SELECT TO SUCCEED: PARETO-CONSISTENT ACTION SELECTION FOR VLAS VIA AUGMENTED TCHEBYCHEFF SCALARIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Scaling test-time computation in Vision-Language-Action (VLA) models necessitates robust arbitration over stochastically generated candidate trajectories. We formalize test-time candidate selection as a finite-set multi-objective optimization problem under a receding-horizon control framework, introducing S2S-VLA (Select to Succeed) for frozen VLA policies. Specifically, we map candidate action prefixes into a predefined cost space comprising the marginal probabilities of task failure and local anomalies, alongside an analytic metric of command roughness. To systematically resolve objective trade-offs, we construct the selection rule via an augmented Tchebycheff scalarization and prove that its exact minimizers are nondominated within the evaluated candidate set in the predicted objective space. To align a lightweight neural evaluator with this selection criterion, we optimize a theoretically compatible dual-objective formulation: a pointwise binary cross-entropy loss anchoring marginal event probabilities to empirical offline branch rollouts, and a pairwise logistic loss enforcing scalarization-induced soft pairwise preferences. Following temperature scaling for post-hoc probability calibration, S2S-VLA deterministically minimizes the scalarized score, enabling mathematically consistent, rollout-free, and computationally efficient test-time scaling for embodied agents. Extensive experiments in simulated and real-world environments demonstrate substantial improvements in task success over baseline methods. Our project is available at S2S-VLA.

1 INTRODUCTION

Scaling test-time computation has emerged as a promising approach to enhance the capabilities of frozen vision-language-action (VLA) models (Kwok et al., 2025; 2026; Team et al., 2025; Ding et al., 2025). By querying a stochastic generative policy (Black et al., 2025; Liu et al., 2025; Li et al., 2024; NVIDIA et al., 2025) to sample multiple candidate action trajectories, embodied agents can provide alternatives to suboptimal single-sample actions. However, this generative scaling increases the importance of candidate selection. The prevailing Best-of-N paradigm typically resolves this by scoring candidates using a learned scalar score. A single aggregate score can obscure how individual execution objectives influence candidate selection.

More critically, selecting the candidate that maximizes a noisy point-estimate exacerbates the optimizer's curse. Because neural evaluators are subject to finite capacity and out-of-distribution uncertainty, selecting from a larger candidate set can favor candidates with overestimated success probabilities or underestimated execution risks. In physical environments, such selections can lead to harmful execution outcomes. These challenges motivate an explicit multi-objective formulation (Lin et al., 2024) of candidate selection, together with empirical supervision of outcome probabilities. The selection criterion should be strictly Pareto-consistent, and the underlying estimator must be rigorously calibrated to empirical probabilities.

To this end, we introduce **S2S-VLA (Select to Succeed)**, a theoretically grounded candidate-selection framework for frozen VLA policies under receding-horizon control. S2S-VLA maps candidate action prefixes into a structured cost space comprising the marginal probabilities of task failure and execution anomalies, alongside an analytic, gradient-free metric of command roughness.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

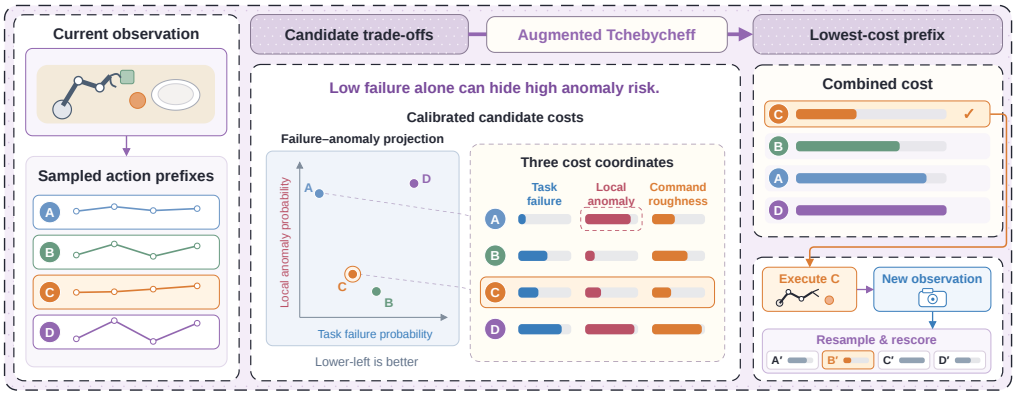


Figure 1: S2S-VLA applies augmented Tchebycheff scalarization to candidate cost vectors and selects the action prefix with the lowest scalarized score.

To systematically navigate objective trade-offs without the computational burden of explicit Pareto pruning, we construct a deterministic selection rule driven by an augmented Tchebycheff scalarization (Wierzbicki, 1980) (Figure 1). We provide a formal proof that the exact minimizers of this scalarization are nondominated within the evaluated candidate set in the predicted objective space.

