VLSA-RL: CAN A SMALL RESIDUAL POLICY ENABLE SAFETY-AWARE VLA ADAPTATION?

Anonymous authors
Paper under double-blind review

ABSTRACT

How can a pretrained Vision-Language-Action (VLA) policy adapt to safety-critical manipulation tasks without updating its billions of parameters? Execution-time safety filters can prevent unsafe actions but do not teach the policy to avoid repeated interventions, while end-to-end reinforcement learning (RL) is costly. We introduce VLSA-RL, which freezes the VLA backbone and learns a 1.3M-parameter residual policy (0.037% of $\pi_{0.5}$) to adjust its actions before control-barrier-function (CBF) safety filtering. A safety-aware GRPO objective uses feedback from the CBF quadratic program, risk-dependent clipping, and task and safety rewards to optimize the residual policy. We further analyze how residual actions can compensate for dynamics mismatch and modulate CBF convergence behavior. On SafeLIBERO, VLSA-RL reaches a 68.0% average safe-success rate across two safety levels. Relative to the CBF-filtered AEGIS baseline, it improves safe success by 21.4 and 22.9 percentage points on Levels I and II, respectively, while reducing average episode length on both levels.

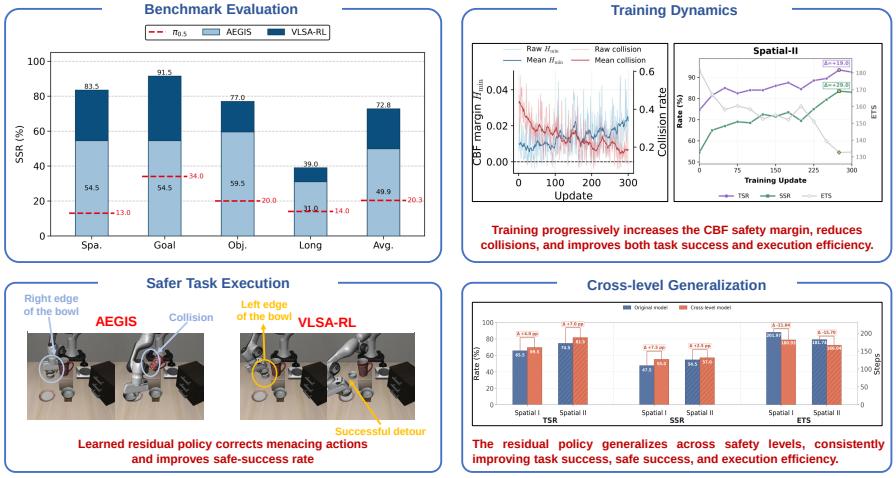

Figure 1: **Overall evaluation of VLSA-RL.** A safety-aware residual RL framework for frozen VLA models that improves upon the pretrained $\pi_{0.5}$ policy and AEGIS, progressively enlarges the CBF safety margin and reduces collisions during training, learns corrective actions for safer task execution, and transfers effectively across safety levels.

1 INTRODUCTION

Vision-Language-Action (VLA) models have advanced language-conditioned robotic manipulation by learning to map visual observations and instructions to actions across multiple tasks (Ma et al., 2026; Kawaharazuka et al., 2025; Sapkota et al., 2025; Zhang et al., 2025b). However, reliable deployment requires these policies to adapt when obstacles and safety constraints differ from those represented in their training data. Most VLA models are trained on offline demonstrations, which provide limited coverage of safety-critical interactions and unforeseen environmental changes

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.

(Shi et al., 2026; Zhang et al., 2025a). Collecting demonstrations for every new condition is costly, while fixed policies cannot improve from deployment-time safety feedback (Hu et al., 2025). This raises a central question: **How can a pretrained VLA policy adapt to safety constraints while preserving its pretrained representation and avoiding costly full-model retraining?**

Execution-time safety filters provide a direct mechanism for handling unsafe VLA actions. For example, AEGIS augments a VLA policy with a control-barrier-function (CBF) layer that projects proposed actions onto a set satisfying modeled safety constraints (Li et al., 2026b; Hu et al., 2025; Yu et al., 2026; Park et al., 2026; Ames et al., 2017). Such projection can prevent immediate constraint violations under the assumptions of the safety model. However, the nominal VLA policy remains unchanged, so similar unsafe actions may require repeated corrections throughout an episode. These interventions may reduce task efficiency and alter the state distribution encountered by the offline-trained policy. Safety filtering therefore addresses execution-time violations but does not, on its own, enable the policy to incorporate safety information into its future action proposals.

Reinforcement learning (RL) can adapt VLA policies from environment interaction, but end-to-end optimization of a large VLA model is computationally demanding. Recent post-training methods have used PPO or GRPO to improve manipulation performance from rollout feedback (Deng et al., 2025; Lu et al., 2026; Sun et al., 2026; Guo et al., 2025; Li et al., 2026a; Schulman et al., 2017; Shao et al., 2024). Updating both the vision-language representation and the action decoder increases the trainable parameter space and can complicate online optimization (Bagaria et al., 2026; Chen et al., 2025a). In addition, objectives dominated by task rewards do not explicitly distinguish actions with different levels of safety risk. Effective VLA adaptation thus requires a parameter-efficient policy update together with an optimization objective that uses safety information during learning.

In this paper, we propose VLSA-RL, a safety-aware residual RL framework that extends VLSA (Hu et al., 2025). VLSA-RL freezes the pretrained VLA backbone and trains a compact residual policy to modify its nominal actions before CBF quadratic-program (CBF-QP) filtering. This design limits the parameters updated by RL while allowing the policy to learn from safety-related feedback. To optimize the residual policy, we incorporate a CBF-QP constraint loss and safety-margin-dependent adaptive clipping into GRPO, together with task and safety rewards, improved advantage estimation, and curriculum episode selection. We also analyze the conditions under which residual actions compensate for dynamics mismatch and their relationship to the CBF convergence rate. Across two safety levels of SafeLIBERO, VLSA-RL improves average safe success over AEGIS by 21.4 and 22.9 percentage points, respectively. Our contributions are summarized as follows.

1) We introduce a residual adaptation framework that freezes the pretrained VLA backbone and trains a 1.3M-parameter policy, equivalent to 0.037% of the parameters in $\pi_{0.5}$.
2) We derive a condition under which residual actions compensate for dynamics mismatch in the CBF constraint. We further show that their contribution is algebraically equivalent to state-dependent modulation of the CBF rate coefficient.
3) We develop safety-aware GRPO with a CBF-QP constraint loss, safety-margin-dependent adaptive clipping, task and safety rewards, improved advantage estimation, and curriculum episode selection.
4) On SafeLIBERO, VLSA-RL reaches a 68.0% average safe-success rate across both safety levels. Compared with AEGIS, it improves safe success by 21.4 and 22.9 percentage points on Levels I and II, respectively, while reducing average episode length by 31.81 and 34.51 steps.

2 Related Work

Safety filtering for robot policies. Safety filters enforce explicit constraints without retraining the nominal policy, typically through CBF-based quadratic programs or predictive filtering (Ames et al., 2017; Wabersich & Zeilinger, 2021; Wabersich et al., 2023). Recent extensions to VLA models include VLSA (Hu et al., 2025), whose AEGIS implementation applies CBF-QP action correction, and Safe-Night VLA (Yu et al., 2026), which combines perception modules with CBF-based filtering. While effective at enforcing safety, reliance on post-hoc corrections may distort nominal behavior and induce distribution shifts during closed-loop execution. Our approach instead integrates CBF-derived information into policy optimization while retaining CBF-QP as a final safeguard.

Reinforcement learning for VLA adaptation. RL post-training methods such as VLA-RL (Lu et al., 2026), TGRPO (Chen et al., 2025c), and SimpleVLA-RL (Li et al., 2026a) improve VLA policies beyond offline demonstrations, while GRAPE (Zhang et al., 2024), SafeVLA (Zhang et al., 2025a), and SafeDojo (Tang et al., 2026) further consider safety-aware objectives. Residual methods such as ResiP (Ankile et al., 2025) and PLD (Xiao et al., 2026) enable parameter-efficient adaptation through additive corrections. However, existing work does not jointly combine lightweight residual adaptation, CBF-derived safety supervision, and risk-dependent policy optimization. VLSA-RL addresses this gap with a compact residual policy trained by safety-aware GRPO, using CBF-QP feedback for action learning and trajectory-level safety to adapt policy updates.

Extended related work and preliminaries are provided in Appendices A and B, respectively.

3 VLSA-RL Framework

VLSA-RL freezes the pretrained VLA backbone and optimizes a residual policy through grouped rollouts, safety-aware GRPO, and curriculum episode selection, as summarized in Figure 2.

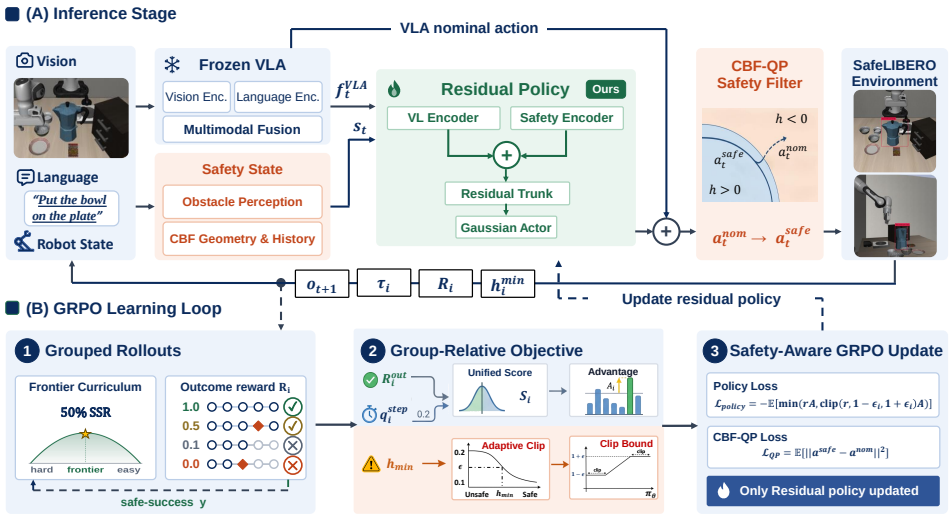

Figure 2: **Overview of VLSA-RL.** (A) During inference stage, a frozen VLA policy produces nominal actions, while the residual policy uses visual-language and safety features to generate action corrections before CBF-QP safety filtering. (B) The GRPO learning loop samples grouped rollouts through frontier curriculum learning, constructs group-relative advantages from task, efficiency, and safety feedback, and updates only the residual policy.

