CAFE-VLA: CONTROLLABLE AVOIDANCE WITH FIELD ENCODING FOR COMPACT VISION-LANGUAGE-ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

We present CAFE-VLA (**C**ontrollable **A**voidance with **F**ield **E**ncoding), a controllable obstacle-avoidance framework for compact vision-language-action (VLA) models. The CAFE-VLA policy is conditioned on two complementary inputs: an avoidance-strength scalar and an end-effector-centric 3D geometry field. The avoidance-strength scalar specifies the requested degree of avoidance, while the 3D geometry field represents local obstacle distances and clearance directions at a fixed set of probes. We inject the scalar condition into the action expert via Feature-wise Linear Modulation (FiLM), while the 3D geometry field conditions the policy through cross-attention to field tokens and visual prompting encoding obstacle distances. This structure-aware design adds less than 5% to the parameter count. A sweep over the avoidance-strength scalar confirms monotonic control over obstacle clearance (Spearman $\rho = 0.953$), while classifier-free guidance enables stable extrapolation to a strength 50% above the training maximum. On risk-stratified SafeLIBERO, CAFE-VLA achieves an average task success rate of 74.32%, compared with 70.34% for the strongest safety-aware baseline. CAFE-VLA retains a task success rate of 82.54% on the original LIBERO benchmark.

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.

1 INTRODUCTION

Vision-Language-Action (VLA) models map visual observations and language instructions to robot actions, offering a path toward generalist manipulation policies (Zitkovich et al., 2023; Jiang et al., 2023). However, deploying them in unstructured environments requires avoiding obstacles without sacrificing task completion.

Avoidance should also adapt to context: tasks involving fragile obstacles or safety-critical workspaces require greater clearance than those in open, low-risk settings. This calls for *controllable obstacle avoidance*, where a single policy adjusts avoidance strength at runtime. Existing approaches do not provide this capability efficiently. Safe reinforcement learning (RL) methods learn policies for predefined constraints and generally require retuning and retraining to change them (Altman, 1999; Thananjeyan et al., 2020; Liu et al., 2023b). Post-hoc safety filters preserve policy parameters but may induce out-of-distribution states through action corrections (Hu et al., 2026). Constrained flow matching applies corrections based on control barrier functions (CBFs) during denoising,

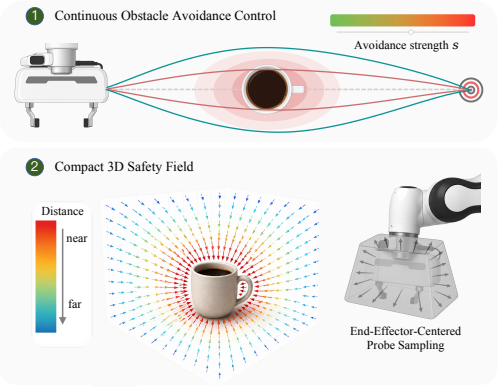

Figure 1: CAFE-VLA continuously adjusts obstacle avoidance through an avoidance-strength scalar s. An action-aligned, end-effector-centric 3D geometry field encodes obstacle distance and clearance direction at local probes, providing compact geometric guidance without full-scene modeling.

however, it still relies on fixed safety margins and CBF parameters (English et al., 2026).

Beyond controllability, existing 3D VLA approaches based on scene-level point clouds or depth inputs (Zhen et al., 2024; Qu et al., 2025; Ze et al., 2024) prioritize manipulation performance over obstacle avoidance, and impose substantial geometric reasoning demands on compact VLAs.

In this work, we treat avoidance preference and local geometry as structurally distinct conditioning signals. Accordingly, we propose CAFE-VLA, which equips a compact VLA with controllable obstacle avoidance while preserving its pretrained manipulation capabilities. It conditions the policy on an avoidance-strength scalar and a compact local 3D geometry field through structure-matched injection. Our contributions are:

1. **Continuous and extrapolatable avoidance control.** An avoidance-strength scalar modulates every action-expert layer through Feature-wise Linear Modulation (FiLM). Its monotonic relationship with clearance enables continuous runtime control, while classifier-free guidance (CFG) extrapolates beyond the training range.
2. **An action-aligned, end-effector-centric 3D geometry field.** Fixed probes sample the end-effector envelope at its current pose and along short-horizon motion, encoding position, obstacle distance, and clearance direction. The representation is independent of scene complexity, supplies explicit geometry to the compact VLA, and supports generalization across obstacle configurations.
3. **Structure-matched dual-pathway geometry injection.** Sparse cross-attention lets action features query probe-level geometry, while visual prompting projects the 3D geometry field onto the RGB observation. These pathways combine local geometry with scene context at minimal parameter overhead and interference with the pretrained policy.

2 RELATED WORK

2.1 COMPACT VISION-LANGUAGE-ACTION MODELS

Generalist VLAs such as OpenVLA (Kim et al., 2025) and π_0 (Black et al., 2025) ground language and vision in robot actions. We adopt SmolVLA (Shukor et al., 2025) as our compact backbone; related compact VLAs likewise target resource-constrained deployment.

2.2 SAFETY AND CONTROLLABLE AVOIDANCE

Safe RL enforces safety through constrained Markov decision processes (Altman, 1999) and control barrier functions (Xiao et al., 2023; Liu et al., 2023a), while Constrained Decision Transformer (Liu et al., 2023b) exposes reward and cost budgets as policy conditions. For VLAs, SafeVLA learns safety-aligned policies (Zhang et al., 2025a); AEGIS, the Vision-Language-Safe Action (VLSA) architecture, applies a post-hoc safety layer (Hu et al., 2026); and Neuro-Symbolic Safety Guidance (NSG) performs CBF-guided correction during flow matching (English et al., 2026). These methods optimize or enforce predefined constraints rather than providing continuous, scene-conditioned avoidance while preserving pretrained manipulation capabilities. We use FiLM (Perez et al., 2018) and classifier-free guidance (Ho & Salimans, 2022), which supports inference-time control in diffusion and flow-based robot policies (Chi et al., 2023; Zhang et al., 2025b; Lu et al., 2025), to expose avoidance strength as a lightweight runtime condition.

2.3 GEOMETRIC CONDITIONING FOR MANIPULATION POLICIES

3D-VLA (Zhen et al., 2024), SpatialVLA (Qu et al., 2025), and point-cloud manipulation policies such as DP3 (Ze et al., 2024), iDP3 (Ze et al., 2025), and PointVLA (Li et al., 2026) derive spatial relations from full-scene geometry. SDF and potential-field methods (Khatib, 1986) provide explicit obstacle geometry but typically operate outside the VLA feature hierarchy, while TraceVLA (Zheng et al., 2024) encodes state–action trajectories as visual prompts. Our fixed-size local field instead represents obstacle distance and clearance direction at 78 action-aligned probes and injects them through cross-attention and visual prompting.

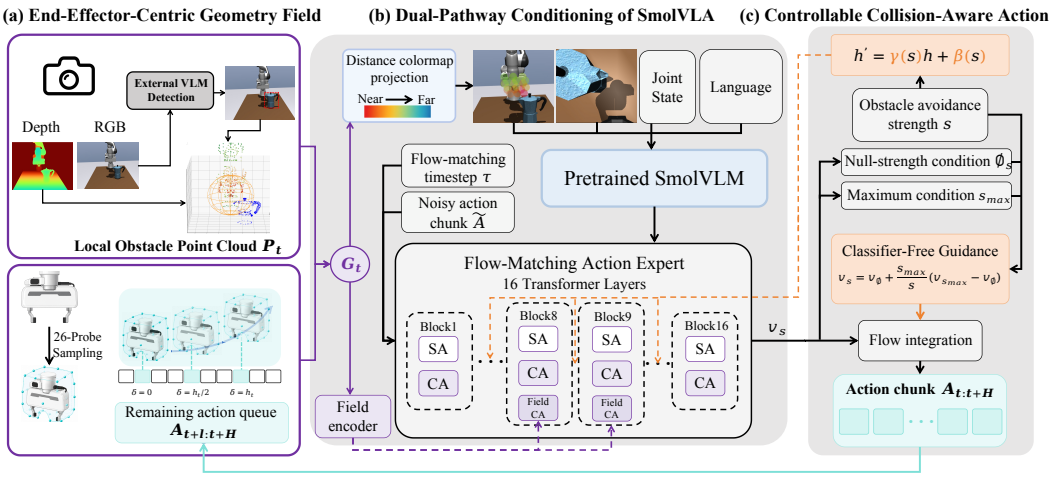

Figure 2: Architecture of CAFE-VLA. The avoidance strength s modulates the action expert, while the 3D geometry field G_t conditions the policy through field tokens and visual prompting.

3 METHODOLOGY

3.1 OVERVIEW

We decompose controllable obstacle avoidance into an avoidance-strength scalar s and an action-aligned, end-effector-centric 3D geometry field G_t. The former controls avoidance at runtime, while the latter encodes local obstacle distances and clearance directions in a fixed-size representation. At control step t, the policy generates the action chunk $A_t = \pi_\theta(I_t, S_t, L, s, G_t)$, where $I_t = \{I_t^{\text{third}}, I_t^{\text{wrist}}\}$ contains the RGB observations, S_t is the proprioceptive state containing the joint positions $\mathbf{q}_t$, L is the instruction, and $A_t = (a_t, \ldots, a_{t+H-1})$ is an H-step action chunk. For each demonstration ξ through offline relabeling, we obtain a trajectory-level avoidance-strength label $s(\xi) \in [0, s_{\max}]$, where $s_{\max} = 10$. During the inference, the conditioning value s is provided by a user or an external vision-language model (VLM). We construct G_t based on the current depth observation and the short-horizon commanded motion.

We adopt SmolVLA (Shukor et al., 2025) as our base policy because its compact, open-source 450M-parameter architecture is suitable for resource-constrained deployment and is pretrained on diverse manipulation tasks. SmolVLA comprises a SmolVLM vision-language backbone and a flow-matching action expert, to which we add the conditioning modules described below.

3.2 ACTION-ALIGNED END-EFFECTOR-CENTRIC 3D GEOMETRY FIELD

Obstacle avoidance primarily requires geometry near the end effector and along its commanded motion. We therefore retain only this local geometry and use a fixed set of probes to extract explicit geometric features, rather than reconstructing the full scene.

3.2.1 LOCAL OBSTACLE POINT SET

At each control step, a third-person RGB-D camera provides aligned RGB and depth images. An external VLM identifies task-relevant obstacles in the RGB image and provides a semantic mask. For each masked pixel with valid depth, we recover its 3D position in the robot base frame and retain the resulting 3D points within radius r of the end-effector center. Following the protective separation distance principle (Marvel & Norcross, 2017), we choose the local filtering radius r to satisfy:

$$r \geq v_{\max}\Delta t_{\text{react}} + \frac{v_{\max}^2}{2a_{\text{brake}}} + l_{\text{EE}}, \tag{1}$$

where Δt_{react}, $v_{\max}$, a_{brake}, and l_{EE} denote the perception-to-action latency, maximum speed, braking deceleration, and enclosing-sphere radius, respectively.

3.2.2 ACTION-ALIGNED END-EFFECTOR ENVELOPE PROBES

We place 26 probes on an oriented bounding box enclosing the end effector (8 vertices, 12 edge midpoints, and 6 face centers), with its width determined by the gripper opening. During SmolVLA's asynchronous action-chunk execution, the remaining actions in the most recent raw prediction, before temporal aggregation, define a short-horizon motion. Let $\{\bar{\mathbf{x}}_i\}_{i=1}^{26}$ be the probes in the current end-effector frame and $\aleph_t$ the number of remaining raw joint-position targets. To represent the short-horizon motion, we evaluate the current, midpoint, and endpoint of the latest raw action chunk, indexed by $\delta \in \{0, \lfloor \aleph_t/2 \rfloor, \aleph_t\}$. Forward kinematics yields the corresponding end-effector pose $\mathbf{T}_{t,\delta}$. Relative to the current measured pose $\mathbf{T}_{t,0}$, each template probe $\bar{\mathbf{x}}_i$ is transformed as $\mathbf{e}_{t,i,\delta} = \mathbf{T}_{t,0}^{-1}\mathbf{T}_{t,\delta}(\bar{\mathbf{x}}_i)$, yielding 78 motion-aligned probes. We transform the probes into the robot base frame using the current end-effector pose when querying the local obstacle points. Thus, raw predicted joint actions provide motion-aligned coverage without a trajectory predictor.

