WHEN TO WARN, NOT WHAT TO DO: CALIBRATED LANGUAGE WARNINGS FOR FROZEN WORLD-ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

World-action models (WAMs) combine future visual prediction and action generation for language-conditioned robot control. Collision safety requires anticipating unsafe contact and guiding responses that preserve task progress, yet collision records do not specify corrective actions. Learning recovery or language-feedback policies requires additional supervision or interaction. We propose risk-triggered language steering, which learns when to warn and guides a frozen WAM with a fixed prompt without corrective-action supervision. A classifier trained on collision-derived K-step warning labels reads risk from predicted-video and action hidden states. Unsafe-class split conformal calibration sets a threshold targeting unsafe recall under the unmodified policy. Alerts trigger regeneration of the pending chunk with the warning appended to the task instruction. Experiments with LingBot-VA on seen tasks across four SafeLIBERO simulation suites show 89.3–99.0% AUROC and mean unsafe recall meeting nominal targets. Online, mean collision avoidance rate rises from 9.4% to 18.8%, versus 3.8% with the same warning always active. Mean task success is 30.0%, versus 29.4% without intervention.

1 INTRODUCTION

World-action models (WAMs) combine future visual prediction with action generation for language-conditioned robot control (Li et al., 2026b; Kim et al., 2026; Ye et al., 2026). During manipulation, pursuing the instructed goal can bring the robot into unintended contact with surrounding obstacles (Hu et al., 2025; Fan et al., 2026a). WAMs' predictive representations offer a potential source of information about these interactions before actions are executed (Li et al., 2026b; Kim et al., 2026). Using this information for collision avoidance requires connecting risk anticipation to behavioral adjustments that preserve task progress.

This connection involves a supervision gap: collision records can supervise risk recognition, but do not specify corrective actions. Gu et al. (2026) detect task failures from VLA representations. Recovery learning trains corrective policies (Thananjeyan et al., 2021), while safety filters constrain actions (Hu et al., 2025). Language can also guide frozen policies. Jeong et al. (2026) learn a language-feedback policy through interactive search and distillation, using a calibrated improvement predictor to decide whether to steer. VLA-Corrector detects deviations from observation history, discards the pending chunk, and re-queries a frozen policy with stage-specific recovery prompts (Song et al., 2026). This motivates a question: *can collision-risk signals from a frozen WAM's predictive representations trigger a fixed warning for safer execution, without learning corrective actions or feedback content?*

To address this question, we propose *risk-triggered language steering*, which learns when to warn and relies on a frozen WAM to determine how to respond (Figure 1). The intuition is that video and action representations available before execution may contain cues about impending collisions. Collision timestamps provide supervision for reading out these cues, but do not specify corrective actions. A fixed caution prompt supplies safety guidance, which the WAM translates into scene-specific action adjustments using the ongoing task context.

Concretely, we train a lightweight MLP on predicted-video and action hidden states using collision-derived K-step warning labels. With the risk head fixed, split conformal calibration on held-out

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.

unsafe examples from the unmodified policy sets an alarm threshold targeting detection of a fraction $1 - \alpha$ of pre-collision moments under that policy. At runtime, an alert triggers regeneration of the pending chunk with the fixed caution prompt appended to the task instruction, preserving the executed visual–action history.

Empirically, we evaluate frozen LingBot-VA on seen tasks across four SafeLIBERO simulation suites. Offline, the risk head achieves AUROC of 89.3–99%, outperforming the evaluated action-consistency baselines. Calibrated triggers attain mean unsafe recall meeting the nominal $1 - \alpha$ targets under the unmodified policy. Online, risk-triggered steering increases mean collision avoidance rate (CAR) from 9.4% to 18.8%, whereas keeping the same warning always active yields 3.8%. Mean task success rate (TSR) is 30.0%, compared with 29.4% without intervention.

Contributions. The key contributions are summarized:

- We propose risk-triggered language steering for frozen WAMs, regenerating chunks with a fixed warning while preserving executed history, without corrective-action supervision.
- We construct a collision-supervised warning rule using K-chunk labels and predicted-video and action representations, with unsafe-class split conformal calibration targeting unsafe recall under the unmodified policy.
- Across four SafeLIBERO suites, we demonstrate effective risk discrimination and mean empirical unsafe recall meeting nominal targets. Online, selective warning-and-regeneration improves mean collision avoidance with similar mean task success, whereas persistent warnings reduce mean collision avoidance.

2 Problem Setup and Motivation

WAM Execution and Notations. A pretrained world-action model (WAM) π_θ couples visual prediction with language-conditioned action generation (Li et al., 2026b; Ye et al., 2026). At chunk step t, its execution context c_t encodes available observations and previously executed actions. Given a task instruction ℓ, the model generates

$$(\hat{V}_t, \hat{A}_t) \sim \pi_\theta(\cdot \mid c_t, \ell), \qquad \hat{A}_t = (\hat{a}_{t,1}, \ldots, \hat{a}_{t,H}),$$

where $\hat{V}_t$ represents predicted future frames in latent space and H counts control actions in the candidate chunk. Thus, t indexes chunks rather than individual control steps. In LingBot-VA, video generation precedes action generation (Li et al., 2026b), making both representations available before execution. Realized observations and applied actions update c_t for subsequent generation.

Collision Safety. Following SafeLIBERO, collision safety concerns unintended contact with surrounding obstacles during benign manipulation, excluding contacts required to manipulate task objects. Obstacle displacement serves as the collision indicator (Hu et al., 2025). For an executed trajectory τ, let $C(\tau) \in \{0, 1\}$ indicate whether a collision occurs under this criterion.

Following deployment-time interventions that reuse frozen robot policies (Hu et al., 2025; Jeong et al., 2026), the WAM backbone parameters θ remain fixed throughout subsequent training and deployment. Drawing on failure detection from policy rollouts (Gu et al., 2026), supervision here is restricted to observations, executed actions, and first-collision annotations from rollouts of the unmodified policy. These annotations support offline risk learning but do not specify corrective actions. At deployment, decisions rely only on information available before the pending actions are executed. The objective is to reduce the collision probability $\mathbb{P}(C(\tau) = 1)$ while maintaining the task success rate.

Motivations. At this pre-execution stage, predictive representations may support collision-risk assessment while the pending actions remain revisable. The next challenge is to use this assessment to adjust pending actions while continuing the task from its current execution context. How can such adjustments be elicited from the frozen policy without corrective-action supervision?

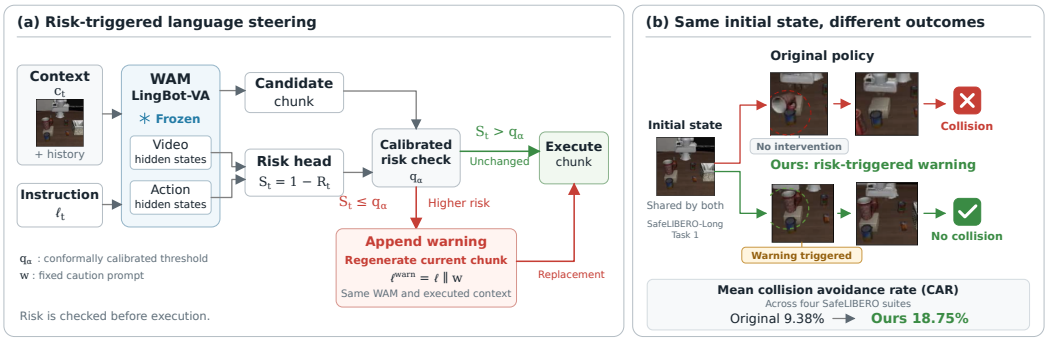


Figure 1: **Risk-triggered language steering for safer execution. (a)** A trained risk head computes S_t from frozen LingBot-VA's predicted-video and action hidden states. A candidate is executed unchanged if S_t exceeds the conformally calibrated threshold q_α. Otherwise, it is regenerated with a fixed warning appended to the task instruction, preserving executed context. **(b)** Rollouts on SafeLIBERO-Long task 1 from the same initial state. The original policy collides without intervention (top), while our method triggers a warning at the marked frame and avoids collision (bottom). CAR denotes collision avoidance rate; the reported values are averaged across four SafeLIBERO suites.