3.1 Overall Framework

As illustrated in Figure 2, we denote the frozen VLA policy by π_{VLA}, which consists of a vision-language encoder $\mathcal{E}_{\text{VLM}}$ and an action decoder $\mathcal{D}_{\text{act}}$. Given the visual observation o_t, language instruction l, and proprioceptive state p_t, the encoder and action decoder produce

$$f_t = \mathcal{E}_{\text{VLM}}(o_t, l), \qquad a_t^{\text{VLA}} = \mathcal{D}_{\text{act}}(f_t, p_t). \tag{1}$$

Both modules remain frozen during residual policy training.

To adapt the frozen VLA policy, we introduce a lightweight residual policy π_θ^{res}. In addition to the visual-language representation f_t, the residual policy receives a 79-D context vector z_t that summarizes the robot state, obstacle geometry, safety information, action history, and episode progress. The complete policy input is defined as $\xi_t = [f_t; z_t]$. A detailed breakdown of z_t is provided in Appendix D.

Conditioned on ξ_t, the residual policy generates an action correction, which is added to the base VLA action to form the nominal action:

$$a_t^{\text{res}} \sim \pi_\theta^{\text{res}}(\cdot \mid \xi_t), \qquad a_t^{\text{nom}} = a_t^{\text{VLA}} + a_t^{\text{res}}. \tag{2}$$

To enforce the safety constraint, the CBF-QP layer projects the nominal action onto the safe feasible set by solving

$$a_t^{\text{safe}} = \arg\min_a \ \|a - a_t^{\text{nom}}\|_2^2, \quad \text{s.t.} \quad \dot{h}(s_t, a) \geq -\alpha(h(s_t)) \,. \tag{3}$$

where h is the control barrier function defining the safe set and α is an extended class-$\mathcal{K}_\infty$ function governing its convergence behavior. The specific CBF used in our implementation is provided in Appendix C. The resulting action a_t^{safe} is sent to the robot as the final control command.

The action modification introduced by the CBF-QP layer is $\delta a_t^{\text{QP}} = a_t^{\text{safe}} - a_t^{\text{nom}} = a_t^{\text{safe}} - (a_t^{\text{VLA}} + a_t^{\text{res}})$. This correction is incorporated into z_{t+1} to provide the residual policy with explicit feedback on the direction and magnitude of the preceding safety intervention. In this framework, the frozen VLA policy provides the pretrained manipulation prior, the residual policy learns task- and safety-aware action corrections, and the CBF-QP layer provides an execution-time safety safeguard.

3.2 Analyzing the Role of the Residual Policy

Delays between high-level VLA action generation and low-level execution, along with unmodeled dynamics, can cause actual robot behavior to deviate from the nominal model (Black et al., 2026; Johannink et al., 2019). Beyond refining VLA actions and guiding nominal trajectories toward task-effective states with larger safety margins, the residual policy can compensate for such discrepancies. We analyze this effect from a CBF perspective. Since the CBF coefficient κ governs the allowable rate of approaching the safety boundary, prior methods have optimized or adapted κ to improve control performance (Xiao et al., 2023; Ma et al., 2022; Zeng et al., 2021; Chriat & Sun, 2023). We show that the residual policy induces an analogous implicit adaptation.

Theorem 1. *Let $\mathcal{C} = \{s \mid h(s) \geq 0\}$ be the safe set defined by a continuously differentiable CBF h. Consider the uncertain system $\dot{s} = \hat{f}(s) + \hat{g}(s)a + d(s,a)$, where $d(s,a)$ denotes the dynamics mismatch, and let the pre-projection action be $a = a^{\text{VLA}} + \pi_\theta^{\text{res}}(\xi)$. Under the exponential CBF condition $\alpha(h) = \kappa h$, define the original VLA safety margin as $M_{\text{VLA}}(s) = L_{\hat{f}}h(s) + L_{\hat{g}}h(s)a^{\text{VLA}} + \kappa h(s)$. The CBF condition under the composite policy satisfies*

$$\dot{h}(s) + \kappa h(s) = M_{\text{VLA}}(s) + L_{\hat{g}}h(s)\pi_\theta^{\text{res}}(\xi) + \nabla h(s)^\top d(s,a). \tag{4}$$

Therefore, if $L_{\hat{g}}h(s)\pi_\theta^{\text{res}}(\xi) \geq -M_{\text{VLA}}(s) - \nabla h(s)^\top d(s,a)$ for all admissible (s,ξ) with $s \in \mathcal{C}$, then $\dot{h}(s) + \kappa h(s) \geq 0$, and $\mathcal{C}$ is forward invariant under the standard CBF regularity conditions.

For $h(s) > 0$, define $\Delta\kappa_{\text{res}} = L_{\hat{g}}h(s)\pi_\theta^{\text{res}}(\xi)/h(s)$, $\Delta\kappa_d = \nabla h(s)^\top d(s,a)/h(s)$, and $\dot{h}_{\text{VLA}}(s) = L_{\hat{f}}h(s) + L_{\hat{g}}h(s)a^{\text{VLA}}$. Eq. (4) is equivalently written as

$$\dot{h}(s) + \kappa h(s) = \dot{h}_{\text{VLA}}(s) + \big(\kappa + \Delta\kappa_{\text{res}} + \Delta\kappa_d\big)h(s). \tag{5}$$

Thus, although the residual policy does not explicitly modify κ, its contribution is algebraically equivalent to a state-dependent modulation of the CBF rate coefficient.

Proof. Substituting $a = a^{\text{VLA}} + \pi_\theta^{\text{res}}(\xi)$ into the uncertain dynamics and differentiating $h(s)$ gives Eq. (4). The sufficient condition makes its right-hand side nonnegative, ensuring forward invariance by CBF results. For $h(s) > 0$, dividing the residual and mismatch terms by $h(s)$ gives Eq. (5). ■

Theorem 1 shows that the residual policy can provide additional safety margin to compensate for model mismatch. It therefore serves not only as an action-refinement module but also as a mechanism for proactive trajectory guidance and learnable safety compensation. These roles motivate incorporating CBF-derived feedback into residual policy optimization so that the nominal policy requires less external correction from the CBF-QP layer.

3.3 Safety-Aware Reward and Group-Relative Advantage

For each selected task episode, we sample a trajectory group $\mathcal{T} = \{\tau_i\}_{i=1}^G$ from the same initial condition using the same stochastic residual policy, where τ_i is the i-th rollout and $T_i = |\tau_i|$ its execution horizon. Early in training, some episodes may achieve moderate task success but few

safe successes. A binary safe-success reward assigns zero to both unsafe successes and failures, providing little relative learning signal when safe successes are scarce. We therefore use a tiered outcome reward that distinguishes task completion and safety:

$$R_i^{\text{out}} = \begin{cases} 1, & \text{safe success,} \\ 0.5, & \text{unsafe success,} \\ 0.1, & \text{safe failure,} \\ 0, & \text{unsafe failure.} \end{cases} \tag{6}$$

This ordering prioritizes safe task completion while retaining credit for unsafe task success and safe incomplete behavior, providing more informative group-relative learning signals.

To additionally encourage efficient execution, successful trajectories receive a step-efficiency reward. The efficiency reward and the resulting trajectory score are defined as

$$q_i^{\text{step}} = \mathbb{I}_i^{\text{succ}} \operatorname{clip}\left(\frac{T_{\text{ref}} - T_i}{T_{\text{ref}}}, -1, 1\right), \quad S_i = R_i^{\text{out}} + \lambda_{\text{step}} q_i^{\text{step}}. \tag{7}$$

where T_{ref} denotes the reference horizon and $\lambda_{\text{step}} = 0.2$.

Given the group mean $\bar{S} = \frac{1}{G}\sum_{i=1}^{G} S_i$, we estimate the trajectory-level group-relative advantage as

$$\hat{A}_i = \frac{S_i - \bar{S}}{c_A}. \tag{8}$$

Under this four-level outcome reward, normalization by the within-group standard deviation would remove absolute differences in reward scale across groups, weakening the intended hierarchy among different task and safety outcomes. We therefore use a fixed scaling factor $c_A = 0.5$ so that larger reward differences produce correspondingly stronger learning signals. The same advantage $\hat{A}_i$ is assigned to all residual actions in τ_i. Groups satisfying $\max_i S_i - \min_i S_i \leq 10^{-6}$ are discarded because they contain no group-relative learning signal.

3.4 Safety-Guided Group Relative Policy Optimization

Safety-margin-dependent adaptive clipping. Standard PPO and GRPO apply the same clipping range to all samples regardless of their safety conditions. To make policy updates responsive to trajectory-level risk, we summarize the safety of τ_i by its minimum CBF margin, $H_{\min}(\tau_i) = \min_{1 \leq t \leq |\tau_i|} h(s_{i,t})$, and define its clipping bound as

$$\epsilon_i = \epsilon_{\min} + (\epsilon_{\max} - \epsilon_{\min})\sigma\left(\frac{h_{\text{clip}} - H_{\min}(\tau_i)}{\tau_h}\right). \tag{9}$$

A smaller $H_{\min}(\tau_i)$ indicates greater safety risk and produces a wider clipping range, allowing larger policy updates for high-risk trajectories. Conversely, safer trajectories receive tighter updates to avoid unnecessary changes to already safe behavior.

For residual action $a_{i,t}^{\text{res}}$, the importance ratio is $r_{i,t}(\theta) = \pi_\theta^{\text{res}}(a_{i,t}^{\text{res}} \mid \xi_{i,t}) / \pi_{\theta_{\text{old}}}^{\text{res}}(a_{i,t}^{\text{res}} \mid \xi_{i,t})$. The resulting safety-adaptive policy loss is

$$\mathcal{L}_{\text{policy}}(\theta) = -\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|\tau_i|}\sum_{t=1}^{|\tau_i|}\min\Big\{r_{i,t}(\theta)\hat{A}_i, \operatorname{clip}\big(r_{i,t}(\theta), 1-\epsilon_i, 1+\epsilon_i\big)\hat{A}_i\Big\}. \tag{10}$$

Differentiable CBF-QP constraint loss. Although the CBF-QP layer provides execution-time safety, post-hoc corrections alone do not teach the residual policy to generate safer nominal actions, limiting adaptation effectiveness. We therefore introduce a differentiable CBF-QP constraint loss that directly penalizes the correction required by the safety layer.

