GEOPHASE: PHASE-AWARE GEOMETRIC SAFETY FILTERING FOR VISION-LANGUAGE-ACTION POLICIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language-action (VLA) policies enable general-purpose robot manipulation, but their task-directed actions lack explicit geometric safety guarantees, leaving the robot vulnerable to physical collisions in cluttered environments. While runtime safety filtering via control barrier function quadratic programs (CBF-QPs) offers an interpretable solution, current approaches are bottlenecked by costly multi-stage geometric perception, phase-agnostic constraint scheduling, and insufficient protection scopes that typically ignore the grasped object and arm body. To address these limitations, we present GeoPhase, a runtime geometric safety layer that operates on a hybrid learned-perception and analytic-control design. GeoPhase bypasses explicit perception pipelines by predicting structured obstacle ellipsoids directly from intermediate VLA visual embeddings. By tracking the manipulation phase (approach, grasp, transport, place) online via proprioceptive and gripper signals, it dynamically adapts constraint activation to optimize the phase-specific balance between safety and task completion. Furthermore, GeoPhase constructs a multi-proxy geometric representation of the robot-object system, encompassing the end-effector, held object, and arm body, to jointly constrain the robot–object geometry via a unified, minimal-deviation CBF-QP action projection. On the SafeLIBERO benchmark, GeoPhase achieves an average collision rate of 12.8% and a safe success rate of 62.6%, outperforming the best reported baseline averages by 14.0 and 5.8 percentage points, respectively. These results demonstrate that GeoPhase substantially improves collision-free task completion without modifying the underlying, frozen VLA policy.

1 INTRODUCTION

Vision-language-action (VLA) policies provide a unified interface for mapping language instructions and visual observations to robot actions (Zhao et al., 2023; Team et al., 2024). Recent systems, such as RT-1/2 and $\pi_{0.5}$, demonstrate remarkable generalization across tasks, objects, and embodiments (Brohan et al., 2022; 2023; Intelligence et al., 2025). However, the actions produced by these models are primarily task-directed rather than explicitly constrained by physical geometry. Semantic competence does not imply geometric safety. In cluttered manipulation scenes, an instruction-appropriate action may still cause the robot-object system to collide with obstacles.

This safety gap is amplified by the phase-dependent nature of manipulation tasks. The safety–task trade-off shifts dynamically across four distinct phases: approach, grasp, transport, and place (Kroemer et al., 2021). During the approach and grasp phases, the end effector must deliberately enter close proximity to or make contact with the target object, rendering overly conservative avoidance rules detrimental to task completion. Conversely, during transport and place, the grasped object becomes part of the moving collision geometry, expanding the necessary protection envelope. These conflicting requirements render phase-invariant safety margins insufficient, as they either suppress task-critical contact or fail to protect newly relevant kinematic bodies as the operation evolves.

Control barrier function quadratic programs (CBF-QPs) offer an interpretable runtime solution by minimally projecting a nominal VLA action onto a safe control space that satisfies local barrier inequalities (Taylor et al., 2020; Robey et al., 2020; Knoedler et al., 2025). While recent methods like AEGIS (Hu et al., 2025) instantiate this for VLA manipulation, their robust deployment is hindered by three critical bottlenecks, as shown in Fig. 1: ❶ They rely on explicit multi-stage geometry

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.

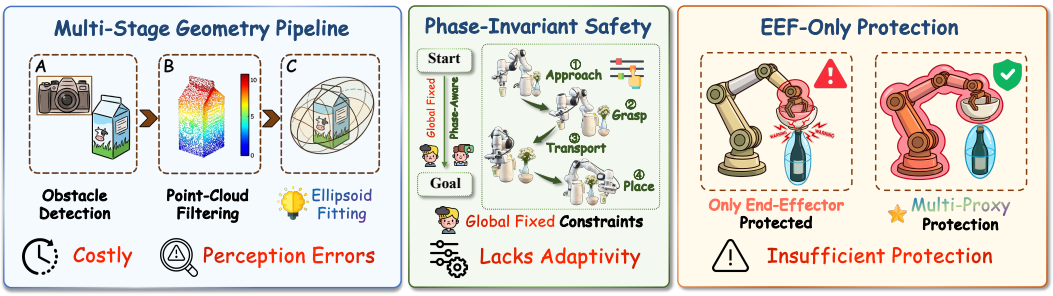


Figure 1: Motivation for GeoPhase. Existing CBF-based safety layers are limited by brittle multi-stage geometry extraction, phase-invariant safety constraints, and end-effector-only protection.

pipelines (e.g., object detection, depth processing, and fitting) that are computationally costly and susceptible to accumulated perception errors. ❷ They are largely phase-agnostic, applying global, static constraints that lack adaptability to the varying requirements of manipulation phases. ❸ They typically represent the robot using a single end-effector (EEF) proxy, providing insufficient protection by ignoring potential collisions involving the grasped object and the arm body.

We present GeoPhase, a phase-aware geometric safety layer that filters VLA actions through a hybrid learned-perception and analytic-control design to systematically resolve the three bottlenecks (Fig. 2). First, it extracts structured obstacle geometry directly from VLA visual embeddings, bypassing the need for an explicit perception pipeline. Second, it infers the current manipulation phase online via a finite-state machine (FSM) driven by proprioceptive and gripper signals, enabling the dynamic scheduling of phase-appropriate safety constraints. Third, it constructs a multi-proxy geometric representation of the robot-object system encompassing the EEF, the held object, and relevant body regions. This formulation jointly constrains the spatial motion of all active proxies through a unified, minimal-deviation CBF-QP projection over the nominal action. Our contributions are:

- ✧ We demonstrate that VLA visual embeddings encode sufficient spatial structure to directly predict structured obstacle ellipsoids. This enables a hybrid design that preserves analytic geometric constraints while bypassing explicit multi-stage perception pipelines.
- ✧ We formulate a phase-aware multi-proxy ellipsoid-pair CBF-QP that dynamically adapts constraint activation. By inferring manipulation phases online via a finite-state machine, it jointly protects the end-effector, held object, and relevant body regions through a unified, minimal-deviation action projection, optimizing the phase-specific safety–task trade-off.
- ✧ We evaluate GeoPhase on the SafeLIBERO benchmark and in real-world scenarios, demonstrating that it reduces average collision rates to 12.8% in simulation and 12.0% in physical trials, substantially improves safe task completion without updating the underlying VLA policy.

2 Related Work

2.1 Vision-Language-Action Models

Vision-language-action (VLA) models demonstrate remarkable generalization in robotic manipulation by mapping visual and linguistic inputs directly to actions (Brohan et al., 2023; Driess et al., 2023; Kim et al., 2024; Intelligence et al., 2025). However, these policies are primarily optimized for semantic task completion and broad applicability rather than geometric safety. Consequently, their behavioral competence does not inherently guarantee a collision-free action space at each control step. To bridge this gap, GeoPhase treats the VLA as a frozen policy and introduces a geometric safety layer, effectively separating semantic action generation from explicit safety enforcement.

2.2 Safety of VLA Models

Safety mechanisms for VLA models broadly fall into (Zhang et al., 2025; Li et al., 2026): **policy-level alignment** and **runtime filtering**. Policy-level methods, such as SafeVLA (Zhang et al., 2026)

and SafeDojo (Tang et al., 2026), train or fine-tune the policy to internalize safety constraints via constrained learning or simulated rollouts (Brunke et al., 2022; Gu et al., 2024). While these approaches reduce unsafe behaviors empirically, they demand costly retraining or simulation environments and fail to explicitly enforce geometric safety constraints on actions during deployment.

In contrast, runtime filtering operates on a frozen VLA, applying interventions only when actions violate predefined safety sets (Wabersich & Zeilinger, 2021). Recent methods like AEGIS (Hu et al., 2025) and KNOWS (Park et al., 2026) utilize CBF-QPs (Ames et al., 2016; 2019; Choi et al., 2021; Zeng et al., 2021; Liu et al., 2025) to minimally modify unsafe actions. GeoPhase advances this runtime filtering paradigm. By extracting structured obstacle representations directly from intermediate VLA embeddings, GeoPhase constructs a phase-aware, multi-proxy geometric constraint system. This design delivers a more adaptive and comprehensive safety layer compared to existing phase-agnostic or EEF-only filtering approaches.

3 Problem Formulation

We consider a VLA model π_{vla} equipped with a safety filter. At each sampled control step t, the robot receives visual observation I_t, language instruction l, and proprioceptive state q_t. π_{vla} produces the nominal action

$$a_t^0 = \pi_{\text{vla}}(I_t, l, q_t) = (\Delta p_t^0, \Delta\theta_t^0, g_t^0), \tag{1}$$

where $\Delta p_t^0 \in \mathbb{R}^3$ and $\Delta\theta_t^0 \in \mathbb{R}^3$ denote the translational and rotational Cartesian commands, g_t^0 is the gripper command. GeoPhase filters only the Cartesian command $u_t^0 = [(\Delta p_t^0)^\top, (\Delta\theta_t^0)^\top]^\top$ while preserving the gripper command, producing a filtered safe action

$$a_t^\star = (\Delta p_t^\star, \Delta\theta_t^\star, g_t^0). \tag{2}$$

The safety filter is conditioned on the predicted obstacle geometry, the geometric proxy of the robot–object system, and the current manipulation phase σ_t. The obstacle is represented by the ellipsoid

$$\mathcal{O}_t = \{y \in \mathbb{R}^3 : \|(y - c_t) \oslash r_t\|_2 \le 1\}, \tag{3}$$

where $c_t \in \mathbb{R}^3$ is the center, $r_t \in \mathbb{R}^3_{>0}$ contains the predicted semi-axis lengths, and $\oslash$ denotes element-wise division. The robot-object system is approximated by a set of ellipsoidal proxies

$$\mathcal{P}_t = \{\mathcal{E}_{i,t}\}_{i=1}^{N_t}, \qquad \mathcal{E}_{i,t} = \{p_{i,t} + R_{i,t}\,\mathrm{diag}(q_i)s : \|s\|_2 \le 1\}, \tag{4}$$

where $p_{i,t}, q_i \in \mathbb{R}^3$ and $R_{i,t} \in SO(3)$ denote center, semi-axis lengths, and orientation of proxy i.

For each proxy–obstacle pair, we formulate a differentiable ellipsoid-pair barrier $h_{i,t}$, whose positivity certifies strict spatial separation. Let $\widehat{\dot{h}}_{i,t}$ denote the local first-order barrier-rate model parameterized by the Cartesian command u. The safety filter performs a minimal-deviation projection of the nominal command by solving a CBF-QP:

$$\begin{aligned} u_t^\star = \arg\min_{u \in \mathbb{R}^6} \quad & \frac{1}{2}\left\|u - u_t^0\right\|_W^2 \\ \text{s.t.} \quad & \widehat{\dot{h}}_{i,t}(u) + \alpha_i(\sigma_t) h_{i,t} \ge 0, \qquad \forall i \in \mathcal{K}_t, \end{aligned} \tag{5}$$

where $\|v\|_W^2 = v^\top W v$ with a positive-definite weight matrix $W \succ 0$. Here, $\alpha_i(\sigma_t) > 0$ is a barrier gain conditioned on the current manipulation phase σ_t, and $\mathcal{K}_t$ is the set of indices for active constraints. The filtered Cartesian command is held for one sampling interval Δt. Proposition 1 characterizes the resulting one-step barrier prediction. If the nominal action inherently satisfies all active barrier conditions, the projection trivially returns $u_t^\star = u_t^0$.

4 Methodology

GeoPhase operates as a runtime geometric safety layer downstream of a frozen VLA model. It assumes no specific VLA architecture, requiring only the intermediate visual representations and the nominal action generated by the policy. At each control step, GeoPhase executes a four-stage pipeline (Fig. 2): (1) predicting structured obstacle ellipsoids directly from VLA visual embeddings, (2) tracking the current manipulation phase, (3) dynamically instantiating a phase-dependent multi-proxy geometric model of the robot-object system, and (4) projecting the nominal action through a unified ellipsoid-pair CBF-QP. This projection yields a minimally modified Cartesian command satisfying the active barrier inequalities, which is executed alongside the preserved gripper command.

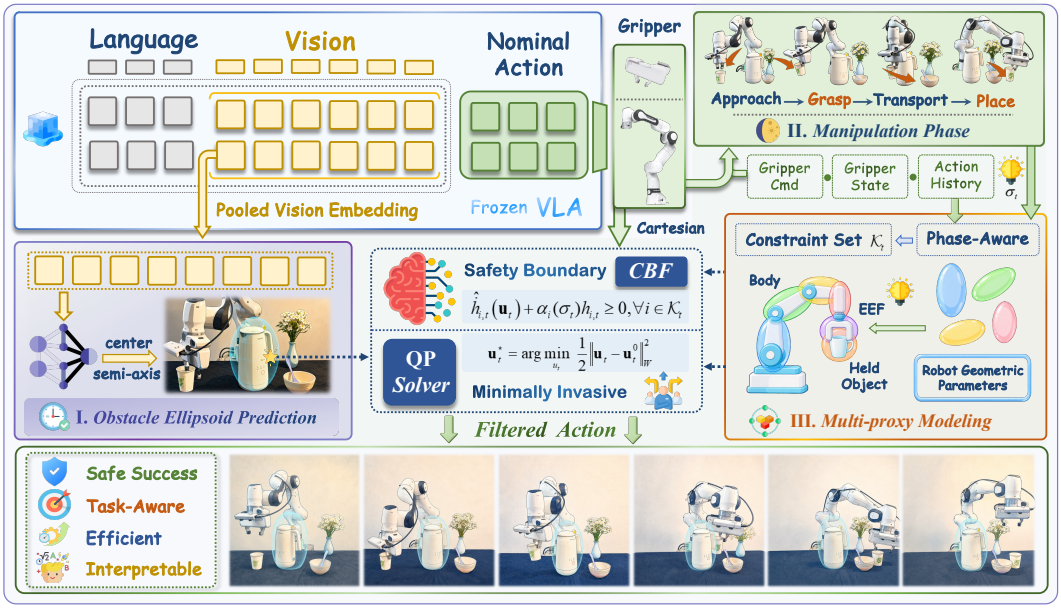


Figure 2: **Overview of the GeoPhase pipeline.** A frozen VLA outputs a nominal action and pooled visual embedding. The embedding is mapped to an obstacle ellipsoid, while gripper signals identify the current manipulation phase. GeoPhase then constructs phase-aware multi-proxy geometric constraints and applies a CBF-QP to minimally project the nominal action, reducing collisions.