3.2.3 PROBE-LEVEL DISTANCE AND CLEARANCE DIRECTION

For each probe j, we find its k_{nn} nearest obstacle points, compute their mean distance $d_{t,j}$, and normalize their mean unit direction toward the probe to obtain the clearance direction $\hat{\mathbf{u}}_{t,j}$. Let $\mathbf{x}_{t,j}^{\mathrm{EE}}$ denote the coordinate of probe j in the end-effector frame. Its feature is $\mathbf{f}_{t,j} = [\mathbf{x}_{t,j}^{\mathrm{EE}}, d_{t,j}, \hat{\mathbf{u}}_{t,j}]$. The ordered 3D geometry field $G_t = (\mathbf{f}_{t,j})_{j=1}^{78}$ explicitly represents control-relevant geometry in the current end-effector frame and is rebuilt at each policy inference step.

3.3 ADJUSTABLE AND EXTRAPOLATABLE AVOIDANCE STRENGTH

Because the same geometry may require different responses across tasks, we use s as an independent control variable. At deployment, classifier-free guidance (CFG) enables bounded extrapolation beyond the training range.

3.3.1 TRAJECTORY-LEVEL AVOIDANCE-STRENGTH ANNOTATION

Let $d_\xi(t)$ be the signed distance from the end-effector collision geometry to the obstacle's initial occupied volume, where $d_\xi(t) < 0$ indicates penetration, and let $W_\xi = \{t : d_\xi(t) < r\}$ be the interaction window. For a demonstration ξ, we assign the strength

$$s(\xi) = s_{\max}\left[\frac{\tilde{d}(\xi) - d_{\mathrm{lo}}}{d_{\mathrm{hi}} - d_{\mathrm{lo}}}\right]_0^1, \qquad \tilde{d}(\xi) = -\tau \log\left(\frac{1}{|W_\xi|}\sum_{t\in W_\xi} e^{-d_\xi(t)/\tau}\right), \tag{2}$$

where $\tilde{d}$ is the soft-minimum signed clearance, approaching the minimum as $\tau \to 0$ and otherwise reflecting penetration severity and duration; d_{lo} and d_{hi} are the minimum and maximum values of $\tilde{d}$ over the training set, and $[\cdot]_0^1$ clips to $[0, 1]$.

3.3.2 GLOBAL MODULATION AND STRENGTH EXTRAPOLATION

A shared FiLM MLP maps the avoidance-strength scalar s to channel-wise scale $\boldsymbol{\gamma}(s)$ and bias $\boldsymbol{\beta}(s)$. For an action-expert hidden state $\mathbf{h}_\ell$, FiLM produces $\mathbf{h}'_\ell = \boldsymbol{\gamma}(s) \odot \mathbf{h}_\ell + \boldsymbol{\beta}(s)$ at every layer and action token. During training, we randomly replace s with a learned null embedding so that the model learns conditional and null-conditioned flow fields (Ho & Salimans, 2022). At inference, we directly use the action-flow prediction $\mathbf{v}_s$ conditioned on the requested strength s within the training range. Beyond this range, we extrapolate from the maximum-strength prediction $\mathbf{v}_{s_{\max}}$ relative to the null-conditioned reference $\mathbf{v}_\varnothing$:

$$\mathbf{v}_{\mathrm{guided}}(s) = \begin{cases} \mathbf{v}_s, & 0 \le s \le s_{\max}, \\ \mathbf{v}_\varnothing + \dfrac{s}{s_{\max}}\left(\mathbf{v}_{s_{\max}} - \mathbf{v}_\varnothing\right), & s > s_{\max}, \end{cases} \tag{3}$$

where all three predictions are evaluated at the same noisy action state and flow-matching timestep, under the same observation, instruction, and geometry field. With the compact VLA backbone, the

two forward passes remain computationally lightweight. This extrapolates the maximum-strength flow increment, and Sec. 4.2 empirically determines its reliable range.

3.4 Dual-Pathway Geometry Injection

We inject the action-aligned, end-effector-centric 3D geometry field G_t through field-token cross-attention for probe-level metric geometry and visual prompting for image-aligned scene context.

3.4.1 Field-Token Pathway: Sparse Cross-Attention

A shared two-layer MLP ϕ_G independently encodes each probe:

$$\mathbf{Z}_t = \big(\phi_G(\mathbf{f}_{t,j})\big)_{j=1}^{78} \in \mathbb{R}^{78\times d_{\text{model}}}. \tag{4}$$

We retain the original attention modules and add field-token cross-attention only to action-expert blocks $\ell \in \{8, 9\}$. The field-conditioned action-expert hidden state $\widetilde{\mathbf{h}}_\ell$ is obtained through the following gated residual update:

$$\widetilde{\mathbf{h}}_\ell = \mathbf{h}_\ell + \alpha_\ell \,\mathrm{CrossAttn}_\ell(\mathbf{h}_\ell, \mathbf{Z}_t, \mathbf{Z}_t). \tag{5}$$

The zero-initialized gate α_ℓ preserves the pretrained policy at initialization. Keeping probes as separate keys and values permits selective access to local geometry, while sparse insertion limits overhead and interference.

3.4.2 Visual-Prompting Pathway: End-Effector Envelope Coloring

To ground the field in the pretrained visual representation, we project each probe onto the third-person image and color it by normalized distance $\bar{d}_{t,j} = \mathrm{clip}(d_{t,j}/d_{\text{vis}}, 0, 1)$ using a fixed red–yellow–green map, where d_{vis} is the maximum displayed distance. We apply a depth-consistency test to remove occluded probes and render the remaining projections as enlarged, alpha-blended markers to preserve the underlying image texture; the wrist image remains unchanged.

3.5 Training Objective

To preserve SmolVLA's original capabilities, we construct a clean branch for 30% of training samples by using the unmodified third-person image and disabling both strength and geometry conditioning. On these clean-branch samples, we match the current policy's flow prediction v_θ to the frozen vanilla model's prediction v_0 under the same noisy action $\widetilde{A}_\tau$ and flow-matching timestep τ:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{FM}} + \lambda_{\text{preserve}} \left\| v_\theta(\widetilde{A}_\tau, \tau) - \mathrm{sg}\Big[v_0(\widetilde{A}_\tau, \tau)\Big] \right\|_2^2. \tag{6}$$

Here, $\mathcal{L}_{\text{FM}}$ is the standard flow-matching loss for learning the target action flow, and sg denotes stop-gradient through the frozen vanilla model. $\mathcal{L}_{\text{total}}$ is the overall training objective, with $\lambda_{\text{preserve}}$ controlling the weight of the capability-preservation term. Together, the FiLM MLP, field encoder, and sparse cross-attention add less than 5% to the original SmolVLA parameter count.

4 Experiments

4.1 Experimental Setup

We evaluate CAFE-VLA in MuJoCo-based LIBERO environments and in real-world experiments. Our primary benchmark, SafeLIBERO (Hu et al., 2026), augments 16 tasks from four LIBERO suites with two safety-critical obstacle scenarios per task. To measure capability retention, we additionally evaluate on the original obstacle-free LIBERO benchmark. Simulation results are averaged over three seeds. Real-world experiments use an AgileX PiPER, a fixed third-person Intel RealSense RGB-D camera pitched 45° downward, and a wrist RGB camera pitched 30° downward (Sec. 4.6). Fig. 3 shows the real-world and simulated environments across the three contact-risk levels used in our evaluation. Following SafeLIBERO, we report collision avoidance rate (CAR), task success rate (TSR), and execution time steps (ETS). CAR is the fraction of rollouts without contact, TSR the fraction satisfying the task objective, and ETS the mean episode length. We additionally report minimum clearance and computational overhead where relevant.

We compare against unmodified SmolVLA and safety-aware baselines and ablate strength selection, 3D representation, and condition injection. All ablations share the backbone, data, and training and evaluation protocols.

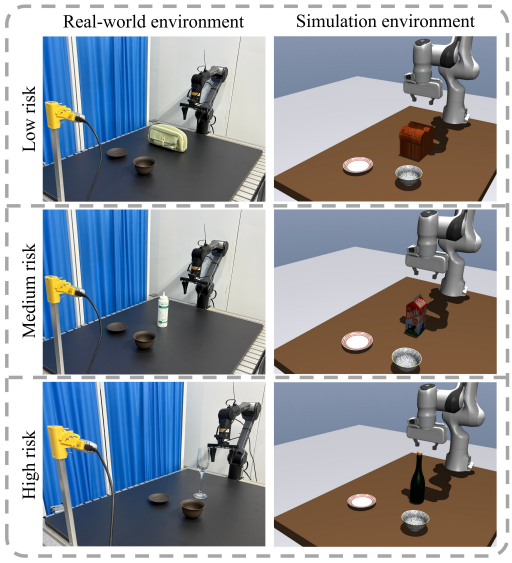

Figure 3: Real-world (left) and simulated (right) environments at low, medium, and high contact risk, defined by the consequence of contact.

4.2 Continuous Control via Avoidance-Strength Conditioning

As shown in Fig. 4, we sweep $s \in [0, 20]$ at unit intervals in a controlled single-obstacle scene and measure mean minimum clearance and TSR to test whether CFG extends the training range $s \in [0, 10]$ without degrading task success. CFG is used only for $s > 10$.

Within the training range, clearance increases consistently with s while TSR remains stable; the relationship remains strongly monotonic over $s \in [0, 18]$ (Spearman's $\rho = 0.953$, with p-value $= 2.5 \times 10^{-11}$). Under CFG, clearance remains monotonic to $s \approx 18$, but TSR is stable only for $s \leq 15$; hence, $s \in (10, 15]$ is the effective extrapolation range. Larger values cause excessive detours and unrecoverable joint states, and the clearance response becomes unstable for $s > 18$, consistent with overguidance moving trajectories outside the demonstration distribution.

4.3 Scene-Adaptive Avoidance Strength

We next test whether scene-adaptive strength improves the clearance–success trade-off. Scenes are grouped by the consequence of contact (Fig. 3): low risk permits contact, medium risk may fail if an obstacle is displaced, and high risk marks any contact as failure. Other task and evaluation conditions are unchanged.

We compare unmodified SmolVLA; AEGIS (Hu et al., 2026), a post-hoc CBF filter whose gain, controlling convergence to the safe-set boundary, is tuned per scene for the highest TSR; NSG (English et al., 2026), trajectory-level CBF guidance during flow matching whose decay rate, controlling safety enforcement, is tuned per scene for the highest TSR; CAFE-VLA-Fixed with $s = 8$; and CAFE-VLA-Oracle, which exhaustively selects s per scene and serves only as an upper bound. The deployable CAFE-VLA uses a VLM to predict $s \in [0, 15]$ for each scene. For a matched comparison, AEGIS and NSG are applied to SmolVLA models fine-tuned on our avoidance demonstrations without CAFE-VLA's strength or geometry injection modules. For CBF distance computation in simulation, AEGIS and NSG receive the exact obstacle geometry from the simulator, whereas CAFE-VLA constructs its 3D geometry field only from the RGB-D point cloud.

As shown in Table 1, compared with CAFE-VLA-Fixed, adaptive selection improves TSR by 5.10% at low risk and 9.17% at high risk, while reducing low-risk ETS by 39 steps. Thus, a fixed strength is overly conservative at low risk yet insufficient at high risk. Despite oracle tuning, at high risk CAFE-VLA outperforms AEGIS and NSG in TSR by 13.96 and 4.17%, respectively. Across low-to-high risk, CAFE-VLA has a CAR span of 64.27%, compared with 20.31% for AEGIS and 22.50% for NSG. VLM-selected strengths achieve an average TSR only 1.28% below the per-scene oracle across the three risk levels.

4.4 Explicit 3D Representation and Dual-Pathway Ablation

We compare CAFE-VLA w/o Geometry, which retains avoidance-strength conditioning but removes the geometry field; QDepth-VLA (Li et al., 2025), reproduced with auxiliary quantized-depth prediction; PointVLA (Li et al., 2026), adapted with an iDP3-style global point-cloud encoder (Ze

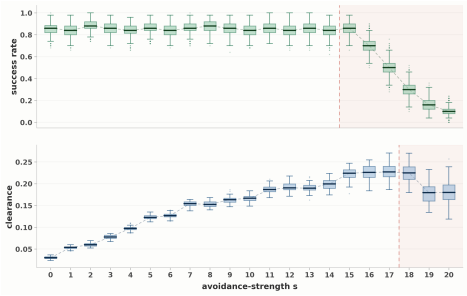

Figure 4: TSR (top) and minimum clearance (bottom) over $s \in [0, 20]$. Training covers $s \in [0, 10]$, while CFG extrapolates over $s \in (10, 20]$. The red dashed lines mark the stable-TSR limit ($s \leq 15$, top) and the monotonic-clearance limit ($s \leq 18$, bottom).