Let $\bar{a}_{i,t}^{\text{nom}}(\theta) = a_{i,t}^{\text{VLA}} + \mu_\theta^{\text{res}}(\xi_{i,t})$ denote the nominal action induced by the mean residual action. Its CBF-QP projection is

$$\bar{a}_{i,t}^{\text{QP}}(\theta) = \text{CBF-QP}\big(s_{i,t}, \bar{a}_{i,t}^{\text{nom}}(\theta)\big). \tag{11}$$

The analytic CBF-QP projection is differentiable almost everywhere, allowing gradients to propagate through both the nominal action and its safety projection (Cheng et al., 2019). Minimizing their

discrepancy therefore encourages the residual policy to generate nominal actions that require less CBF-QP correction (See Appendix C). Let P_{QP} select the action components constrained by the CBF-QP layer. The constraint loss is defined as

$$\mathcal{L}_{\mathrm{QP}}(\theta) = \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|\tau_i|}\sum_{t=1}^{|\tau_i|}\frac{1}{d_{\mathrm{QP}}}\left\|P_{\mathrm{QP}}\left(\bar{a}_{i,t}^{\mathrm{QP}}(\theta) - \bar{a}_{i,t}^{\mathrm{nom}}(\theta)\right)\right\|_2^2, \tag{12}$$

where d_{QP} denotes the number of CBF-QP-constrained action dimensions, excluding the unconstrained gripper action. The factor $1/|\tau_i|$ normalizes each trajectory by its horizon, ensuring equal contribution to the constraint loss. We omit the KL-divergence regularization term from the standard GRPO objective. The complete training objective is

$$\mathcal{L}(\theta) = \mathcal{L}_{\mathrm{policy}}(\theta) + \lambda_{\mathrm{QP}}\mathcal{L}_{\mathrm{QP}}(\theta), \tag{13}$$

where $\lambda_{\mathrm{QP}} = 0.05$ in our implementation.

3.5 Curriculum Episode Selection

Inspired by the bandit-based EXP3 algorithm (Auer et al., 2002), we prioritize informative episodes. Tasks are sampled uniformly, while episode i of task k is sampled with

$$p_i^k = (1-\gamma)\frac{w_i^k}{\sum_{j=1}^{M} w_j^k} + \frac{\gamma}{M}, \tag{14}$$

where w_i^k is initialized to one, M is the number of candidate episodes, and γ controls uniform exploration. After G rollouts, the safe-success rate, informativeness score, and weight update are

$$y_i^k = \frac{1}{G}\sum_{g=1}^{G}\mathbb{I}[\tau_{i,g} \text{ is a safe success}], \qquad c_i^k = 4y_i^k(1-y_i^k), \tag{15}$$

$$w_{i,n+1}^k = w_{i,n}^k \exp\left(\eta c_i^k\right). \tag{16}$$

where η is the update rate. Since c_i^k peaks at $y_i^k = 0.5$, the strategy favors intermediate-difficulty episodes with mixed outcomes. We periodically reset the weights to prevent excessive concentration.

4 Experiments

4.1 Experimental Setup

Benchmark. We evaluate VLSA-RL on SafeLIBERO (Hu et al., 2025), a safety-aware extension of LIBERO (Liu et al., 2023) implemented in Robosuite (Zhu et al., 2020) with a 7-DoF Franka Emika Panda arm. SafeLIBERO contains four task suites (Spatial, Goal, Object, and Long), each with four tasks evaluated at two safety levels. Level I places obstacles near the target object, whereas Level II places them farther from the target along the robot's motion path.

Baselines. We compare VLSA-RL with five VLA policies: $\pi_{0.5}$ (Intelligence et al., 2025), OpenVLA-OFT (Kim et al., 2025), OptimusVLA (Li et al., 2026c), UniVLA (Bu et al., 2025), and GR00T N1.7 (Bjorck et al., 2025). We also include AEGIS (Hu et al., 2025), which applies a CBF-based safety layer at execution time. We replace its original point-cloud extraction and filtering module with a MobileSAM pipeline (Zhang et al., 2023) for more accurate and robust obstacle extraction. We further include SafeDojo (Tang et al., 2026), a world-model-based safe RL method.

Evaluation Metrics. We report **Task Success Rate (TSR)**, the percentage of episodes that complete the task within the time limit, and **Safe Success Rate (SSR)**, the percentage that additionally avoid collisions. Collisions do not terminate an episode. **Execution Time Steps (ETS)** is the average number of steps across all episodes, including those that reach the time limit.

Training Details. The pretrained $\pi_{0.5}$-LIBERO policy is frozen, and only a six-layer MLP residual policy (1.36M parameters), mapping a 2,127-dimensional input to a seven-dimensional residual action, is optimized. We train a separate policy for each suite and safety level. Each configuration

Table 1: **Results on SafeLIBERO Level I.** Δ denotes VLSA-RL minus AEGIS.

Method	TSR (%) ↑					SSR (%) ↑					ETS ↓				
	Spa.	Goal	Obj.	Long	Avg.	Spa.	Goal	Obj.	Long	Avg.	Spa.	Goal	Obj.	Long	Avg.
$\pi_{0.5}$	65.0	54.0	40.0	59.5	54.6	14.5	3.0	10.5	13.5	10.4	188.90	224.90	253.53	408.25	268.90
OpenVLA-OFT	43.5	24.0	0.0	1.0	17.1	10.0	2.0	0.0	0.0	3.0	221.65	251.23	300.00	547.88	330.19
GR00T-N1.7	49.5	12.5	3.0	5.5	17.6	13.0	0.0	0.0	1.0	3.5	216.25	282.58	297.17	537.59	333.40
UniVLA	33.0	14.0	0.0	4.5	12.9	12.5	1.5	0.0	0.5	3.6	239.28	274.46	300.00	541.56	338.83
OptimusVLA	67.5	51.5	40.0	**63.0**	55.5	12.0	1.0	15.5	14.0	10.6	187.06	236.17	249.71	**397.96**	267.73
SafeDojo	59.5	80.0	73.5	45.0	64.5	51.5	78.5	**49.0**	**34.0**	53.3	209.25	171.12	227.13	470.15	269.41
AEGIS	65.5	**88.5**	58.5	56.5	67.3	47.5	79.5	19.5	20.5	41.8	201.97	162.58	240.79	422.57	256.98
VLSA-RL	**93.0**	87.5	**78.0**	61.5	**80.0**	**74.0**	**86.5**	**63.0**	**29.0**	**63.1**	**143.15**	**149.57**	**194.76**	**413.21**	**225.17**
Δ	+27.5	−1.0	+19.5	+5.0	+12.8	+26.5	+7.0	+43.5	+8.5	+21.4	−58.82	−13.01	−46.03	−9.36	−31.81

Table 2: **Results on SafeLIBERO Level II.** Δ denotes VLSA-RL minus AEGIS.

Method	TSR (%) ↑					SSR (%) ↑					ETS ↓				
	Spa.	Goal	Obj.	Long	Avg.	Spa.	Goal	Obj.	Long	Avg.	Spa.	Goal	Obj.	Long	Avg.
$\pi_{0.5}$	53.5	66.0	75.0	**49.0**	60.9	13.0	34.0	20.0	14.0	20.3	210.10	174.58	193.53	**432.78**	252.75
OpenVLA-OFT	29.5	24.5	57.5	29.5	35.3	2.5	4.5	18.0	8.5	8.4	248.63	246.78	219.83	475.55	297.70
GR00T-N1.7	45.0	26.5	46.0	23.0	35.1	10.0	2.5	15.0	4.5	8.0	225.31	252.84	233.60	497.27	302.26
UniVLA	21.0	23.5	36.5	7.0	22.0	7.5	7.5	15.0	1.5	7.9	263.06	256.31	255.98	537.00	328.09
OptimusVLA	54.5	72.5	78.5	47.0	63.1	12.0	21.5	19.0	10.0	15.6	210.73	167.57	**187.98**	441.99	252.07
SafeDojo	74.5	64.0	60.0	29.5	57.0	60.5	56.0	55.5	26.5	49.6	187.50	190.23	225.99	504.29	277.00
AEGIS	74.5	82.0	**79.0**	41.0	69.1	54.5	54.5	59.5	31.0	49.9	181.74	158.91	197.84	470.44	252.23
VLSA-RL	**93.5**	**94.5**	**89.5**	**53.5**	**82.8**	**83.5**	**91.5**	**77.0**	**39.0**	**72.8**	**132.68**	**122.09**	**166.78**	**449.35**	**217.73**
Δ	+19.0	+12.5	+10.5	+12.5	+13.6	+29.0	+37.0	+17.5	+8.0	+22.9	−49.06	−36.82	−31.06	−21.09	−34.51

contains four tasks with 50 training episodes per task. At each of 300 optimization updates, the curriculum samples four episodes per task, and eight parallel rollouts are collected for each episode. We use a learning rate of 3×10^{-5}, one optimization epoch per rollout batch, adaptive clipping bounds in $[0.1, 0.2]$, and a maximum gradient norm of 0.1. Training on a single NVIDIA RTX 5090 GPU and an Intel Xeon Gold CPU takes approximately 48 hours for the Spatial, Goal, and Object suites. The Long suite requires more time because its tasks contain multiple stages.

4.2 Main Results

Overall performance. Tables 1 and 2 report the checkpoint with the highest SSR for each suite and safety level, with detailed task-level results in Appendix E. Relative to AEGIS, VLSA-RL improves average TSR from 67.3% to 80.0% and SSR from 41.8% to 63.1% on Level I, while reducing ETS from 256.98 to 225.17. On Level II, average TSR increases from 69.1% to 82.8%, SSR from 49.9% to 72.8%, and ETS decreases from 252.23 to 217.73. Among baselines other than AEGIS, SafeDojo attains the highest average SSR at both levels. VLSA-RL exceeds SafeDojo by 9.9 and 23.2 percentage points on Levels I and II, respectively, with higher average TSR and lower ETS. AEGIS marginally exceeds SafeDojo in average SSR on Level II (49.9% versus 49.6%).

Performance across suites. Relative to AEGIS, SSR increases and ETS decreases across all four suites at both safety levels. The largest SSR gains occur on Object I ($+43.5$ percentage points) and Goal II ($+37.0$ percentage points). TSR also increases in seven of the eight settings. The exception is Goal I, where TSR decreases by 1.0 point while SSR increases by 7.0 points. These comparisons show improved collision-free task completion across the evaluated suites, together with higher task success in all but one setting and shorter average episodes throughout.

Evaluation curves. Policies are evaluated every 25 optimization updates. Figures 3 and 4 track TSR, SSR, and ETS across training for all four suites at Levels I and II, respectively. The curves show how performance changes during adaptation rather than only at the selected checkpoints reported in Tables 1 and 2.