3 Risk-Triggered Language Steering

We first describe collision-supervised risk readout (Section 3.1) and calibration of warning triggers (Section 3.2). We then present language steering, including candidate regeneration, history preservation, and warning persistence (Section 3.3).

3.1 Collision-Supervised Risk Readout

To determine when to request an adjustment from the frozen policy, we learn a risk score from the predictive representations available before action execution. A lightweight classifier, termed the *risk head*, is supervised by collision annotations from unmodified-policy rollouts.

Collision-derived warning targets. Warning decisions concern whether the unmodified policy leads to contact in the near term. We therefore expand each first-collision annotation into a short window of decision points. For an unmodified-policy episode e, let ν_e index the chunk containing its first collision, with $\nu_e = \infty$ for a completed collision-free episode. Given a K-chunk horizon, we define

$$Y_{e,t}^{(K)} = \mathbf{1}\{0 \le \nu_e - t < K\}. \tag{1}$$

The target is positive when the first collision occurs in the current or next $K-1$ chunks. Completed collision-free episodes provide negative labels, including terminal windows shorter than K chunks. We exclude post-collision chunks and windows truncated before their outcomes are resolved.

Predictive representations. Collision risk depends on how the proposed motion interacts with the surrounding scene. We therefore combine the predicted-video and action representations of the pending candidate. Let U_t^v and U_t^a denote the corresponding hidden states at the final denoising evaluation, before output projection and after classifier-free guidance combines the text-conditioned and unconditioned branches. To obtain a chunk-level input, we average predicted-frame tokens in the video branch and executable-action tokens in the action branch, excluding the conditioning-frame prefix and its associated action tokens. We concatenate the pooled features:

$$Z_t = \mathrm{Norm}([\mathrm{Pool}(U_t^v); \mathrm{Pool}(U_t^a)]), \tag{2}$$

where Norm uses statistics estimated from training episodes.

Risk-head training. A multilayer perceptron g_ψ followed by a sigmoid produces the risk score

$$R_t = \sigma(g_\psi(Z_t)). \tag{3}$$

We optimize only ψ by minimizing mean binary cross-entropy over labeled training chunks using the targets in Eq. equation 1. Training, calibration, and evaluation use disjoint episodes. The trained head and normalization statistics remain fixed during threshold calibration (Section 3.2) and deployment.

Algorithm 1 Risk-triggered language steering

Require: Frozen WAM π_θ, risk head g_ψ, threshold q_α, task instruction ℓ, warning w, durations $L_{\mathrm{w}}, L_{\mathrm{c}}$, b, d *count remaining warning and cooldown chunks; instruction switches preserve* c_t.

Initialize c_1; $b \leftarrow 0$; $d \leftarrow 0$; $\ell^{\mathrm{warn}} \leftarrow \ell \,\|\, w$
for $t = 1, 2, \ldots$ until termination **do**
  $\ell_t \leftarrow \ell^{\mathrm{warn}}$ if $b > 0$, else ℓ
  Generate $(\widehat{V}_t, \widehat{A}_t) \sim \pi_\theta(\cdot \mid c_t, \ell_t)$
  Extract Z_t; $S_t \leftarrow 1 - \sigma(g_\psi(Z_t))$
  if $b = 0$ **and** $d = 0$ **and** $S_t \le q_\alpha$ **then**
    $b \leftarrow L_{\mathrm{w}}$; $d \leftarrow L_{\mathrm{c}}$; $\ell_t \leftarrow \ell^{\mathrm{warn}}$
    Regenerate $(\widehat{V}_t, \widehat{A}_t) \sim \pi_\theta(\cdot \mid c_t, \ell_t)$
  end if
  Execute $\widehat{A}_t$; update c_{t+1} from realized interaction under ℓ_t
  if $b > 0$ **then**
    $b \leftarrow b - 1$
  else if $d > 0$ **then**
    $d \leftarrow d - 1$
  end if
end for

3.2 Unsafe-Class Conformal Calibration

To turn the learned risk score into a warning decision, we select a threshold targeting a missed-detection level $\alpha \in (0, 1)$ within the unsafe class $Y^{(K)} = 1$.

Threshold construction. Define $S_t = 1 - R_t$, so lower scores indicate higher risk. From held-out unmodified-policy episodes, collect the n_u scores with $Y^{(K)} = 1$ and order them as $S_{(1)} \le \cdots \le S_{(n_u)}$. We choose q_α so that the warning region $S_t \le q_\alpha$ covers unsafe scores at the target level $1 - \alpha$. With a finite-sample correction for a new inference, unsafe-class split conformal calibration (Vovk et al., 2005; Angelopoulos & Bates, 2022) sets

$$k_\alpha = \lceil (n_u + 1)(1 - \alpha) \rceil, \qquad q_\alpha = S_{(k_\alpha)}, \tag{4}$$

where $S_{(n_u+1)} = 1$ handles the case $k_\alpha = n_u + 1$. The warning rule is

$$D_t = \mathbf{1}\{S_t \le q_\alpha\}. \tag{5}$$

Theorem 3.1 (Unsafe missed-detection bound)**.** *Fix the trained risk head and normalization statistics. Let S_{new} be the score of a new test inference from the unmodified policy, with warning target $Y^{(K)}_{\mathrm{new}}$. Suppose that, conditional on n_u and $Y^{(K)}_{\mathrm{new}} = 1$, the unsafe calibration scores and S_{new} are jointly exchangeable. Then the threshold in Eq. 4 satisfies*

$$\mathbb{P}\Big(S_{\mathrm{new}} > q_\alpha \,\Big|\, Y^{(K)}_{\mathrm{new}} = 1\Big) \le \alpha. \tag{6}$$

Remark 3.2. Theorem 3.1 guarantees unsafe recall of at least $1 - \alpha$ under the unmodified policy, averaged over calibration and test draws. Exchangeability is required for the unsafe scores themselves; splitting episodes alone does not establish it for pooled within-episode windows. Smaller α favors unsafe recall, potentially increasing intervention frequency.

The trigger determines when to request an adjustment. The next subsection describes how a fixed caution prompt guides regeneration of the pending chunk.

3.3 Language Steering with Preserved Execution History

Warning-conditioned regeneration. We request the adjustment through language, leaving the specific motion to the frozen WAM. To keep the task goal explicit, we append a fixed, task-independent warning w to ℓ, forming $\ell^{\mathrm{warn}} = \ell \| w$:

> *Avoid colliding with obstacles, but keep making progress on the task. Do not stop or freeze; complete the instruction carefully.*

When neither warning nor cooldown is active, $D_t = 1$ triggers regeneration of a video–action pair from the same execution context c_t under ℓ^{warn}, replacing the pending candidate. Otherwise, the candidate is retained. The replacement action chunk is executed without another trigger check.