4.1 Predict Obstacle Ellipsoid from VLA Embeddings

To formulate geometric barrier constraints, the safety layer requires spatial representations of the obstacles. As shown in Fig. 1, conventional perception-to-control pipelines extract obstacle geometry through a multi-stage process of language-grounded detection, point-cloud reconstruction, and shape fitting (Tong et al., 2022; Abdi et al., 2023; Tabbara & Sibai, 2024). This sequential approach incurs substantial computational overhead and accumulates errors across perception modules.

To circumvent these bottlenecks, we exploit the observation that the intermediate visual representations extracted from VLA backbones already encode task-relevant spatial information (Fu et al., 2025; Shkolnikov, 2026; Gu et al., 2026). Let Z_t denote the pooled visual embedding at control step t. A lightweight multilayer perceptron (MLP) head, f_θ, is trained to directly map this embedding to the center and semi-axis lengths of the primary safety-critical obstacle, details see Appendix C.1 :

$$(c_t, r_t) = f_\theta(Z_t), \qquad c_t \in \mathbb{R}^3, \quad r_t \in \mathbb{R}^3_{>0}. \tag{6}$$

Learning strictly concludes at this geometric interface. The predicted parameters (c_t, r_t) provides the necessary parameters to construct the obstacle ellipsoid $\mathcal{O}_t$ required for the safety layer, as shown in Fig. 6. Consequently, the subsequent safety constraints and action corrections are computed entirely analytically within the CBF-QP. This hybrid learned-perception and analytic-control design preserves strict mathematical interpretability, ensuring that geometry predictions remain explicitly verifiable through standard spatial metrics.

4.2 Manipulation Phase Tracking

We observe that different stages of robotic manipulation inherently exhibit distinct geometric risks and control requirements. A fixed constraint configuration can therefore be either overly conservative when intentional proximity is required or insufficiently protective when the collision-relevant geometry expands. To dynamically adapt to these shifting conditions, GeoPhase categorizes the task execution into four manipulation phases (see Fig. 7):

$$\mathcal{S} = \{\text{approach}, \text{grasp}, \text{transport}, \text{place}\}.$$

The current phase $\sigma_t \in \mathcal{S}$ is estimated online by a finite-state machine (FSM) driven by the gripper command and recent proprioceptive history. ***Approach*** characterizes the initial motion toward the

target, during which the filter preserves sufficient freedom to reach the object. ***Grasp*** corresponds to the gripper-closing interval, where the safety layer avoids excessive corrections to maintain proper alignment and controlled closure. Upon confirming a stable grasp, the FSM transitions to ***transport***, activating the held-object proxy to protect the geometrically expanded collision volume of the combined robot-object system. Finally, ***Place*** initiates when the release process is detected, allowing precise placement while maintaining clearance from unrelated obstacles. The FSM can seamlessly repeat to accommodate tasks requiring multiple sequential grasps.

Crucially, this coarse phase estimation extracts the necessary risk structure without relying on a detailed symbolic task planner. The phase tracker neither generates nor directly modifies the robot's Cartesian action; its sole function is to schedule the activation logic of the geometric proxy constraints and modulate their phase-appropriate intervention strengths.

4.3 Multi-Proxy Ellipsoid-Pair Constraints

A single end-effector proxy is insufficient to capture the complete collision geometry encountered during manipulation. The gripper, the grasped object, and the robot's body links may approach an obstacle from disparate directions, even when the end-effector center itself remains collision-free (Cai et al., 2026). To address this, GeoPhase models the robot-side geometry using a candidate set of multiple ellipsoidal proxies, $\mathcal{P}_t$, see Fig. 3. The set includes an end-effector envelope, a held-object envelope activated after grasp, and configurable body envelopes. For a proxy attached to the end effector, its center is given by

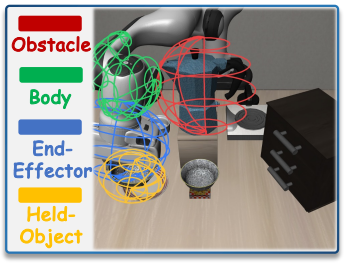


Figure 3: Multi-proxy ellipsoidal constraint representation.

$$p_{i,t} = p_{e,t} + R_{e,t} d_i, \tag{7}$$

where $p_{e,t}$ and $R_{e,t}$ denote the end-effector position and orientation, and d_i is the proxy offset in the end-effector frame. More generally, the pose of any proxy attached to a specific robot link $\ell(i)$ is obtained via forward kinematics. This formulation intrinsically binds the spatial twist of each proxy to the shared Cartesian command through a link-to-action pose-transfer map (the complete derivation is provided in Appendix B.1). During execution, the phase-aware schedule first selects the relevant proxy subset $\mathcal{P}_{\sigma_t}$, from which the geometric activation rule determines the active constraint set $\mathcal{K}_t$. All active proxy–obstacle pairs concurrently constrain the same nominal Cartesian action.

Following the differentiable ellipsoid-pair construction by Funada et al. (2024), we associate each proxy–obstacle pair with a dynamically optimized separation direction $z_{i,t} \in \mathbb{S}^2$. Using the ellipsoid parameters defined in Section 3, we introduce the shape matrices $\bar{Q}_{i,t} = R_{i,t}\,\mathrm{diag}(q_i)R_{i,t}^\top$ and $\bar{Q}_{o,t} = \mathrm{diag}(r_t)$. The minimum signed separation from the obstacle ellipsoid $\mathcal{O}_t$ to the supporting hyperplane of proxy $\mathcal{E}_{i,t}$ defines the pairwise barrier function:

$$\begin{aligned} h_{i,t}(z_{i,t}) &:= \min_{x \in \mathcal{O}_t} n_i(z_{i,t})^\top \big(x - m_i(z_{i,t})\big) \\ &= \frac{-\|\bar{Q}_{o,t}\bar{Q}_{i,t}^{-1} z_{i,t}\|_2 + (c_t - p_{i,t})^\top \bar{Q}_{i,t}^{-1} z_{i,t} - 1}{\|\bar{Q}_{i,t}^{-1} z_{i,t}\|_2}. \end{aligned} \tag{8}$$

Here, $m_i(z_{i,t})$ and $n_i(z_{i,t})$ denote the point on the proxy surface and its outward unit normal associated with the direction $z_{i,t}$. A positive $h_{i,t}(z_{i,t})$ certifies that the supporting hyperplane strictly separates $\mathcal{E}_{i,t}$ from $\mathcal{O}_t$. A nonpositive value fails to certify separation (Ding et al., 2024). By dynamically optimizing $z_{i,t}$, the separating hyperplane continuously adapts to the evolving relative geometry, avoiding the restrictive conservatism of a fixed separation direction.

4.4 Unified Phase-Aware QP

GeoPhase integrates the predicted obstacle geometry, manipulation phase, and multi-proxy barriers into a unified minimal-deviation projection of the nominal VLA action. At each control step, the safety layer filters the constraints based on a phase-conditioned geometric activation rule: it first isolates the phase-relevant proxy subset $\mathcal{P}_{\sigma_t} \subseteq \mathcal{P}_t$, and then selects the active constraint indices $\mathcal{K}_t$ based on proximity thresholds (detailed in Appendix C.4). If $\mathcal{K}_t = \varnothing$, GeoPhase executes the nominal action unchanged.

When $\mathcal{K}_t$ is nonempty, the QP jointly optimizes the shared Cartesian command u_t and a pair-specific separation-direction update $u_{z,i,t} \in \mathbb{R}^3$ for each active constraint. Let $\mathbf{u}_t = [u_t^\top, \{u_{z,i,t}^\top\}_{i\in\mathcal{K}_t}]^\top$ denote the augmented decision variable, with its nominal counterpart $\mathbf{u}_t^0 = [(u_t^0)^\top, \{0^\top\}_{i\in\mathcal{K}_t}]^\top$. Because the barrier derivative $\widehat{\dot{h}}_{i,t}(\mathbf{u}_t)$ is strictly affine in $\mathbf{u}_t$ (see Appendix B.1), GeoPhase efficiently solves the following unified CBF-QP:

$$\begin{aligned} \mathbf{u}_t^\star = \arg\min_{\mathbf{u}_t} \quad & \frac{1}{2}\left\|\mathbf{u}_t - \mathbf{u}_t^0\right\|_W^2 \\ \text{s.t.} \quad & \widehat{\dot{h}}_{i,t}(\mathbf{u}_t) + \alpha_i(\sigma_t) h_{i,t} \geq 0, \qquad \forall i \in \mathcal{K}_t. \end{aligned} \tag{9}$$

Here, the positive-definite weight matrix $W \succ 0$ balances the preservation of the semantic VLA command against the regularization of the auxiliary direction updates, and $\alpha_i(\sigma_t) > 0$ is the phase-conditioned barrier gain. If the QP fails to return a feasible solution, GeoPhase defaults to a conservative emergency stop, explicitly setting all Cartesian increments to zero.

GeoPhase enforces the local continuous-time CBF inequalities in Eq. 9 at each sampling instant and executes the filtered Cartesian command under zero-order hold. Proposition 1 establishes a local first-order certificate for this projection. Appendix B.3 provides a sufficient condition for intersample separation under the stated assumptions and a second-order remainder bound.

Proposition 1 (One-step barrier prediction) *Consider one sampling interval $[t, t+\Delta t]$ with a fixed active set $\mathcal{K}_t$. For each $i \in \mathcal{K}_t$, define the first-order predicted barrier value as:*

$$\widehat{h}_{i,t+\Delta t} = h_{i,t} + \Delta t\, \widehat{\dot{h}}_{i,t}(\mathbf{u}_t^\star), \tag{10}$$

where $\mathbf{u}_t^\star$ is a feasible solution to Eq. 9. If the initial state is safe and the phase-conditioned gain is properly bounded such that

$$h_{i,t} \geq 0, \qquad 0 < \alpha_i(\sigma_t)\Delta t \leq 1, \qquad \forall i \in \mathcal{K}_t,$$

then the predicted barrier value remains non-negative at the next step:

$$\widehat{h}_{i,t+\Delta t} \geq (1 - \alpha_i(\sigma_t)\Delta t) h_{i,t} \geq 0, \qquad \forall i \in \mathcal{K}_t.$$

Hence, under the local first-order model, every active constraint preserves a non-negative predicted separation over the subsequent sampling interval (Proof provided in Appendix B.2).

5 Experiments

We evaluate GeoPhase to determine whether it improves geometric safety without sacrificing task completion, and whether its learned geometry, phase-aware scheduling, and multi-proxy action projection remain minimally invasive and computationally efficient.

5.1 Experimental Setup

Benchmark. We conduct simulation experiments on SafeLIBERO (Hu et al., 2025). The benchmark contains 16 tasks selected from four LIBERO (Liu et al., 2023) suites: Spatial, Goal, Object, and Long. Each task is augmented with two levels of obstacle interference. Object poses are randomized over 50 episodes for each task–level combination, resulting in 32 safety-critical scenarios and 1,600 evaluation episodes. Details see Appendix D.

Baselines. We compare GeoPhase with the following representative VLA policy and safety methods. $\pi_{0.5}$ (Intelligence et al., 2025) is a flow-matching VLA pretrained through heterogeneous co-training and serves as the unfiltered nominal policy for our primary comparison. **AEGIS** (Hu et al., 2025) is an inference-time CBF safety layer that obtains obstacle geometry through explicit visual grounding, depth reconstruction, and ellipsoid fitting before filtering the VLA action. **SafeDojo** (Tang et al., 2026) is a training-based safe-RL method that optimizes task reward and safety cost through imagined rollouts in an interactive world model. **KNOWS** (Park et al., 2026) is a runtime safety filter that identifies the active target from real-time object tracking with CBF-based action correction to avoid collisions. **GeoPhase** is constructed on top of the same $\pi_{0.5}$ policy used

Table 1: Main results on SafeLIBERO. Bold and underline denote the best and second-best results.