Training dynamics. Figure 5 examines optimization on SafeLIBERO Spatial II in greater detail. Evaluation TSR and SSR generally improve over training, while ETS declines. The clipping bound varies with the observed CBF margin, as prescribed by the adaptive update rule. Higher CBF margins and lower collision rates accompany the improvement in safe task completion.

Qualitative behavior. Figure 6 compares representative trajectories on two bowl-placement tasks. VLSA-RL grasps the bowl at its left rim and moves it past the obstacle without a visible collision. The trajectories illustrate the collision-free executions reflected in the aggregate SSR results.

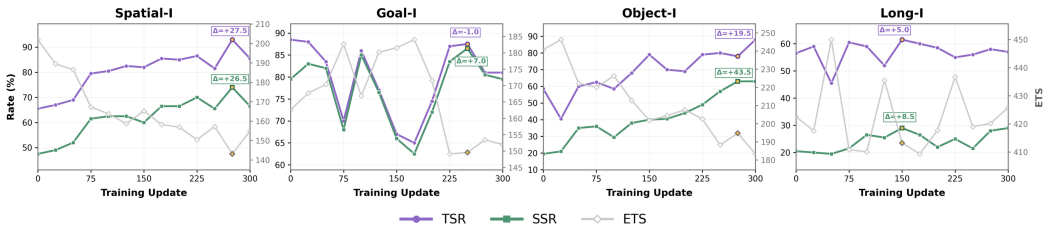

Figure 3: **Training curves on the SafeLIBERO Level I benchmark.** TSR, SSR, and ETS are reported over evaluation checkpoints for the Spatial, Goal, Object, and Long suites.

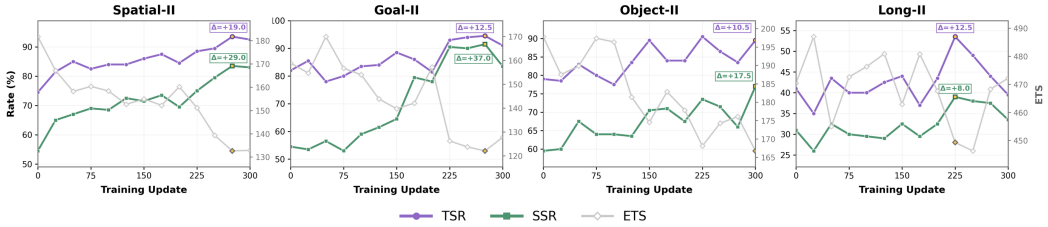

Figure 4: **Training curves on the SafeLIBERO Level II benchmark.** TSR, SSR, and ETS are reported over evaluation checkpoints for the Spatial, Goal, Object, and Long suites.

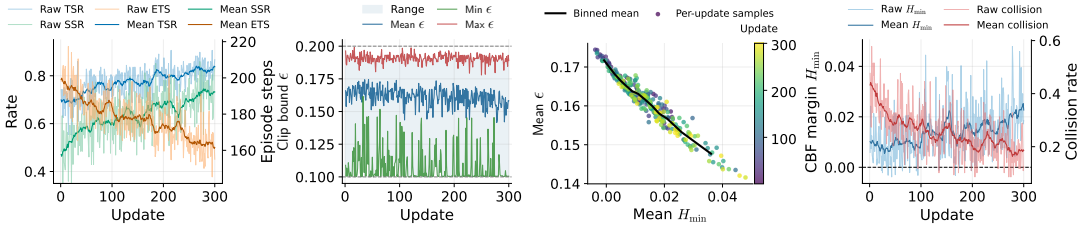

Figure 5: **Training dynamics on SafeLIBERO Spatial II.** (a) Evaluation TSR, SSR, and ETS. (b) Adaptive clipping bounds during optimization. (c) Empirical relationship between the mean CBF margin $H_{\min}$ and clipping bound ϵ. (d) CBF margin and collision rate over training.

4.3 Ablation Study

We evaluate the main design choices of VLSA-RL on SafeLIBERO Spatial I. Table 3 shows that the full method achieves the highest TSR and SSR and the lowest ETS, with 93.0% task success, 74.0% safe success, and an average episode length of 143.1 steps.

Adaptive clipping. Replacing the CBF-guided clip bound with any fixed $\epsilon \in \{0.10, 0.15, 0.20\}$ lowers both TSR and SSR in this setting. The best fixed variant, $\epsilon = 0.10$, reaches 90.0% TSR and 72.5% SSR, compared with 93.0% and 74.0% for the full method. This supports adapting the policy-update range to the trajectory-level CBF margin.

Table 3: **Ablation on SafeLIBERO Spatial I.**

Variant	TSR (%) ↑	SSR (%) ↑	ETS ↓
Full method	**93.0**	**74.0**	**143.1**
Fixed $\epsilon = 0.10$	90.0	72.5	145.6
Fixed $\epsilon = 0.15$	89.0	72.0	151.9
Fixed $\epsilon = 0.20$	85.0	69.5	158.7
Std. normalization	85.0	70.0	153.8
No graded reward	86.5	73.0	155.9
No curriculum	89.0	70.0	146.0
No CBF-QP loss	87.5	69.5	150.5
KL ($\beta = 0.1$)	82.0	60.0	162.4

Reward and advantage design. Removing the graded reward reduces TSR to 86.5% and increases ETS to 155.9 steps. Replacing fixed-scale advantage normalization with within-group standard-deviation normalization yields 85.0% TSR and 70.0% SSR. Both choices contribute to performance on this ablation setting.

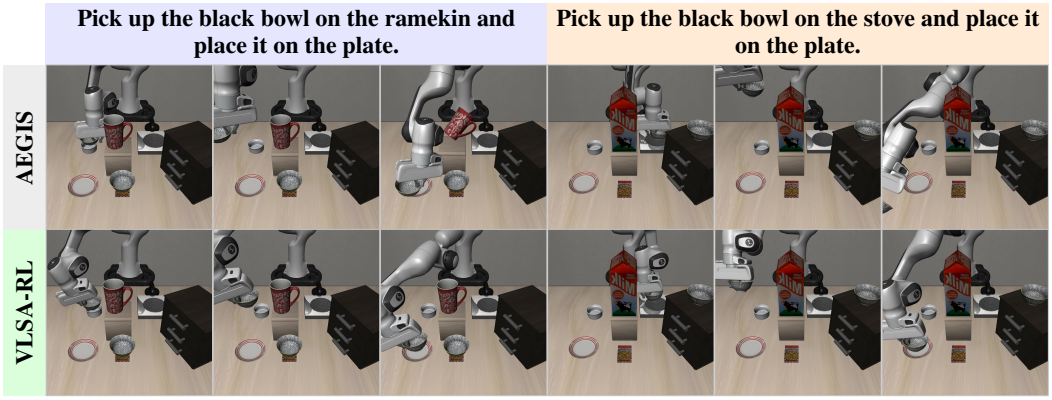

Figure 6: **Qualitative comparison of AEGIS and VLSA-RL.** Each row shows three consecutive frames for each of two bowl-placement tasks.

Curriculum and CBF-QP loss. Without curriculum episode sampling, TSR and SSR decline to 89.0% and 70.0%. Removing the CBF-QP constraint loss reduces them to 87.5% and 69.5%. These comparisons indicate that both components contribute to task completion and safe success.

KL regularization. Adding a fixed KL penalty with $\beta = 0.1$ yields 82.0% TSR, 60.0% SSR, and 162.4 ETS. This result suggests that the added penalty restricts adaptation excessively under the evaluated configuration.

4.4 Generalization Analysis

We evaluate cross-level transfer in SafeLIBERO Spatial by applying policies trained on Levels I and II to the other level. As shown in Figure 7, I-to-II transfer improves TSR from 74.5% to 81.5% and SSR from 54.5% to 57.0% relative to AEGIS, while II-to-I transfer improves TSR from 65.5% to 69.5% and SSR from 47.5% to 55.0%. These results provide evidence of cross-level generalization between the two evaluated safety levels within the same task suite.

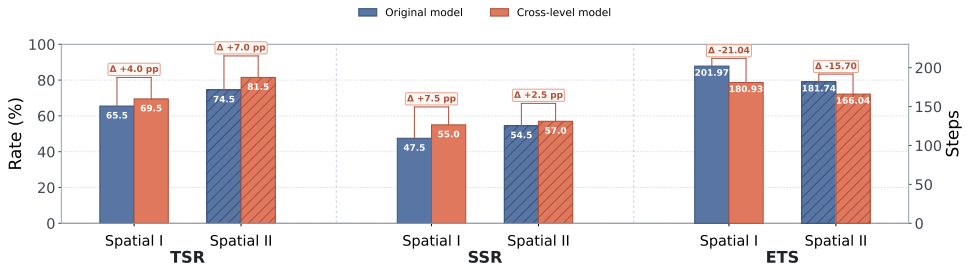

Figure 7: **Cross-level transfer on SafeLIBERO Spatial.** Original model is compared with the residual policy transferred from the other safety level using TSR, SSR, and ETS.

5 Conclusion

We presented VLSA-RL for safety-aware adaptation of pretrained VLA policies. It freezes the VLA backbone and optimizes a compact residual policy via CBF-informed GRPO, with CBF-QP action correction at execution. We establish a sufficient condition for residual actions to compensate for dynamics mismatch and characterize their effect as state-dependent modulation of the CBF convergence rate. On SafeLIBERO, VLSA-RL achieved a 68.0% average safe-success rate across two safety levels, improving over CBF-filtered AEGIS by 21.4 and 22.9 pp on Levels I and II, respectively, while shortening episodes. These results support residual RL as a parameter-efficient approach to safety-aware VLA adaptation under the evaluated simulation conditions. Limitations and future work are discussed in Appendix F.

AI USE STATEMENT

Generative AI tools were used to assist with language polishing and to improve the clarity and readability of the manuscript. They were also used to assist with some aspects of code development, including code completion, debugging, and formatting. The authors reviewed all AI-assisted content and take full responsibility for the final content of the paper and the correctness of the code.

REPRODUCIBILITY STATEMENT

To support reproducibility, Section 4.1 specifies the benchmark, baselines, evaluation metrics, training protocol, and computational setup. Appendix C provides the complete CBF-QP formulation and theoretical derivations, while Appendix D describes the residual-policy architecture, action parameterization, hyperparameters, and training algorithm. Appendix E further reports detailed task-level results across all SafeLIBERO suites and safety levels.