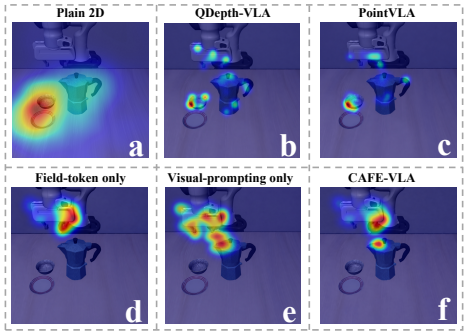

Figure 5: Mean visual attention maps over action-denoising steps for the same observation and a shared color scale.

et al., 2025); field-token-only and visual-prompting-only variants; and the full CAFE-VLA model. All variants use the same SmolVLA backbone, avoidance demonstrations, training protocol, risk-stratified evaluation, and strength selection. The resulting comparison is reported in Table 2.

Explicit 3D representation. Although CAFE-VLA w/o Geometry attains the highest low-risk TSR, its TSR drops to 48.96% at medium risk and 45.63% at high risk. Compared with this variant, QDepth improves TSR by 14.48% at medium risk and 15.72% at high risk by supplying depth supervision. Relative to QDepth, PointVLA further improves TSR by 4.37% at medium risk and 2.83% at high risk because it encodes real 3D point-cloud observations rather than predicting depth. Compared with PointVLA, our 3D geometry field raises TSR by another 6.15% at medium risk and 7.29% at high risk by explicitly representing end-effector–obstacle relations.

Dual-pathway geometry injection. Both single pathways underperform their combination at medium and high risk. Compared with the field-token-only variant, adding visual prompting improves TSR by 12.50% at medium risk and 13.13% at high risk by supplying image-aligned scene context. Compared with the visual-prompting-only variant, adding field tokens improves TSR by 5.63% at medium risk and 8.65% at high risk by supplying explicit local metric geometry. Their

Table 1: Scene-adaptive versus fixed avoidance on risk-stratified SafeLIBERO. The best TSR among deployable methods is highlighted in bold; the strongest safety-aware baseline is marked with an underline.

Method	Risk	CAR (%)	TSR (%)↑	ETS (steps)
CAFE-VLA	Low	20.73	77.50	248
	Medium	81.25	**73.96**	302
	High	85.00	**71.46**	323
CAFE-VLA-Fixed	Low	79.90	72.40	287
	Medium	80.73	71.15	290
	High	80.10	62.29	289
CAFE-VLA-Oracle (upper bound)	Low	21.56	78.75	250
	Medium	82.50	75.31	306
	High	86.35	72.71	327
AEGIS	Low	55.63	<u>73.23</u>	259
	Medium	72.81	69.38	271
	High	75.94	57.50	286
NSG	Low	60.21	72.81	276
	Medium	78.44	<u>70.94</u>	284
	High	82.71	<u>67.29</u>	307
SmolVLA-Base	Low	16.35	**80.63**	244
	Medium	16.77	47.40	248
	High	16.88	9.58	251

Table 2: 3D-representation and dual-pathway ablations under a common backbone, dataset, training setup, and evaluation protocol. The best TSR is highlighted in bold.

Representation	Risk	CAR (%)	TSR (%)↑	ETS (steps)
CAFE-VLA	Low	20.73	77.50	248
	Medium	81.25	**73.96**	302
	High	85.00	**71.46**	323
Field-token pathway only	Low	14.48	64.38	275
	Medium	68.75	61.46	316
	High	79.38	58.33	356
Visual-prompting pathway only	Low	20.83	77.29	254
	Medium	76.46	68.33	298
	High	74.69	62.81	355
PointVLA	Low	16.77	71.25	266
	Medium	77.40	67.81	296
	High	81.67	64.17	343
QDepth-VLA	Low	18.44	71.25	259
	Medium	72.19	63.44	305
	High	74.58	61.35	359
CAFE-VLA w/o Geometry	Low	16.15	**79.79**	252
	Medium	54.69	48.96	327
	High	58.33	45.63	376

Table 3: Avoidance-strength injection. ρ denotes the strength–clearance correlation; s_{sat} denotes the stable-TSR range; s_{mono} denotes the monotonic-clearance limit; and $\Delta P, \Delta L$ denote the parameter and latency overheads.

Metric	**FiLM**	Additive embedding	In-context token
$\rho \uparrow$	**0.953**	0.914	0.858
s_{sat}	**[0,15]**	[0,12]	[0,10]
s_{mono}	**≈18**	≈14	≈12
$\Delta P \downarrow$	+0.31%	**+0.29%**	+0.42%
$\Delta L \downarrow$	+1.2%	**+1.1%**	+1.8%

Table 4: Geometry-field injection (FiLM fixed) on SafeLIBERO at $s = 8$ and original LIBERO. $\Delta P, \Delta L$ denote the additional parameter and latency overheads relative to the shared FiLM-conditioned backbone.

Metric	**Sparse field-token cross-attention**	In-context concatenation	Attention pooling	Max pooling
Safe. CAR (%)	80.21	**81.04**	66.88	64.17
Safe. TSR (%)	**68.65**	67.40	60.10	57.92
LIB. TSR (%)	**82.54**	79.42	82.42	82.00
$\Delta P \downarrow$	+1.01%	**+0.63%**	+0.88%	+0.71%
$\Delta L \downarrow$	+5.1%	+18.9%	+2.1%	**+1.9%**

Original SmolVLA: SafeLIBERO CAR 16.67 and TSR 45.83; LIBERO TSR 82.91; $\Delta P{=}\Delta L{=}0$.

combination therefore captures complementary scene context and end-effector–obstacle relations, and also achieves the highest CAR at medium and high risk. Fig. 5 shows that CAFE-VLA w/o Geometry and alternative geometry representations yield absent, fragmented, or disconnected responses to the obstacle, while either pathway alone produces incomplete or diffuse responses. The full CAFE-VLA model focuses on the end-effector–obstacle interaction region. Each map averages action-to-third-person-image attention over all flow-matching denoising steps for the same observation.

4.5 CONDITION INJECTION: CONTROLLABILITY AND CAPABILITY PRESERVATION

We separately ablate injection of the avoidance-strength scalar s and the geometry field G_t. For s, with field-token cross-attention fixed, we compare layer-wise FiLM (Perez et al., 2018), additive embedding (Ho et al., 2020), and an in-context token (Peebles & Xie, 2023). For G_t, with FiLM fixed, we compare sparse cross-attention, in-context concatenation (Peebles & Xie, 2023), attention pooling (Lee et al., 2019), and max pooling (Qi et al., 2017). Each variant is trained independently.

Scalar variants are swept over $s \in [0, 20]$, with direct conditioning applied for $s \in [0, 10]$ and identical CFG thereafter. Field variants are evaluated at $s = 8$ on risk-stratified SafeLIBERO and on original LIBERO for capability retention.

Avoidance-strength injection. Table 3 shows that FiLM provides the strongest monotonicity ($\rho = 0.953$), stable TSR over $s \in [0, 15]$, and monotonic clearance over $s \in [0, 18]$. For additive embedding, the corresponding ranges are $s \in [0, 12]$ for stable TSR and $s \in [0, 14]$ for monotonic clearance; for the in-context token, the avoidance-strength ranges are $s \in [0, 10]$ and $s \in [0, 12]$, respectively. Because all variants share CFG, layer-wise FiLM better preserves the strength signal and its guidance direction throughout the action expert than input-only conditioning.

Geometry-field injection. As shown in Table 4, compared with sparse field-token cross-attention, attention pooling reduces SafeLIBERO CAR by 13.33% and TSR by 8.55% because it collapses the spatially ordered probe features into a single vector. Max pooling discards more probe-level information, reducing CAR by 16.04% and TSR by 10.73% relative to sparse field-token cross-attention. In-context concatenation retains all probes and comparable avoidance, but reduces original-LIBERO TSR by 3.12% relative to sparse field-token cross-attention and adds 18.9% latency because 78 extra tokens enter every self-attention layer. Sparse field-token cross-attention instead preserves the original token sequence and injects separate probe keys and values through gated updates in selected blocks, balancing avoidance, capability retention, and overhead. Thus, FiLM matches the scalar condition, whereas sparse field-token cross-attention matches the field structure.

4.6 REAL-WORLD VALIDATION

Protocol. We evaluate the AgileX PiPER in the three contact-risk levels of Sec. 4.3. For training, 12 obstacles are approximated by cuboidal or cylindrical volumes and placed at ten sampled poses. An operator demonstrates the task once at low, medium, and high avoidance levels for each placement configuration, yielding 360 raw demonstrations, all of which are used for model training. Deployment evaluation uses six unseen obstacles.

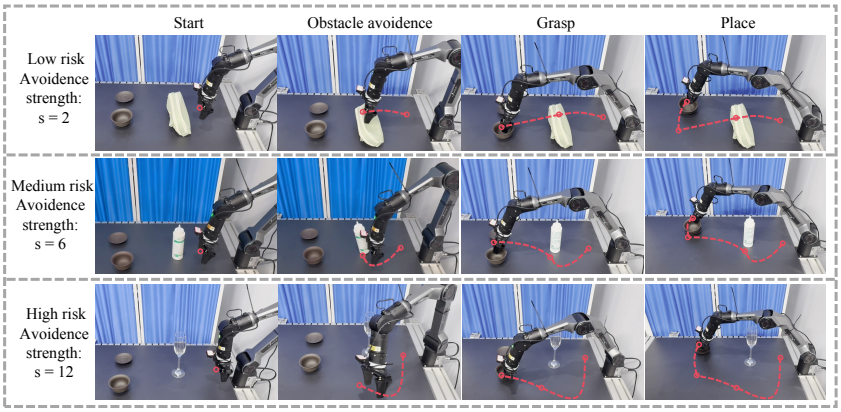

Figure 6: Real-world executions at low, medium, and high risk with predicted strengths $s = 2, 6, 12$. Columns show start, avoidance, grasp, and placement; dashed curves trace end-effector trajectories.

We compute end-effector poses by forward kinematics and combine them with each obstacle's measured initial pose and volume to obtain the signed clearance $d_\xi(t)$ at 30 Hz. Equation (2) provides avoidance-strength labels. Synchronized third-person depth constructs G_t, and SmolVLA is fine-tuned for 20,000 steps.

We compare NSG using its per-scene selected CBF parameter, CAFE-VLA-Fixed at $s = 8$, and adaptive CAFE-VLA. Thirty rollouts per method and risk level yield 270 trials. For NSG, CBF distances are computed from the measured poses and fitted volumes of the physical obstacles.

Table 5: Real-world results (30 rollouts per entry).

Method	Risk	CAR (%)	TSR (%)↑	ETS (steps)
NSG	Low	63.33	63.33	553.7
	Medium	76.67	56.67	571.4
	High	86.67	56.67	589.8
CAFE-VLA-Fixed	Low	83.33	60.00	564.2
	Medium	76.67	60.00	561.9
	High	80.00	50.00	577.6
CAFE-VLA	Low	26.67	**66.67**	**518.5**
	Medium	**83.33**	**63.33**	584.1
	High	**90.00**	**60.00**	601.7

Results. Table 5 summarizes the real-world results. Despite having fewer demonstrations than the simulation dataset and a narrower distribution of avoidance behaviors due to manual collection, CAFE-VLA preserves risk-adaptive behavior: CAR increases from 26.67% to 83.33% and 90.00% across risk levels, yielding a CAR span of 63.33%, compared with 23.34% for NSG. We also observe that CFG remains effective when s exceeds the training range. CAFE-VLA achieves the highest TSR at each level and improves mean TSR by 4.44% over NSG and 6.66% over CAFE-VLA-Fixed. Fig. 6 illustrates larger detours as predicted strength rises from $s = 2$ to 6 and 12.

5 CONCLUSION

We present CAFE-VLA, which enables compact VLAs to perform controllable obstacle avoidance through a continuous avoidance-strength scalar and an action-aligned, end-effector-centric 3D geometry field. FiLM conditions the action expert on the avoidance-strength scalar, while the cross-attention and visual-prompting pathways inject the field with less than 5% additional parameters. Experiments show monotonic strength–clearance control ($\rho = 0.953$), 74.32% average TSR on SafeLIBERO versus 70.34% for the strongest safety-aware baseline), and 82.54% TSR on LIBERO. Real-world results further demonstrate risk-adaptive avoidance on a PiPER manipulator.