Preserving executed context. To continue from the current stage of the task, instruction switches change the text conditioning for new generation while retaining the executed history in c_t. Discarded candidates never enter this history; only realized observations and applied actions update the context after execution.

Warning persistence and cooldown. An adjustment may span several chunks, so each trigger starts a fixed window of L_{w} executed chunks, including the regenerated chunk. Generation uses ℓ^{warn} throughout this window. The controller then restores ℓ for a cooldown of L_{c} chunks to limit intervention frequency. Risk scoring continues throughout both windows, but scores cannot trigger another regeneration or refresh the warning window. Interventions are enabled again after the cooldown. The threshold q_α remains fixed, but the bound in Section 3.2 does not cover the altered execution distribution after intervention.

Algorithm 1 summarizes the online execution loop. The controller state determines the active instruction and whether the scored candidate may be replaced. After executing the selected chunk, the controller updates the realized interaction history and advances the warning or cooldown counter. The loop runs until episode termination. If execution continues after a collision, subsequent scores fall outside the first-collision training target.

4 Experiments

Our experiments address four questions: (I) Where is collision risk encoded, and how well can it be decoded? (II) Does empirical unsafe recall meet the nominal targets? (III) Can our method improve safety without fine-tuning? (IV) How efficient and practical is deployment?

4.1 Experimental Setup

Environment and Task. We evaluate on SafeLIBERO (Hu et al., 2025), a safety-oriented benchmark extending LIBERO (Liu et al., 2023), with four tasks per suite: Long (long-horizon planning), Spatial (spatial reasoning), Goal (goal-conditioned tasks), and Object (object manipulation). Each task includes multiple initial scene configurations, with an obstacle randomly selected from a diverse object set and placed near the target object.

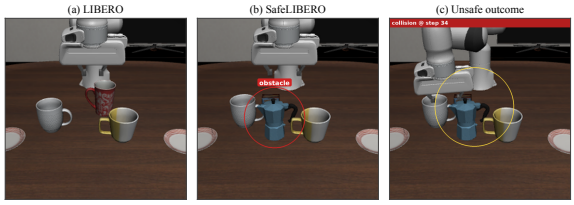


Figure 2: "Mug on Plates" in LIBERO- and SafeLIBERO-Long. The first detected obstacle displacement marks an unsafe event. Evaluation uses held-out, in-distribution initial scene configurations.

Datasets and Implementation Details. We collect 50 rollouts per task in each suite, treating any positional displacement of an obstacle as a collision and assigning chunk-level safety labels using a K-chunk look-ahead labeling rule. The collected rollouts are randomly partitioned into three disjoint subsets for risk-head training, conformal calibration, and testing. We use the frozen LingBot-VA (Li et al., 2026a) as the backbone for all experiments, with suite-specific LingBot-VA checkpoints[1]. We conduct in-task evaluations in simulation, with all experiments run on a single NVIDIA GH200 GPU with 120 GB of memory. More details can be seen in Appendix A.2

4.2 Results

4.2.1 Where is the collision-risk signal, and how well can it be decoded?

Evaluation Protocol: Risk Readout Baselines. To locate collision-risk signals, all detectors are trained and evaluated on the same K-window collision labels and the same train/test datasets.

[1]Long, Spatial, Goal, and Object.

(1) *Action consistency.* Following SAFE Gu et al. (2026), we assign higher risk to greater disagreement among $N{=}4$ sampled actions. (2) *Different features.* We test SAFE-style features Gu et al. (2026)—the last-layer hidden state of the final action token at each step—with the same readout (MLP). Replacing SAFE's episode-level task-failure targets with our step-level collision labels isolates the feature source. We also evaluate current observations (frozen visual encoder, no imagined rollout) and world-model action features with the same readout. (3) *Different readouts.* We compare the MLP with cosine and Euclidean k-NN embedding-distance methods adapted to collision risk (Appendix A.2). (4) The *label-free reference* t/T tests whether episode timing alone predicts collision.

Results: video–action features support collision-risk discrimination across readouts. Table 1 shows that rotation variance achieves only 45.0–61.0% AUROC, suggesting that disagreement among sampled actions is a weak proxy for imminent collision. Holding the MLP fixed, video–action features outperform the SAFE-style features on every suite, by 1.6–6.3 AUROC and 0.6–9.3 AUPRC points. Observation-only and action-only features also underperform the joint representation, supporting the benefit of combining predicted-video and action information. On the same video–action features, cosine and Euclidean k-NN perform comparably to the trained MLP, showing that collision-risk information is also accessible through feature-space comparisons. This comparison checks the representation across different readouts; the MLP remains our risk head for online steering.

Table 1: **Collision-risk discrimination.** AUROC and AUPRC on held-out samples from four suites, varying feature sources and, in the final block, readouts on world-model video–action features. Chance AUPRC equals each suite's unsafe prevalence. Best in **bold**, second-best underlined. Action consistency shows limited discrimination. (Only the strongest action-consistency baseline is shown; others appear in Appendix A.3.) With matched labels, splits, and MLP architecture, video–action features outperform SAFE-style last action-token states Gu et al. (2026) across all suites. On video–action features, k-NN performs comparably to the MLP.

	Long		Spatial		Goal		Object	
Feature source / readout	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC
Label-free reference								
Chance (AUPRC = unsafe prevalence)	50.0%	24.2%	50.0%	61.1%	50.0%	68.2%	50.0%	45.7%
Time index t/T	9.9%	14.5%	40.9%	54.3%	49.0%	75.8%	5.5%	28.7%
Action consistency								
Rotation variance	45.0%	29.3%	61.0%	70.6%	58.9%	75.8%	55.6%	55.4%
Different features								
SAFE-style features	96.0%	86.2%	83.0%	87.3%	95.1%	97.2%	96.8%	96.7%
Current observation only	90.9%	74.2%	67.7%	82.0%	87.0%	94.0%	97.9%	98.2%
World-model action feature only	97.2%	91.2%	89.0%	93.7%	95.6%	97.4%	96.6%	96.8%
***Ours** world-model video–action features, different readout*								
k-NN, cosine	**98.8%**	**96.8%**	90.0%	93.9%	98.2%	99.1%	98.8%	98.5%
k-NN, Euclidean	98.4%	95.3%	**90.3%**	94.0%	**98.5%**	**99.2%**	98.8%	98.4%
MLP risk head	98.7%	95.5%	89.3%	**94.1%**	96.7%	97.8%	**99.0%**	**98.8%**

4.2.2 Does empirical unsafe recall meet the nominal targets?

Evaluation Protocol: Conformal Metrics. Let b denote evaluation repetition and $\mathcal{I}_{\mathrm{u}}$ index the valid unsafe test steps. To evaluate whether the detector attains the expected risk detection level across four suites, we report the marginal coverage rate (Angelopoulos & Bates, 2022):

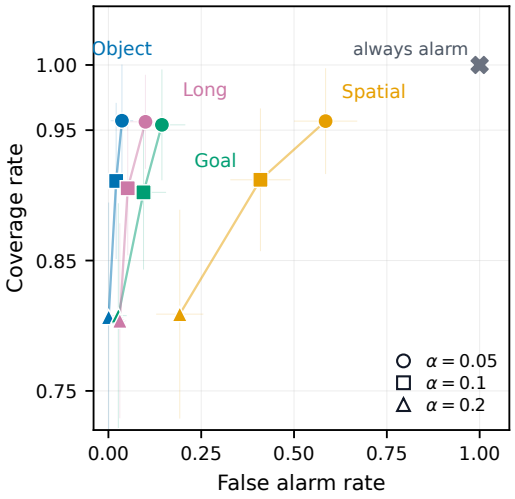
$$\mathrm{cov}_{\alpha}^{(b)} = \frac{1}{|\mathcal{I}_{\mathrm{u}}^{(b)}|} \sum_{i \in \mathcal{I}_{\mathrm{u}}^{(b)}} \mathbf{1}\Big\{ S_i^{(b)} \le q_{\alpha}^{(b)} \Big\}. \tag{7}$$

where q_α denote the corresponding calibration threshold on the $S = 1 - R$ scale with R denoting the risk score of detector.