REFERENCES

Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and J Zico Kolter. Differentiable convex optimization layers. *Advances in neural information processing systems*, 32, 2019.

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, 2017. doi: 10.1109/TAC.2016.2638961. URL https://arxiv.org/abs/1609.06408.

Brandon Amos and J. Zico Kolter. OptNet: Differentiable optimization as a layer in neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 136–145, 2017. URL https://proceedings.mlr.press/v70/amos17a.html.

Lars Ankile, Anthony Simeonov, Idan Shenfeld, Marcel Torne, and Pulkit Agrawal. From imitation to refinement—residual RL for precise assembly. In *IEEE International Conference on Robotics and Automation*, 2025. URL https://arxiv.org/abs/2407.16677.

Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002. doi: 10.1137/S0097539701398375. URL https://doi.org/10.1137/S0097539701398375.

Vaidehi Bagaria, Nikshep Grampurohit, and Pulkit Verma. Learn where outcomes diverge: Efficient vla rl via probabilistic chunk masking. *arXiv preprint arXiv:2605.16154*, 2026.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Kevin Black, Manuel Galliker, and Sergey Levine. Real-time execution of action chunking flow policies. *Advances in Neural Information Processing Systems*, 38:33383–33407, 2026.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2505.06111*, 2025.

Kang Chen, Zhihao Liu, Tonghe Zhang, Zhen Guo, Si Xu, Hao Lin, Hongzhi Zang, Xiang Li, Quanlu Zhang, Zhaofei Yu, et al. π_{RL}: Online rl fine-tuning for flow-based vision-language-action models. *arXiv preprint arXiv:2510.25889*, 2025a.

Yuhui Chen, Shuai Tian, Shugao Liu, Yingting Zhou, Haoran Li, and Dongbin Zhao. ConRFT: A reinforced fine-tuning method for VLA models via consistency policy. In *Proceedings of Robotics: Science and Systems*, 2025b. doi: 10.15607/RSS.2025.XXI.019. URL https://www.roboticsproceedings.org/rss21/p019.html.

Zengjue Chen, Runliang Niu, He Kong, Qi Wang, Qianli Xing, and Zipei Fan. Tgrpo :fine-tuning vision-language-action model via trajectory-wise group relative policy optimization, 2025c. URL https://arxiv.org/abs/2506.08440.

Richard Cheng, Gábor Orosz, Richard M. Murray, and Joel W. Burdick. End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3387–3395, 2019. doi: 10.1609/aaai.v33i01.33013387. URL https://ojs.aaai.org/index.php/AAAI/article/view/4213.

Alaa Eddine Chriat and Chuangchuang Sun. On the optimality, stability, and feasibility of control barrier functions: An adaptive learning-based approach. *IEEE Robotics and Automation Letters*, 8(11):7865–7872, 2023.

Haoyuan Deng, Zhenyu Wu, Haichao Liu, Wenkai Guo, Yuquan Xue, Ziyu Shan, Chuanrui Zhang, Bofang Jia, Yuan Ling, Guanxing Lu, et al. A survey on reinforcement learning of vision-language-action models for robotic manipulation. 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, 2025. doi: 10.1109/TCST.2024.3467809. URL https://arxiv.org/abs/2308.12073.

Sicong Gao, Maurice Pagnucco, Tomasz Bednarz, and Yang Song. Nvidia isaac sim: Enabling scalable, gpu-accelerated simulation for robotics. *arXiv preprint arXiv:2606.03551*, 2026.

Yanjiang Guo, Jianke Zhang, Xiaoyu Chen, Xiang Ji, Yen-Jen Wang, Yucheng Hu, and Jianyu Chen. Improving vision-language-action model with online reinforcement learning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 15665–15672. IEEE, 2025.

Kai-Chieh Hsu, Haimin Hu, and Jaime F Fisac. The safety filter: A unified view of safety-critical control in autonomous systems. *Annual Review of Control, Robotics, and Autonomous Systems*, 7, 2023.

Changhao Hu, Zeyi Liu, Songqiao Hu, Shuang Liu, Zihan Meng, and Xiao He. Predictive relative-velocity steering for safe robotic manipulator teleoperation in dynamic environments. *arXiv preprint arXiv:2608.13284*, 2026.

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

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Tobias Johannink, Shikhar Bahl, Ashvin Nair, Jianlan Luo, Avinash Kumar, Matthias Loskyll, Juan Aparicio Ojea, Eugen Solowjow, and Sergey Levine. Residual reinforcement learning for robot control. In *2019 international conference on robotics and automation (ICRA)*, pp. 6023–6029. IEEE, 2019.

Tobias Jülg, Wolfram Burgard, and Florian Walter. Refined policy distillation: From vla generalists to rl experts. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 11677–11684. IEEE, 2025.

Kento Kawaharazuka, Jihoon Oh, Jun Yamada, Ingmar Posner, and Yuke Zhu. Vision-language-action models for robotics: A review towards real-world applications. *IEEE Access*, 13:162467–162504, 2025.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

Haozhan Li, Yuxin Zuo, Jiale Yu, Yuhao Zhang, Yang Zhaohui, Kaiyan Zhang, Xuekai Zhu, Yuchen Zhang, Tianxing Chen, Ganqu Cui, et al. Simplevla-rl: Scaling vla training via reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 125846–125869, 2026a.

Xiao Li, Xiang Zheng, Yifeng Gao, Xinyu Xia, Yixu Wang, Xin Wang, Ye Sun, Yunhan Zhao, Ming Wen, Jiayu Li, et al. Safety in embodied ai: A survey of risks, attacks, and defenses. *arXiv preprint arXiv:2605.02900*, 2026b.

Zaijing Li, Bing Hu, Rui Shao, Gongwei Chen, Dongmei Jiang, Pengwei Xie, Jianye Hao, and Liqiang Nie. Global prior meets local consistency: Dual-memory augmented vision-language-action model for efficient robotic manipulation. *arXiv preprint arXiv:2602.20200*, 2026c.

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.

Guanxing Lu, Wenkai Guo, Chubin Zhang, Yuheng Zhou, Haonan Jiang, Zifeng Gao, Yansong Tang, and Ziwei Wang. VLA-RL: Towards masterful and general robotic manipulation with scalable reinforcement learning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2026. URL https://arxiv.org/abs/2505.18719.

Hengbo Ma, Bike Zhang, Masayoshi Tomizuka, and Koushil Sreenath. Learning differentiable safety-critical control using control barrier functions for generalization to novel environments. In *2022 European Control Conference (ECC)*, pp. 1301–1308. IEEE, 2022. URL https://arxiv.org/abs/2201.01347.

Yueen Ma, Zixing Song, Yuzheng Zhuang, Jianye Hao, and Irwin King. A survey on vision–language–action models for embodied ai. *IEEE Transactions on Neural Networks and Learning Systems*, 37(7):3031–3051, 2026.

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. URL https://arxiv.org/abs/2606.09749.

Federico Pizarro Bejarano, Lukas Brunke, and Angela P. Schoellig. Safety filtering while training: Improving the performance and sample efficiency of reinforcement learning agents. *IEEE Robotics and Automation Letters*, 10(1):788–795, 2025. doi: 10.1109/LRA.2024.3512374. URL https://doi.org/10.1109/LRA.2024.3512374.

Ranjan Sapkota, Yang Cao, Konstantinos I Roumeliotis, and Manoj Karkee. Vision-language-action (vla) models: Concepts, progress, applications and challenges. *arXiv preprint arXiv:2505.04769*, 2025.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

Liangzhi Shi, Shuaihang Chen, Feng Gao, Yinuo Chen, Kang Chen, Tonghe Zhang, Hongzhi Zang, Jiakai Zhou, Weinan Zhang, Chao Yu, et al. Beyond imitation: Reinforcement learning-based sim-real co-training for vla models. *arXiv preprint arXiv:2602.12628*, 2026.

Kasra Sinaei, Hung-Chieh Wu, and Donald Ebeigbe. Safe vision language action models via barrier enhanced flow matching. *arXiv preprint arXiv:2607.29569*, 2026.

Oswin So, Zachary Serlin, Makai Mann, Jake Gonzales, Kwesi Rutledge, Nicholas Roy, and Chuchu Fan. How to train your neural control barrier function: Learning safety filters for complex input-constrained systems. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11532–11539. IEEE, 2024.

Haoran Sun, Yongjian Guo, Zhong Guan, Shuai Di, Xiaodong Bai, Jing Long, Tianyun Zhao, Mingxi Luo, Hongke Zhao, Likang Wu, et al. Rl-vla ^{3}: A flexible and asynchronous reinforcement learning framework for vla training. *arXiv preprint arXiv:2602.05765*, 2026.

Shuhan Tan, Kairan Dou, Yue Zhao, and Philipp Krähenbühl. Interactive post-training for vision-language-action models. *arXiv preprint arXiv:2505.17016*, 2025. URL https://arxiv.org/abs/2505.17016.

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.

Ben Tearle, Kim P Wabersich, Andrea Carron, and Melanie N Zeilinger. A predictive safety filter for learning-based racing control. *IEEE Robotics and Automation Letters*, 6(4):7635–7642, 2021.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033. IEEE, 2012.

Maximilian Tölle, Theo Gruner, Daniel Palenicek, Tim Schneider, Jonas Günster, Joe Watson, Davide Tateo, Puze Liu, and Jan Peters. Towards safe robot foundation models using inductive biases. *arXiv preprint arXiv:2505.10219*, 2025. URL https://arxiv.org/abs/2505.10219.

Kim P Wabersich, Andrew J Taylor, Jason J Choi, Koushil Sreenath, Claire J Tomlin, Aaron D Ames, and Melanie N Zeilinger. Data-driven safety filters: Hamilton-jacobi reachability, control barrier functions, and predictive methods for uncertain systems. *IEEE Control Systems Magazine*, 43(5):137–177, 2023.

Kim Peter Wabersich and Melanie N. Zeilinger. A predictive safety filter for learning-based control of constrained nonlinear dynamical systems. *Automatica*, 129:109597, 2021. doi: 10.1016/j.automatica.2021.109597. URL https://arxiv.org/abs/1812.05506.

Nolan C. Wagener, Byron Boots, and Ching-An Cheng. Safe reinforcement learning using advantage-based intervention. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 10630–10640. PMLR, 2021. URL https://proceedings.mlr.press/v139/wagener21a.html.

Wei Xiao, Tsun-Hsuan Wang, Ramin Hasani, Makram Chahine, Alexander Amini, Xiao Li, and Daniela Rus. Barriernet: Differentiable control barrier functions for learning of safe robot control. *IEEE Transactions on Robotics*, 39(3):2289–2307, 2023.