Limitations include the quasi-static setting, reliance on calibrated RGB-D and an external VLM, end-effector-only probes, bounded CFG extrapolation, and single-robot, small-scale real-world evaluation. Future work will address dynamic obstacles, perception uncertainty, and full-body collisions with various robot embodiments and manipulation settings.

AI USE STATEMENT

OpenAI ChatGPT and Codex were used during manuscript preparation to assist with literature discovery, English-language editing, translation, terminology and notation consistency checks, LaTeX formatting, and compilation debugging. Codex also assisted with the implementation and debugging of the research code; all AI-assisted code was reviewed and tested by the authors. Generative AI was used to create the planar illustrations of the Franka robot end effector shown in Figs. 1 and 2; these illustrations are used only for schematic presentation and do not constitute experimental observations or results. Generative AI was not used to generate experimental data, determine evaluation results, or replace the authors' analysis. The authors independently verified the cited papers against their original publication sources and manually reviewed all AI-assisted text, equations, tables, references, code changes, and visual materials. All methodological decisions, experimental design, data collection, implementation, result interpretation, and scientific claims were made and validated by the authors, who take full responsibility for the final manuscript.

REFERENCES

Eitan Altman. *Constrained Markov Decision Processes*. CRC Press / Chapman & Hall, 1999.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. In *Proceedings of Robotics: Science and Systems (RSS)*, 2025.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

William English, Hao Zheng, and Rickard Ewetz. Neuro-symbolic safety guidance for Vision-Language-Action models via constrained flow matching. *arXiv preprint arXiv:2607.01378*, 2026. URL https://arxiv.org/abs/2607.01378.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. URL https://arxiv.org/abs/2207.12598.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 6840–6851, 2020.

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. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2026. Accepted; arXiv:2512.11891.

Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. VIMA: Robot manipulation with multimodal prompts. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pp. 14975–15022. PMLR, 2023.

Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research (IJRR)*, 5(1):90–98, 1986.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P. Foster, Pannag R. Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713, 2025. URL https://proceedings.mlr.press/v270/kim25c.html.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R. Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3744–3753. PMLR, 2019.

Chengmeng Li, Junjie Wen, Yaxin Peng, Yan Peng, and Yichen Zhu. PointVLA: Injecting the 3D world into vision-language-action models. *IEEE Robotics and Automation Letters*, 11(3): 2506–2513, 2026. doi: 10.1109/LRA.2026.3653303. arXiv:2503.07511.

Yixuan Li, Yuhui Chen, Zhengtao Zhang, Dongbin Zhao, Mingcai Zhou, and Haoran Li. QDepth-VLA: Quantized depth prediction as auxiliary supervision for vision-language-action models. *arXiv preprint arXiv:2510.14836*, 2025. URL https://arxiv.org/abs/2510.14836.

Puze Liu, Kuo Zhang, Davide Tateo, Snehal Jauhri, Zhiyuan Hu, Jan Peters, and Georgia Chalvatzaki. Safe reinforcement learning of dynamic high-dimensional robotic tasks: Navigation, manipulation, interaction. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9449–9456, 2023a. doi: 10.1109/ICRA48891.2023.10161548. URL https://ieeexplore.ieee.org/document/10161548.

Zuxin Liu, Zijian Guo, Yihang Yao, Zhepeng Cen, Wenhao Yu, Tingnan Zhang, and Ding Zhao. Constrained decision transformer for offline safe reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pp. 21611–21630. PMLR, 2023b.

Yuang Lu, Song Wang, Xiao Han, Xuri Zhang, Yucong Wu, and Zhicheng He. Enhancing diffusion policy with classifier-free guidance for temporal robotic tasks. *arXiv preprint arXiv:2510.09786*, 2025.

Jeremy A. Marvel and Rick Norcross. Implementing speed and separation monitoring in collaborative robot workcells. *Robotics and Computer-Integrated Manufacturing*, 44:144–155, 2017. doi: 10.1016/j.rcim.2016.08.001.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4195–4205, 2023.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 32, 2018.

Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 652–660, 2017.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, and Xuelong Li. SpatialVLA: Exploring spatial representations for visual-language-action model. In *Proceedings of Robotics: Science and Systems (RSS)*, 2025. arXiv:2501.15830.

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, Simon Alibert, Matthieu Cord, Thomas Wolf, and Remi Cadene. SmolVLA: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025.

Brijen Thananjeyan, Ashwin Balakrishna, Ugo Rosolia, Felix Li, Rowan McAllister, Joseph E. Gonzalez, Sergey Levine, Francesco Borrelli, and Ken Goldberg. Safety augmented value estimation from demonstrations (SAVED): Safe deep model-based RL for sparse cost robotic tasks. *IEEE Robotics and Automation Letters (RA-L)*, 5(2):3612–3619, 2020. doi: 10.1109/LRA.2020.2976272. arXiv:1905.13402.

Wei Xiao, Ramin Hasani, Xiao Li, and Daniela Rus. BarrierNet: Differentiable control barrier functions for learning of safe robot control. *IEEE Transactions on Robotics (T-RO)*, 39(3):2289–2305, 2023. arXiv:2111.11277.

Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3D diffusion policy: Generalizable visuomotor policy learning via simple 3D representations. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024. doi: 10.15607/RSS.2024.XX.067. arXiv:2403.03954.

Yanjie Ze, Zixuan Chen, Wenhao Wang, Tianyi Chen, Xialin He, Ying Yuan, Xue Bin Peng, and Jiajun Wu. Generalizable humanoid manipulation with 3D diffusion policies. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025. arXiv:2410.10803.

Borong Zhang, Yuhao Zhang, Jiaming Ji, Yingshan Lei, Yishuai Cai, Josef 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*, 2025a.

Qinglun Zhang, Zhen Liu, Haoqiang Fan, Guanghui Liu, Bing Zeng, and Shuaicheng Liu. FlowPolicy: Enabling fast and robust 3D flow-based policy via consistency flow matching for robot manipulation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2025b. arXiv:2412.04987.

Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3D-VLA: A 3D vision-language-action generative world model. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 61229–61245. PMLR, 2024.

Ruijie Zheng, Yongyuan Liang, Shuaiyi Huang, Jianfeng Gao, Hal Daumé III, Andrey Kolobov, Furong Huang, and Jianwei Yang. TraceVLA: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. *arXiv preprint arXiv:2412.10345*, 2024.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183, 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

A APPENDIX

This appendix provides implementation and evaluation details omitted from the main paper for space. It follows the notation and experimental organization of the main text. Appendix A.1 summarizes the end-to-end inference and training procedures; Appendix A.2 reports training and inference settings; Appendix A.3 gives expanded evaluation and system details; Appendix A.4 provides additional analyses and visual evidence; and Appendix A.5 discusses limitations.

Table 6: Notation used throughout the paper.

Symbol	Meaning	Symbol	Meaning
t	Control step	$I_t = \{I_t^{\text{third}}, I_t^{\text{wrist}}\}$	Third-person and wrist RGB observations
D_t	Aligned third-person depth image	$S_t, \mathbf{q}_t$	Proprioceptive state and its joint positions
L	Language instruction	A_t, H	Predicted action chunk and its length
$s, s_{\max}$	Avoidance strength and its training maximum	G_t	Ordered action-aligned 3D geometry field
$d_\xi(t), W_\xi$	Signed clearance and its interaction window	$\tilde{d}, d_{\text{lo}}, d_{\text{hi}}$	Soft-minimum clearance and its minimum–maximum bounds
P_t, r	Local obstacle point set and filtering radius	$\bar{\mathbf{x}}_i$	ith probe-template coordinate in the end-effector frame
$\aleph_t, \delta$	Remaining raw-action count and look-ahead step	$\mathbf{T}_{t,\delta}, \Delta\mathbf{T}_{t,\delta}$	End-effector pose and relative pose transform
$\mathbf{e}_{t,i,\delta}$	Transformed probe coordinate	$\mathbf{x}_{t,j}^{\text{EE}}$	Coordinate of probe j in the current end-effector frame
$d_{t,j}, \hat{\mathbf{u}}_{t,j}$	Probe distance and clearance direction	$\mathbf{f}_{t,j}, \mathbf{Z}_t$	Probe feature and encoded field-token sequence
k_{nn}	Number of nearest obstacle points queried per probe	d_{vis}	Maximum distance displayed in the visual prompt
$\boldsymbol{\gamma}(s), \boldsymbol{\beta}(s)$	FiLM scale and bias	$\mathbf{h}_\ell, \alpha_\ell$	Action-expert features and cross-attention residual gate at layer ℓ
$\widetilde{A}_\tau, \tau$	Noisy action chunk and flow-matching timestep	K	Number of flow-integration steps
v_θ, v_0	Current policy and frozen vanilla policy	$\mathcal{L}_{\text{FM}}, \mathcal{L}_{\text{preserve}}$	Flow-matching and capability-preservation losses
$\lambda_{\text{preserve}}$	Capability-preservation loss weight	$\text{sg}[\cdot]$	Stop-gradient operator
$\mathcal{Q}_t$	Temporally aggregated action queue	B_t, M_t	VLM obstacle boxes and SAM2 pixel mask
CAR, TSR, ETS	Collision avoidance rate, task success rate, and execution time steps	CFG	Classifier-free guidance

A.1 END-TO-END TRAINING AND INFERENCE PROCEDURES

We follow the formulation in Sec. 3. Algorithms 1 and 2 summarize the complete inference and training procedures, respectively. Component-specific implementation details and hyperparameters are provided in Appendix A.2.

A.1.1 END-TO-END INFERENCE PROCEDURE

Algorithm 1 details one asynchronous policy update. It uses the latest obstacle boxes and requested strength supplied by the external VLM, while SAM2 provides the current pixel mask. The 3D geometry field is reconstructed from the latest RGB-D frame.

A.1.2 TRAINING PROCEDURE

Algorithm 2 gives the complete optimization procedure. Trajectory-level labels are computed once before training; geometry and visual prompts are reconstructed from the sampled observation during training. Null conditioning and the clean preservation branch are sampled independently.

A.2 TRAINING AND INFERENCE CONFIGURATION

A.2.1 BASE POLICY AND OPTIMIZATION

We initialize from `lerobot/smolvla_base` and record the exact checkpoint commit with the released configuration. SmolVLA uses a SmolVLM2-500M-Video-Instruct backbone with a SigLIP-B/16 visual encoder and a 16-block, 720-dimensional flow-matching action expert. The SigLIP encoder and SmolVLM language backbone are frozen; the action expert and all newly added

Algorithm 1 Asynchronous CAFE-VLA inference update

Require: Policy v_θ; instruction L; aligned RGB image I_t^{third}, depth D_t, wrist image I_t^{wrist}, proprioception S_t; action queue $\mathcal{Q}_t$
Require: Most recent unaggregated joint-action chunk $\hat{A}_{t^-}^{\text{raw}}$ (initialized by repeating the current joint state)
Require: Latest VLM obstacle boxes B_t and requested strength s
Require: Camera calibration, probe template $\mathcal{X}$, local radius r, k_{nn}, training limit $s_{\max}$, and K flow steps
Ensure: Updated action queue $\mathcal{Q}_{t+1}$

Obtain the current pixel mask M_t from B_t with SAM2
$P_t \leftarrow \varnothing$
for each pixel (u, v) with $M_t(u, v) = 1$ and valid depth $z = D_t(u, v)$ **do**
  Back-project (u, v, z) and transform the point to the robot base frame
  if the point is within radius r of the current end-effector center **then**
    Insert the point into P_t
  end if
end for
Let h_t be the number of remaining actions in $\hat{A}_{t^-}^{\text{raw}}$ and $\Delta_t \leftarrow \{0, \lfloor \aleph_t/2 \rfloor, \aleph_t\}$
Use the measured joint state at $\delta = 0$ and forward kinematics on the raw joint-position targets at the midpoint and endpoint to obtain the poses
Transform the 26-probe template at these poses to form 78 ordered probes
for each probe $j \in \{1, \ldots, 78\}$ **do**
  Transform probe j to the base frame and query its k_{nn} neighbors in P_t
  Compute mean distance $d_{t,j}$ and normalized clearance direction $\hat{\mathbf{u}}_{t,j}$
  Store $\mathbf{f}_{t,j} = [\mathbf{x}_{t,j}^{\text{EE}}, d_{t,j}, \hat{\mathbf{u}}_{t,j}]$ using the prescribed fallback if no depth is valid
end for
$G_t \leftarrow (\mathbf{f}_{t,j})_{j=1}^{78}$ and $Z_t \leftarrow \phi_G(G_t)$
Project the probes onto I_t^{third}, apply the depth test, and render I_t^{prompt}
Sample $\widetilde{A}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $k = 0, \ldots, K-1$ **do**
  Set flow-matching timestep $\tau_k = k/K$
  if $0 \le s \le s_{\max}$ **then**
    $v_k \leftarrow v_\theta(\widetilde{A}_k, \tau_k \mid I_t^{\text{prompt}}, I_t^{\text{wrist}}, S_t, L, Z_t, s)$
  else
    Jointly evaluate $v_\varnothing$ and $v_{s_{\max}}$ under the same inputs and noisy action $\widetilde{A}_k$
    $v_k \leftarrow v_\varnothing + \frac{s}{s_{\max}}(v_{s_{\max}} - v_\varnothing)$
  end if
  $\widetilde{A}_{k+1} \leftarrow \widetilde{A}_k + K^{-1} v_k$
end for
Decode and denormalize $\widetilde{A}_K$ to obtain the H-step chunk $\hat{A}_t$
Store $\hat{A}_t$ as the raw chunk for the next look-ahead field
Temporally aggregate overlapping predictions and update $\mathcal{Q}_t$
return $\mathcal{Q}_{t+1}$