Results: our method achieves empirical recall and false-alarm trade-offs. Figure 3 (a) shows that mean empirical recall closely tracks the expected level across suites, supporting empirical agreement with the theoretical risk detection levels under different desired α. The paired false-alarm rates (Figure 3 (b)) characterize the empirical recall–false-alarm trade-off under the current evaluation protocol. Notably, the mean and standard deviation summarize variability across repartitions; they do not establish a lower bound for each individual split.

	α / target $(1-\alpha)$		
	0.05 / 0.95	0.10 / 0.90	0.20 / 0.80
Object	$0.9572^{\checkmark}$ ± 0.04	$0.9110^{\checkmark}$ ± 0.06	$0.8066^{\checkmark}$ ± 0.09
Spatial	$0.9569^{\checkmark}$ ± 0.04	$0.9119^{\checkmark}$ ± 0.05	$0.8089^{\checkmark}$ ± 0.08
Goal	$0.9541^{\checkmark}$ ± 0.04	$0.9023^{\checkmark}$ ± 0.06	$0.8077^{\checkmark}$ ± 0.09
Long	$0.9564^{\checkmark}$ ± 0.04	$0.9054^{\checkmark}$ ± 0.05	$0.8038^{\checkmark}$ ± 0.07
Avg.	$0.9562^{\checkmark}$	$0.9077^{\checkmark}$	$0.8068^{\checkmark}$

(a) Unsafe recall



(b) Recall–false alarm trade-off

Figure 3: **Empirical unsafe recall and false-alarm trade-offs.** (a) Mean ± std coverage rate (cov) over 100 random 50:50 splits of the test data. The reported mean recall meets the nominal $1-\alpha$ target (✓) in each setting. (b) Same thresholds on the same splits: coverage vs. false-alarm rate $P(\text{alarm} \mid \text{safe})$. Always-alarm sits at $(1, 1)$.

4.2.3 Can our method achieve safety enhancement without fine-tuning?

Evaluation Protocol: Metrics. We assess safety and task performance with three metrics (Hu et al., 2025; Yeke et al., 2026): (1) *collision avoidance rate* (CAR), the percentage of collision-free episodes; (2) *task success rate* (TSR), the percentage of episodes completing the task within the time limit; and (3) *success-safety harmonic mean* (SSH), $\text{SSH} = 2\,\text{CAR}\,\text{TSR}/(\text{CAR}+\text{TSR})$. SSH emphasizes the lower rate, penalizing imbalance and rewarding jointly high CAR and TSR.

Evaluation Protocol: Baselines. We compare three inference-time baseline families on the same frozen backbone, without baseline-specific training: (1) *Second Chance* uniformly selects and executes a resampled candidate, following random-selection controls in prior VLA studies (Kwok et al., 2025; Sun et al., 2026a). It tests resampling without risk-guided selection. (2) *Best-of-N* follows sample-and-rank (Qi et al., 2026; Du & Song, 2026), reusing our trained collision-risk head to select the lowest-risk candidate and test the benefit of risk-guided selection. *Prompt Intervention*, inspired by fixed safety reminders evaluated in RoboJailBench (Yeke et al., 2026), retains the warning of CP-prompt in the task instruction throughout each episode, without conformal gating. It thus tests whether the gains of CP-prompt stem from the risk-aligned timing of the warning rather than its content alone. The sampling baselines generate $N = 4$ candidates under the unchanged instruction at Bernoulli-triggered events at least five action chunks apart. We calibrate the trigger probability so that expected resampling events per episode match the mean re-inference count of CP-prompt. Action- and video-resampling variants vary one source of randomness while holding the other fixed.

Results: improved collision avoidance and safety–success across all four suites. Table 2 shows that our method achieves the highest CAR and SSH across all four suites, increasing mean CAR by 9.37 percentage points over LingBot-VA with similar mean TSR. The SSH gains indicate a better aggregate safety–success trade-off. These gains are not reproduced by the evaluated sampling baselines or by keeping the same warning active throughout execution. *Second Chance* yields suite-dependent gains, while *Best-of-N* does not consistently outperform random selection. Together with the offline detection results, this suggests that strong risk discrimination does not necessarily translate into safer candidate selection. *Prompt Intervention* lowers both CAR and SSH relative to the base policy in most suites, suggesting that persistent warnings can be counterproductive in this setting. Clean-LIBERO success rates are reported in Appendix A.3.

Table 2: **Simulation evaluation on SafeLIBERO.** Metrics are collision avoidance rate (CAR), task success rate (TSR), and success-safety harmonic mean (SSH). Avg. is the unweighted suite mean. Evaluation uses 10 episodes per task, with matched initial states and a shared seed across methods. Best results are **bold**. Our method achieves the highest CAR and SSH in every suite. Paired bootstrap intervals (Appendix A.3) show that the aggregate gains are robust to episode-level resampling.

SafeLIBERO Suites	Metric	Backbone (Lingbot-VA)	Second Chance		Best-of-N		Prompt Intervention	**Ours**
			Action	Vision	Action	Vision		
Long	CAR ↑	12.50%	15.00%	22.50%	20.00%	22.50%	7.50%	**25.00%**
	TSR ↑	17.50%	17.50%	12.50%	22.50%	**27.50%**	20.00%	25.00%
	SSH ↑	14.58%	16.15%	16.07%	21.18%	24.75%	10.91%	**25.00%**
Spatial	CAR ↑	12.50%	12.50%	12.50%	7.50%	12.50%	2.50%	**20.00%**
	TSR ↑	62.50%	67.50%	**72.50%**	57.50%	52.50%	47.50%	55.00%
	SSH ↑	20.83%	21.09%	21.32%	13.27%	20.19%	4.75%	**29.33%**
Goal	CAR ↑	2.50%	2.50%	0.00%	0.00%	0.00%	2.50%	**15.00%**
	TSR ↑	27.50%	25.00%	30.00%	**37.50%**	22.50%	32.50%	30.00%
	SSH ↑	4.58%	4.55%	0.00%	0.00%	0.00%	4.64%	**20.00%**
Object	CAR ↑	10.00%	7.50%	12.50%	12.50%	7.50%	2.50%	**15.00%**
	TSR ↑	**10.00%**	2.50%	5.00%	2.50%	2.50%	5.00%	**10.00%**
	SSH ↑	10.00%	3.75%	7.14%	4.17%	3.75%	3.33%	**12.00%**
Avg.	CAR ↑	9.38%	9.38%	11.88%	10.00%	10.63%	3.75%	**18.75%**
	TSR ↑	29.38%	28.13%	**30.00%**	**30.00%**	26.25%	26.25%	**30.00%**
	SSH ↑	12.50%	11.38%	11.13%	9.66%	12.17%	5.91%	**21.58%**