	Method	SPATIAL			GOAL			OBJECT			LONG		
		TSR↑	CR↓	SSR↑	TSR↑	CR↓	SSR↑	TSR↑	CR↓	SSR↑	TSR↑	CR↓	SSR↑
Level I	$\pi_{0.5}$	**67.5**	86.5	13.5	51.0	95.0	5.0	43.0	83.0	14.0	**58.5**	86.0	13.5
	AEGIS	55.0	<u>27.5</u>	45.0	<u>83.0</u>	<u>6.5</u>	<u>80.0</u>	55.5	60.0	25.0	41.0	61.5	23.0
	SafeDojo	59.5	30.5	51.5	80.0	11.5	78.5	**73.5**	42.5	49.0	45.0	<u>24.5</u>	34.0
	KNOWS	<u>63.0</u>	32.5	<u>53.0</u>	**87.0**	10.0	**81.0**	<u>71.0</u>	**6.5**	**70.5**	49.0	45.5	<u>35.0</u>
	Ours	59.0	**16.0**	**57.0**	81.0	**5.5**	77.5	66.0	<u>8.5</u>	<u>63.5</u>	<u>54.5</u>	**12.5**	**48.5**
Level II	$\pi_{0.5}$	49.0	88.0	11.0	66.0	64.5	29.0	69.0	72.0	25.5	**47.0**	83.5	14.5
	AEGIS	74.5	31.5	61.5	71.0	39.5	52.5	71.5	39.0	50.0	34.0	35.5	26.0
	SafeDojo	74.5	**20.5**	60.5	64.0	<u>33.5</u>	<u>56.0</u>	60.0	<u>17.0</u>	55.5	29.5	<u>34.0</u>	<u>26.5</u>
	KNOWS	**87.5**	27.0	<u>70.0</u>	**82.5**	41.0	52.0	<u>79.5</u>	17.5	<u>69.0</u>	35.0	39.5	24.0
	Ours	<u>75.5</u>	<u>22.5</u>	**73.5**	<u>74.0</u>	**24.0**	**64.5**	**82.0**	**2.0**	**80.5**	<u>39.0</u>	**11.0**	**35.5**

as its nominal baseline, without updating the VLA backbone; thus, the comparison between $\pi_{0.5}$ and GeoPhase directly isolates the effect of our safety layer.

Metrics. We report four primary episode-level metrics. **Task Success Rate (TSR)** measures the percentage of episodes that complete the instructed task within the time limit, irrespective of collisions. **Collision Rate (CR)** measures the percentage of episodes that experience at least one collision, irrespective of task completion. **Safe Success Rate (SSR)** measures the percentage of episodes that both complete the task and remain collision-free, thereby capturing the joint safety–utility objective. **Execution Time Steps (ETS)** is the average episode length, including timed-out episodes, where a lower value indicates more efficient execution.

5.2 MAIN RESULTS

As shown in Table 1, we compared GeoPhase with the unfiltered $\pi_{0.5}$ policy and three representative safety-enhanced methods on SafeLIBERO. As the official codebases for SafeDojo and KNOWS remain closed-source, their results are reported directly from the respective publications under the aligned SafeLIBERO evaluation protocol.

Safety performance. GeoPhase consistently provides the strongest overall safety performance. Averaged across the eight settings, GeoPhase achieves a CR of 12.8% ($\pm 1.6\%$ at 95% confidence interval), reducing collisions by 14.0 percentage points relative to the strongest reported baseline average of 26.8% obtained by SafeDojo. GeoPhase also achieves the highest average SSR of 62.6% ($\pm 2.4\%$), outperforming the strongest reported results of KNOWS by 5.8 percentage points. In particular, GeoPhase obtains the lowest CR and highest SSR in six of the eight settings. These improvements are especially pronounced under Level-II interference, where GeoPhase achieves an average CR of 14.9% ($\pm 2.5\%$) and an SSR of 63.5% ($\pm 3.3\%$), improving over the best baseline averages by 11.4 and 9.8 percentage points, respectively. On the Level-II Object suite, GeoPhase further reduces CR to 2.0% while attaining an SSR of 80.5%.

Execution efficiency. Despite its substantial safety gains, GeoPhase does not introduce excessive trajectory detours. Among the methods reporting ETS, GeoPhase achieves the lowest average execution length of 264.3 steps, compared with 268.9 for AEGIS and 273.2 for SafeDojo. Together with its competitive TSR, these results indicate that GeoPhase improves collision avoidance and safe task completion while preserving the execution efficiency of the nominal policy.

5.3 ABLATION STUDIES

We conduct ablation studies to examine the contributions of the three key components of GeoPhase: learned obstacle geometry, phase-aware safety scheduling, and multi-proxy robot modeling. We first evaluate their overall contributions and then analyze each component individually.

Table 2: Overall component ablation results on SafeLIBERO.

Method	Level I				Level II			
	TSR ↑	CR ↓	SSR ↑	ETS ↓	TSR ↑	CR ↓	SSR ↑	ETS ↓
$\pi_{0.5}$ (Base)	55.0	87.6	11.5	290.9	57.8	77.0	20.0	273.8
w/o Learned Geometry	59.4	39.4	42.5	275.8	63.8	29.4	48.8	267.7
w/o Phase Awareness	59.4	17.5	55.6	281.9	68.1	15.0	63.1	257.4
w/o Multi-Proxy	**68.8**	28.1	59.4	**261.6**	**71.9**	30.6	53.1	**248.9**
+ ALL (Ours)	65.1	**10.6**	**61.6**	270.6	67.6	**14.9**	**63.5**	257.9

Table 3: Geometric representation quality.

Geometry	Center MAE ↓	Radii MAE ↓	Coverage ↑	IoU ↑
Oracle (Base)	0.000	0.000	1.000	1.000
AEGIS	0.578	0.912	0.529	0.217
Learned (Ours)	**0.201**	**0.198**	**0.931**	**0.846**

Table 4: Collision rate and proxy intervention rates (**IR**).

Proxy Configuration	CR ↓	EEF IR	Held IR	Body IR
EEF Only	29.4	72.9	–	–
EEF + Held	21.9	71.3	39.0	–
EEF + Held + Body (Ours)	**12.8**	71.6	38.7	24.0

Overall Component Ablation. Table 2 compares the complete GeoPhase model with the unfiltered base policy and three variants in which one component is replaced or removed. Specifically, *w/o Learned Geometry* replaces our learned geometry predictor with the traditional fitting pipeline (AEGIS), *w/o Phase Awareness* applies a static safety configuration throughout execution, and *w/o Multi-Proxy* retains only the end-effector (EEF) proxy.

GeoPhase achieves the lowest CR and the highest SSR under both interference levels. Compared with the unfiltered $\pi_{0.5}$ policy, it reduces CR by 77.0 and 62.1 percentage points and improves SSR by 50.1 and 43.5 percentage points under Levels I and II, respectively. Replacing the learned geometry predictor causes the largest safety degradation, increasing CR to 39.4% and 29.4%. Removing phase awareness produces a smaller but consistent decline, particularly under Level I. Although the EEF-only variant attains a higher TSR and lower ETS, its substantially higher CR leads to a lower SSR. These results demonstrate that the three components play complementary roles in balancing task completion, safety, and execution efficiency.

Geometric Representation. We compare our learned geometry predictor with oracle obstacle geometry and the explicit reconstruction pipeline adopted by AEGIS. The "Oracle" denotes the inflated ellipsoid constructed from ground-truth simulator geometry. Table 3 evaluates the quality of the predicted ellipsoids (in centimeters), while Fig. 4(a) reports their safety control performance.

Oracle geometry achieves perfect reconstruction by definition and provides better performance. Compared with AEGIS geometry, our learned predictor substantially reduces both center and semi-axis length errors, increasing the mean reference-ellipsoid coverage from 0.529 to 0.931 and the mean IoU from 0.217 to 0.846. These results indicate that the predictions cover most of the inflated reference volume while maintaining moderate volumetric agreement. These gains translate into a marked improvement in control performance, reducing CR from 34.4% to 12.8% and increasing SSR from 45.6% to 62.6%. The small gap between learned and oracle geometry—0.5 percentage points in CR and 0.6 percentage points in SSR—provides evidence of effective safety filtering.

Phase-Aware Safety Scheduling. Fig. 4(b) compares phase-aware scheduling with a static configuration using the same learned geometry and multi-proxy representation. The static baseline retains gripper-based held-object activation but applies a constant transport-phase barrier gain ($\alpha_i = 15\,\mathrm{s}^{-1}$) throughout execution. Phase-aware scheduling increases TSR from 63.8% to 66.4% and SSR from 59.4% to 62.6%, while reducing CR from 16.2% to 12.8%. These results demonstrate that dynamically adapting constraint activation and barrier gains improves the safety–task trade-off.

Multi-Proxy Robot Modeling. We progressively augment the robot representation from an EEF-only proxy to an EEF–held-object configuration and, finally, to the complete representation containing EEF, held-object, and body proxies. As shown in Fig. 4(c), adding the held-object proxy reduces CR from 29.4% to 21.9%, while the complete proxy set further reduces it to 12.8%. Correspondingly, SSR increases monotonically from 56.2% to 59.7% and 62.6%, which means our method achieves the strongest joint safety–utility performance.

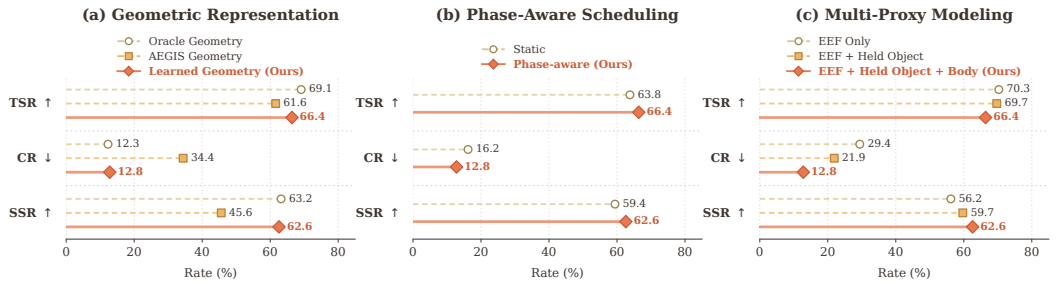


Figure 4: Component-wise ablation results for (a) Geometric Representation, (b) Phase-Aware Scheduling, and (c) Multi-Proxy robot modeling.



Figure 5: Real-world execution comparison. The base $\pi_{0.5}$ policy collides with the kettle during transport, whereas GeoPhase minimally projects the action to safely navigate around the obstacle.

Table 4 further reports proxy-specific intervention rates. The EEF intervention rate remains approximately constant across configurations, whereas the held-object and body proxies introduce additional interventions in 38.7% and 24.0% of the evaluated steps, respectively. This pattern suggests that the additional proxies cover complementary collision modes rather than merely increasing the frequency of EEF-based corrections.

5.4 Real-World Experiments

To evaluate the real-world performance of GeoPhase, we deploy both the base policy $\pi_{0.5}$ and GeoPhase on a Franka Research 3 robot. We construct five manipulation tasks involving diverse objects, receptacles, and obstacle configurations, and conduct 10 independent trials for each task. Fig. 5 presents a representative rollout for **Task 5** (placing the water cup next to the yellow bowl). The complete per-task results, rollouts, and analysis are provided in Appendix E.6.

Across the aggregate 50 trials of the five tasks, GeoPhase improves the average TSR from 80.0% ($\pm 11.1\%$ at 95% CI) to 86.0% ($\pm 9.6\%$), reduces the average CR from 64.0% ($\pm 13.3\%$) to 12.0% ($\pm 9.0\%$), and increases the average SSR from 18.0% ($\pm 10.6\%$) to 74.0% ($\pm 12.1\%$). These results show that GeoPhase substantially improves collision-free task completion in physical manipulation while preserving the execution capability of the underlying VLA policy.

6 Conclusion

We presented GeoPhase, a phase-aware geometric safety layer that augments a frozen VLA policy through learned obstacle geometry and analytic action filtering. GeoPhase predicts structured obstacle ellipsoids from VLA visual embeddings, infers manipulation phases from gripper signals, and jointly constrains end-effector, held-object, and relevant body proxies through an ellipsoid-pair CBF-QP over the shared Cartesian command. Across 1,600 SafeLIBERO episodes, GeoPhase achieves SOTA collision rate and safe success rate improvements. These findings establish GeoPhase as an effective and interpretable runtime safety interface for improving the safety–task trade-off of VLA manipulation without modifying the underlying policy.

AI USE STATEMENT

In this work, we used generative AI tools solely to edit the manuscript for grammatical correctness, clarity, conciseness, and readability. We did not use generative AI to develop the research methodology, formulate theoretical or mathematical claims, construct proofs, design experiments, implement methods, generate or process data, analyze results, or interpret findings; the remaining disclosure categories are not applicable to this work. All AI-assisted revisions were carefully reviewed and approved by the authors to ensure technical accuracy, originality, and consistency with the underlying research. The authors take full responsibility for the final content of this paper, including all text and claims.

ETHICS STATEMENT

This work does not involve human subjects, personally identifiable information, or sensitive data. All real-world robot experiments were conducted in a controlled laboratory environment with appropriate safety precautions. The proposed method is intended to improve the safety of VLA policies by reducing collisions during robotic manipulation.

REPRODUCIBILITY STATEMENT

To support reproducibility, the main paper specifies the GeoPhase formulation, model architecture, phase-aware safety scheduling, multi-proxy ellipsoid-pair CBF-QP, and training and evaluation procedures. The appendix provides the theoretical assumptions and derivations, implementation details, hyperparameters, data-collection and geometry-supervision procedures, SafeLIBERO benchmark configuration, and evaluation protocol. Upon publication, we will release the source code required to reproduce the reported results.

REFERENCES

Hossein Abdi, Golnaz Raja, and Reza Ghabcheloo. Safe control using vision-based control barrier function (v-cbf). In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 782–788. IEEE, 2023.