Algorithm 2 Training CAFE-VLA with strength and geometry conditioning

Require: Demonstrations $\mathcal{D} = \{\xi_i\}$; pretrained policy v_{θ_0}; maximum strength $s_{\max}$; null probability p_{null}; clean-branch probability p_{clean}; preservation weight $\lambda_{\text{preserve}}$

Ensure: Trained policy parameters θ

for each demonstration $\xi_i \in \mathcal{D}$ **do**
  Compute $d_{\min}(\xi_i)$, $t_{\text{coll}}(\xi_i)$, and $t_{\text{exec}}(\xi_i)$
  Assign $s_i \leftarrow s(\xi_i)$ using Eq. (2)
end for
Initialize θ from θ_0; initialize FiLM, ϕ_G, field-token attention, residual gates, and the learned null embedding
Freeze a copy of v_{θ_0} as the preservation teacher v_0
repeat
  Sample a minibatch of observations, actions A, instructions, and labels s_i from $\mathcal{D}$
  Sample $\tau \sim \mathcal{U}[0, 1]$ and noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  Form $\widetilde{A}_\tau = (1 - \tau)\epsilon + \tau A$ and target flow $u = A - \epsilon$
  Reconstruct G_t and the prompted third-person image for each sample
  Independently sample $b_{\text{null}} \sim \text{Bernoulli}(p_{\text{null}})$ and $b_{\text{clean}} \sim \text{Bernoulli}(p_{\text{clean}})$
  if $b_{\text{clean}} = 1$ **then**
    Use the unmodified image, null strength, and no geometry tokens
  else
    Use the prompted image and geometry tokens; replace s_i by the null embedding only if $b_{\text{null}} = 1$
  end if
  Predict $\hat{u} \leftarrow v_\theta(\widetilde{A}_\tau, \tau \mid \text{selected conditions})$
  $\mathcal{L}_{\text{FM}} \leftarrow \|\hat{u} - u\|_2^2$
  if $b_{\text{clean}} = 1$ **then**
    Evaluate $u_0 \leftarrow v_0(\widetilde{A}_\tau, \tau \mid I_t^{\text{third}}, I_t^{\text{wrist}}, S_t, L)$ under the same noisy action and flow-matching timestep
    $\mathcal{L}_{\text{preserve}} \leftarrow \|\hat{u} - \text{sg}[u_0]\|_2^2$
  else
    $\mathcal{L}_{\text{preserve}} \leftarrow 0$
  end if
  $\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_{\text{FM}} + \lambda_{\text{preserve}}\mathcal{L}_{\text{preserve}}$
  Update the action expert and conditioning modules using $\nabla_\theta \mathcal{L}_{\text{total}}$; keep the visual-language backbone and teacher frozen
until the prescribed number of optimization steps is reached
return θ

Table 7: Training configuration.

Setting	Value
Optimizer	AdamW; $\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon = 10^{-8}$
Peak learning rate	1.0×10^{-4}
Schedule	1,000-step linear warmup; cosine decay to 2.5×10^{-6} over 30,000 steps
Weight decay / gradient clipping	10^{-10} / global max-norm 10
Simulation batch	16 per GPU; 64 global on four GPUs; no accumulation
Simulation training	50,000 steps; three seeds $\{42, 3407, 2024\}$
Real-world fine-tuning	20,000 steps; batch 16 on one GPU
Checkpoint selection	Last checkpoint; no validation-based selection
Precision	bfloat16 mixed precision for training, inference, and teacher
Hardware	Four RTX 4090 GPUs (simulation); one RTX 4090 (real world)
Training time	Approximately 21 h (simulation) and 5.5 h (real world)
Trainable components	Action expert, FiLM, field encoder, and field-token cross-attention
Frozen components	SigLIP, SmolVLM language backbone, and preservation teacher
Added parameters	5.93M (1.32%): 1.4M FiLM and 4.53M geometry pathway
Null-condition probability	0.10
Clean-image probability	0.30
Preservation weight	$\lambda_{\text{preserve}} = 0.03$

conditioning modules are optimized. The frozen original SmolVLA serves as the teacher for capability preservation.

The 4.53M geometry-pathway parameters comprise the shared field encoder (0.37M), two independent 720-dimensional eight-head cross-attention modules (4.15M), and their channel-wise residual gates; visual prompting introduces no trainable parameters. The 1.4M FiLM parameters are reported separately because the geometry-injection ablation fixes FiLM across all of its variants.

A.2.2 Actions and asynchronous execution

Each prediction produces a joint-action chunk of length $H = 50$. The action expert transforms an initial noisy chunk into the prediction using ten uniform Euler steps over the flow-matching timestep $\tau \in [0, 1]$. Actions are normalized using the statistics stored with the training dataset and denormalized before execution.

The real robot executes at 30 Hz and simulation at 20 Hz. The policy runs asynchronously at approximately 8 Hz. A new prediction is requested when the remaining queue contains at most 15 actions. Overlapping predictions are temporally aggregated: predictions $\{\hat{a}_t^{(i)}\}$ covering time t are combined using $w_i = \exp(-0.05 m_i)$, where m_i is the offset of the prediction inside its action chunk. The executed action is $a_t = \sum_i w_i \hat{a}_t^{(i)} / \sum_i w_i$. Look-ahead geometry is constructed before this aggregation: it uses the current measured end-effector pose, while forward kinematics maps the midpoint and endpoint joint-position targets in the most recent raw predicted chunk to their end-effector poses. Flow initialization is $\widetilde{A}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$; subsequent integration is deterministic.

During training, we construct future probe poses from the demonstration future joint targets to match the queued-action construction used at deployment. To account for imperfect predicted action chunks, we perturb only the six arm-joint targets on 50% of training examples; gripper commands remain unchanged. For each chunk, we sample a shared offset $\mathbf{b} \sim \mathcal{N}(\mathbf{0}, 2.25 \times 10^{-4}\mathbf{I})$ rad and use

$$\tilde{\mathbf{q}}_{t+\delta} = \mathrm{clip}\big(\mathbf{q}_{t+\delta} + \mathbf{b} + \boldsymbol{\epsilon}_\delta, \mathbf{q}_{\min}, \mathbf{q}_{\max}\big), \qquad \boldsymbol{\epsilon}_\delta = 0.8\boldsymbol{\epsilon}_{\delta-1} + \boldsymbol{\eta}_\delta, \quad \boldsymbol{\eta}_\delta \sim \mathcal{N}(\mathbf{0}, 2.25 \times 10^{-4}\mathbf{I})\ \mathrm{rad}, \tag{7}$$

where clipping enforces the robot joint limits. The shared offset models chunk-level prediction bias, and the temporally correlated residual avoids unrealistic frame-to-frame jitter while exposing the field encoder to look-ahead uncertainty.

Table 8: Geometry-field and visual-prompting configuration.

Setting	Value
RGB-D stream	RealSense D435; 640×480 at 30 Hz
Valid depth range	0.15–1.50 m
Local radius / displayed distance	$r = d_{\text{vis}} = 0.20$ m
Radius-model quantities	$\Delta t_{\text{react}} = 0.10$ s, $v_{\max} = 0.30$ m/s, $a_{\text{brake}} = 1.5$ m/s^2, $l_{\text{EE}} = 0.06$ m
Probes	26 per pose; current, midpoint, endpoint; 78 total
End-effector OBB	0.16 m along approach, 0.10 m high; width equals gripper opening plus 0.04 m
Nearest neighbors	$k_{\text{nn}} = 8$ using PyTorch3D `knn_points`
Soft-minimum temperature	$\tau = 0.01$ m
Strength-label bounds	$d_{\text{lo}} \approx -0.015$ m; $d_{\text{hi}} \approx 0.14$ m
Real-world bounds	$d_{\text{lo}} \approx -0.02$ m; $d_{\text{hi}} \approx 0.15$ m
Label-error bound	≈ 3 mm, or ≈ 0.2 strength units
Probe-coordinate normalization	Divide end-effector-frame coordinates by r
Distance normalization	$\bar{d} = \text{clip}(d/d_{\text{vis}}, 0, 1)$
Field feature	7D: coordinate (3), normalized distance (1), direction (3)
Empty-neighborhood encoding	Missing distances set to d_{vis}; directions set to zero
Field encoder	Shared MLP $7 \to 512 \to 720$ with GELU
Cross-attention blocks	Action-expert blocks 8 and 9 (1-indexed)
Cross-attention	8 heads, head dimension 90; blocks do not share parameters
Residual gate	Channel-wise 720D vector initialized to zero
Attention dropout	0
Colormap	`RdYlGn`: near red, intermediate yellow, far green
Depth-test tolerance	0.02 m
Prompt marker	Circular, radius 9 px; alpha 0.65
Visual input	Resize prompted RGB to 256×256 and apply SigLIP normalization

A.2.3 Geometry-field configuration

The clearance-direction normalization uses $\epsilon = 10^{-8}$. When fewer than k_{nn} valid neighbors are available, missing entries use the maximum displayed distance and a zero direction. If no valid local depth is available, all distances and directions receive this fallback encoding while probe coordinates are retained.

A.2.4 Strength conditioning and preservation

For strength labeling, we query signed clearance at 30 Hz with $\tau = 0.01$ m in Eq. (2). FCL represents the gripper and wrist with two capsules and one box, and each obstacle with its measured initial cuboidal or cylindrical volume. The measured 2.0 mm reprojection error and $0.15°$ hand–eye error bound the induced label error by approximately 3 mm, or 0.2 strength units. When W_ξ is empty, we use the learned null strength embedding rather than assigning a scalar strength label.

The scalar is divided by $s_{\max}$ and encoded with a 32-dimensional sinusoidal embedding. A shared $32 \to 64 \to 64$ GELU MLP feeds 16 layer-specific linear heads, each producing the 720D FiLM scale and bias. Head weights are zero-initialized, with scale bias one and offset bias zero. A 32-dimensional learnable null embedding is initialized from $\mathcal{N}(\mathbf{0}, 4 \times 10^{-4}\mathbf{I})$ and is used on 10% of training samples.

The clean branch is sampled independently per example with probability 0.30. It uses the unmodified third-person image, the null strength, and no geometry conditioning. The frozen teacher is evaluated online under `torch.no_grad` in bfloat16. The preservation loss is averaged over action tokens, flow-matching timestep, and action dimensions. Null and maximum-strength CFG evaluations are combined into one batch of size two. VLM strengths are clamped to $[0, 15]$ and rounded to increments of 0.5; parsing failure falls back to $s = 8$.

A.2.5 RGB-D PROCESSING AND EXTERNAL PERCEPTION

We align depth to color with `librealsense2`'s `rs2::align(RS2_STREAM_COLOR)`. Depth is converted from millimeters to meters, filtered with the SDK spatial, temporal, and hole-filling filters, and clipped to 0.15–1.50 m. The released calibration file stores the intrinsics of each device. Camera-to-base calibration uses a ChArUco board and Tsai–Lenz hand–eye calibration, with measured reprojection error below 2.0 mm and $0.15°$.

The obstacle mask contains approximately 9,800 valid depth points on average (2,000–25,000), and radius filtering retains approximately 1,600 (120–4,100). Sets larger than 4,096 points are reduced by farthest-point sampling.