To operationalize this selection rule efficiently, we design a lightweight neural evaluator trained via a theoretically compatible dual-objective formulation. The evaluator is supervised by offline empirical branch rollouts via a pointwise binary cross-entropy loss, supervising its outputs with empirical event frequencies. Simultaneously, the network is optimized via a pairwise logistic loss (Burges et al., 2005) to enforce the soft pairwise preferences induced by the scalarization. We show that exact matching of empirical event probabilities also matches the induced pairwise preference targets. By applying post-hoc temperature scaling (Guo et al., 2017) for post-hoc probability calibration, the receding-horizon controller minimizes the calibrated scalarized score deterministically, eliminating the need for computationally expensive online rollouts.

The primary contributions of this paper are summarized as follows: (1) We reformulate test-time candidate arbitration for VLA models as a finite-set multi-objective receding-horizon control problem, mapping generated controls into a predefined cost space of task failure probability, local anomaly probability, and command roughness. (2) We construct a deterministic test-time selection rule via an augmented Tchebycheff scalarization and formally prove that its exact minimizers are nondominated within the evaluated candidate set in the predicted objective space. (3) We develop a dual-objective training paradigm that theoretically aligns pointwise probability anchoring with pairwise ordinal preference learning, yielding a mathematically consistent, rollout-free, and computationally efficient test-time scaling framework for embodied agents.

2 Method

We introduce **S2S-VLA (Select to Succeed)**, a rigorously grounded candidate-selection framework for scaling test-time computation in frozen vision-language-action (VLA) policies. An overview of the complete framework is provided in Figure 2. Under a receding-horizon control paradigm, the generative policy proposes multiple candidate action chunks at each decision step. To guide candidate selection, S2S-VLA systematically evaluates these candidate prefixes in a predefined multi-objective cost space, comprising predicted task failure probability, local anomaly probability, and analytic command roughness. We resolve objective trade-offs via an augmented Tchebycheff scalarization and train a lightweight neural evaluator using a theoretically compatible dual-objective formulation. Following post-hoc probability calibration, the controller deterministically executes the prefix that minimizes the scalarized score, operating entirely rollout-free online.

2.1 Candidate prefixes and empirical branch supervision

Let $x_t = (o_t, l, a_{t-1}, d_t)$ denote the decision context, where o_t contains the available observations and any required history, l is the language instruction, a_{t-1} is the preceding control command, and d_t

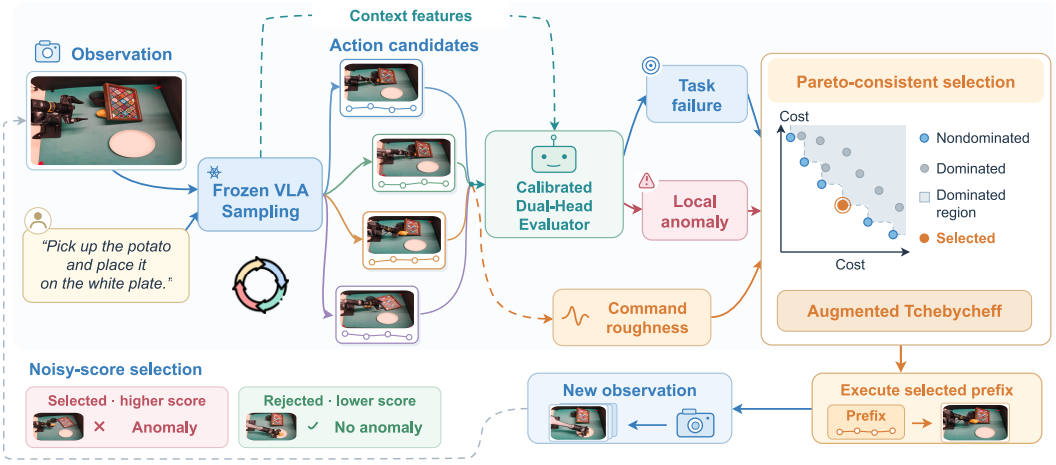


Figure 2: Overview of S2S-VLA. A frozen VLA samples multiple action candidates from the current observation and language instruction. Each candidate is mapped to a multi-objective cost space consisting of calibrated task-failure and local-anomaly probabilities and analytic command roughness. S2S-VLA resolves these objectives via augmented Tchebycheff scalarization, yielding a Pareto-consistent selection whose exact minimizer is nondominated within the evaluated candidate set. The selected prefix is executed and the process repeats under receding-horizon control. The bottom-left example contrasts this structured selection with noisy single-score arbitration, which may favor an undesirable candidate.