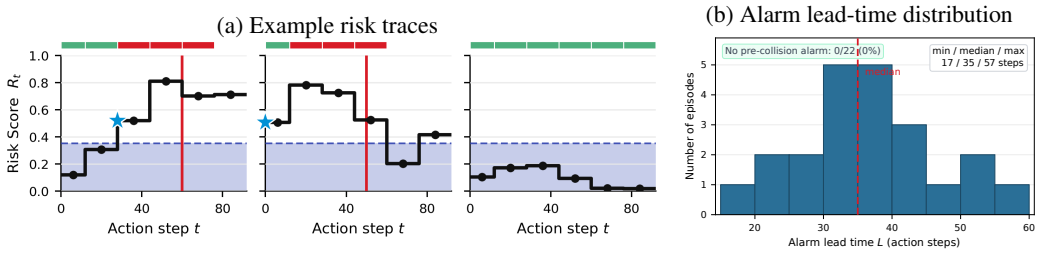


Figure 4: **Safety warnings align with robot safety.** (a) Randomly selected monitoring-only rollouts. Scores above the blue CP band trigger alarms. Blue stars mark first alarms; red/green strips label unsafe/safe chunks. *Left:* risk exits the band shortly before collision. *Middle:* a first-chunk alarm precedes an early collision. *Right:* risk stays within the band throughout a collision-free rollout. (b) Alarm lead-time distribution for 22 colliding test episodes. Additional results are in Appendix A.3.

4.2.4 How efficient and practical is our method to deploy?

Results: our method maintains competitive execution times, while achieving higher safety performance. Table 3 shows that our method remains competitive with the sampling-based baselines, achieving the lowest execution time on Long and running faster than action-based Best-of-N across all four suites. Compared with persistent Prompt Intervention, our method requires more execution time in three suites but achieves higher CAR and SSH in every suite (Table 2). These results demonstrate an empirical trade-off: selective re-inference incurs additional computation while delivering more

Table 3: Mean execution time (s/ep). Our method maintains competitive execution times.

Method	Long	Spatial	Goal	Object
SC (A)	318.8	185.6	309.7	191.7
SC (V)	343.5	192.4	323.9	183.0
BoN (A)	516.0	201.4	207.2	274.7
BoN (V)	453.5	167.8	206.8	274.3
Prompt Interv.	388.9	**144.0**	**149.3**	**166.9**
Ours	**315.4**	170.6	163.2	192.7

SC: Second Chance; BoN: Best-of-N. A/V: action/vision resampling.

effective safety intervention than an always-on warning. Notably, both risk-head training and conformal calibration are computationally lightweight, involving only the optimization of a small MLP and quantile-based threshold estimation, respectively, with the backbone kept frozen throughout.

Results: the calibrated risk signal can provide an timely intervention before a collision, while remaining inactive in the illustrated safe rollout. We visualize rollouts under safety detection to examine whether our method can issue warnings before collisions. In Fig. 4 (a), the risk score crosses the conformal threshold before the observed collision, providing advance warning as the rollout evolves, while the collision-free rollout remains below the threshold throughout execution, producing no false alarms. Fig. 4 (b) shows that each of the colliding episodes triggers an alarm before its first collision, with a median lead time of 35 action steps (approximately 1.75 seconds).

5 Related Work

Failure Detection and Risk Calibration. Failure monitors track action consistency and task progress (Agia et al., 2024) or probe policy representations (Gu et al., 2026; Park et al., 2026; Rajaprakash et al., 2026). Foresight uses action-conditioned world-model latents and terminal task labels (Zhang et al., 2026b); FARM probes frozen VLA-JEPA predictive states (Pei et al., 2026a). SAFE calibrates on successful rollouts to control false alarms (Gu et al., 2026), while Luo et al. (2024) calibrate warnings on unsafe examples to bound missed warnings. SAFECAST perturbs training and calibration data for robustness under shift (Rajaprakash et al., 2026). Trajectory-level calibration targets temporal specifications (Lindemann et al., 2023). We predict collisions within K chunks and calibrate unsafe scores for marginal per-inference missed-detection control under the nominal policy, assuming exchangeable unsafe calibration and test scores.

Safety Interventions and Recovery. SafeVLA uses constrained reinforcement learning; SafeDojo uses world-model rollouts (Zhang et al., 2026a; Tang et al., 2026). Other interventions use geometric filters (Hu et al., 2025), activation steering (Hong et al., 2026), or latent safety policies learned without corrective demonstrations (Nakamura et al., 2025). Pre-VLA verifies chunks for resampling (Sun et al., 2026b); DreamAvoid gates candidate generation and learned future evaluation during critical phases (Fan et al., 2026b). AEGIS uses probes of frozen activations and conformal-style thresholds to invoke a stronger policy (Chen, 2026); SafeLoop predicts hazards and rolls back before requerying (Lou et al., 2026). CheckVLA repairs chunk suffixes using a separate action-conditioned world model with episode-level false-alarm control (Liu et al., 2026). We regenerate pending chunks through the same frozen WAM's language interface.

Language Steering. RT-H supports human corrections through learned language-motion hierarchies (Belkhale et al., 2024). For frozen policies, Jeong et al. (2026) learn closed-loop language feedback and calibrate an initial-state improvement gate to limit harmful steering. VLA-Corrector uses a learned history-based verifier to trigger stage-specific recovery prompts and replace pending chunks through the same frozen policy (Song et al., 2026). SafeStage reports limited benefits from generic safety prompts and reduced task success with scenario-specific prompts (Luo et al., 2026). We use collision timestamps to learn a chunk-level trigger from WAM video and action states, with a fixed warning and no corrective-action supervision.

6 Conclusion and Limitations

We present risk-triggered language steering for frozen WAMs. The contrast between selective warning-and-regeneration and persistent prompting highlights the importance of evaluating a warning together with the procedure that applies it. This finding motivates studying warning triggers together with the responses they elicit, particularly whether intervention benefit can guide warnings more effectively than nominal collision risk alone.

Evaluation uses LingBot-VA on seen SafeLIBERO tasks in simulation with one hand-written warning. Displacement serves as a proxy for unsafe contact. Mean collision avoidance improves but remains low, and similar mean task success does not establish preservation across suites. Risk-triggered regeneration without warnings and matched risk-independent warning schedules remain future work. Under score exchangeability, calibration bounds marginal per-inference missed-detection probability under the unmodified policy. It does not guarantee collision avoidance after intervention.

AI USE STATEMENT

In this work, we used generative AI tools to assist with manuscript writing and language polishing, and to support literature retrieval and identification of related work. The authors reviewed and revised all AI-assisted text to preserve the intended technical meaning and verified suggested references against the original sources for accuracy and relevance. We take responsibility for the final content of this work, including all text, claims, citations, and artifacts produced with AI assistance.

ETHICS STATEMENT

This work aims to improve safety in benign robot manipulation by reducing unintended collisions. All experiments are conducted in simulation using SafeLIBERO, without deployment on physical robots.

REPRODUCIBILITY STATEMENT

Our method is described in Section 3 and Algorithm 1. Experimental protocols and implementation details are provided in Section 4 and Appendix A.2, and the proof of Theorem 3.1 appears in Appendix A.1. We provide anonymized code in the supplementary material to support reproduction of our results.

REFERENCES

Christopher Agia, Rohan Sinha, Jingyun Yang, Zi-ang Cao, Rika Antonova, Marco Pavone, and Jeannette Bohg. Unpacking failure modes of generative policies: Runtime monitoring of consistency and progress. *arXiv preprint arXiv:2410.04640*, 2024.