Wenli Xiao, Haotian Lin, Andy Peng, Haoru Xue, Tairan He, Yuqi Xie, Fengyuan Hu, Jimmy Wu, Zhengyi Luo, Linxi Fan, Guanya Shi, and Yuke Zhu. Self-improving vision-language-action models with data generation via residual RL. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=eUGoqrZ6Ea.

Lizhi Yang, Blake Werner, Massimiliano de Sa, and Aaron D Ames. Cbf-rl: Safety filtering reinforcement learning in training with control barrier functions. *arXiv preprint arXiv:2510.14959*, 2025.

Dian Yu, Qingchuan Zhou, Bingkun Huang, Majid Khadiv, and Zewen Yang. Safe-Night VLA: Seeing the unseen via thermal-perceptive vision-language-action models for safety-critical manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2026. URL https://arxiv.org/abs/2603.05754.

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. In *Advances in Neural Information Processing Systems*, volume 38, pp. 153335–153373, 2025a. doi: 10.52202/085713-5128. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/e185c7be603426028c32ae1003a59d78-Abstract-Conference.html.

Chaoning Zhang, Dongshen Han, Yu Qiao, Jung Uk Kim, Sung-Ho Bae, Seungkyu Lee, and Choong Seon Hong. Faster segment anything: Towards lightweight sam for mobile applications. *arXiv preprint arXiv:2306.14289*, 2023.

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*, 2025b.

Zijian Zhang, Kaiyuan Zheng, Zhaorun Chen, Joel Jang, Yi Li, Siwei Han, Chaoqi Wang, Mingyu Ding, Dieter Fox, and Huaxiu Yao. Grape: Generalizing robot policy via preference alignment. *arXiv preprint arXiv:2411.19309*, 2024.

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.

APPENDIX

A EXTENDED RELATED WORK

A.1 SAFETY FILTERS AND POLICY LEARNING

Safety filtering provides a modular approach to enforcing explicit constraints in robot control (Hsu et al., 2023; Wabersich et al., 2023). A filter modifies candidate actions before execution when they conflict with modeled safety requirements. Common designs use CBF-based quadratic programs (Ames et al., 2017; So et al., 2024; Sinaei et al., 2026) or predictive safety filtering (Wabersich & Zeilinger, 2021; Tearle et al., 2021; Hu et al., 2026). Because a filter can be attached to an existing controller, the nominal policy need not be retrained. However, safety interventions do not directly improve the policy's subsequent action proposals.

Recent work has extended safety filtering to generalist robot policies and VLA models. Tölle et al. (2025) add an ATACOM-based safety layer to a robot foundation model for collision avoidance and joint-space constraints. VLSA (Hu et al., 2025) introduces a safety layer for VLA models. Its AEGIS implementation combines vision-language obstacle assessment with CBF-QP action correction. Safe-Night VLA (Yu et al., 2026) integrates thermal perception and a CBF-based filter to address workspace constraints. In these methods, safety is primarily enforced at execution time. Repeated interventions may therefore be required when the nominal policy continues to propose actions that violate the modeled constraints.

Several studies use safety-filter interventions as learning signals. RL-CBF uses CBF corrections to guide exploration (Cheng et al., 2019), while SAILR treats interventions as penalized learning signals (Wagener et al., 2021). Pizarro Bejarano et al. (2025) penalize the magnitude of filter corrections during training, and CBF-RL incorporates CBF filtering into policy optimization (Yang et al., 2025). These approaches link safety filtering and policy learning. VLSA-RL applies this idea to a frozen VLA model by using CBF-derived feedback to train a residual policy while retaining CBF-QP filtering during execution.

A.2 RL FOR VLA MODELS

RL post-training allows VLA policies to improve through environment interaction beyond what is available from offline demonstrations. Existing approaches differ in their reward design and the components they update. iRe-VLA (Guo et al., 2025) alternates RL updates of a lightweight action head with supervised updates using successful trajectories. VLA-RL (Lu et al., 2026) combines PPO with a VLM-based process reward model, whereas RIPT-VLA (Tan et al., 2025) uses sparse success rewards with improved rollout sampling and advantage estimation. TGRPO (Chen et al., 2025c) combines trajectory-level and step-level advantages, and SimpleVLA-RL (Li et al., 2026a) applies GRPO with VLA-specific exploration strategies. ConRFT (Chen et al., 2025b) updates an action head and auxiliary critic while keeping the visual encoders and Transformer backbone frozen. These methods address different aspects of VLA optimization, but do not jointly consider residual adaptation and CBF-guided safety optimization.

Another line of work limits trainable parameters by learning a policy around a pretrained controller. Refined Policy Distillation (RPD) (Jülg et al., 2025) uses a teacher VLA to guide the training of compact RL experts. Residual RL retains the base policy during execution and learns additive action corrections. ResiP (Ankile et al., 2025) augments a frozen action-chunked imitation policy with a residual controller for manipulation. PLD (Xiao et al., 2026) uses residual RL to generate recovery trajectories that are subsequently distilled into a VLA model. VLSA-RL instead deploys the residual policy alongside the frozen VLA and optimizes its corrections using CBF-derived safety information.

Safety-aware VLA adaptation has also been studied through preference and constrained optimization. GRAPE (Zhang et al., 2024) uses trajectory-level preferences with configurable constraints, including safety and efficiency objectives. SafeVLA (Zhang et al., 2025a) formulates safety alignment as a constrained learning problem, while SafeDojo (Tang et al., 2026) integrates task rewards and safety costs through constrained GRPO. VLSA-RL uses CBF-QP feedback to optimize a compact

residual policy and retains a CBF-QP layer for execution-time action correction. Safety information also determines the adaptive clipping range during policy optimization.

B PRELIMINARIES

We review the residual-control setting and a trajectory-level GRPO objective used as the starting point for VLSA-RL. The safety-aware modifications are described in Section 3.

B.1 RESIDUAL RL WITH A FROZEN VLA

Residual RL adapts a fixed base policy by learning additive action corrections. At time step t, let $s_t \in \mathcal{S}$ be the environment state, o_t the visual observation, l the language instruction, and p_t the robot proprioceptive state. The frozen VLA policy π_{VLA} comprises a vision-language encoder $\mathcal{E}_{\mathrm{VLM}}$ and an action decoder $\mathcal{D}_{\mathrm{act}}$. These modules produce the representation f_t and the base action a_t^{VLA} according to

$$f_t = \mathcal{E}_{\mathrm{VLM}}(o_t, l), \qquad a_t^{\mathrm{VLA}} = \mathcal{D}_{\mathrm{act}}(f_t, p_t). \tag{17}$$

The residual policy is conditioned on the VLA representation and a context vector z_t that captures the current control and safety state. We write its input as

$$\xi_t = [f_t; z_t], \tag{18}$$

where $z_t \in \mathbb{R}^{79}$ contains robot-state, obstacle, safety, action-history, and episode-progress features. One of these features is the previous CBF-QP correction, defined by $\delta a_{t-1}^{\mathrm{QP}} = a_{t-1}^{\mathrm{safe}} - a_{t-1}^{\mathrm{nom}}$ and initialized to zero at the start of each episode. The context vector stores its six non-gripper components. The complete feature specification is given in Appendix D.

Let $\pi_\theta^{\mathrm{res}}$ be the trainable residual policy. A residual action is sampled in the VLA action space and added to the base action. The resulting nominal action is passed through the CBF-QP layer before execution:

$$a_t^{\mathrm{res}} \sim \pi_\theta^{\mathrm{res}}(\cdot \mid \xi_t), \qquad a_t^{\mathrm{nom}} = a_t^{\mathrm{VLA}} + a_t^{\mathrm{res}}, \qquad a_t^{\mathrm{safe}} = \text{CBF-QP}(s_t, a_t^{\mathrm{nom}}). \tag{19}$$

Only the residual-policy parameters θ are updated during RL. The VLA encoder and action decoder remain frozen.

The executed action a_t^{safe} induces the next state according to the environment transition kernel $\mathcal{P}$:

$$s_{t+1} \sim \mathcal{P}(\cdot \mid s_t, a_t^{\mathrm{safe}}). \tag{20}$$

Task and safety outcomes from this interaction provide the feedback used to optimize the residual policy.

Let $\mathcal{D}$ be a distribution over task episodes, each specifying a language instruction and an initial environment configuration. For $e \sim \mathcal{D}$, repeated interaction yields a trajectory τ with score $S(\tau)$. The residual policy is trained to maximize the expected trajectory score:

$$J_{\mathrm{res}}(\theta) = \mathbb{E}_{e\sim\mathcal{D},\, \tau\sim\mathbb{P}_\theta(\cdot|e)}\left[S(\tau)\right], \tag{21}$$

where $\mathbb{P}_\theta$ denotes the trajectory distribution induced by the frozen VLA policy, residual policy, CBF-QP layer, and environment dynamics. The score used in VLSA-RL is specified in Eq. (7).

B.2 GROUP RELATIVE POLICY OPTIMIZATION

Group Relative Policy Optimization (GRPO) (Shao et al., 2024) estimates policy advantages from within-group comparisons rather than a separately trained value function. We use a trajectory-level formulation, following its application to VLA post-training (Li et al., 2026a), for the residual policy $\pi_\theta^{\mathrm{res}}(a_t^{\mathrm{res}} \mid \xi_t)$.

For a selected episode, the behavior policy $\pi_{\theta_{\mathrm{old}}}^{\mathrm{res}}$ generates a group $\mathcal{T} = \{\tau_i\}_{i=1}^G$ of G trajectories from the same initial condition. Let $|\tau_i|$ be the number of decision steps in trajectory i and $S_i = S(\tau_i)$ its score. A group-standardized trajectory advantage is defined as

$$\bar{S} = \frac{1}{G}\sum_{j=1}^{G} S_j, \qquad \sigma_S = \sqrt{\frac{1}{G}\sum_{j=1}^{G}(S_j - \bar{S})^2}, \qquad \hat{A}_i^{\mathrm{std}} = \frac{S_i - \bar{S}}{\sigma_S + \varepsilon_A}. \tag{22}$$

The constant $\varepsilon_A > 0$ ensures a nonzero denominator. Every residual action in τ_i is assigned the same trajectory-level advantage.