Qwen3-VL-Flash (API model ID: `qwen3-vl-flash`; temperature zero) identifies task-relevant obstacles, selects the requested strength, and returns bounding boxes. We use SAM2.1 Hiera Small for mask tracking. SAM2 mask tracking runs asynchronously with a per-frame latency of 15.2 ± 2.1 ms. Since this is below the camera-frame interval of 33.3 ms, masks are updated at the camera rate of 30 Hz. The tracker uses the VLM boxes to produce pixel-aligned masks, and each policy update consumes the latest available mask. Masks from multiple obstacles are unioned and regions smaller than 100 pixels are removed. Tracking failure triggers reinitialization from the most recent boxes. If the VLM fails, all non-table depth points within r are treated as obstacles and the strength is set to $s = 8$.

A.2.6 VLM PROMPT AND OUTPUT PARSING

The VLM receives the current third-person RGB image together with the task instruction and infers the avoidance strength from visible scene evidence. We use the following prompt template; bracketed fields are populated at runtime.

System message.

```
You are a safety reasoning assistant for a tabletop manipulation robot.
Given one RGB observation and a manipulation instruction, identify
task-relevant obstacles and select an obstacle-avoidance strength.
Base the decision only on visible evidence and the instruction.
Return valid JSON only, with no
Markdown or additional text.
```

User message template.

```
Task instruction:
[TASK_INSTRUCTION]

Image size:
width = [IMAGE_WIDTH] pixels, height = [IMAGE_HEIGHT] pixels

Analyze the attached third-person RGB image for the specified task.

1. Identify every visible object that can obstruct the robot end
   effector or its task-relevant motion. Do not mark the robot,
   gripper, table, background, manipulation target, or destination
   as an obstacle.

2. For every task-relevant obstacle, output a tight bounding box
   [x_min, y_min, x_max, y_max] in image-pixel coordinates. Set
   "avoid" to true only when the object should constrain the motion.

3. Select an avoidance strength s in [0, 15] from the visible spatial
   relation and object properties in the image.
   Larger s requests greater clearance but can cause longer detours
   or task failure. Consider obstacle proximity to the likely motion,
   visible stability or fragility cues, and available free space. Prefer
   the smallest value consistent with the visible risk and spatial
   constraints. Values above 10 invoke
```

```
   classifier-free-guidance extrapolation and should be used only
   when necessary. Never exceed 15. Use increments of 0.5.

4. If multiple obstacles are present, include all relevant boxes and
   choose the strength required by the most restrictive task-relevant
   obstacle. If no relevant obstacle is visible, return an empty
   obstacle list and strength 0.

5. Provide one concise assessment of at most 40 words. State the visible
   obstacle relation and visual basis for the requested clearance; do not provide
   hidden chain-of-thought or unsupported physical properties.

Return exactly this JSON schema:
{
  "obstacles": [
    {
      "label": "short object name",
      "avoid": true,
      "bbox": [x_min, y_min, x_max, y_max]
    }
  ],
  "strength": 0.0,
  "assessment": "one concise evidence-based sentence"
}
```

We reject responses with invalid JSON, coordinates outside the image, or a nonnumeric strength. A single repair request is then issued:

```
Your previous response did not satisfy the required schema.
Return JSON only. Preserve the same scene assessment, clip every
bounding-box coordinate to the image bounds, and use a numeric
strength in [0, 15] at increments of 0.5.
Do not add new objects.
```

After parsing, bounding boxes initialize SAM2 and multiple masks are unioned. The strength is clipped to $[0, 15]$ and rounded to the nearest 0.5. If both the initial response and repair fail, all non-table depth points within r are treated as obstacles and the conditioning strength is set to $s = 8$. The exact API version and access date are stored with the released configuration. Predicted strength has mean absolute error 0.8 relative to the oracle, and 78% of predictions are within one strength unit.

A.3 Expanded Experimental Protocol and System Details

We follow the evaluation protocol in Sec. 4 and provide additional dataset, implementation, aggregation, and system details below.

A.3.1 SafeLIBERO data and aggregation

SafeLIBERO contains four tasks from each of LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-Long. Each task has two obstacle scenarios: one places an obstacle along the start–goal path, and the other places it near the grasp or placement region. We collect 50 avoidance demonstrations per task–scenario pair, for 1,600 demonstrations. Training and evaluation obstacle poses are disjoint. Demonstrations are generated by adding obstacle-side waypoints to the original LIBERO trajectories, smoothing and retiming the path, and resampling trajectories that miss their target clearance band. We discard trajectories that fail the task, violate a high-risk contact constraint, or contain an emergency-stop or reset segment.

Each risk-stratified simulation table entry aggregates 16 tasks, two obstacle scenarios per task, ten episodes per task–scenario pair, and three independently trained seeds, giving 960 rollouts. Tables aggregated across the three risk levels use 2,880 rollouts. We first average rollouts within a task–scenario pair, then scenarios within a task, and finally tasks within a risk level. Original-LIBERO retention uses all 40 tasks, 20 episodes per task, and the same three seeds.

Table 9: Per-seed variability for Table 1.

Method	Risk	CAR (%)	TSR (%)	ETS
CAFE-VLA	Low	20.73 ± 1.76	77.50 ± 2.09	248 ± 6.5
CAFE-VLA	Medium	81.25 ± 1.51	73.96 ± 1.77	302 ± 8.1
CAFE-VLA	High	85.00 ± 1.36	71.46 ± 2.11	323 ± 9.0
CAFE-VLA-Fixed	Low	79.90 ± 1.29	72.40 ± 1.52	287 ± 6.6
CAFE-VLA-Fixed	Medium	80.73 ± 1.43	71.15 ± 1.72	290 ± 6.0
CAFE-VLA-Fixed	High	80.10 ± 1.89	62.29 ± 1.87	289 ± 7.9
CAFE-VLA-Oracle	Low	21.56 ± 1.28	78.75 ± 1.9	250 ± 5.3
CAFE-VLA-Oracle	Medium	82.50 ± 1.61	75.31 ± 1.75	306 ± 6.8
CAFE-VLA-Oracle	High	86.35 ± 1.38	72.71 ± 2.05	327 ± 7.8
AEGIS	Low	55.63 ± 1.66	73.23 ± 1.95	259 ± 6.0
AEGIS	Medium	72.81 ± 1.37	69.38 ± 1.54	271 ± 6.4
AEGIS	High	75.94 ± 1.55	57.50 ± 2.19	286 ± 8.5
NSG	Low	60.21 ± 2.27	72.81 ± 1.98	276 ± 8.2
NSG	Medium	78.44 ± 1.57	70.94 ± 1.87	284 ± 7.0
NSG	High	82.71 ± 1.45	67.29 ± 2.27	307 ± 6.6
SmolVLA-Base	Low	16.35 ± 1.41	80.63 ± 1.95	244 ± 5.1
SmolVLA-Base	Medium	16.77 ± 1.71	47.40 ± 1.78	248 ± 6.5
SmolVLA-Base	High	16.88 ± 1.51	9.58 ± 1.23	251 ± 6.4

Low-risk obstacles permit contact, although task failure still counts against TSR. Medium-risk contact is permitted unless obstacle displacement makes the task fail. At high risk, any obstacle contact is prohibited; simulation contact is registered when collision bodies are within 1 mm for at least one control step. Episodes terminate on success, failure, or a horizon of 600 simulation steps (20 Hz) or 900 real-world steps (30 Hz). Minimum clearance is computed from the full fitted collision geometry, rather than from probe distances.

The strength sweep uses 20 rollouts at every integer strength. Spearman's test uses the mean clearance at the 19 levels $s = 0, \ldots, 18$ and a two-sided test. This is the only hypothesis test, so no multiple-comparison correction is applied.

A.3.2 Per-Seed Variability

We report mean $\pm$ standard deviation over three independently trained seeds. Main-paper tables report means only.

A.3.3 Baseline and ablation implementation

AEGIS uses $h(x) = d_{\min} - d_{\text{margin}}$ with a 0.03 m margin and an OSQP filter. Its gain is selected per scene from $\{0.1, 0.3, 0.5, 1, 2, 5\}$. NSG applies CBF guidance at all ten flow-integration steps, uses a 0.02 m margin, and selects its decay rate per scene from $\{0.1, 0.5, 1, 2, 5\}$. CAFE-VLA-Fixed uses $s = 8$. CAFE-VLA-Oracle searches integer strengths $\{0, \ldots, 15\}$ and breaks ties by higher CAR and then lower ETS. For AEGIS and NSG, we first fine-tune SmolVLA on the same avoidance demonstrations used by CAFE-VLA, without the strength or geometry injection modules, and then apply the respective CBF mechanism at inference.

A.3.4 Rule-mapped strength baseline

To separate the value of VLM-based scene interpretation from a trivial risk-stratum heuristic, we additionally evaluate CAFE-VLA with a fixed rule mapping from the evaluation contact-risk label to the avoidance strength: low risk maps to $s = 2$, medium risk to $s = 6$, and high risk to $s = 12$. This baseline is given the risk stratum specified by the evaluation protocol, whereas the deployable CAFE-

Table 10: Per-seed variability for Table 2.

Representation	Risk	CAR (%)	TSR (%)	ETS
CAFE-VLA	Low	20.73 ± 1.76	77.50 ± 2.09	248 ± 6.5
CAFE-VLA	Medium	81.25 ± 1.51	73.96 ± 1.77	302 ± 8.1
CAFE-VLA	High	85.00 ± 1.36	71.46 ± 2.11	323 ± 9.0
Field-token pathway only	Low	14.48 ± 1.39	64.38 ± 1.92	275 ± 7.9
Field-token pathway only	Medium	68.75 ± 1.48	61.46 ± 1.83	316 ± 9.0
Field-token pathway only	High	79.38 ± 1.63	58.33 ± 2.35	356 ± 7.2
Visual-prompting pathway only	Low	20.83 ± 1.64	77.29 ± 1.72	254 ± 5.2
Visual-prompting pathway only	Medium	76.46 ± 1.41	68.33 ± 2.29	298 ± 8.3
Visual-prompting pathway only	High	74.69 ± 1.9	62.81 ± 2.36	355 ± 9.0
PointVLA	Low	16.77 ± 1.71	71.25 ± 1.51	266 ± 5.5
PointVLA	Medium	77.40 ± 1.31	67.81 ± 2.05	296 ± 7.5
PointVLA	High	81.67 ± 1.24	64.17 ± 1.66	343 ± 9.7
QDepth-VLA	Low	18.44 ± 1.47	71.25 ± 1.51	259 ± 6.0
QDepth-VLA	Medium	72.19 ± 2.23	63.44 ± 1.8	305 ± 6.3
QDepth-VLA	High	74.58 ± 1.95	61.35 ± 2.31	359 ± 10.4
CAFE-VLA w/o Geometry	Low	16.15 ± 1.46	79.79 ± 1.54	252 ± 6.2
CAFE-VLA w/o Geometry	Medium	54.69 ± 1.91	48.96 ± 1.81	327 ± 7.8
CAFE-VLA w/o Geometry	High	58.33 ± 1.52	45.63 ± 2.16	376 ± 11.2

Table 11: Per-seed variability for Table 4.

Geometry injection	Safe CAR (%)	Safe TSR (%)	LIBERO TSR (%)
Sparse field-token cross-attention	80.21 ± 1.65	68.65 ± 2.23	82.54 ± 0.92
In-context concatenation	81.04 ± 1.4	67.40 ± 2.02	79.42 ± 0.96
Attention pooling	66.88 ± 1.73	60.10 ± 2.1	82.42 ± 1.14
Max pooling	64.17 ± 1.53	57.92 ± 2.35	82.00 ± 0.91

VLA selector receives the RGB observation and task instruction. Table 12 reports the corresponding comparison under the same 960-rollout SafeLIBERO protocol used for the main results.

In simulation, AEGIS and NSG use exact simulator obstacle geometry to compute their CBF distances, while CAFE-VLA uses only the point cloud reconstructed from RGB-D observations. In the real-world comparison, NSG computes distances from measured obstacle poses and fitted cuboidal or cylindrical volumes.

QDepth-VLA predicts 256 uniformly spaced depth bins over 0.15–1.50 m with an auxiliary cross-entropy loss weighted by 0.1. PointVLA samples 4,096 global points with farthest-point sampling and uses a three-stage PointNet++-style encoder to produce 64 prefix tokens. The in-context geometry variant inserts all 78 field tokens with learned positional embeddings into every self-attention layer.

Table 12: VLM-based scene-adaptive strength selection versus a trivial risk-stratum rule mapping on SafeLIBERO. The rule uses $s = 2$, 6, and 12 for low-, medium-, and high-risk scenes, respectively. CAR/TSR are in %; ETS is in steps.