Anastasios N. Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification, 2022. URL https://arxiv.org/abs/2107.07511.

Suneel Belkhale, Tianli Ding, Ted Xiao, Pierre Sermanet, Quon Vuong, Jonathan Tompson, Yevgen Chebotar, Debidatta Dwibedi, and Dorsa Sadigh. Rt-h: Action hierarchies using language, 2024. URL https://arxiv.org/abs/2403.01823.

Josef Chen. Aegis: A backup reflex for physical ai. *arXiv preprint arXiv:2606.06660*, 2026.

Maximilian Du and Shuran Song. Dynaguide: Steering diffusion polices with active dynamic guidance. *Advances in Neural Information Processing Systems*, 38:44192–44221, 2026.

Jialiang Fan, Weizhe Xu, Oleg Sokolsky, Insup Lee, and Fanxin Kong. Safevla-bench: A benchmark for the success-safety gap in vision-language-action models. *arXiv preprint arXiv:2606.00773*, 2026a.

Xianzhe Fan, Yuxiang Lu, Shenyuan Gao, Xiaoyang Wu, Ruihua Han, Manling Li, and Hengshuang Zhao. Dreamavoid: Critical-phase test-time dreaming to avoid failures in vla policies. *arXiv preprint arXiv:2605.11750*, 2026b.

Qiao Gu, Yuanliang Ju, Shengxiang Sun, Igor Gilitschenski, Haruki Nishimura, Masha Itkina, and Florian Shkurti. Safe: Multitask failure detection for vision-language-action models. *Advances in Neural Information Processing Systems*, 38:40041–40076, 2026.

Jihoon Hong, Julian Skifstad, Qiyue Dai, Alice Chan, and Glen Chou. Steering robustness into world action models via mechanistic interpretability and optimal control. *arXiv preprint arXiv:2607.14943*, 2026.

Songqiao Hu, Zeyi Liu, Shuang Liu, Jun Cen, Zihan Meng, Shihefeng Wang, Xiang Li, and Xiao He. Vlsa: Vision-language-action models with plug-and-play safety constraint layer. *arXiv preprint arXiv:2512.11891*, 2025.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael 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, 2025. URL https://arxiv.org/abs/2504.16054.

Hyun Joe Jeong, Gokul Swamy, and Andrea Bajcsy. Learning what to say to your vla: Mostly harmless vision language action model steering. *arXiv preprint arXiv:2606.12299*, 2026.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model, 2024. URL https://arxiv.org/abs/2406.09246.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, et al. Cosmos policy: Fine-tuning video models for visuomotor control and planning. *arXiv preprint arXiv:2601.16163*, 2026.

Jacky Kwok, Christopher Agia, Rohan Sinha, Matt Foutter, Shulu Li, Ion Stoica, Azalia Mirhoseini, and Marco Pavone. Robomonkey: Scaling test-time sampling and verification for vision-language-action models, 2025. URL https://arxiv.org/abs/2506.17811.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control, 2026a. URL https://arxiv.org/abs/2601.21998.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, et al. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026b.

Lars Lindemann, Xin Qin, Jyotirmoy V Deshmukh, and George J Pappas. Conformal prediction for stl runtime verification. In *Proceedings of the ACM/IEEE 14th International Conference on Cyber-Physical Systems (with CPS-IoT Week 2023)*, pp. 142–153, 2023.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning, 2023. URL https://arxiv.org/abs/2306.03310.

Yushan Liu, Peibo Sun, Xintao Chao, Zhenyang Yang, Yifan Xie, Lingfeng Zhang, Shoujie Li, Chenyu Tang, Fang Chen, Xiao-Ping Zhang, et al. Checkvla: Execution-time verification with action-conditioned world model for long-horizon mobile manipulation. *arXiv preprint arXiv:2607.26789*, 2026.

Zeyu Lou, Tianran Zhang, Xinquan Yue, Ya Jing, and Chenyang Si. Safeloop: Risk-aware rollback for vision-language-action manipulation, 2026. URL https://arxiv.org/abs/2609.26313.

Jinzhu Luo, Qi Zhang, Wei Wang, and Wei Jiang. Safestage: Evaluating safety before, during, and after vision-language-conditioned robot manipulation, 2026. URL https://arxiv.org/abs/2609.21223.

Rachel Luo, Shengjia Zhao, Jonathan Kuck, Boris Ivanovic, Silvio Savarese, Edward Schmerling, and Marco Pavone. Sample-efficient safety assurances using conformal prediction. *The International Journal of Robotics Research*, 43(9):1409–1424, 2024.

Kensuke Nakamura, Lasse Peters, and Andrea Bajcsy. Generalizing safety beyond collision-avoidance via latent-space reachability analysis. *arXiv preprint arXiv:2502.00935*, 2025.

Seongheon Park, Wendi Li, Changdae Oh, Samuel Yeh, Zsolt Kira, Michael Hagenow, and Sharon Li. Hide-and-seek in trajectories: Discovering failure signals for vla runtime monitoring. *arXiv preprint arXiv:2605.30834*, 2026.

Haoran Pei, Mingrui Luo, Senbao Wang, Haoran Lv, Jie Guo, Sheng Zhong, and Ruixi Ci. Farm: Reading failure signals from the internal predictive states of a frozen robotic world model, 2026a. URL https://arxiv.org/abs/2609.11445.

Haoran Pei, Mingrui Luo, Senbao Wang, Haoran Lv, Jie Guo, Sheng Zhong, and Ruixi Ci. Farm: Reading failure signals from the internal predictive states of a frozen robotic world model, 2026b. URL https://arxiv.org/abs/2609.11445.

Han Qi, Haocheng Yin, Aris Zhu, Yilun Du, and Heng Yang. Inference-time enhancement of generative robot policies via predictive world modeling. *IEEE Robotics and Automation Letters*, 2026.

Harshitha Rajaprakash, Aditeya Prajapati, Rong Xue, Abrar Anwar, and Jesse Thomason. Safecast: Robust failure detection for vla policies with contrast-set training and calibration. *arXiv preprint arXiv:2608.04246*, 2026.

Chang Song, Bin Qian, Yan Feng, and Zhijie Song. Vla-corrector: Stage-aware observable state understanding for prompt-based closed-loop recovery of vision-language-action policies. *arXiv preprint arXiv:2609.06508*, 2026.

Zhen Sun, Yongjian Guo, Haoran Sun, Luqiao Wang, Wei Lu, Jiachi Ji, Shengzhe Ji, Junwu Xiong, and Zhijun Meng. Pre-vla: Preemptive runtime verification for reliable vision-language-action and world-model rollouts. *arXiv preprint arXiv:2605.22446*, 2026a.

Zhen Sun, Yongjian Guo, Haoran Sun, Luqiao Wang, Wei Lu, Jiachi Ji, Shengzhe Ji, Junwu Xiong, and Zhijun Meng. Pre-vla: Preemptive runtime verification for reliable vision-language-action and world-model rollouts, 2026b. URL https://arxiv.org/abs/2605.22446.

Kai Tang, Peidong Jia, Zhong Chu, Jixian Wu, Rui Ma, Jiajun Cao, Fangyuan Zhao, Sixiang Chen, Yichen Guo, Xiaowei Chi, et al. Safedojo: Safe reinforcement learning for vla via interactive world model. *arXiv preprint arXiv:2606.20698*, 2026.

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*, 6(3): 4915–4922, 2021.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*. Springer, 2005.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026.

