable parameter count, and query cost require measurement; a qualitative description of a lightweight actor is insufficient.

Reproducibility particularly requires the task actor aggregation operator $\mathcal{A}_{\text{RLT}}$, target-mask probability ρ_{boot}, parameter version used to sample target actions, and target-network update rule. These implementation settings remain unspecified and must not be filled in from assumptions about common practice.

APPENDIX D EXPERIMENTAL AND REPRODUCIBILITY PROTOCOLS

D.1 EVALUATION SCOPE AND INFORMATION ACCESS

Simulation evaluation covers LIBERO-Safety and twelve independently constructed manipulation scenarios, with results reported separately. LIBERO-Safety (Cui et al., 2026) contains four physical-safety suites: Affordance-Aware Grasping (AAG), Human–Robot Interaction (HRI), Tabletop Spatial Avoidance (TSA), and Free-Space Hand–Object Avoidance (FSHOA). Each suite includes five tasks at three difficulty levels. Semantic refusal tasks are outside the evaluation scope.

L0/L1 are used for adaptation and in-distribution evaluation. L2 is excluded from parameter updates, expert-library construction, risk calibration, and model selection. Training, validation, and test instances and seeds are registered separately, with independent splits. Evaluation freezes learned parameters and statistics while allowing monitor caches, information age, validity, and ACC smoothing states to evolve causally.

The main comparisons include $\pi_{0.5}$ supervised fine-tuning, GR00T N1.5 supervised fine-tuning, RLT, SafeVLA, SafeDojo, COX-Q, and SANCTUARY. RLT-based methods use the $\pi_{0.5}$ interface. Checkpoints, code versions, task adapters, demonstration splits, and trainable parameter counts must be recorded for each method. Comparisons across different backbones evaluate complete systems and are not pure algorithmic ablations on a common model.

The main simulation comparison retains each method's own safety mechanism rather than applying SANCTUARY Hard Constraints to every baseline. SFT and basic RLT do not use our $\mathcal{H}$; RLT + Hard Constraints is a separate controlled configuration. Access to sensors, demonstrations, safety labels, world-model data, and pretraining resources is disclosed separately. Hyperparameter-search and physical-interaction budgets are matched or reported.

An independent evaluator records completion, violations, and truncation according to native task objectives, contact rules, and event definitions. Neither g nor a safety value output serves as the sole safety criterion. Generic collision checks cannot replace task-specific conditions such as safe-grasping requirements. Event detection remains active through the frame in which the objective is achieved and is not disabled before success is recorded.

D.2 METRICS AND STATISTICAL REPORTING

Let y_i indicate task completion in test episode i, and let v_i^{vio} indicate at least one predefined physical violation. Task success rate, safe success rate, and violation episode rate are

$$\begin{aligned} \mathrm{TSR} &= \frac{1}{N}\sum_i y_i, \qquad \mathrm{VR} = \frac{1}{N}\sum_i v_i^{\mathrm{vio}}, \\ \mathrm{SSR} &= \frac{1}{N}\sum_i y_i(1 - v_i^{\mathrm{vio}}). \end{aligned} \tag{D.1}$$

Tables report percentages. SSR requires linking success and violation records within each episode and generally cannot be obtained by subtracting VR from TSR. An episode containing both completion and a violation can contribute to TSR but not to SSR.

LIBERO-Safety SR follows its native success criteria, which include task-specific safety conditions. It is equivalent to SSR in Eq. (D.1) only when the independent evaluator's violation scope matches those native conditions. The success column for custom scenarios uses TSR. VR is computed only over the applicable task set with independent violation definitions, with its denominator specified. Tasks lacking an event definition are marked not applicable rather than assigned zero violations.

Physical budget. The per-task online budget in actual control steps, evaluation interval, and control frequency remain to be specified. Braking, holding, and waiting count toward actual interaction time. Predicted chunk counts and optimizer steps cannot replace physical time. Offline training, generated data, human takeover, and computational overhead are reported separately.