For each sampled residual action, the importance ratio compares its density under the current and behavior policies:

$$r_{i,t}(\theta) = \frac{\pi_\theta^{\mathrm{res}}(a_{i,t}^{\mathrm{res}} \mid \xi_{i,t})}{\pi_{\theta_{\mathrm{old}}}^{\mathrm{res}}(a_{i,t}^{\mathrm{res}} \mid \xi_{i,t})}. \tag{23}$$

Given a fixed clipping radius $\epsilon > 0$, the clipped surrogate contribution at step t of trajectory i is

$$C_{i,t}(\theta) = \min\left\{ r_{i,t}(\theta)\hat{A}_i^{\mathrm{std}}, \mathrm{clip}\big(r_{i,t}(\theta), 1-\epsilon, 1+\epsilon\big)\hat{A}_i^{\mathrm{std}} \right\}. \tag{24}$$

A trajectory-level clipped GRPO objective with optional reference-policy regularization can then be written as

$$\mathcal{L}_{\mathrm{GRPO}}(\theta) = -\mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G} \frac{1}{|\tau_i|} \sum_{t=1}^{|\tau_i|} \left(C_{i,t}(\theta) - \beta \mathcal{K}_{i,t}(\theta) \right) \right]. \tag{25}$$

The expectation is over episode groups and trajectories sampled by the behavior policy. The coefficient $\beta \geq 0$ controls reference-policy regularization, with the per-step term defined as

$$\mathcal{K}_{i,t}(\theta) = D_{\mathrm{KL}}\left(\pi_\theta^{\mathrm{res}}(\cdot \mid \xi_{i,t}) \,\|\, \pi_{\mathrm{ref}}^{\mathrm{res}}(\cdot \mid \xi_{i,t}) \right). \tag{26}$$

The reference policy $\pi_{\mathrm{ref}}^{\mathrm{res}}$ remains fixed during optimization, whereas $\pi_{\theta_{\mathrm{old}}}^{\mathrm{res}}$ is the behavior policy that generated the current rollout group. VLSA-RL replaces the group-standardized advantage and fixed clipping radius with the formulations in Eqs. (8) and (9), respectively. It also omits the KL term and adds the CBF-QP constraint loss in Eq. (13).

C Detailed CBF-QP Formulation

This section specifies the geometric barrier function and the CBF-QP projection used to correct nominal actions. The former measures separation between the end effector and an obstacle, while the latter enforces the corresponding constraint under the nominal dynamics model.

C.1 Ellipsoidal CBF Construction

Following VLSA (Hu et al., 2025), the end effector and the selected obstacle are approximated by minimum-volume enclosing ellipsoids. For $j \in \{\mathrm{ef}, \mathrm{ob}\}$, let $p_j \in \mathbb{R}^3$ be the center of ellipsoid j in the world frame. Here, ef and ob denote the end-effector and obstacle ellipsoids, respectively. Let $Q_j = \mathrm{diag}(q_{j,1}, q_{j,2}, q_{j,3}) \succ 0$ contain its local semi-axis lengths, and let $R_j \in \mathrm{SO}(3)$ specify its orientation. With $\bar{Q}_j = R_j Q_j R_j^\top$, the ellipsoid is

$$\mathcal{E}_j = \left\{ p_j + \bar{Q}_j w : \|w\|_2 \leq 1 \right\}, \tag{27}$$

The obstacle parameters are estimated from a filtered point cloud. If p_{ee} is the physical end-effector position, the center of its ellipsoidal proxy is $p_{\mathrm{ef}} = p_{\mathrm{ee}} + R_{\mathrm{ef}}\Delta p$, where Δp is a fixed offset expressed in the end-effector frame.

We construct a separating plane using the rotating supporting-hyperplane formulation (Funada et al., 2025). A virtual state $p_s \in \mathbb{S}^2$, with $\|p_s\|_2 = 1$, parameterizes the surface point $p_b = p_{\mathrm{ef}} + \bar{Q}_{\mathrm{ef}} p_s$ on the end-effector ellipsoid. The plane tangent at p_b is

$$\mathcal{T}(p_s) = \left\{ q : n^\top (q - p_{\mathrm{ef}}) = 1 \right\}, \qquad n = \bar{Q}_{\mathrm{ef}}^{-1} p_s. \tag{28}$$

The barrier value is the minimum signed distance from the obstacle ellipsoid to this tangent plane, measured along its normal direction:

$$h(s) = \min_{q \in \mathcal{E}_{\mathrm{ob}}} \frac{n^\top (q - p_{\mathrm{ef}}) - 1}{\|n\|_2} = \frac{(p_{\mathrm{ob}} - p_{\mathrm{ef}})^\top \bar{Q}_{\mathrm{ef}}^{-1} p_s - \left\| \bar{Q}_{\mathrm{ob}} \bar{Q}_{\mathrm{ef}}^{-1} p_s \right\|_2 - 1}{\left\| \bar{Q}_{\mathrm{ef}}^{-1} p_s \right\|_2}. \tag{29}$$

The second equality follows by minimizing a linear functional over $\mathcal{E}_{\mathrm{ob}}$. A nonnegative value of $h(s)$ places the obstacle ellipsoid on the nonpenetrating side of the tangent plane, so the ellipsoid interiors do not overlap. Equality permits boundary contact, whereas $h(s) > 0$ gives strict separation. The virtual state p_s allows the tangent plane to adapt to the relative geometry and can reduce conservatism. We include p_s in the augmented state s, with its prescribed evolution included in the nominal state dynamics.

C.2 Differentiable CBF-QP Safety Projection

The CBF-QP safety layer is defined using the nominal control-affine dynamics

$$\dot{s} = \hat{f}(s) + \hat{g}(s)a. \tag{30}$$

The effect of mismatch between this model and the actual dynamics is analyzed in Theorem 1. Along the nominal dynamics, the time derivative of the barrier function is

$$\dot{h}(s,a) = L_{\hat{f}}h(s) + L_{\hat{g}}h(s)a. \tag{31}$$

For an exponential CBF with $\kappa > 0$, the nominal safety condition is $\dot{h}(s,a) + \kappa h(s) \geq 0$ (Ames et al., 2017). Define

$$w_s = \big(L_{\hat{g}}h(s)\big)^\top, \qquad b_s = -L_{\hat{f}}h(s) - \kappa h(s). \tag{32}$$

Substituting these quantities into the CBF condition gives the halfspace $w_s^\top a \geq b_s$ in action space. For a nominal action a^{nom}, the safety layer selects the closest action in this halfspace by solving

$$\Pi_s(a^{\mathrm{nom}}) = \arg\min_a \frac{1}{2}\|a - a^{\mathrm{nom}}\|_2^2 \quad \text{s.t.} \quad w_s^\top a \geq b_s, \tag{33}$$

The executed action is $a^{\mathrm{safe}} = \Pi_s(a^{\mathrm{nom}})$. Because the barrier function is independent of gripper motion, the gripper component of w_s is zero and is unchanged by the projection.

For a fixed state, this QP is a Euclidean projection onto a halfspace. Its closed-form solution and derivative with respect to the nominal action are given below (Amos & Kolter, 2017; Agrawal et al., 2019).

Theorem 2. *Fix a state s with $w_s \neq 0$. The minimizer of Eq. (33) is unique and has the form*

$$\Pi_s(a^{\mathrm{nom}}) = a^{\mathrm{nom}} + \frac{\max\big(b_s - w_s^\top a^{\mathrm{nom}}, 0\big)}{\|w_s\|_2^2} w_s. \tag{34}$$

The projection is continuous in a^{nom} and differentiable except on the boundary $w_s^\top a^{\mathrm{nom}} = b_s$. Its Jacobian away from this boundary is

$$\frac{\partial a^{\mathrm{safe}}}{\partial a^{\mathrm{nom}}} = \begin{cases} I_{d_a}, & w_s^\top a^{\mathrm{nom}} > b_s, \\ I_{d_a} - \dfrac{w_s w_s^\top}{\|w_s\|_2^2}, & w_s^\top a^{\mathrm{nom}} < b_s, \end{cases} \tag{35}$$

where d_a denotes the action dimension.

Proof. Because $w_s \neq 0$, the constraint in Eq. (33) defines a nonempty halfspace. The objective is strictly convex, so the minimizer is unique. Introducing a multiplier $\lambda \geq 0$ for $b_s - w_s^\top a \leq 0$, the KKT stationarity condition yields $a^{\mathrm{safe}} = a^{\mathrm{nom}} + \lambda w_s$. If $w_s^\top a^{\mathrm{nom}} \geq b_s$, the nominal action is feasible and $\lambda = 0$. Otherwise, the constraint is active and $\lambda = (b_s - w_s^\top a^{\mathrm{nom}})/\|w_s\|_2^2$. Combining these cases gives Eq. (34). The projection is affine on either side of the constraint boundary. Differentiating each branch gives Eq. (35). The boundary is a measure-zero hyperplane, so Π_s is differentiable almost everywhere. ■

For sampled states, this almost-everywhere differentiability permits gradients from the CBF-QP constraint loss to pass through the projection. The loss can therefore encourage nominal actions that require smaller execution-time corrections.

D Implementation Details

Residual policy architecture. At each control step, the frozen VLA backbone produces a 2048-D visual-language representation f_t. The residual policy also receives the 79-D context vector z_t, which contains robot-state, obstacle, and safety features together with action history and normalized episode progress. The combined input has 2,127 dimensions. As shown in Figure 8, separate MLP branches encode f_t and z_t. Their outputs form a 384-D representation that is processed by a two-layer, 256-D fusion network to predict the mean of a 7-D residual action distribution.

Table 4 specifies the feature order of z_t. Orientations are encoded by the continuous 6-D representation $\rho_6(R) = \mathrm{vec}(R_{:,1:2})$. The relative obstacle direction is $d_t^{\mathrm{obs}} = (p_{\mathrm{ob}} - p_{\mathrm{ef}})/\|p_{\mathrm{ob}} - p_{\mathrm{ef}}\|_2$, where p_{ef} is the center of the end-effector ellipsoid defined in Appendix C.1. When no obstacle is active, the obstacle center and size are zero padded, its rotation is set to the identity, and the relative-direction feature is set to zero. Only the six non-gripper components of the previous QP correction are included in z_t.

Table 4: Composition of the 79-D context vector for the residual policy. The index ranges follow the implementation order.