Doguhuan Yeke, Yanming Zhou, Leo Y. Lin, Hongyu Cai, Antonio Bianchi, and Z. Berkay Celik. Robojailbench: Benchmarking adversarial attacks and defenses in embodied robotic agents, 2026. URL https://arxiv.org/abs/2605.19328.

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. *Advances in Neural Information Processing Systems*, 38:153335–153373, 2026a.

Haoran Zhang, Yifu Lu, Boyang Wang, Xuhui Kang, Yen-Ling Kuo, Zezhou Cheng, Mengdi Wang, and Odest Chadwicke Jenkins. Foresight: Failure detection for long-horizon robotic manipulation with action-conditioned world model latents, 2026b. URL https://arxiv.org/abs/2606.23085.

A APPENDIX

A.1 PROOF OF THEOREM 3.1

Theorem (Re) 1 (Theorem 3.1 restated, unsafe missed-detection bound). Fix the trained risk head and normalization statistics. Let S_{new} be the score of a new test inference from the unmodified policy, with warning target $Y_{\text{new}}^{(K)}$. Suppose that, conditional on n_u and $Y_{\text{new}}^{(K)} = 1$, the unsafe calibration scores and S_{new} are jointly exchangeable. Then the threshold in Eq. 4 satisfies

$$\mathbb{P}\Big(S_{\text{new}} > q_\alpha \,\Big|\, Y_{\text{new}}^{(K)} = 1\Big) \le \alpha. \tag{8}$$

Proof. (of Theorem 3.1)

We use the standard split-conformal rank argument (Vovk et al., 2005; Angelopoulos & Bates, 2022). Fix the trained risk head and normalization statistics, and condition on n_u and $Y_{\text{new}}^{(K)} = 1$. Let $S_1, \ldots, S_{n_u}$ be the unsafe calibration scores and S_{new} the new unsafe score from the unmodified policy. Since $S_t = 1 - \sigma(g_\psi(Z_t))$, all scores lie in $[0, 1]$.

Write $k_\alpha = \lceil (n_u + 1)(1 - \alpha) \rceil$. If $k_\alpha = n_u + 1$, the convention $S_{(n_u+1)} = 1$ gives $q_\alpha = 1$, so $S_{\text{new}} > q_\alpha$ is impossible. This also covers $n_u = 0$.

Otherwise, let $\widetilde{S}_{(k_\alpha)}$ be the k_α-th order statistic of the augmented sample $(S_1, \ldots, S_{n_u}, S_{\text{new}})$. Adding the test score gives $\widetilde{S}_{(k_\alpha)} \le q_\alpha$. At most $n_u + 1 - k_\alpha$ augmented scores strictly exceed $\widetilde{S}_{(k_\alpha)}$. Since this order statistic is permutation-invariant, exchangeability makes each augmented score equally likely to exceed it. Therefore,

$$\begin{aligned}\mathbb{P}\Big(S_{\text{new}} > q_\alpha \,\Big|\, n_u, Y_{\text{new}}^{(K)} = 1\Big) &\le \mathbb{P}\Big(S_{\text{new}} > \widetilde{S}_{(k_\alpha)} \,\Big|\, n_u, Y_{\text{new}}^{(K)} = 1\Big) \\ &\le \frac{n_u + 1 - k_\alpha}{n_u + 1} \le \alpha.\end{aligned}$$

The counting argument remains valid with tied scores. Averaging over n_u conditional on $Y_{\text{new}}^{(K)} = 1$ yields the stated bound. Since the warning rule triggers when $S_{\text{new}} \le q_\alpha$, the bounded event is exactly a missed detection. The probability averages over calibration and test draws. □

A.2 ADDITIONAL EXPERIMENTAL DETAILS

Prompt design. CP-prompt intervenes at inference time by temporarily augmenting the task instruction, without updating the backbone parameters. Given the original instruction x, we construct the warning prompt by text concatenation. The warning combines an explicit collision-avoidance objective with instructions to maintain progress and complete the task. It is designed to discourage stopping or freezing as a trivial response to a safety warning, while preserving the original task instruction verbatim.

Risk-head Configuration. The detector is a two-layer MLP with hidden dimension 256, trained using binary cross-entropy while LingBot-VA remains frozen. For each inferred chunk, its input is the concatenation of the mean-pooled vision and action hidden states. Each modality is standardized independently with mean/std fitted on the training split only, then concatenated. We optimize with Adam ($l_r = 3 \times 10^{-4}$, batch size 64, epochs 100, l_2 penalty $\lambda = 0.1$; spatial uses $l_r = 10^{-4}$ and $\lambda = 0.5$), with seed 0. Supervision in Eq. 1 follows a $K = 3$ pre-collision early-warning window: the first-collision chunk ν_e and its two preceding chunks are labeled unsafe. Completed collision-free episodes provide negative labels, including terminal windows shorter than K chunks. Post-collision chunks and windows truncated before their outcomes are resolved are excluded from the loss.

Noise sampling. During warning-triggered regeneration of the current chunk, the video branch resamples its initial Gaussian noise z_v, and the action branch independently resamples its initial Gaussian noise z_a. Neither branch reuses the noise from the nominal proposal. For the online resampling controls, action resampling uses N action-noise draws with a shared video-noise draw, whereas video resampling uses N video-noise draws with a shared action-noise draw. The shared noise is fixed across candidates for the same chunk.

Online intervention workflow. The controller is designed around two principles. First, a warning-triggered regeneration should be given time to take effect: after the replacement chunk is executed, the warning instruction $\tilde{x}$ is kept active for a window of A chunks so that the policy can re-plan and clear the risky region rather than being interrupted chunk by chunk. Second, the subsequent cooldown of L_C chunks acts as hysteresis, preventing residual high risk scores immediately after x is restored from re-firing the risk band.

The controller alternates among three stages:

1. **Monitoring.** The policy generates an action chunk under the original instruction x, and the non-conformity score is calculated at each chunk t. If the non-conformity score S_t exceeds the conformal threshold ($S_t > q_\alpha$), the proposal is executed unchanged. Otherwise, the controller discards the unexecuted proposal, switches the instruction to $\tilde{x}$, and re-infers the current chunk once. The replacement chunk is then executed. The conformal threshold is designed at $\alpha = 0.1$ and remains fixed throughout online evaluation.
2. **Sticky warning.** After executing the replacement chunk, the controller retains $\tilde{x}$ for A additional chunks. These chunks are generated and executed under the active warning, without additional risk-triggered regeneration. Thus, the warning applies to the replacement chunk and the following A chunks. We use $A = 5$ in all evaluations, corresponding to $L_w = A + 1 = 6$ in Algorithm 1.
3. **Cooldown.** Once the sticky window expires, the controller restores x and enters a cooldown of L_C chunks: alarm checking is suspended, the policy runs under the original instruction, and no gate-triggered regeneration occurs. Monitoring resumes afterward. We use $L_C = 5$ on all four suites.

Evaluation configuration. We evaluate this protocol on the four tasks in SafeLIBERO-Long, SafeLIBERO-Spatial, SafeLIBERO-Object and SafeLIBERO-Goal. Each task is evaluated on 10 initial states, yielding 40 online evaluation episodes. Each episode begins with a 10-step settling phase, followed by a maximum of 300 steps for Spatial/Goal/Object and 550 for Long. The original Lingbot-VA uses the same policy checkpoint and initial states with the original task instruction and no prompt intervention. Task success and obstacle collision are scored independently, and are checked after every environment control step, including during sticky warning and cooldown. An episode terminates immediately upon task success or when its control-step budget is exhausted. A detected collision is recorded for safety evaluation but does not terminate the episode. An episode is marked unsafe when the active obstacle's position leaves its episode-start pose by more than $\tau \in 10^{-3}$.