Training safety. The quantity N_{vio} counts training episodes with a violation within a fixed budget. Event counts and durations per thousand physical control steps are also reported. Rules for merging

continuous contact and adjacent triggers are fixed before evaluation. For a specified protection mechanism, let N_{sub} count submitted original chunks and N_{int} count those that trigger at least one intervention. The protection intervention rate is

$$\text{IR} = \frac{N_{\text{int}}}{N_{\text{sub}}}. \tag{D.2}$$

Each IR must identify its protection mechanism and coverage. Resampling or recovery commands do not add extra counts to the intervention numerator. Methods without that mechanism are marked not applicable. IR values from different mechanisms are not direct comparisons under a shared protector. Geometric boundary violations by proposals, actual physical events, and interventions are recorded separately.

Learning efficiency. Frozen checkpoints are evaluated under a fixed validation configuration. Planned reporting includes the normalized area under the SSR curve (AUC) over a common physical-interaction interval. The SSR threshold and sustained-attainment rule are specified before testing; runs that do not attain the threshold are marked NR. Training event curves, AUC, and attainment times still require actual logs and cannot be inferred from final test VR. Shadow queries generate no physical samples. Their additional call counts and inference latency are recorded separately, with at least the median and 95th percentile of chunk-processing latency reported.

Statistical hierarchy. Simulation rates are first computed by task and difficulty within each training seed, followed by equal weighting across tasks. L0 and L1 receive equal weight when combined; custom scenarios also receive equal task weights. The hardware comparison uses one task. VR is aggregated only over the declared applicable set. Simulation seed and episode counts and the interval-estimation method remain to be specified. Analyses should preserve the hierarchy of training seeds, tasks, and paired test instances. Repeated testing is not independent training replication; a single checkpoint supports only uncertainty estimates for its own evaluation.

Scope of the results. Baselines with the same names have different values in the main comparison and ablation tables, so module gains cannot currently be inferred across those tables. Run identifiers, training conditions, and test instances must be reconciled before interpreting the differences. SANCTUARY's reported simulation VR of 0.0% describes observations under the reported evaluation conditions. Without sample counts and event-coverage information, it neither quantifies the population violation probability nor establishes risk-free training.

SFT does not perform online reinforcement learning, so online-training event statistics are not applicable. This absence of an online training phase should not be described as risk-free learning.

D.3 Definitions of the Seven Ablations

In Table D1, "Risk Potential Field" refers to g used for learning, excluding geometric checks internal to Hard Constraints. Soft Constraints comprise long-term safety value learning and safety-guided structured action-chunk exploration. LRM remains value-network context in every applicable variant.

Comparing RLT + Soft Constraints against SANCTUARY tests execution protection within the complete safety-learning system. Comparing RLT + Hard Constraints against SANCTUARY tests the complete safety-learning module under the same protection mechanism. These comparisons vary component combinations and cannot isolate the contribution of learning the correlation matrix.

The Risk-Potential-Field-only variant retains Expert Support, Boundary Margin, safety value networks, structured exploration, and Hard Constraints. It removes all LRM inputs, including derived validity, age, and coverage features, and retrains the safety value networks.

The LRM-only variant retains LRM, safety value networks, structured exploration, and Hard Constraints. It removes every contribution of g to the cost, value inputs, and action gradients, and learns with $c = e$. LRM does not replace event labels or directly serve as a cost. Removing the Risk Potential Field does not disable geometric checks within Hard Constraints. The reference action and its regularizer remain unchanged.

Table D1: Component settings for the seven ablations. Structured denotes Safety-Guided Structured Action-Chunk Exploration. Hard Constraints check modeled constraints and do not guarantee a zero physical-event rate; N/A denotes a component that is not applicable.