Aaron D Ames, Xiangru Xu, Jessy W Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE transactions on automatic control*, 62(8): 3861–3876, 2016.

Aaron D Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. Control barrier functions: Theory and applications. In *2019 18th European control conference (ECC)*, pp. 3420–3431. Ieee, 2019.

Franco Blanchini. Set invariance in control. *Automatica*, 35(11):1747–1767, 1999.

Joseph Breeden, Kunal Garg, and Dimitra Panagou. Control barrier functions in sampled-data systems. *IEEE Control Systems Letters*, 6:367–372, 2021.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Lukas Brunke, Melissa Greeff, Adam W Hall, Zhaocong Yuan, Siqi Zhou, Jacopo Panerati, and Angela P Schoellig. Safe learning in robotics: From learning-based control to safe reinforcement learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 5(1):411–444, 2022.

Wenqi Cai, John Abanes, Nikolaos Evangeliou, and Anthony Tzes. Safe human-to-humanoid motion imitation using control barrier functions. *arXiv preprint arXiv:2604.11447*, 2026.

Jason J Choi, Donggun Lee, Koushil Sreenath, Claire J Tomlin, and Sylvia L Herbert. Robust control barrier–value functions for safety-critical control. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 6814–6821. IEEE, 2021.

Xuda Ding, Han Wang, Yi Ren, Yu Zheng, Cailian Chen, and Jianping He. Online control barrier function construction for safety-critical motion control of manipulators. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 54(8):4761–4771, 2024.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

Stephanie Fu, Tyler Bonnen, Devin Guillory, and Trevor Darrell. Hidden in plain sight: Vlms overlook their visual representations. *arXiv preprint arXiv:2506.08008*, 2025.

Riku Funada, Koju Nishimoto, Tatsuya Ibuki, and Mitsuji Sampei. Collision avoidance for ellipsoidal rigid bodies with control barrier functions designed from rotating supporting hyperplanes. *IEEE Transactions on Control Systems Technology*, 33(1):148–164, 2024.

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.

Shangding Gu, Long Yang, Yali Du, Guang Chen, Florian Walter, Jun Wang, and Alois Knoll. A review of safe reinforcement learning: Methods, theories, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):11216–11235, 2024.

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, et al. pi0.5: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Luzia Knoedler, Oswin So, Ji Yin, Mitchell Black, Zachary Serlin, Panagiotis Tsiotras, Javier Alonso-Mora, and Chuchu Fan. Safety on the fly: Constructing robust safety filters via policy control barrier functions at runtime. *IEEE Robotics and Automation Letters*, 2025.

Oliver Kroemer, Scott Niekum, and George Konidaris. A review of robot learning for manipulation: Challenges, representations, and algorithms. *Journal of machine learning research*, 22(30):1–82, 2021.

Qi Li, Bo Yin, Weiqi Huang, Ruhao Liu, Bojun Zou, Runpeng Yu, Jingwen Ye, Weihao Yu, and Xinchao Wang. Vision-language-action safety: Threats, challenges, evaluations, and mechanisms. *arXiv preprint arXiv:2604.23775*, 2026.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

Shuo Liu, Yihui Mao, and Calin A Belta. Safety-critical planning and control for dynamic obstacle avoidance using control barrier functions. In *2025 American Control Conference (ACC)*, pp. 348–354. IEEE, 2025.

Ilya Loshchilov, Frank Hutter, et al. Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101*, 5(5):5, 2017.

Seongbin Park, Fan Zhang, Baharan Mirzasoleiman, Shahriar Talebi, and Nader Sehatbakhsh. Your model already knows: Attention-guided safety filter for vision-language-action models. *arXiv preprint arXiv:2606.09749*, 2026.

Roger Penrose. A generalized inverse for matrices. In *Mathematical proceedings of the Cambridge philosophical society*, volume 51, pp. 406–413. Cambridge University Press, 1955.

Alexander Robey, Haimin Hu, Lars Lindemann, Hanwen Zhang, Dimos V Dimarogonas, Stephen Tu, and Nikolai Matni. Learning control barrier functions from expert demonstrations. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 3717–3724. Ieee, 2020.

Yakov Pyotr Shkolnikov. Do foundation models know geometry? probing frozen features for continuous physical measurement. *arXiv preprint arXiv:2603.06459*, 2026.

Bartolomeo Stellato, Goran Banjac, Paul Goulart, Alberto Bemporad, and Stephen Boyd. Osqp: an operator splitting solver for quadratic programs: B. stellato et al. *Mathematical Programming Computation*, 12(4):637–672, 2020.

Ihab Tabbara and Hussein Sibai. Learning ensembles of vision-based safety control filters. *arXiv preprint arXiv:2412.02029*, 2024.

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.

Andrew Taylor, Andrew Singletary, Yisong Yue, and Aaron Ames. Learning for safety-critical control with control barrier functions. In *Learning for dynamics and control*, pp. 708–717. PMLR, 2020.

Andrew J Taylor, Victor D Dorobantu, Ryan K Cosner, Yisong Yue, and Aaron D Ames. Safety of sampled-data systems with control barrier functions via approximate discrete time models. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pp. 7127–7134. IEEE, 2022.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

Mukun Tong, Charles Dawson, and Chuchu Fan. Enforcing safety for vision-based controllers via control barrier functions and neural radiance fields. *arXiv preprint arXiv:2209.12266*, 2022.

Kim Peter Wabersich and Melanie N Zeilinger. A predictive safety filter for learning-based control of constrained nonlinear dynamical systems. *Automatica*, 129:109597, 2021.

Jun Zeng, Bike Zhang, Zhongyu Li, and Koushil Sreenath. Safety-critical control using optimal-decay control barrier function with guaranteed point-wise feasibility. In *2021 American Control Conference (ACC)*, pp. 3856–3863. IEEE, 2021.

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, 2026.

Dapeng Zhang, Jing Sun, Chenghui Hu, Xiaoyan Wu, Zhenlong Yuan, Rui Zhou, Fei Shen, and Qingguo Zhou. Pure vision language action (vla) models: A comprehensive survey. *arXiv preprint arXiv:2509.19012*, 2025.

Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Kevin Lin, Abhiram Maddukuri, Soroush Nasiriany, and Yifeng Zhu. robosuite: A modular simulation framework and benchmark for robot learning. *arXiv preprint arXiv:2009.12293*, 2020.

A LIMITATIONS

Current evaluations follow the single-obstacle SafeLIBERO protocol; scaling to multi-obstacle environments requires set-based geometric prediction and temporal tracking. Proposition 1 provides only a conditional one-step certificate under a local first-order kinematic model with fixed geometry and active constraints, and does not account for higher-order discretization, prediction variation, kinematic mismatch, operational-space tracking, actuator error, or phase transitions that may activate previously unverified proxies. A deployment-level or probabilistic guarantee would require calibrated barrier-level uncertainty bounds, online intersample verification and bounded action and tracking errors, which remain for future research. Furthermore, while aggregate metrics across evaluated tasks demonstrate statistically significant improvements, the per-task sample size in real-world evaluations ($n = 10$) intrinsically exhibits high statistical variance. Future deployments require larger-scale physical rollouts to precisely calibrate task-level reliability bounds.

B THEORETICAL DERIVATIONS

We derive the affine dependence of the ellipsoid-pair barrier dynamics on the GeoPhase control variables and establish one-step and intersample barrier bounds stated in Section 4.4. For notational clarity, we omit the time index t whenever it is unambiguous from the context.

B.1 AFFINE BARRIER DYNAMICS AND QP COEFFICIENTS

We first derive the affine barrier dynamics utilized by the unified CBF-QP. For compactness, we drop the time index for the shape and state variables. Define the following intermediate geometric quantities:

$$\eta_i = \bar{Q}_i^{-1} z_i, \qquad s_i = \|\eta_i\|_2, \qquad \beta_i = \|\bar{Q}_o \eta_i\|_2, \qquad n_i = \frac{\eta_i}{s_i}. \tag{11}$$

The gradient of the pairwise barrier h_i with respect to η_i is given by:

$$g_i := \nabla_{\eta_i} h_i = \frac{1}{s_i}\left(c_t - p_i - \frac{\bar{Q}_o^\top \bar{Q}_o \eta_i}{\beta_i}\right) - \frac{h_i}{s_i^2}\eta_i. \tag{12}$$

Let $u_t = [u_{v,t}^\top, u_{\omega,t}^\top]^\top \in \mathbb{R}^6$ denote the shared Cartesian command, unifying the translational and rotational increments. To accommodate proxies attached to the end-effector, the held object, or an arbitrary body link, let $\ell(i)$ denote the supporting robot frame of proxy i. The spatial twist of this frame relates to the shared Cartesian command via the velocity map:

$$\begin{bmatrix} v_{\ell(i)} \\ \omega_{\ell(i)} \end{bmatrix} = T_{\ell(i)\leftarrow e} \frac{u_t}{\Delta t}, \qquad T_{\ell(i)\leftarrow e} = J_{\ell(i)} J_e^\dagger. \tag{13}$$

where $J_{l(i)}$ and J_e are the spatial Jacobians of the supporting link frame and the action frame, respectively, and $J_e^\dagger$ denotes the Moore-Penrose pseudoinverse (Penrose, 1955). This velocity mapping constitutes a first-order minimum-norm kinematic approximation, which explicitly assumes that the low-level operational-space controller executes the primary Cartesian command without inducing significant auxiliary null-space motion (i.e., $(I - J_e^\dagger J_e)\nu \approx 0$).

If the proxy center has a fixed offset d_i^ℓ in its supporting-link frame, its world-frame offset is $\rho_i = R_{\ell(i)} d_i^\ell$, and the proxy-center twist becomes:

$$\begin{bmatrix} v_i \\ \omega_i \end{bmatrix} = S_i \begin{bmatrix} v_{\ell(i)} \\ \omega_{\ell(i)} \end{bmatrix}, \qquad S_i = \begin{bmatrix} I & -[\rho_i]_\times \\ 0 & I \end{bmatrix}. \tag{14}$$

Consequently, the proxy center evolves as $\dot{p}_i = v_i$, while the proxy shape parameterization evolves according to:

$$\dot{\eta}_i = \left([\omega_i]_\times \bar{Q}_i^{-1} - \bar{Q}_i^{-1}[\omega_i]_\times\right) z_i + \bar{Q}_i^{-1}\left(I - z_i z_i^\top\right) u_{z,i,t}, \tag{15}$$

where $u_{z,i,t}$ is the pair-specific separation-direction update.

Defining the rotational barrier coefficient $\Gamma_i \in \mathbb{R}^3$ by:

$$[\Gamma_i]_k = g_i^\top \left([e_k]_\times \bar{Q}_i^{-1} - \bar{Q}_i^{-1}[e_k]_\times\right) z_i, \qquad k \in \{1,2,3\}, \tag{16}$$

where e_k is the k-th Euclidean basis vector, the chain rule yields the local barrier rate:

$$\widehat{\dot{h}}_i = \begin{bmatrix} -n_i \\ \Gamma_i \end{bmatrix}^\top \begin{bmatrix} v_i \\ \omega_i \end{bmatrix} + a_{z,i}^\top u_{z,i,t}, \tag{17}$$

with $a_{z,i} = \left(I - z_i z_i^\top\right) \bar{Q}_i^{-\top} g_i$.

Combining Eqs. 13 and 14, we express the barrier rate entirely in terms of the shared Cartesian command:

$$\begin{bmatrix} a_{v,i} \\ a_{\omega,i} \end{bmatrix} = \frac{1}{\Delta t} \left(S_i T_{\ell(i)\leftarrow e}\right)^\top \begin{bmatrix} -n_i \\ \Gamma_i \end{bmatrix}. \tag{18}$$

which yields the strictly affine barrier derivative model:

$$\widehat{\dot{h}}_i = a_{v,i}^\top u_{v,t} + a_{\omega,i}^\top u_{\omega,t} + a_{z,i}^\top u_{z,i,t}. \tag{19}$$

For an end-effector-attached proxy, $\mathcal{T}_{\ell(i)\leftarrow e} = I$, and this formulation elegantly reduces to the standard offset-coupled dynamics $\dot{p}_i = u_{v,t}/\Delta t + (u_{\omega,t}/\Delta t) \times \rho_i$. Thus, Eq. 19 applies uniformly across all candidate proxy types.

Recall from Section 4.4 the augmented QP decision variable:

$$\mathbf{u}_t = \left[u_{v,t}^\top, u_{\omega,t}^\top, \{u_{z,j,t}^\top\}_{j\in\mathcal{K}_t}\right]^\top. \tag{20}$$

The local CBF constraint associated with active proxy $i \in \mathcal{K}_t$ can therefore be written in the standard linear inequality form:

$$A_i^{\mathrm{cbf}} \mathbf{u}_t \geq b_i^{\mathrm{cbf}}, \tag{21}$$

where

$$A_i^{\mathrm{cbf}} = \left[a_{v,i}^\top, a_{\omega,i}^\top, 0, \ldots, a_{z,i}^\top, \ldots, 0\right], \qquad b_i^{\mathrm{cbf}} = -\alpha_i(\sigma_t) h_i. \tag{22}$$

Stacking these row vectors and scalar bounds for all $i \in \mathcal{K}_t$ constructs the unified linear constraint matrix executed by the phase-aware QP.