Indices	Feature	Dim.	Description
1–3	p_{ee}	3	Physical end-effector Cartesian position
4–7	q_{ef}	4	End-effector orientation quaternion
8–14	q^{joint}	7	Robot joint positions
15–21	$\dot{q}^{\mathrm{joint}}$	7	Robot joint velocities
22–23	q^{grip}	2	Two gripper joint positions
24–26	p_{ef}	3	End-effector proxy-ellipsoid center
27–29	$\mathrm{diag}(Q_{\mathrm{ef}})$	3	End-effector ellipsoid semi-axis parameters
30–35	$\rho_6(R_{\mathrm{ef}})$	6	End-effector ellipsoid rotation in 6-D form
36–38	p_{ob}	3	Obstacle ellipsoid center
39–41	$\mathrm{diag}(Q_{\mathrm{ob}})$	3	Obstacle ellipsoid semi-axis parameters
42–47	$\rho_6(R_{\mathrm{ob}})$	6	Obstacle ellipsoid rotation in 6-D form
48–50	d_t^{obs}	3	Unit direction from p_{ef} to p_{ob}
51	$h(s_t)$	1	Current CBF value
52–57	$\delta a_{t-1}^{\mathrm{QP}}$	6	Previous QP correction for non-gripper action
58–64	a_t^{VLA}	7	Current frozen-VLA action
65–71	a_{t-1}^{safe}	7	Previously executed safe action
72–78	a_{t-1}^{res}	7	Previous residual action
79	$t/T_{\max}$	1	Normalized episode progress
Total		**79**	

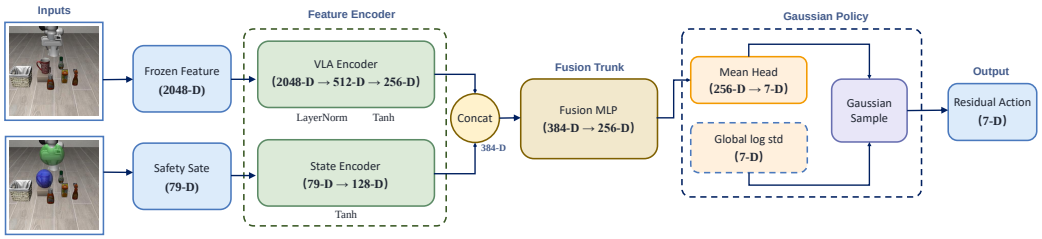

Figure 8: Residual policy architecture of VLSA-RL. The Gaussian mean is conditioned on frozen VLA features and low-dimensional state and safety inputs, while the log standard deviation is a learned global, state-independent parameter.

Action distribution and safety integration. The residual policy uses seven trainable, state-independent log standard deviations, initialized to -1.5 and clipped to $[-5, 1]$. A Gaussian latent action is sampled, transformed by $\tanh$, and added to the frozen VLA action. The mean head is initialized to zero, so the initial residual distribution is centered on zero correction, although individual sampled corrections may be nonzero. Before execution, the nominal action is projected by the CBF-QP layer in Eq. (3). The resulting QP correction is included in the residual-policy input at the next control step.

Optimization. A separate residual policy is trained for each task-suite and safety-level configuration, while the VLA backbone remains frozen. Training uses 300 updates of Adam with a learning rate of 3×10^{-5} and one epoch per rollout batch. At each update, $B = 4$ episodes are sampled per task and $G = 8$ rollouts are collected from each selected initial condition. The policy loss uses the

fixed-scale group-relative advantage and adaptive clipping defined in Sections 3.3 and 3.4. The auxiliary QP loss excludes the gripper dimension, and the gradient norm is clipped to 0.1. Algorithm 1 summarizes the training procedure.

Algorithm 1: VLSA-RL

Input: Frozen VLA policy π_{VLA}, residual policy $\pi_\theta^{\mathrm{res}}$, episode sets $\{\mathcal{E}_k\}$, $N = 300$, $B = 4$, $G = 8$, $\epsilon_{\min} = 0.1$, $\epsilon_{\max} = 0.2$, $\lambda_{\mathrm{QP}} = 0.05$.
Output: Trained residual policy $\pi_\theta^{\mathrm{res}}$.

Initialize θ and $w_i^k \leftarrow 1$, and freeze π_{VLA}.
for $n = 1, \ldots, N$ **do**
  Set $\theta_{\mathrm{old}} \leftarrow \theta$.
  for each task k **do**
    Sample B episodes by p_i^k in Eq. (14).
    for each sampled episode **do**
      Collect G rollouts from the same initial state using $a_t^{\mathrm{res}} \sim \pi_{\theta_{\mathrm{old}}}^{\mathrm{res}}(\cdot \mid \xi_t)$ and the CBF-QP projection in Eq. (3).
      Compute trajectory scores and advantages using Eqs. (7)–(8).
      Set the adaptive clipping bound ϵ_i by Eq. (9).
      Update θ with Eqs. (10), (12), and (13).
      Update w_i^k with Eqs. (15)–(16).
    end for
  end for
  Periodically reset all $w_i^k \leftarrow 1$.
end for
return $\pi_\theta^{\mathrm{res}}$.

E Detailed Task-Level Results

We report the complete task-level evaluation of VLSA-RL on all four SafeLIBERO suites and both obstacle-interference levels. Results are reported at the best-SSR checkpoint for each safety level. When multiple checkpoints attain the same suite-level SSR, we use the earliest one. Each entry is computed from 50 evaluation episodes. CAR denotes the collision avoidance rate, TSR the task success rate, SSR the safe success rate, and ETS the mean number of execution steps. Higher is better for CAR, TSR, and SSR, whereas lower is better for ETS.

Table 5: **Detailed results on the SafeLIBERO-Spatial suite.**

Task description	Level	CAR ↑	TSR ↑	SSR ↑	ETS ↓
Task 0: Pick up the black bowl between the plate and the ramekin and place it on the plate.	I	86.0	98.0	84.0	106.7
	II	80.0	100.0	80.0	101.3
Task 1: Pick up the black bowl on the ramekin and place it on the plate.	I	70.0	82.0	68.0	157.3
	II	82.0	84.0	82.0	143.0
Task 2: Pick up the black bowl on the stove and place it on the plate.	I	70.0	100.0	70.0	155.8
	II	88.0	94.0	84.0	151.6
Task 3: Pick up the black bowl on the wooden cabinet and place it on the plate.	I	78.0	92.0	74.0	152.8
	II	92.0	96.0	88.0	134.8
Suite average	I	76.0	93.0	74.0	143.1
	II	85.5	93.5	83.5	132.7

Table 6: **Detailed results on the SafeLIBERO-Goal suite.**

Task description	Level	CAR ↑	TSR ↑	SSR ↑	ETS ↓
Task 0: Put the bowl on the plate.	I	100.0	100.0	100.0	100.3
	II	98.0	96.0	94.0	139.6
Task 1: Put the bowl on top of the cabinet.	I	100.0	86.0	86.0	125.2
	II	100.0	100.0	100.0	85.8
Task 2: Put the bowl on the stove.	I	98.0	92.0	90.0	134.9
	II	96.0	90.0	86.0	141.0
Task 3: Open the top drawer and put the bowl inside.	I	96.0	72.0	70.0	237.9
Task 3: Put the cream cheese in the bowl.	II	90.0	92.0	86.0	121.9
Suite average	I	98.5	87.5	86.5	149.6
	II	96.0	94.5	91.5	122.1

Table 7: **Detailed results on the SafeLIBERO-Object suite.**

Task description	Level	CAR ↑	TSR ↑	SSR ↑	ETS ↓
Task 0: Pick up the orange juice and place it in the basket.	I	86.0	92.0	84.0	138.2
	II	92.0	88.0	84.0	171.7
Task 1: Pick up the chocolate pudding and place it in the basket.	I	60.0	76.0	58.0	231.7
	II	78.0	100.0	78.0	151.2
Task 2: Pick up the milk and place it in the basket.	I	76.0	52.0	44.0	259.1
	II	98.0	78.0	76.0	180.5
Task 3: Pick up the BBQ sauce and place it in the basket.	I	70.0	92.0	66.0	150.1
	II	70.0	92.0	70.0	163.7
Suite average	I	73.0	78.0	63.0	194.8
	II	84.5	89.5	77.0	166.8

Table 8: **Detailed results on the SafeLIBERO-Long suite.**

Task description	Level	CAR ↑	TSR ↑	SSR ↑	ETS ↓
Task 0: Put both the alphabet soup and the cream cheese box in the basket.	I	42.0	76.0	30.0	400.1
	II	92.0	58.0	56.0	475.3
Task 1: Put both the alphabet soup and the tomato sauce in the basket.	I	44.0	2.0	2.0	546.6
	II	84.0	90.0	74.0	363.3
Task 2: Put the white mug on the left plate and put the yellow and white mug on the right plate.	I	82.0	78.0	66.0	374.0
	II	22.0	44.0	10.0	457.1
Task 3: Put the white mug on the plate and put the chocolate pudding to the right of the plate.	I	22.0	90.0	18.0	332.1
	II	82.0	22.0	16.0	501.7
Suite average	I	47.5	61.5	29.0	413.2
	II	70.0	53.5	39.0	449.3

F LIMITATIONS AND FUTURE WORK

The current evaluation does not establish generalization across the full range of safety-critical manipulation conditions. Training uses a limited range of obstacle positions and sizes, so the reported results may not extend to substantially different layouts, scales, or shapes. In addition, the CBF formulation represents the end effector and a selected obstacle with ellipsoidal proxies. Collisions involving other robot links or multiple obstacles are not explicitly addressed by the current constraint model. The scope of the safety claim is also limited by the nominal dynamics model. The residual policy is not independently certified to satisfy the CBF condition. Safety therefore depends on the execution-time CBF-QP layer and on the validity of its modeling assumptions. Moreover, the evaluation is confined to simulation and does not account for perception errors, communication delays, calibration uncertainty, or unmodeled actuator dynamics on physical robots. Although only a compact residual policy is optimized, grouped online rollouts remain computationally expensive because they rely on CPU-intensive MuJoCo simulation (Todorov et al., 2012; Zhu et al., 2020; Liu et al., 2023; Hu et al., 2025).

Future work will assess generalization under broader obstacle variation, using environment randomization and curriculum learning to cover different positions, sizes, shapes, and layouts. Extending the CBF formulation to multiple robot links and obstacles would broaden the set of collisions represented by the safety layer. We also plan to investigate verifiable residual-policy structures and explicit CBF constraints during learning to reduce reliance on execution-time filtering alone.

Another direction is to develop a more principled adaptive clipping rule that relates the policy-update range to CBF margins, QP correction magnitudes, and dynamics uncertainty. This would clarify the role of safety information in policy optimization beyond the current predefined mapping. To reduce training time, we will examine GPU-parallel simulation platforms such as NVIDIA Isaac Sim (Gao et al., 2026). Finally, physical-robot experiments will be needed to evaluate the method under realistic perception, communication, and dynamics uncertainty.