Configuration	Online RL	Risk Potential Field	LRM	Hard Constraints	Exploration
$\pi_{0.5}$	No	N/A	N/A	No	N/A
RLT	Yes	No	No	No	Independent
RLT + Soft Constraints	Yes	Yes	Yes	No	Structured
RLT + Hard Constraints	Yes	No	No	Yes	Independent
SANCTUARY	Yes	Yes	Yes	Yes	Structured
Soft Constraints (Risk Potential Field only)	Yes	Yes	No	Yes	Structured
Soft Constraints (LRM only)	Yes	No	Yes	Yes	Structured

All variants that change objectives or risk inputs require independent training. Disabling an input only at test time is not an equivalent ablation. Structured-exploration variants share the marginal scale, chunk length, identity-correlation initialization, and event and termination definitions. Terminating an episode after an event does not prevent that event.

Isolating the contribution of correlation structure additionally requires an independent-noise control with fixed marginal variance, the same Hard Constraints, and an identical budget. This control fixes $R = I_D$ while retaining the other learning components. It remains an additional experiment; the existing seven results do not establish that this test has been completed. Some variants with Hard Constraints still have nonzero VR. Their event logs should be examined for constraint-coverage and execution failures rather than rewriting those rates as zero.

D.4 Single-Task Real-Robot Evaluation

The hardware comparison uses one FR3 pick-and-place task and 50 trials per method (Table 3). Success rate is the fraction of successful trials; dangerous events count actual collisions. VLA's 19 collisions occurred in unsuccessful trials. SANCTUARY stopped hazardous motions before contact; these interventions are not counted as collisions. Detailed hardware settings, task criteria, and trial-level records remain to be documented.

D.5 Twelve Custom Simulation Scenarios

The twelve scenarios (S01–S12) follow the order in Figure 2 of the main paper, from left to right and top to bottom: cluttered pick-and-place; rigid-cup pouring; block manipulation; fragile-egg handling; drawer opening; soft-cup pouring; gear assembly; carton cutting (80); test-tube insertion; carton cutting (36); curved-path gluing; and banana pick-and-place. They are summarized separately from LIBERO-Safety. Figure descriptions establish these names, but demonstration frames do not determine the success thresholds, allowed-contact rules, or data splits used in evaluation.

Each scenario configuration must specify the task objective, success-maintenance condition, time limit, event-detection source and thresholds, allowed contact pairs and stages, protection coverage, recovery rules, ranges of initial states and payload parameters, training/validation/test splits, and environment version. Several scenarios may share a risk category; twelve scenarios do not imply twelve mutually exclusive risks.

The scenario set should be determined independently of the final method ranking. Per-task results must link at least to run identifiers, success and violation counts, timeouts, interventions, and valid sample counts. Aggregate means do not replace these auditable records. The archived demonstrations in Figure 2 illustrate scenario assets and do not constitute evidence of reinforcement-learning performance.

D.6 Reproducibility and Remaining Measurements

Fixed design choices and values that still require confirmation from actual configurations must remain distinct. Unspecified settings are neither defaults nor measurements. Every final result should be traceable through run identifiers to its configuration, checkpoint, and event logs.

Experimental records to be completed. Training safety and efficiency statistics, random seeds and confidence intervals, per-task simulation counts, real-robot trial and intervention logs, complete scenario specifications, and the correspondence between main-comparison and ablation runs must come from actual configurations or logs. Zero observed violations cannot replace these records.

Failure and missing-data analysis. At minimum, distinguish failures in task understanding, risky original proposals, risk-evaluation errors, gaps or failures in protection, perception anomalies, and timeouts. Record rates of invalid geometric queries, missing event observations, missing LLMD, and absent or expired ACC overlap. Report retention rates for each of the three valid learning sets. Complete costs are computed only for episodes with complete information, alongside the number of excluded episodes.

Related references. The visual readout and online-adaptation interface follow RLT (Xu et al., 2026), as cited in the main paper. LRM draws on VLA-FAIL (Seligmann et al., 2026), and distance-field queries draw on cuRoboV2 (Sundaralingam et al., 2026). References for LIBERO-Safety and the comparison methods are shared with the main paper.