B.2 Proof of the One-Step Sampled-Data Condition

Proof. By the operational feasibility of the unified CBF-QP (Eq. 9), the optimized solution $\mathbf{u}_t^\star$ satisfies:

$$\widehat{\dot{h}}_{i,t}(\mathbf{u}_t^\star) + \alpha_i(\sigma_t) h_{i,t} \geq 0, \qquad \forall i \in \mathcal{K}_t.$$

Substituting this inequality into the local first-order sampled-data prediction model,

$$\widehat{h}_{i,t+\Delta t} = h_{i,t} + \Delta t\, \widehat{\dot{h}}_{i,t}(\mathbf{u}_t^\star),$$

we obtain:

$$\widehat{h}_{i,t+\Delta t} \geq h_{i,t} - \alpha_i(\sigma_t)\Delta t\, h_{i,t} = \big(1 - \alpha_i(\sigma_t)\Delta t\big) h_{i,t}.$$

Given the premise that the initial state is safe ($h_{i,t} \geq 0$) and the scheduled barrier gain is bounded ($0 < \alpha_i(\sigma_t)\Delta t \leq 1$), the scalar multiplier $(1 - \alpha_i(\sigma_t)\Delta t)$ strictly resides in $[0, 1)$. Consequently, the right-hand side remains non-negative:

$$\widehat{h}_{i,t+\Delta t} \geq \big(1 - \alpha_i(\sigma_t)\Delta t\big) h_{i,t} \geq 0.$$

This completes the proof, confirming that the discrete-time projection preserves the non-negative predicted separation for all active constraints over the subsequent interval (Blanchini, 1999).

B.3 Intersample barrier bound

We analyze the interval $[t, t+\Delta t]$ with the obstacle geometry (c_t, r_t), proxy attachments, and active set $\mathcal{K}_t$ fixed. For each $i \in \mathcal{K}_t$, let $\bar{h}_{i,t}(\tau)$ denote the ellipsoid-pair barrier along the ideal continuous proxy motion induced by the held command $u_t^\star$, for $\tau \in [0, \Delta t]$. The separation direction follows the normalized interpolation of Eq. 39, obtained by replacing Δt with τ in its right-hand side (Breeden et al., 2021; Taylor et al., 2022).

Assumption (Regular interval evolution). The proxy motion realizes the calibrated kinematics in Appendix B.1. For each active pair, $\bar{h}_{i,t}$ is twice continuously differentiable and admits a finite constant $M_{i,t} \geq 0$ such that

$$\bar{h}_{i,t}(0) = h_{i,t}, \qquad \bar{h}'_{i,t}(0) = \widehat{\dot{h}}_{i,t}(\mathbf{u}_t^\star), \qquad \bar{h}''_{i,t}(\tau) \geq -M_{i,t} \quad \text{for all } \tau \in [0, \Delta t]. \tag{23}$$

Here, $M_{i,t}$ bounds the adverse second-order variation of the composed barrier along the interval, including nonlinear proxy motion and the normalized separation-direction update.

Proposition 2 (Intersample separation condition) *Under the regular interval evolution assumption, a feasible solution $\mathbf{u}_t^\star$ to Eq. 9 satisfies, for every $i \in \mathcal{K}_t$,*

$$\bar{h}_{i,t}(\tau) \geq h_{i,t} + \tau \widehat{\dot{h}}_{i,t}(\mathbf{u}_t^\star) - \frac{1}{2} M_{i,t} \tau^2, \qquad 0 \leq \tau \leq \Delta t. \tag{24}$$

If

$$h_{i,t} \geq 0, \qquad \widehat{h}_{i,t+\Delta t} \geq \frac{1}{2} M_{i,t} \Delta t^2, \tag{25}$$

then $\bar{h}_{i,t}(\tau) \geq 0$ throughout the sampling interval. Strict inequalities in Eq. 25 yield strict separation of the corresponding ellipsoids throughout the interval.

Proof. Taylor's formula with integral remainder gives

$$\bar{h}_{i,t}(\tau) = h_{i,t} + \tau \widehat{\dot{h}}_{i,t}(\mathbf{u}_t^\star) + \int_0^\tau (\tau - s) \bar{h}''_{i,t}(s) \, \mathrm{d}s. \tag{26}$$

Applying Eq. 23 yields Eq. 24. Its right-hand side is concave in τ, with endpoint values $h_{i,t}$ and $\widehat{h}_{i,t+\Delta t} - \frac{1}{2} M_{i,t} \Delta t^2$. Condition 25 makes both endpoint values nonnegative, and concavity establishes nonnegativity over the entire interval. Strict endpoint inequalities establish strict positivity.

Combining Proposition 1 with Eq. 25 gives the sufficient clearance condition

$$\big(1 - \alpha_i(\sigma_t)\Delta t\big) h_{i,t} \geq \frac{1}{2} M_{i,t} \Delta t^2. \tag{27}$$

Eq. 25 additionally uses any surplus barrier rate provided by the QP solution. The quadratic remainder connects the one-step prediction to the full interval evolution. QP feasibility alone does not imply Eq. 25; applying the intersample certificate additionally requires a valid bound $M_{i,t}$ and verification of this condition for every active pair. Strict ellipsoid separation implies separation of the represented bodies provided that the obstacle and robot-side ellipsoids contain their corresponding collision geometries.

C GeoPhase Implementation Details

This section provides the implementation details for the obstacle geometry prediction module, the finite-state manipulation phase tracker, the multi-proxy robot construction, and the online CBF-QP solver utilized in GeoPhase.

C.1 Obstacle Ellipsoid Prediction Head

Visual Token Aggregation. Let $H_t = [h_{t,1}, \ldots, h_{t,L_t}]$ denote the patch-level visual-token features extracted from the frozen VLA backbone at control step t, where $h_{t,j} \in \mathbb{R}^D$. Retaining only the valid image-branch tokens, we compute the pooled visual embedding as:

$$Z_t = \frac{1}{|\mathcal{V}_t|} \sum_{j \in \mathcal{V}_t} h_{t,j}, \qquad Z_t \in \mathbb{R}^D, \tag{28}$$

where $\mathcal{V}_t$ represents the set of valid visual-token indices determined by the VLA input processing pipeline. For the $\pi_{0.5}$ policy deployed in our experiments, the feature dimension is $D = 2048$.

Input Normalization and Prediction Network. The pooled visual embedding is normalized using dataset-level statistics computed from the geometry-head training split:

$$\bar{Z}_t = \frac{Z_t - \mu_Z}{\sigma_Z + \epsilon}, \tag{29}$$

where μ_Z and σ_Z denote the per-dimension mean and standard deviation vectors, respectively.

As introduced in Section 4.1, the ellipsoid predictor f_θ is parameterized as a lightweight MLP with hidden dimensions of 512 and 256. Layer normalization is applied to the input, and each hidden layer utilizes a SiLU activation function. Driven by the safety-critical boundary loss during training, the network learns to isolate the primary obstacle from the target manipulation objects encoded within the global representation. The network directly maps the normalized embedding to the spatial parameters of this primary safety-critical obstacle:

$$(c_t, r_t) = f_\theta(\bar{Z}_t), \qquad c_t \in \mathbb{R}^3, \quad r_t \in \mathbb{R}^3_{>0}. \tag{30}$$

To prevent degenerate or unrealistically scaled obstacle representations, the predicted semi-axis lengths are clamped to a physically plausible range during inference. Fig. 6 visualizes the resulting obstacle ellipsoids predicted by the learned geometry head across various object categories, demonstrating its capability to capture structured collision geometry directly from visual embeddings.



Figure 6: Visualization of learned obstacle ellipsoids across diverse object categories. The predictor successfully extracts tight, structured geometric representations directly from visual embeddings.

Geometry Supervision and Dataset Construction. Ground-truth obstacle geometry is constructed directly from the exact collision meshes provided by the SafeLIBERO simulator. For each obstacle instance, we fit an axis-aligned outer ellipsoid

$$\mathcal{E}_t^\star = \left\{ y \in \mathbb{R}^3 : \| (y - c_t^\star) \oslash r_t^\star \|_2 \le 1 \right\}, \tag{31}$$

whose center $c_t^\star$ and semi-axis lengths $r_t^\star$ are determined from the complete collision geometry. The semi-axis lengths are enlarged as needed until $\mathcal{G}_t \subseteq \mathcal{E}_t^\star$, where $\mathcal{G}_t$ denotes the obstacle's collision geometry. This containment check is performed separately for each obstacle instance when constructing the supervision labels; only verified enclosing ellipsoids are retained as training targets. At deployment, the geometry head directly predicts the obstacle ellipsoid $\mathcal{O}_t$ from VLA visual embeddings. Offline evaluation confirmed that the predicted ellipsoids contained the corresponding obstacle collision geometries in the evaluated scenes.

The geometry-head dataset consists of 63,187 frame-level samples collected exclusively from dedicated data-collection rollouts. To prevent temporal data leakage, the dataset is partitioned into 90% training and 10% validation sets grouped strictly at the rollout level. No RGB observations, proprioceptive states, robot trajectories, or simulator geometry labels from any evaluation rollout are exposed during training or model selection. While the training and evaluation sets share the standard obstacle categories and geometric templates, the object poses, scene layouts, visual occlusions, and executed trajectories remain strictly independent across rollouts. This split rigorously evaluates the predictor's spatial and configurational generalization—its ability to robustly extract structured geometry under novel spatial arrangements and viewpoints—within a defined operational domain.

Optimization and Runtime. The geometry head is trained via a composite objective function that weights both geometric regression (center and semi-axis lengths) and safety-critical metrics (barrier value and barrier-gradient direction). Let (c_t, r_t) denote the predicted obstacle geometry parameters and $(c_t^\star, r_t^\star)$ their ground-truth counterparts. The loss function $\mathcal{L}$ is defined as:

$$\mathcal{L} = w_{\text{bdry}} \Big[\ell_{\text{s1}}(c_t, c_t^\star) + \ell_{\text{s1}}(r_t, r_t^\star) + \lambda_h \ell_{\text{s1}}(h_t, h_t^\star) + \lambda_\nabla \big(1 - \cos(\nabla h_t, \nabla h_t^\star)\big) \Big], \tag{32}$$

where ℓ_{s1} denotes the Smooth-L1 loss. The terms h_t and ∇h_t represent the scalar barrier value and its spatial gradient derived from the predicted geometry, supervised by the ground-truth targets $h_t^\star$ and $\nabla h_t^\star$. The loss components are balanced using $\lambda_h = 0.5$ and $\lambda_\nabla = 0.5$. To prioritize collision avoidance, samples located near the safety boundary are assigned a higher weight of $w_{\mathrm{bdry}} = 4.0$, while other samples use $w_{\mathrm{bdry}} = 1.0$. The VLA backbone remains strictly frozen throughout this training phase.

The network incorporates a dropout rate of 0.05 and is optimized using AdamW (Loshchilov et al., 2017) with an initial learning rate of 10^{-3} and weight decay of 10^{-4}. Training proceeds for 100 epochs with a batch size of 1024, employing a cosine annealing learning rate schedule and gradient norm clipping at 1.0.

At deployment, the compact geometry head infers the obstacle parameters in a single forward pass. Amortized over the VLA action-chunk execution, this visual-geometry processing requires an average of $12.63\,\mathrm{ms}$ per control step across the complete SafeLIBERO evaluation suite, ensuring minimal latency overhead (detailed in Table 6).

C.2 Finite-State Manipulation Phase Tracker

As introduced in Section 4.2, GeoPhase estimates the current manipulation phase $\sigma_t \in \mathcal{S}$ online using a continuous FSM driven by the nominal gripper command and recent proprioceptive history. The FSM executes the following cyclic transition sequence:

$$\mathrm{approach} \rightarrow \mathrm{grasp} \rightarrow \mathrm{transport} \rightarrow \mathrm{place} \rightarrow \mathrm{approach} \cdots$$

The tracker initializes and remains in *approach* until the nominal action issues a gripper-closure command, which triggers the transition to *grasp*. Once closure is physically complete and translational arm motion resumes, the FSM advances to *transport*, systematically activating the held-object proxy $\mathcal{P}_{\mathrm{held}}$. Importantly, the ***grasp-to-transport*** transition introduces the previously inactive held-object proxy. Before each newly activated proxy is admitted to the active constraint set, GeoPhase evaluates its current barrier value using the current robot and obstacle poses. If the required initialization condition is not satisfied, the corresponding Cartesian command is suppressed until a valid initialization is obtained. Thus, proxy activation is explicitly checked at the transition that enlarges the protected geometry. A subsequent gripper-opening command initiates the transition to *place*. Upon successful release, $\mathcal{P}_{\mathrm{held}}$ is deactivated. If the task horizon has not terminated, the FSM automatically resets to *approach* to begin the next manipulation cycle.

To mitigate spurious state switches caused by transient or delayed gripper-state observations, the tracker incorporates a temporal robustness mechanism: a phase transition is executed only after the triggering condition is consistently observed across several consecutive control steps. If a transition condition is not sustained, the FSM conservatively retains the current phase.

This robust cyclic architecture allows the four phases to repeat seamlessly within a single episode. For instance, tasks within the SafeLIBERO LONG suite require the sequential manipulation of two distinct objects, thereby naturally exercising two complete *approach–grasp–transport–place* cycles. All phase transitions are inferred strictly online from execution-time states, eliminating any reliance on predefined, task-specific symbolic annotations.