Selector	Risk	CAR (%) ↑	TSR (%) ↑	ETS ↓
VLM-based	Low	20.73 ± 1.76	77.50 ± 2.09	248 ± 6.5
	Medium	81.25 ± 1.51	73.96 ± 1.77	302 ± 8.1
	High	85.00 ± 1.36	71.46 ± 2.11	323 ± 9.0
Rule mapping	Low ($s = 2$)	28.23 ± 1.6	75.73 ± 2.15	261 ± 5.3
	Medium ($s = 6$)	78.13 ± 1.56	73.33 ± 1.59	307 ± 6.6
	High ($s = 12$)	84.17 ± 1.32	69.69 ± 1.51	319 ± 9.6

Table 13: Runtime on an RTX 4090 and Intel Core i9-13900K. Local components are measured over 1,000 repetitions and reported as mean ± standard deviation.

Component	Latency
Obstacle VLM API (asynchronous)	≈ 1.2 s
SAM2 mask tracking (asynchronous)	15.2 ± 2.1 ms/frame
Depth alignment, back-projection, and filtering	6.1 ± 0.8 ms
Probe KNN and field construction	4.0 ± 0.3 ms
Visual prompting	2.2 ± 0.4 ms
Policy, FiLM + 3D geometry conditioning	64.9 ms
Policy, batched two-branch CFG	118 ± 8 ms
End-to-end, direct / CFG	$\approx$ 77 / 130 ms

Attention pooling uses four learned queries and eight heads. Max pooling reduces the 78 probe features to one 720D vector. The additive scalar baseline injects a 720D embedding at every action-expert block, while the in-context scalar baseline prepends one strength token.

A.3.5 Real-world data collection

The AgileX PiPER is a six-DoF arm with a two-finger gripper. A RealSense D435 is mounted approximately 1.0 m above the table and 0.7 m horizontally from the base, pitched 45° downward. A 640×480 wrist RGB camera is pitched 30° downward. Demonstrations are collected through leader–follower teleoperation at 30 Hz.

Data collection uses 12 obstacles, four per risk level, each at ten sampled poses in a 40×40 cm workspace. One low-, medium-, and high-avoidance demonstration is recorded per pose. This yields 360 raw demonstrations; after the trajectory-quality filtering above, 300 are used for training. Deployment uses six held-out obstacles, two per risk level, at unseen poses. Evaluation contains 30 rollouts for every method–risk pair, or 270 trials in total.

Robot, camera, and action streams use host-clock timestamps with measured synchronization error below 5 ms. Software limits end-effector speed to 0.30 m/s and enforces workspace walls; any emergency stop or human intervention counts as a failed rollout. Object and pose order is randomized, and the operator interface hides the evaluated method.

A.3.6 Runtime and resource use

The VLM API and SAM2 tracking run asynchronously and are therefore outside the serialized policy-update path. End-to-end p95 latency is 100 ms for direct conditioning, which defines Δt_{react}. Inference uses 9.8 GB GPU memory and training peaks at 21.4 GB. FiLM adds 1.2% latency relative to SmolVLA, and the geometry pathway adds 5.1% relative to the FiLM-conditioned policy. Thus, the full CAFE-VLA policy requires $61 \times 1.012 \times 1.051 \approx 64.9$ ms per forward pass, compared with 61 ms for the SmolVLA baseline; the batched CFG path costs approximately $1.93\times$ the baseline.

A.3.7 Artifacts and reproducibility

The anonymized release contains configuration files, all three simulation seeds, simulation and real-world checkpoints, SafeLIBERO task definitions, and HDF5 demonstrations containing observations, actions, masks, and strength labels. Per-rollout CSV files record method, risk, task, obstacle, placement, predicted strength, collision and duration, task success, clearance, ETS, and failure category. Code is released under MIT and demonstrations under CC BY-NC 4.0. The release also records the exact SmolVLA, Qwen3-VL-Flash, and SAM2 versions and the VLM access date.

A.4 Additional Evidence and Analyses

This section provides additional implementation visualizations, benchmark details, quantitative analyses, and qualitative evidence not included in the main paper. To avoid duplication, no main-paper figures or result tables are repeated here.

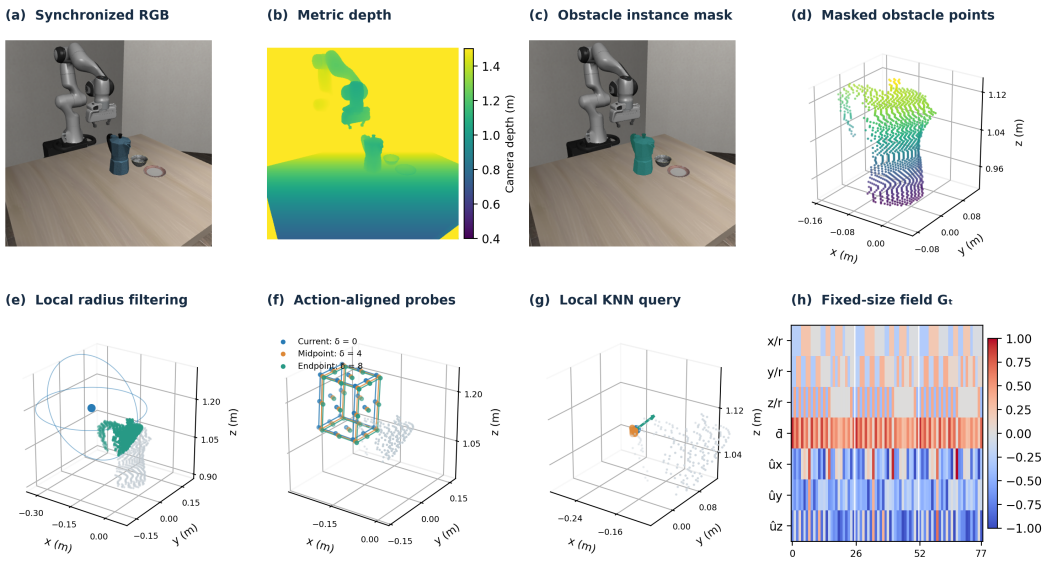

Figure 7: Construction of the local 3D geometry field. Synchronized RGB and metric depth yield an obstacle instance mask and masked 3D points. Radius filtering retains local obstacle geometry; the current, queue-midpoint, and queue-endpoint probes query this geometry through KNN to form the fixed-size field G_t.

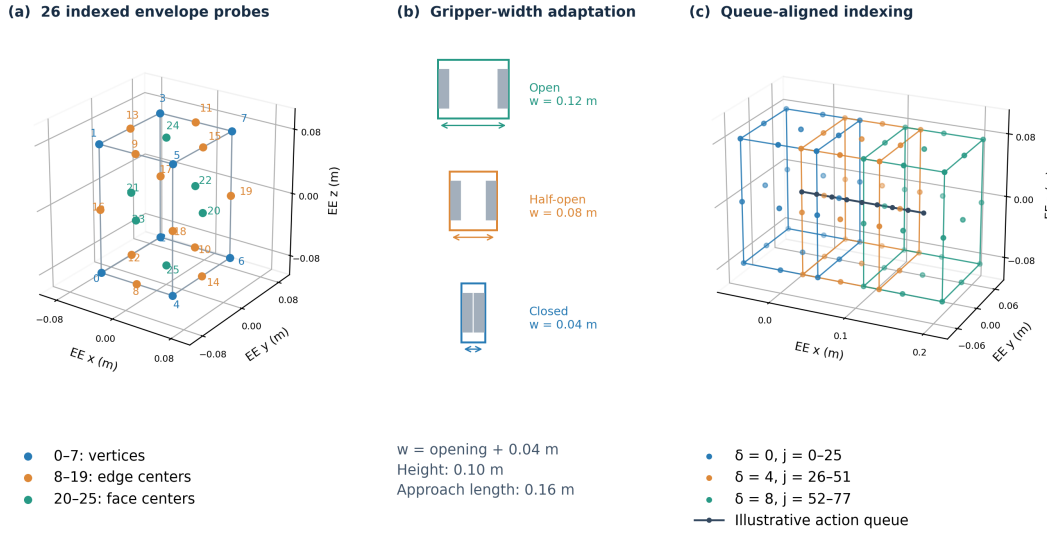

Figure 8: Probe layout and action-aligned look-ahead construction. Each end-effector envelope contains 8 vertex, 12 edge-center, and 6 face-center probes. Its width follows the gripper opening. Replicating the ordered 26-probe template at the current, queue-midpoint, and queue-endpoint poses produces 78 probes with index $j = 26k + i$. The queue in (c) is illustrative.

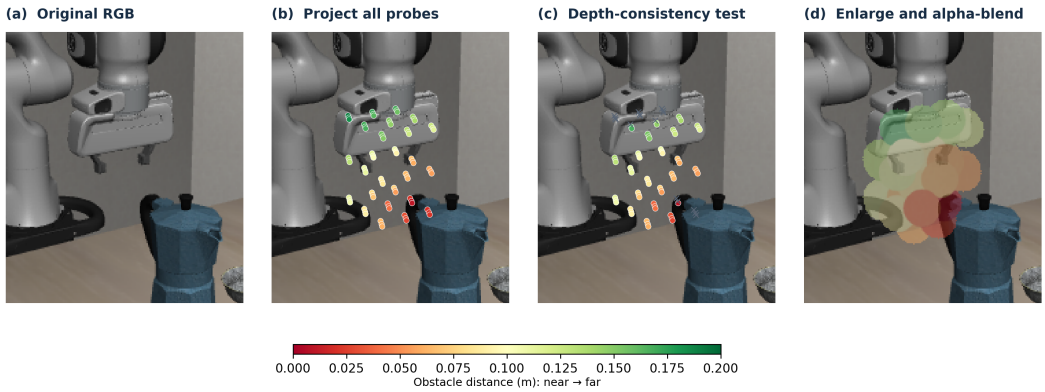

Figure 9: Visual-prompting construction. We project all probes onto the original third-person RGB image, remove depth-inconsistent projections, and enlarge and alpha-blend the remaining regions. Color encodes obstacle distance from near (red) to far (green) over the displayed range.

A.4.1 Geometry-field construction

A.4.2 Dataset composition and evaluation coverage

Table 14: SafeLIBERO task and obstacle-scenario inventory. SafeLIBERO uses one obstacle from its released object pool (moka pot, storage box, milk carton, wine bottle, mug, or book). Level I places it near the target object, whereas Level II places it farther from the target but across the nominal motion path. The exact object identity and randomized pose are defined by the released BDDL and `.pruned_init` files. In our evaluation, each task–scenario pair is run for ten episodes per seed; SafeLIBERO itself does not define a training split.

Suite	Task / instruction	Obstacle scenarios	Coverage
Spatial	Pick up the black bowl between the plate and the ramekin and place it on the plate	I: near the target; II: obstructing the motion path	2×10/seed
Spatial	Pick up the black bowl on the ramekin and place it on the plate	I: near the target; II: obstructing the motion path	2×10/seed
Spatial	Pick up the black bowl on the stove and place it on the plate	I: near the target; II: obstructing the motion path	2×10/seed
Spatial	Pick up the black bowl on the wooden cabinet and place it on the plate	I: near the target; II: obstructing the motion path	2×10/seed
Goal	Put the bowl on the plate	I: near the target; II: obstructing the motion path	2×10/seed
Goal	Put the bowl on top of the cabinet	I: near the target; II: obstructing the motion path	2×10/seed
Goal	Put the bowl on the stove	I: near the target; II: obstructing the motion path	2×10/seed
Goal	Open the top drawer and put the bowl inside / put the cream cheese in the bowl	I: drawer-and-bowl task; II: cream-cheese-and-bowl task	2×10/seed
Object	Pick up the orange juice and place it in the basket	I: near the target; II: obstructing the motion path	2×10/seed
Object	Pick up the chocolate pudding and place it in the basket	I: near the target; II: obstructing the motion path	2×10/seed
Object	Pick up the milk and place it in the basket	I: near the target; II: obstructing the motion path	2×10/seed
Object	Pick up the bbq sauce and place it in the basket	I: near the target; II: obstructing the motion path	2×10/seed
Long	Put both the alphabet soup and the cream cheese box in the basket	I: near the target; II: obstructing the motion path	2×10/seed
Long	Put both the alphabet soup and the tomato sauce in the basket	I: near the target; II: obstructing the motion path	2×10/seed
Long	Put the white mug on the left plate and the yellow-and-white mug on the right plate	I: near the target; II: obstructing the motion path	2×10/seed
Long	Put the white mug on the plate and the chocolate pudding to the right of the plate	I: near the target; II: obstructing the motion path	2×10/seed



Figure 10: Real-world obstacle inventory grouped by contact risk. High-risk objects are fragile or safety-critical, medium-risk objects can be displaced and interfere with task completion, and low-risk objects permit incidental contact. The collection/deployment split is specified separately in Table 15.