Embedding-distance baselines Reference banks E_{safe} and E_{unsafe} are built from $\mathcal{D}_{\text{train}}$ embeddings using K-window safety labels rather than rollout-level success/failure labels; with $\bar{d}_k(\mathbf{h}, E)$ the mean distance from a query embedding $\mathbf{h}$ to its k=3 nearest neighbours in bank E, the score is $r_d(\mathbf{h}) = \bar{d}_k(\mathbf{h}, E_{\text{safe}}) - \bar{d}_k(\mathbf{h}, E_{\text{unsafe}})$, larger meaning riskier.

A.3 Additional Experimental Results

Clean-LIBERO Results On clean LIBERO without added obstacles, LingBot-VA achieves success rates of 98.5%, 97.2%, 99.6%, and 98.5% on Spatial, Goal, Object, and Long, respectively (98.5% on average), demonstrating strong task competence across the original suites.

Action Consistency Results. Full results of action-consistency baselines can be seen in Table 4. Note that Cluster Entropy is trivially degenerate: under the original threshold of 0.05 from SAFE Gu et al. (2026), retained without task-specific retuning, every set of $N = 4$ samples forms four singleton clusters, yielding a constant entropy of $ln4$ and hence an AUROC of exactly 50.00.

Collision-Risk Perception across Policy Representations We supplement the main evaluation with a comparison of the representations produced by LingBot-VA (Li et al., 2026b), OpenVLA (Kim et al., 2024), and $\pi_{0.5}$ (Intelligence et al., 2025). OpenVLA and the $\pi_{0.5}$ policies build on pretrained vision-language models, using multimodal representations of visual observations (VLM-derived) and task instructions to condition action generation. In contrast, LingBot-VA explicitly models future visual states together with actions through interacting video and action streams. Its vision-action

Table 4: Full results of action-consistency baselines.

Method	Long		Spatial		Goal		Object	
	AUROC↑	AUPRC↑	AUROC↑	AUPRC↑	AUROC↑	AUPRC↑	AUROC↑	AUPRC↑
Action Variance	45.63	21.95	61.69	65.90	44.24	70.47	44.30	39.73
Translation Variance	48.06	23.16	63.17	68.41	35.66	60.49	50.48	48.86
Rotation Variance	45.02	29.30	60.97	70.62	58.94	75.76	55.59	55.37
Gripper Variance	45.33	22.91	56.60	65.59	55.40	76.99	58.28	47.09
Cluster Entropy	50.00	24.21	50.00	61.11	50.00	68.22	50.00	45.71

Table 5: **Comparison of collision-risk representations of different policies on SafeLIBERO-Spatial.** We report AUROC on test samples using $K = 3$ collision-warning labels. Each policy is evaluated on its own rollouts and corresponding collision labels to assess how well its internal representations predict collision risk arising from its own actions. Bold indicates the highest reported value in each column.

Policy / readout	AUROC ↑
LingBot-VA MLP	**89.3%**
$\pi_{0.5}$ MLP	85.6%
OpenVLA MLP	81.0%

representations are consequently shaped by both visual-dynamics prediction and action generation. This distinction motivates examining whether they exhibit more accessible collision risk information.

Table 5 quantifies collision-risk decodability from the extracted hidden states. Video–action states within LingBot-VA achieve highest AUROC compared to other states within OpenVLA and $\pi_{0.5}$. This observation is consistent with our hypothesis that future-predictive vision-action representations provide a useful basis for collision-risk assessment. A plausible explanation is that predicting visual transitions encourages the model to retain information about evolving robot-obstacle interactions, making collision-related cues more accessible to a downstream risk detector. Recent work FARM Pei et al. (2026b) shows that frozen world-model states encode eventual task failure. In contrast, we show that they encode imminent collisions, decodable from video-action states before a chunk is executed.

Analysis of Safety Warning We examine whether our method always intervenes before a collision occurs. The analysis is conducted using trajectories disjoint from the training and calibration sets. For a colliding episode i, let a_i denote the first chunk at which an alert is raised, and let b_{i,a_i} denote its starting control step. The pre-collision lead time is

$$L_i = t_i^{\mathrm{col}} - b_{i,a_i}, \tag{9}$$

where t_i^{col} is the first recorded collision step, identified by obstacle displacement. Scores are computed before the corresponding action chunks are executed, and an alarm is considered pre-collision when $L_i > 0$.

As shown in Table 6, all 22 colliding test episodes trigger an alarm before the first recorded collision. The median lead time is 35 action steps, with a range of 17-57 steps; the corresponding chunk-level summary is a median of two chunks and a range of one to three. These results indicate that our method can provide a timely opportunity to revise action generation rather than merely responding after a collision.

Uncertainty in online performance. We estimate uncertainty in the performance differences between our method and the backbone using $10{,}000$ paired bootstrap resamples of episodes within each task, with shared resampling indices for both methods and 40 episodes per suite. All differences are reported in percentage points, with 95% confidence intervals in brackets. Averaging the suite-level metrics with equal weight over the four suites gives $+9.4$ [$+4.4$, $+14.4$] for CAR, $+0.6$ [-6.2, $+7.5$] for TSR, and $+9.1$ [$+3.1$, $+14.9$] for SSH. These intervals support aggregate gains in CAR and SSH, whereas we find no evidence that intervention changes task success-the TSR interval is centered near zero, although a shift of a few points in either direction cannot be ruled out at this sample size. At the suite level (Table 7), CAR and SSH point estimates are positive on all four suites, although some

Table 6: **Our method alerts before collision.** *Pre-collision alarm*: first alarm chunk index $\leq$ collision chunk. Lead time is measured from the start of the first alarmed chunk to the first displacement step.

Quantity	Value	Notes
Colliding episodes	22	monitoring-only episodes
Pre-collision alarm	22/22 (100%)	alarm chunk $\leq$ collision chunk
Lead time (chunks)	1 / 2 / 3	min / median / max
Lead time (action steps)	17 / 35 / 57	min / median / max
Lead time (seconds)	0.85 / 1.75 / 2.85	min / median / max

confidence intervals include zero; TSR changes range from -7.5 on Spatial to $+7.5$ on Long, with all suite-level TSR intervals including zero.

Table 7: **Online performance differences of our method relative to the frozen LingBot-VA backbone.** Changes in CAR, TSR, and SSH are reported in percentage points, with 95% confidence intervals in brackets from 10,000 paired bootstrap resamples of episodes within each task (40 episodes per suite). Positive values favor our method.

Suite	ΔCAR	ΔTSR	ΔSSH
Long	+12.5 [+0.0, +25.0]	+7.5 [-2.5, +17.5]	+10.4 [+1.7, +20.7]
Spatial	+7.5 [-5.0, +20.0]	-7.5 [-27.5, +12.5]	+8.5 [-7.6, +24.6]
Goal	+12.5 [+5.0, +22.5]	+2.5 [-7.5, +12.5]	+15.4 [+5.5, +24.8]
Object	+5.0 [+0.0, +12.5]	+0.0 [-12.5, +12.5]	+2.0 [-9.4, +11.3]
Avg.	+9.4 [+4.4, +14.4]	+0.6 [-6.2, +7.5]	+9.1 [+3.1, +14.9]