C.3 Robot-Side Proxy Construction

GeoPhase constructs the candidate multi-proxy set $\mathcal{P}_t$ from distinct geometric envelopes: the end-effector ($\mathcal{P}_{\mathrm{eef}}$), the dynamically instantiated held object ($\mathcal{P}_{\mathrm{held}}$), and relevant robot body links ($\mathcal{P}_{\mathrm{body}}$). As defined in Section 3, each proxy $\mathcal{E}_{i,t}$ is parameterized by its center $p_{i,t}$, orientation $R_{i,t}$, and semi-axis lengths q_i.

Each proxy is rigidly attached to a supporting kinematic frame $\ell(i)$. For the evaluated manipulator, the body proxies $\mathcal{P}_{\mathrm{body}}$ are assigned to the primary collision-prone links (e.g., the forearm and wrist links). The corresponding local center offsets d_i^ℓ, orientations R_i^{local}, and semi-axis lengths q_i are statically derived directly from the publicly available robot URDF collision meshes, ensuring standard reproducibility. Given these fixed local parameters, the world-frame pose updates strictly via forward kinematics:

$$p_{i,t} = p_{\ell(i),t} + R_{\ell(i),t} d_i^\ell, \qquad R_{i,t} = R_{\ell(i),t} R_i^{\mathrm{local}}. \tag{33}$$

For an end-effector proxy, the supporting frame is $\ell(i) = e$ with local offset $d_i^{\ell} = d_i$. As derived in Appendix B.1, the spatial twist of any configured body link or geometric proxy is analytically mapped to the shared Cartesian command, natively accounting for offset-induced translational–rotational coupling.

The held-object proxy $\mathcal{P}_{\text{held}}$ is activated exclusively when the FSM transitions into the *transport* phase. Rather than estimating the geometry of each manipulated object online, we use one universal conservative ellipsoid in the end-effector frame. Its semi-axis lengths, q_{held}, are selected as the coordinate-wise maximum envelope over all objects. Containment against the collision meshes of all evaluated held objects is explicitly verified offline. Its world-frame pose is rigidly coupled to the end-effector:

$$p_{held,t} = p_{e,t} + R_{e,t} d_{held}, \quad R_{held,t} = R_{e,t} R_{held}^{local}. \tag{34}$$

The identity orientation $R_{\text{held}}^{\text{local}} = I$ is utilized when the held-object proxy is aligned with the end-effector frame. The identical fixed parameters (d_{held}, $R_{\text{held}}^{\text{local}}$, q_{held}) are deployed uniformly across all tasks and objects in both simulation and real-world experiments.



Figure 7: Representative rollouts illustrating the dynamic activation of phase-aware multi-proxy constraints across the four distinct manipulation phases.

As established in Section 4.2, distinct manipulation phases exhibit varying geometric risk profiles and operational tolerances. GeoPhase dynamically schedules the proxy subset $\mathcal{P}_{\sigma_t}$ to maintain continuous body coverage across all phases, while restricting the held-object proxy exclusively to the transport phase to reflect the shifting kinematic boundaries of the robot-object system:

$$\mathcal{P}_{\sigma_t} = \begin{cases} \mathcal{P}_{\text{eef}} \cup \mathcal{P}_{\text{body}}, & \sigma_t \in \{\text{approach}, \text{grasp}, \text{place}\}, \\ \mathcal{P}_{\text{eef}} \cup \mathcal{P}_{\text{held}} \cup \mathcal{P}_{\text{body}}, & \sigma_t = \text{transport}. \end{cases} \tag{35}$$

Under our execution convention, a gripper-opening command initiates the *place* phase. To account for mechanical actuation latency, $\mathcal{P}_{\text{held}}$ is deactivated only after a brief temporal buffer ensures complete physical separation. GeoPhase subsequently maintains continuous end-effector and body protection, assuming the designated target region provides sufficient clearance for the object's free-fall.

Table 5: Phase-dependent safety parameters used by GeoPhase.

Phase	$\alpha_i(\sigma_t)$ [s^{-1}]	$h_{\mathrm{free}}(\sigma_t)$ [m]
Approach	10	0.10
Grasp	10	0.06
Transport	15	0.10
Place	5	0.03

With this continuous-coverage schedule, *grasp*-to-*transport* becomes the sole transition introducing a previously inactive proxy ($\mathcal{P}_{\mathrm{held}}$). GeoPhase evaluates this proxy–obstacle barrier online prior to activation; in all evaluated rollouts, the newly activated proxy consistently satisfied $h_{i,t} \geq 0$.

Fig. 7 visually demonstrates this dynamic scheduling, illustrating how the multi-proxy constraints adapt across the four manipulation stages during representative task rollouts. Following this phase-aware schedule, the active constraint set $\mathcal{K}_t$ is subsequently filtered from $\mathcal{P}_{\sigma_t}$ using the free-space geometric activation thresholds detailed in the following subsection.

C.4 Constraint Activation and Solver Details

At each control step, GeoPhase solves Eq. 9 using the current barrier gradients and the calibrated action mapping in Appendix B.1, then holds the filtered command for $\Delta t = 0.05\,\mathrm{s}$. The proxy poses, manipulation phase, and active constraint set are recomputed at the subsequent control step. Consequently, the implementation utilizes a strictly one-step horizon, deliberately avoiding multi-step rollouts of the action chunk within the QP.

For each candidate proxy $\mathcal{E}_{i,t} \in \mathcal{P}_{\sigma_t}$, GeoPhase evaluates the corresponding barrier value $h_{i,t}$. The proxy–obstacle pair is activated if this value falls below a phase-dependent free-space threshold:

$$\mathcal{K}_t = \{i \mid \mathcal{E}_{i,t} \in \mathcal{P}_{\sigma_t}, \quad h_{i,t} < h_{\mathrm{free}}(\sigma_t)\}\,. \tag{36}$$

This set $\mathcal{K}_t$ indexes the active constraints incorporated into the unified CBF-QP. The threshold $h_{\mathrm{free}}(\sigma_t)$ determines when a proxy–obstacle pair enters the QP, whereas $\alpha_i(\sigma_t)$ controls the admissible barrier decrease. For $h_{i,t} > 0$, a larger gain permits a faster decrease; the resulting intervention depends jointly on the gain, activation threshold, and eligible proxy set. The specific scalar values deployed in our simulation experiments are detailed in Table 5; Appendix E.4 provides a comprehensive ablation of these dynamic parameters.

When $\mathcal{K}_t \neq \emptyset$, the optimization is solved using OSQP (Stellato et al., 2020). Because the Cartesian commands are executed as zero-order-held increments over Δt, the corresponding continuous-time spatial rates are defined as:

$$v_t = \frac{\Delta p_t}{\Delta t}, \qquad \omega_t = \frac{\Delta \theta_t}{\Delta t}. \tag{37}$$

The increment-to-rate scaling factor is directly absorbed into the affine QP coefficients, allowing the solver to optimize the command increments natively. Consistent with Section 4.4, the one-step sampled-data prediction is enforced as:

$$\widehat{h}_{i,t+\Delta t} = h_{i,t} + \Delta t\, \widehat{\dot{h}}_{i,t}(\mathbf{u}_t^\star),$$

where $\widehat{\dot{h}}_{i,t}$ denotes the affine barrier derivative. No multi-step prediction is computed.

Each proxy–obstacle pair maintains an auxiliary separation direction $z_{i,t} \in \mathbb{S}^2$. When a pair is newly considered, the direction is initialized, whenever $\|c_t - p_{i,t}\|_2 > 0$, as

$$z_{i,t} = \frac{c_t - p_{i,t}}{\|c_t - p_{i,t}\|_2}. \tag{38}$$

This initialization supplies a well-defined starting direction but does not itself certify $h_{i,t} \geq 0$, because a non-positive barrier value may result from the selected supporting direction. GeoPhase therefore evaluates the resulting barrier online for every newly activated pair. For a persistent constraint that remains active across consecutive steps, the direction from the preceding step is retained.

Following a feasible QP solve, each active direction is updated in the tangent space of $\mathbb{S}^2$ and renormalized:

$$z_{i,t+\Delta t} = \text{normalize}\left(z_{i,t} + \Delta t \left(I - z_{i,t} z_{i,t}^\top\right) u^\star_{z,i,t}\right), \qquad i \in \mathcal{K}_t. \tag{39}$$

To enhance solver efficiency, the auxiliary decision variable $u_{z,i,t}$ is warm-started from the preceding solution for persistent constraints, and initialized to zero for newly activated ones. If the solver fails to return a feasible solution, GeoPhase triggers an emergency stop by setting all Cartesian increments to zero.

D Experimental Setup and Evaluation Protocol

D.1 Hardware Environment

Simulation. All simulation experiments were conducted on an Ubuntu 20.04 server equipped with an Intel Xeon Gold 6326 CPU, 4× NVIDIA RTX 4090 (24GB) GPUs, and 512GB RAM.

Real-World. Our real-world robotic experiments are conducted using a Franka Research 3 robotic arm, equipped with a Franka Hand gripper and two ZED cameras for visual input. The robotic arm is controlled by a NUC mini laptop featuring an Intel Core i9-12900H CPU, which is responsible for the real-time execution of low-level control tasks.

D.2 SafeLIBERO Benchmark

We evaluate GeoPhase on SafeLIBERO (Hu et al., 2025), a safety-critical manipulation benchmark derived from LIBERO (Liu et al., 2023) and implemented in Robosuite (Zhu et al., 2020). The benchmark uses a 7-DoF Franka Emika Panda manipulator and contains 16 tasks drawn from the Spatial, Goal, Object, and Long suites, with four tasks from each suite. Each task is instantiated under two obstacle-interference levels. Level I places an obstacle near the task-relevant interaction region, requiring the robot to maintain clearance without preventing the proximity needed for grasping or placement. Level II positions the obstacle along the nominal manipulation path, requiring a more substantial correction of the policy trajectory. The LONG suite additionally evaluates repeated phase transitions because its tasks require two sequential object manipulations within the same episode.

For each task–level combination, we evaluate 50 episodes with randomized object poses, yielding 32 scenarios and 1,600 episodes in total. All locally evaluated methods use the same task definitions, initial-state seeds, obstacle configurations, episode horizons, observation streams, and low-level controller. GeoPhase and the unfiltered policy additionally share the same pretrained $\pi_{0.5}$-LIBERO checkpoint, ensuring that their difference is attributable solely to the proposed runtime safety layer.

D.3 Baselines and Reproduction

We compare GeoPhase with the unfiltered $\pi_{0.5}$ policy and three representative safety-enhanced methods: AEGIS, SafeDojo, and KNOWS. The unfiltered $\pi_{0.5}$ policy provides the nominal task-execution baseline and receives no action-space safety correction.

AEGIS (Hu et al., 2025) is reproduced using the authors' publicly released implementation, retaining its perception and CBF-based control pipeline. We tune AEGIS and GeoPhase on the same validation instances with comparable hyperparameter-search budgets. Each method's selected configuration is fixed before evaluation on the held-out test instances. Both methods use the same nominal policy, test initializations, episode horizons, and low-level controller.

At the time of evaluation, official implementations of SafeDojo (Tang et al., 2026) and KNOWS (Park et al., 2026) were not publicly available. To avoid misrepresenting these methods via unverified surrogate implementations, we compare against the metrics directly reported in their original publications. While we structurally align our evaluation with their SafeLIBERO protocols (identical task suites, obstacles, and metric definitions), the baselines differ in their policy configurations. Specifically, SafeDojo utilizes a different VLA architecture (OpenVLA (Kim et al., 2024)) and training pipeline, rendering its comparison system-level. Conversely, KNOWS applies safety filtering to the same $\pi_{0.5}$ base policy used by GeoPhase. To ensure strict comparability, our local

evaluation of the unfiltered $\pi_{0.5}$ baseline adopts the identical policy checkpoint and deterministic environment seeds utilized in the KNOWS benchmark, yielding the exactly matched baseline metrics. Only directly comparable task suites, interference levels, and metric definitions are included. When a source reports collision avoidance rate, we convert it to collision rate as $\mathrm{CR} = 1 - \mathrm{CAR}$, so that all results follow the same metrics.

D.4 METRICS

For each method and evaluation split, let N denote the number of evaluated episodes. For episode n, let $s_n \in \{0, 1\}$ denote the benchmark task-success indicator, and let $c_n \in \{0, 1\}$ indicate whether at least one safety-relevant collision is recorded under the evaluation protocol. We report three primary episode-level metrics:

$$\mathrm{TSR} = \frac{1}{N}\sum_{n=1}^{N} s_n, \qquad \mathrm{CR} = \frac{1}{N}\sum_{n=1}^{N} c_n, \qquad \mathrm{SSR} = \frac{1}{N}\sum_{n=1}^{N} s_n(1 - c_n). \tag{40}$$

Task Success Rate (TSR) measures the fraction of episodes that complete the task, irrespective of collision. Collision Rate (CR) measures the fraction containing at least one collision. Safe Success Rate (SSR) measures the fraction that complete the task without collision and serves as our primary metric of safe task completion. All three rates are reported as percentages.

We additionally report the mean execution length in environment time steps (ETS):

$$\mathrm{ETS} = \frac{1}{N}\sum_{n=1}^{N} T_n, \tag{41}$$

where T_n is the number of environment steps executed until episode termination. ETS is averaged over the same set of episodes as the three rate-based metrics.