Table 15: Real-world obstacle split shown in Fig. 10. The first four objects in each row were used for data collection and the last two were held out for deployment. Object names, materials, and sizes follow the collection records; "approx." denotes an approximate physical dimension recorded for reproducibility.

Split	Object	Risk	Dimensions / material	Collision primitive	Pose range
Collection	Open-top container	High	Metal; approx. 10 cm dia. $\times$14 cm	Cylinder	10 poses; 40 $\times$ 40 cm, full yaw
Collection	Metal desk tool	High	Metal; approx. 15 $\times$ 6 $\times$ 4 cm	Cuboid	10 poses; 40 $\times$ 40 cm, full yaw
Collection	Glass mug	High	Glass; approx. 8 cm dia. $\times$10 cm	Cylinder	10 poses; 40 $\times$ 40 cm, full yaw
Collection	Lidded goblet	High	Glass/metal; approx. 9 cm dia. $\times$16 cm	Cylinder	10 poses; 40 $\times$ 40 cm, full yaw
Deployment	Stemmed glass	High	Glass; approx. 7 cm dia. $\times$18 cm	Cylinder	Unseen poses in the same workspace
Deployment	Lithium-ion battery	High	Metal/plastic; approx. 20 $\times$ 14 $\times$ 10 cm	Cuboid	Unseen poses in the same workspace
Collection	Phone stand	Medium	Plastic; approx. 12 $\times$ 12 $\times$ 15 cm	Cuboid	10 poses; 40 $\times$ 40 cm, full yaw
Collection	Sealed tea tin	Medium	Metal; approx. 10 cm dia. $\times$12 cm	Cylinder	10 poses; 40 $\times$ 40 cm, full yaw
Collection	Tape roll	Medium	Plastic/paper; approx. 12 cm dia. $\times$5 cm	Cylinder	10 poses; 40 $\times$ 40 cm, full yaw
Collection	Plastic bottle	Medium	Plastic; approx. 7 cm dia. $\times$20 cm	Cylinder	10 poses; 40 $\times$ 40 cm, full yaw
Deployment	White carton	Medium	Paperboard; approx. 10 $\times$ 7 $\times$ 15 cm	Cuboid	Unseen poses in the same workspace
Deployment	Squeeze bottle	Medium	Plastic; approx. 7 cm dia. $\times$18 cm	Cylinder	Unseen poses in the same workspace
Collection	Tissue pack	Low	Soft plastic/paper; approx. 22 $\times$ 11 $\times$ 8 cm	Cuboid	10 poses; 40 $\times$ 40 cm, full yaw
Collection	Soft pad pack	Low	Foam/fabric; approx. 20 $\times$ 15 $\times$ 8 cm	Cuboid	10 poses; 40 $\times$ 40 cm, full yaw
Collection	Paper bag	Low	Paper; approx. 25 $\times$ 20 $\times$ 8 cm	Cuboid	10 poses; 40 $\times$ 40 cm, full yaw
Collection	Coiled cord	Low	Flexible cord; approx. 12 cm dia. $\times$4 cm	Cylinder	10 poses; 40 $\times$ 40 cm, full yaw
Deployment	Plush ball toy	Low	Plush fabric; approx. 10 cm dia. $\times$8 cm	Cylinder	Unseen poses in the same workspace
Deployment	Plush rabbit	Low	Plush fabric; approx. 12 $\times$ 10 $\times$ 20 cm	Cuboid	Unseen poses in the same workspace

A.4.3 ADDITIONAL QUANTITATIVE ANALYSES

We compare three variants. *SmolVLA base* is the unmodified pretrained policy. *No preservation branch* is CAFE-VLA trained with avoidance-strength and 3D geometry conditioning but without unmodified preservation samples or teacher matching. *Full objective* is the complete CAFE-VLA training objective: 30% of samples use the unmodified third-person image with strength and geometry conditioning disabled, and the policy prediction is matched to the frozen SmolVLA teacher using $\lambda_{\text{preserve}} = 0.03$.

Table 16: Original-LIBERO capability-retention TSR (%) by suite under the same evaluation protocol. Best adapted-policy results are highlighted in bold.

Variant	Spatial	Object	Goal	Long	Mean
SmolVLA base	88.00 ± 1.30	90.33 ± 1.79	84.83 ± 2.09	68.50 ± 2.20	82.92 ± 1.62
No preservation branch	86.50 ± 1.49	88.33 ± 1.75	82.17 ± 1.92	63.33 ± 2.07	80.08 ± 1.58
Full objective	$\mathbf{87.50} \pm 1.38$	$\mathbf{90.00} \pm 1.43$	$\mathbf{84.67} \pm 2.25$	$\mathbf{68.00} \pm 2.09$	$\mathbf{82.54} \pm 1.55$

Table 17: Sensitivity of the 3D geometry field and soft-minimum temperature. CAR and TSR are measured on SafeLIBERO; retention is the original-LIBERO TSR; calibration error is the signed-clearance error; and latency is single-policy inference time. Default settings are highlighted in bold.

Factor	Setting	CAR (%)	TSR (%)	Retention (%)	Calib. err. (cm)	Latency
Soft-min. τ	0.005 m	78.92	67.53	—	0.92	64.9 ms
	0.01 m	**80.24**	**68.65**	—	**0.71**	**64.9 ms**
	0.02 m	79.38	68.06	—	1.14	64.9 ms
Filtering radius r	0.15 m	75.49	66.22	82.63	—	64.7 ms
	0.20 m	**80.24**	**68.65**	**82.54**	—	**64.9 ms**
	0.25 m	80.83	67.57	82.29	—	65.8 ms
Neighbors k_{nn}	4	76.53	66.04	82.42	—	64.8 ms
	8	**80.24**	**68.65**	**82.54**	—	**64.9 ms**
	16	78.06	67.47	82.50	—	65.1 ms
Probe poses	Current	72.53	65.03	82.58	—	64.6 ms
	Current + end-point	77.47	67.05	82.50	—	64.8 ms
	All three	**80.24**	**68.65**	**82.54**	—	**64.9 ms**
Probes per pose	14	76.04	66.49	82.71	—	64.7 ms
	26	**80.24**	**68.65**	**82.54**	—	**64.9 ms**
	50	80.63	68.44	82.21	—	66.4 ms
Injection blocks	6–7	78.54	66.98	82.50	—	64.9 ms
	8–9	**80.24**	**68.65**	**82.54**	—	**64.9 ms**
	10–11	79.03	67.81	82.29	—	64.9 ms
Prompt radius	5 px	76.49	66.81	82.83	—	64.9 ms
	9 px	**80.24**	**68.65**	**82.54**	—	**64.9 ms**
	13 px	80.80	68.19	81.54	—	65.0 ms
Prompt alpha	0.40	78.02	67.47	82.75	—	64.9 ms
	0.65	**80.24**	**68.65**	**82.54**	—	**64.9 ms**
	0.80	80.38	68.02	81.83	—	64.9 ms

The default configuration provides the strongest overall CAR–TSR trade-off. Reducing r, k_{nn}, the number of probe poses, or probe density removes local or look-ahead geometry and lowers avoidance performance. Larger neighborhoods and denser fields provide little additional CAR while increasing latency or reducing TSR. Small or faint visual markers weaken the image-aligned cue, whereas larger or more opaque markers obscure pretrained visual features and reduce capability retention.

A.4.4 QUALITATIVE RESULTS

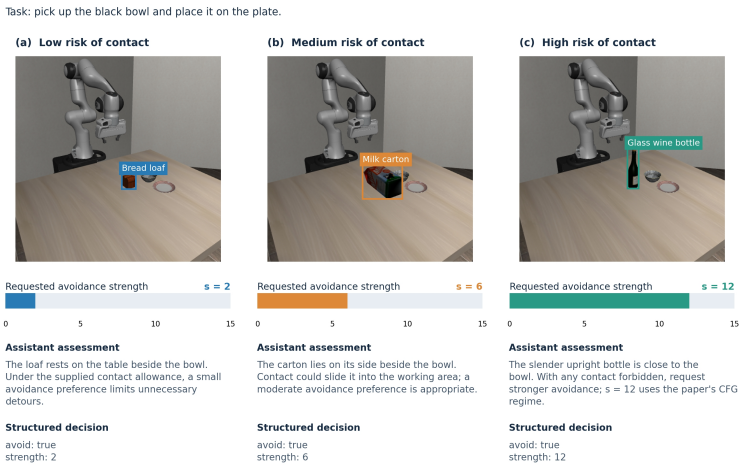

Figure 11: Illustrative scene-adaptive strength recommendations under different visible obstacle properties, obstacle–motion relationships, and available free space.

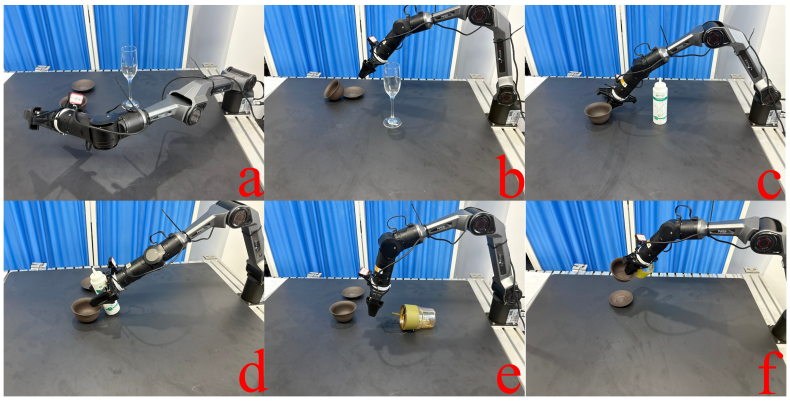

Figure 12: Representative real-world failures. (a) Excessive avoidance drives the arm into a joint-limit configuration from which it cannot recover. (b) The target is placed inaccurately. (c) The target is grasped inaccurately. (d) Obstacle displacement blocks the nominal grasping motion. (e) A high-risk obstacle is contacted and knocked over. (f) The target and a nearby obstacle are grasped together. These examples expose failures from over-avoidance, manipulation imprecision, obstacle motion, residual collision, and insufficient separation of adjacent target and obstacle geometry.

A.4.5 TRAINING DIAGNOSTICS

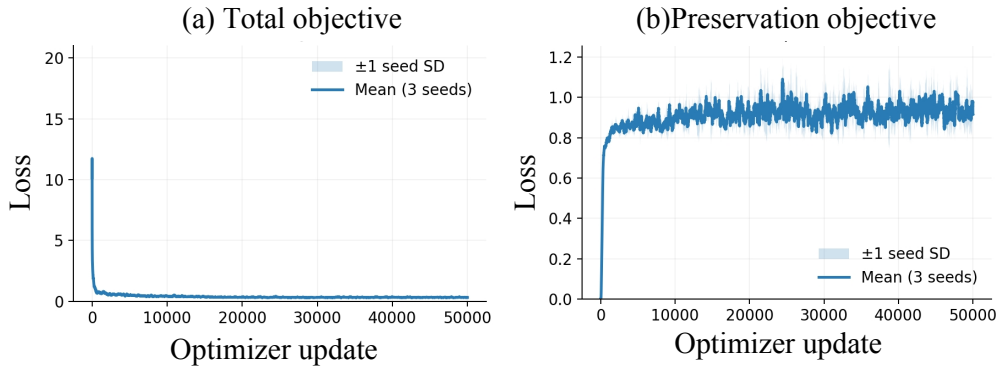

Figure 13: Training diagnostics over 50,000 optimizer updates: (a) total training objective and (b) preservation objective. Solid curves show the mean over three random seeds, and shaded regions indicate one standard deviation.

A.5 LIMITATIONS AND SCOPE

The current method assumes quasi-static obstacles, calibrated RGB-D sensing, and an external VLM for obstacle localization and strength prediction. The probe set covers the end-effector envelope rather than the full robot body; dynamic obstacles, perception uncertainty, and full-body collisions are not modeled. CFG extrapolation is reliable only over a bounded interval, and the real-world study uses one robot and a relatively small manually collected dataset. Future work should extend the field to dynamic scenes and full-body geometry, model uncertainty explicitly, and evaluate broader robot embodiments and manipulation settings.