In addition, for each frame n, let G_n denote the inflated ground-truth reference ellipsoid and P_n the predicted ellipsoid. Coverage measures the fraction of the reference ellipsoid's volume covered by the prediction, and intersection-over-union (IoU) measures their volumetric agreement:

$$\mathrm{Coverage}_n = \frac{\mathrm{Vol}(G_n \cap P_n)}{\mathrm{Vol}(G_n)}, \qquad \mathrm{IoU}_n = \frac{\mathrm{Vol}(G_n \cap P_n)}{\mathrm{Vol}(G_n \cup P_n)}. \tag{42}$$

The reported values are macro averages over all evaluation frames, with each frame weighted equally. The volumes are estimated using a fixed voxelization procedure. Because the reference ellipsoid includes conservative padding, a prediction can fully contain the obstacle without covering the entire reference volume.

E ADDITIONAL EXPERIMENTAL RESULTS AND ANALYSES

E.1 ADDITIONAL OUTCOME AND PROXY ANALYSES

Task success and collision outcomes. Figure 8 visualizes the results from Table 1 by partitioning episode terminations into four mutually exclusive categories: safe success, success with collision, failure with collision, and safe failure. Averaged across both interference levels, GeoPhase reduces the fraction of episodes that succeed with a collision from 40.63% (base policy) to 3.81%, while increasing the safe success rate from 15.75% to 62.56%. This demonstrates that GeoPhase effectively converts unsafe task completions and collision-induced failures into safe successes.

Proxy-specific intervention patterns. Figure 9 further analyzes the intervention distributions from Table 4. The end-effector (EEF) intervention rate remains consistently near 72% as the protected geometry expands, indicating that the held-object and body proxies trigger interventions for complementary collision modes rather than simply duplicating EEF constraints. Consequently, adding the held-object proxy reduces the collision rate (CR) by 7.5 percentage points, and incorporating body proxies provides an additional 9.1 percentage-point reduction.

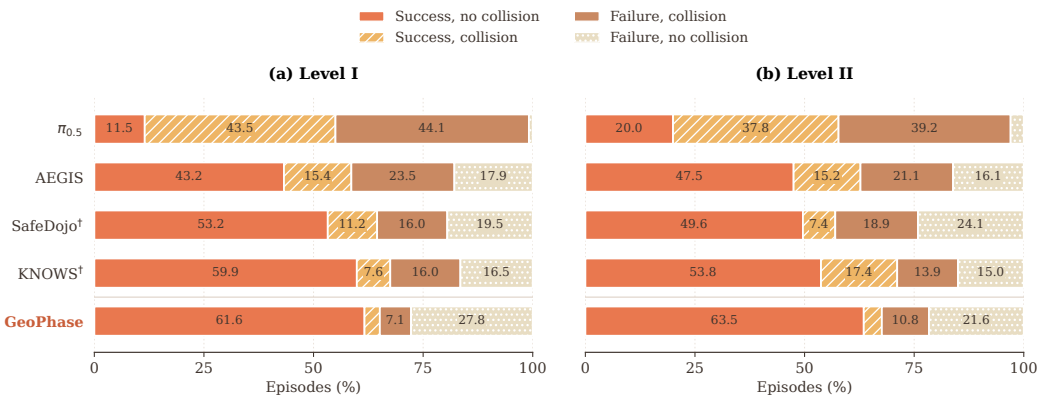


Figure 8: Outcome composition of simulated episodes under Level I and Level II interference. GeoPhase attains the highest safe success proportion by effectively minimizing collision-induced task failures.

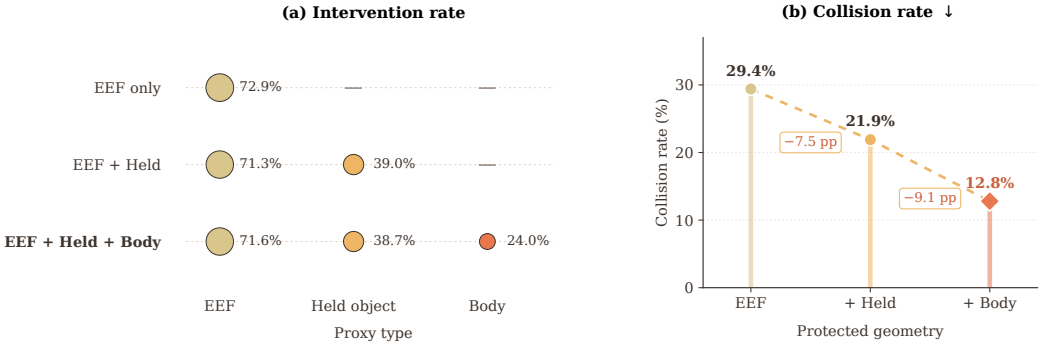


Figure 9: Intervention patterns and collision rates under varied proxy configurations. Progressively incorporating held-object and body proxies covers complementary collision modes, yielding monotonic reductions in the overall collision rate.

E.2 System Runtime and Efficiency Analysis

We measure end-to-end runtime on a profiling sample comprising 1,600 control steps and 323 policy queries drawn from the four SafeLIBERO task suites. To prevent latency spikes during policy updates, VLA inference and geometry processing are executed asynchronously once per policy query. Their computational costs are reported as amortized values across the corresponding control steps to reflect the overall system throughput. Conversely, the local phase-aware safety layer—including phase tracking, proxy updates, constraint construction, QP solution, and post-processing—runs synchronously within the low-level control loop at every control step. Table 6 summarizes the resulting runtime profile.

The QP is activated on 86.6% of the profiled control steps. Its mean solution time is 7.98 ms when conditioned on activation and 6.92 ms when amortized over all control steps. Constraint construction contributes 2.79 ms per step, while phase tracking, proxy updates, and post-processing jointly require 0.50 ms. The local safety layer therefore incurs an average latency of 10.21 ms per control step. Table 7 provides the complete breakdown.

Combining the amortized geometry-processing latency (12.63 ms) with the local safety-layer latency (10.21 ms), GeoPhase introduces an average safety overhead of **22.84** ms per control step. Because the VLA and geometry modules run asynchronously, they do not block the strict 50 ms control deadline during policy updates. Instead, the synchronous local safety layer (requiring only 10.21 ms) guarantees that geometric constraints are consistently enforced at **21.3** Hz, filtering the active action chunk using the most recently updated obstacle geometry.

Table 6: Runtime profile of GeoPhase. Latency is reported in ms, and control rate is reported in Hz.

Suite	VLA	Geometry	Safety Layer	Control Compute	Rate
Spatial	14.57	10.97	11.25	43.23	23.1
Object	14.31	13.57	7.01	44.90	22.3
Goal	14.49	13.58	11.65	49.33	20.3
Long	14.28	12.41	10.94	50.15	19.9
Overall	**14.41**	**12.63**	**10.21**	**46.90**	**21.4**

Table 7: Average per-step latency of the local safety layer. QP/Active denotes the mean solver latency conditioned on QP activation, whereas QP/Step denotes its amortized contribution over all profiled control steps. All values are reported in ms.

Suite	Phase	Proxy	Constraint	QP/Active	QP/Step	Post.	Total
Spatial	0.03	0.34	2.92	7.95	7.81	0.15	11.25
Object	0.03	0.37	2.30	8.20	4.22	0.09	7.01
Goal	0.03	0.35	3.12	8.26	7.99	0.16	11.65
Long	0.03	0.31	2.82	7.64	7.64	0.14	10.94
Overall	**0.03**	**0.34**	**2.79**	**7.98**	**6.92**	**0.13**	**10.21**

E.3 QP Solver Reliability.

We evaluate QP solver reliability at both the episode and individual-solve levels. A QP failure is recorded when the solver does not return a valid solution or when the returned solution fails the runtime feasibility check. **Across the full evaluation of** $1{,}600$ **episodes and** $\sim$ $422{,}800$ **control steps, the runtime monitor recorded no QP failures, and no episode triggered a solver-related emergency stop.**

For solve-level analysis, we use the $1{,}600$-step profiling sample described in Section E.2. The safety filter activated the QP on $1{,}386$ control steps, and every solve passed the runtime feasibility check, corresponding to an observed failure rate of $0/1{,}386$ (0.000%). The activated QPs contained between one and four safety constraints: 520, 529, 141, and 196 solves involved one, two, three, and four constraints, respectively. **These measurements establish the observed numerical reliability of the QP solver across the evaluated scenarios.**

E.4 Ablation of Phase-Dependent Safety Parameters

We conduct a lightweight ablation study of the two phase-dependent safety parameters on independent validation instances: the barrier gain α_i and the free-space activation threshold $h_{\mathrm{free},i}$. Unless otherwise specified, the remaining parameters are fixed to the values reported in Table 5. We evaluate each configuration using TSR, CR, SSR, and ETS.

Phase-dependent barrier gain. Fig. 11 reports a phase-wise sweep over $\alpha_i \in \{2.5, 5, 10, 15, 20\}$. In each sweep, only the gain of the selected phase is varied, while the gains of the other phases remain fixed. The resulting curves show phase-dependent sensitivity and motivate the final configuration

$$(\alpha_{\mathrm{approach}}, \alpha_{\mathrm{grasp}}, \alpha_{\mathrm{transport}}, \alpha_{\mathrm{place}}) = (10, 10, 15, 5).$$

Thus, their default points correspond to the same evaluation result: $\mathrm{TSR} = 66.5\%$, $\mathrm{CR} = 12.7\%$, $\mathrm{SSR} = 62.8\%$, and $\mathrm{ETS} = 265.2$ steps. These trends support phase-specific gains for balancing collision avoidance and task completion.

Phase-dependent free-space activation threshold. With the selected barrier gains fixed, Fig. 12 compares one-at-a-time threshold reductions from the reference configuration $(0.10, 0.06, 0.10, 0.05)\,\mathrm{m}$, ordered as approach, grasp, transport, and place. The reference configuration achieves a validation SSR of 62.5%. Reducing the approach, grasp, or transport threshold yields SSR values of 61.8%, 61.4%, and 62.4%, respectively. Reducing the place threshold from 0.05 to $0.03\,\mathrm{m}$ achieves the highest validation SSR among the evaluated configurations, with TSR

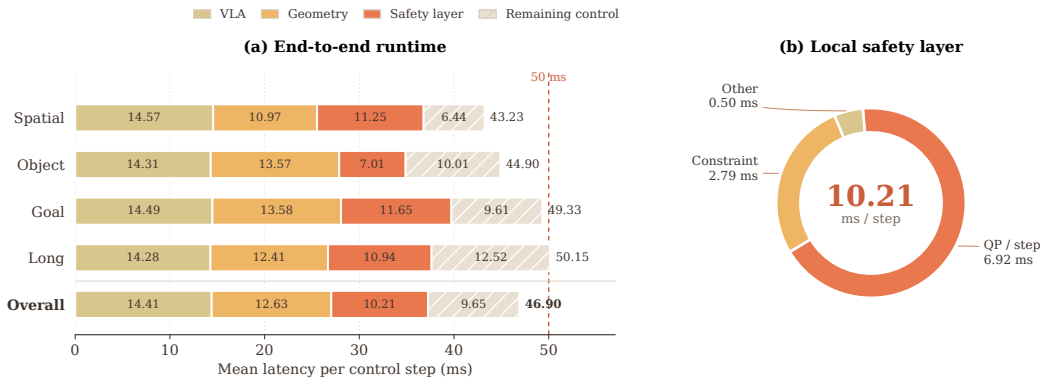


Figure 10: Runtime decomposition of the GeoPhase system. (a) Mean per-control-step latency across task suites. (b) Detailed breakdown of the local safety-layer latency.

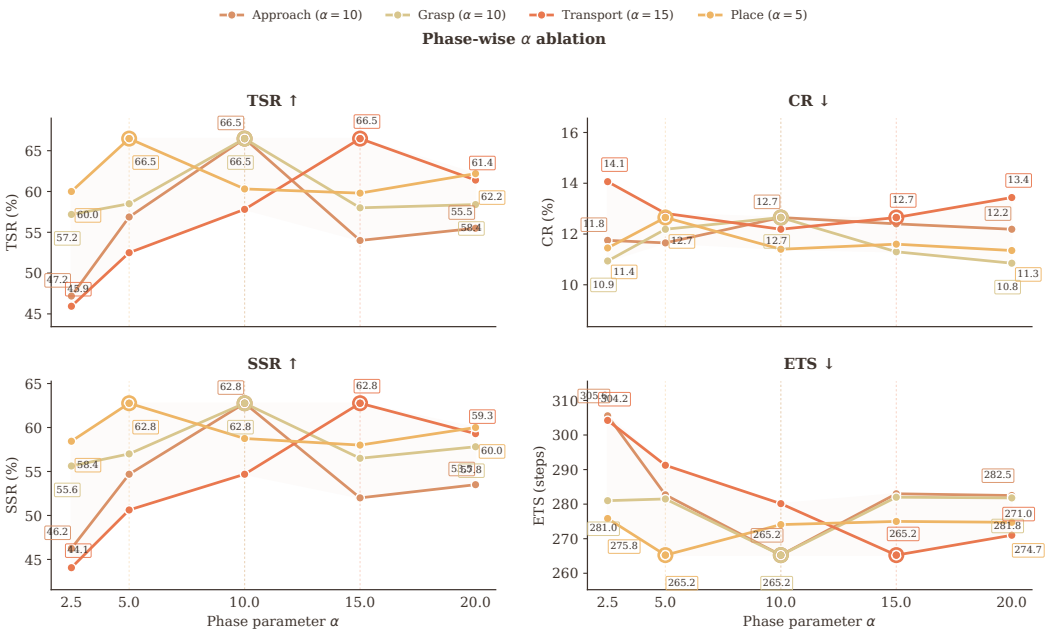


Figure 11: Phase-wise sensitivity to the barrier gain α_i. Each sweep varies the gain of one phase while keeping the remaining parameters fixed. The curves characterize how phase-specific gain choices affect task completion, collision rate, and execution length.

66.5%, CR 12.7%, SSR 62.8%, and ETS 265.2 steps. We therefore use

$$(h_{\text{free,approach}}, h_{\text{free,grasp}}, h_{\text{free,transport}}, h_{\text{free,place}}) = (0.10, 0.06, 0.10, 0.03)\,\text{m}.$$

in the final GeoPhase configuration.

E.5 Fixed Robust-Margin Calibration

To quantify the effect of a nonzero robustness margin, we conduct an additional calibration ablation in which the CBF constraint is augmented by a fixed phase-dependent margin. Specifically, we use $\Delta t = 0.05$ s and estimate the phase-wise curvature proxy $\overline{M}_{i,\sigma}$ from the 99th percentile of EEF accelerations collected from frozen-policy trajectories. We set $\varepsilon_{i,\sigma} = 0.0005$ and obtain

$$\rho_{\text{approach}} = 0.002720, \quad \rho_{\text{grasp}} = 0.003102, \quad \rho_{\text{transport}} = 0.007768, \quad \rho_{\text{place}} = 0.003463.$$

The resulting margin is incorporated into the QP constraint as $\rho_{i,\sigma}/\Delta t$.

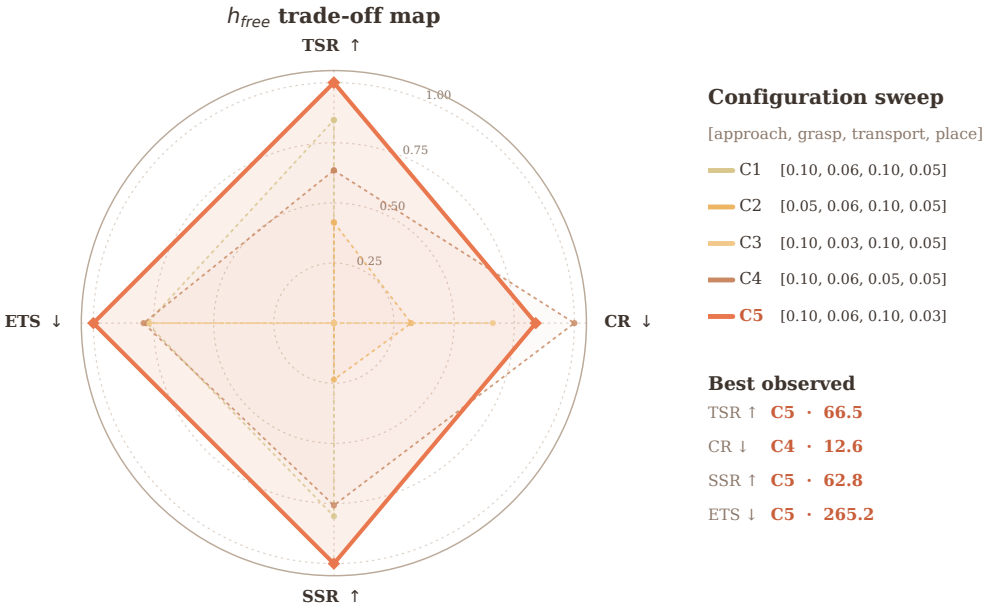


Figure 12: Ablation of the phase-wise activation threshold $h_{free,i}$. The radar chart demonstrates the safety-utility trade-off across different threshold reductions relative to the unified baseline.

We evaluate this variant on all four SafeLIBERO suites and both interference levels, using four tasks and ten episodes per suite–level combination (320 episodes in total). Relative to the zero-margin configuration, the fixed-margin variant reduces CR from 11.6% to 9.1%, but also reduces TSR from 63.4% to 54.1% and SSR from 60.3% to 52.5%, while increasing ETS from 272.4 to 286.6 steps. These results indicate that the fixed margin improves empirical collision avoidance at the cost of additional intervention and execution delay. They therefore support the existence of a safety–utility trade-off, rather than a uniformly improved performance profile.

E.6 Real-World Experiment Details

We evaluate the base policy $\pi_{0.5}$ and GeoPhase on five real-world manipulation tasks using a Franka Research 3 robot. The tasks are designed to rigorously evaluate spatial generalization across varied object poses, distinct physical placement requirements, and complex obstacle-induced trajectory corrections within common household scenarios. Fig. 13 shows representative time-ordered execution frames for each task.

Hardware Setup and Geometry Predictor Training. The physical workspace is observed via two ZED cameras providing RGB streams directly to the frozen $\pi_{0.5}$ VLA backbone. Because our learned geometry head operates directly on the VLA's intermediate pooled visual embeddings, it inherently bypasses explicit multi-view depth fusion. Under a fixed camera and direct base-frame supervision, it eliminates the need for explicit runtime extrinsic transformations, as the network implicitly internalizes the fixed spatial mapping. The predicted obstacle parameters (ellipsoid center and semi-axis lengths) are regressed strictly within the robot's base coordinate frame.

To deploy GeoPhase without simulator-induced domain shift, the real-world geometry head was trained from scratch on a modestly sized, manually annotated dataset of physical household objects. We utilized the exact same network architecture and loss formulation as the simulation model, with ground-truth bounding ellipsoids derived from actual physical measurements and calibrated robot kinematics. Crucially, the physical evaluation encompasses an out-of-distribution (OOD) novel object not present in this physical training set. For instance, the wine bottle acting as an obstacle in Task 4 was never seen during geometry training. Successfully avoiding this object provides preliminary evidence of the predictor's zero-shot geometric generalization to an unseen obstacle instance

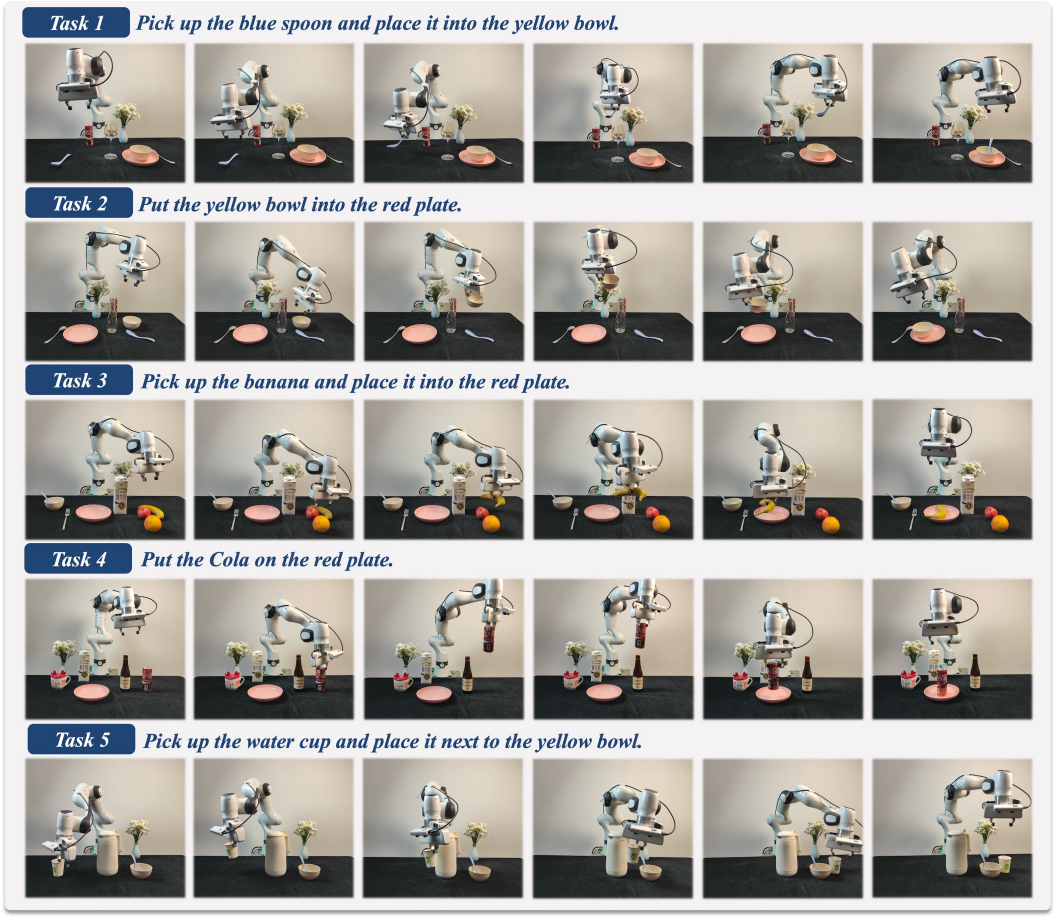


Figure 13: Sequential execution frames of GeoPhase across five diverse real-world manipulation tasks. The system successfully adapts to varying object geometries and obstacle-induced trajectory corrections.

based purely on VLA embeddings. Offline static validation for this novel object indicated a reliable geometric extraction, with an estimated center error of approximately 2 cm.

Task Design and Evaluation Protocol. For each task, we conduct 10 independent trials with randomized initial poses of the manipulated object and surrounding obstacles. A task is successful (TSR) if the commanded object reaches the designated target region within the episode horizon. A collision (CR) is recorded upon any unintended contact between the robot or held object and an obstacle. SSR denotes the fraction of trials satisfying both task completion and collision-free execution. The five designated tasks are:

- **Task 1: Pick up the blue spoon and place it into the yellow bowl.** Requires grasping and transporting a thin, elongated object through tabletop clutter, testing safe clearance alongside intentional target interaction.
- **Task 2: Put the yellow bowl onto the red plate.** Involves transporting an object with a footprint substantially larger than the end-effector, highlighting the necessity of the held-object proxy.
- **Task 3: Pick up the banana and place it onto the red plate.** Introduces an asymmetric carried-object geometry with upright obstacles obstructing the direct transport route.
- **Task 4: Put the Cola on the red plate.** Involves lifting and transporting a tall cylindrical object among upright distractors, notably including the unseen out-of-distribution wine bottle, testing zero-shot geometric extraction.

Table 8: Real-world manipulation performance across five designated tasks. GeoPhase consistently yields higher safe success rates compared to the unfiltered baseline.

	$\pi_{0.5}$ (Base)			GeoPhase		
Task	**TSR ↑**	**CR ↓**	**SSR ↑**	**TSR ↑**	**CR ↓**	**SSR ↑**
Task 1	70.0	60.0	10.0	70.0	10.0	60.0
Task 2	90.0	70.0	30.0	90.0	10.0	80.0
Task 3	80.0	70.0	10.0	90.0	20.0	70.0
Task 4	80.0	60.0	20.0	80.0	10.0	70.0
Task 5	80.0	60.0	20.0	100.0	10.0	90.0
Average	80.0	64.0	18.0	**86.0**	**12.0**	**74.0**

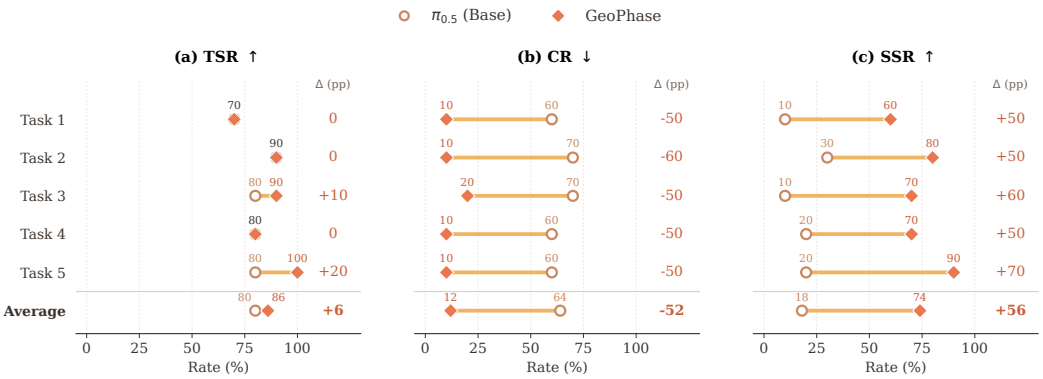


Figure 14: Real-world task performance comparison. GeoPhase consistently improves both TSR and SSR while significantly reducing CR across all manipulation tasks relative to the nominal base policy.

- **Task 5: Pick up the water cup and place it next to the yellow bowl.** Requires substantial trajectory deviation to safely guide the grasped cup around a large kettle obstacle.

Quantitative Results and Analysis. As detailed in Table 8 and Fig. 14, GeoPhase substantially outperforms the unfiltered baseline across all metrics. By dynamically enforcing multi-proxy geometric constraints, GeoPhase reduces the average collision rate (CR) from 64.0% to 12.0% and increases the safe success rate (SSR) from 18.0% to 74.0%.

Crucially, this safety improvement does not compromise the policy's semantic competence; rather, it actively prevents collision-induced task failures (e.g., knocking over targets or dropping objects due to harsh contacts), thereby elevating the overall task success rate (TSR) from 80.0% to 86.0%. For instance, in Task 5, safely routing the water cup around the kettle raises TSR from 80.0% to 100.0%. Furthermore, the successful execution of Task 4 (CR reduced from 60.0% to 10.0%) confirms that the learned geometry head robustly identifies and avoids OOD obstacles (the novel wine bottle) in a zero-shot manner. Fig. 13 visually corroborates these findings, demonstrating GeoPhase's capability to smoothly coordinate phase-aware constraints and multi-proxy kinematics in complex physical environments.