is the number of low-level control steps remaining before the episode timeout. A frozen generative policy π_0 samples $K \geq 2$ action chunks (Pertsch et al., 2025) of horizon H. To facilitate receding-horizon execution, the controller evaluates and executes a prefix of length $h_t = \min\{h, d_t\}$, where $h \leq H$:

$$A_t^{(k)} \sim \pi_0(\cdot \mid x_t), \qquad B_t^{(k)} = A_{t,0:h_t-1}^{(k)}, \qquad k = 1, \ldots, K. \tag{1}$$

The execution horizon is strictly capped by the remaining budget. Anomaly labels and command roughness are evaluated over the executed prefix, whereas task-success labels additionally account for the subsequent execution of the continuation policy.

To rigorously anchor our supervisory signals, we collect empirical data in a resettable simulator. For each training context x_n, we store a simulator snapshot and evaluate each candidate prefix across R independent branch rollouts. Each rollout restores the identical physical state, independently resamples the prescribed rollout randomness, executes $B_n^{(k)}$, and then follows a fixed continuation policy π_{ref} (we use $\pi_{\text{ref}} = \pi_0$) until task completion or timeout. Let $S_{nkr} \in \{0, 1\}$ indicate task completion within the remaining budget, and let $D_{nkr} \in \{0, 1\}$ indicate a predefined harmful event during prefix execution. Event definitions are task-specific and exclude intended contacts. Because an anomaly may be physically recoverable, S_{nkr} and D_{nkr} need not be mutually exclusive.

The evaluator targets the marginal event probabilities:

$$p_s(x, B) = \Pr(S = 1 \mid x, B, \pi_{\text{ref}}), \qquad p_d(x, B) = \Pr(D = 1 \mid x, B), \tag{2}$$

using the empirical proportions:

$$\bar{S}_{nk} = \frac{1}{R} \sum_{r=1}^{R} S_{nkr}, \qquad \bar{D}_{nk} = \frac{1}{R} \sum_{r=1}^{R} D_{nkr}. \tag{3}$$

These proportions provide noisy but unbiased supervision under the collection protocol. Simulator snapshots support state restoration and labeling. The evaluator receives only its available decision context. Crucially, p_s evaluates a prefix followed by π_{ref}, and does not represent the overall success

probability of the final selector's closed loop. Counterfactual branch outcomes must be collected explicitly. They cannot be extracted from ordinary demonstration trajectories. Data partitions are separated by collection episode or initial-condition group, keeping related snapshots and all their branches in the same partition.

2.2 Dual-head evaluator and cost space representation

To ensure test-time computational efficiency, the evaluator reuses the frozen visual-language features $e_t = \phi_{\mathrm{VLA}}(o_t, l)$ (Karamcheti et al., 2024; Beyer et al., 2024). A trainable action encoder and multilayer perceptron combine these features with the candidate prefix and control context to produce two logits:

$$(u_s, u_d) = V_\psi(e_t, B, a_{t-1}, d_t), \qquad \hat{p}_s = \sigma(u_s), \quad \hat{p}_d = \sigma(u_d), \tag{4}$$

where $\sigma(u) = (1 + \exp(-u))^{-1}$. Only the parameters ψ are optimized. By maintaining separate heads, we estimate marginal probabilities directly, imposing no artificial conditional independence assumptions between task progression and localized risks. The evaluator is shared across all selection preferences.

To penalize kinematically abrupt transitions without requiring additional predictive capacity, we compute command roughness analytically for a prefix $B = (b_0, \ldots, b_{|B|-1})$ as:

$$s(B, a_{t-1}) = \frac{1}{|B|} \sum_{u=0}^{|B|-1} \|C(b_u - b_{u-1})\|_2^2, \qquad b_{-1} = a_{t-1}. \tag{5}$$

Here, C is a fixed scaling operator over continuous control coordinates, while discrete commands are excluded from this distance. The boundary term explicitly penalizes discontinuity from the preceding command. This quantity measures variation in commanded controls, rather than realized physical jerk.

We systematically map all candidates into a predefined, unified cost space to be minimized:

$$\begin{aligned} \mathbf{f}(x, B) &= (1 - p_s(x, B),\ p_d(x, B),\ s(B, a_{t-1})), \\ \bar{\mathbf{f}}_{nk} = (1 - \bar{S}_{nk},\ \bar{D}_{nk},\ s_{nk}), &\qquad \hat{\mathbf{f}}_{nk} = (1 - \hat{p}_{s,nk},\ \hat{p}_{d,nk},\ s_{nk}), \end{aligned} \tag{6}$$

where the three coordinates represent task failure probability, local anomaly probability, and command roughness, respectively. A candidate is Pareto nondominated if no other candidate is no worse on every objective and strictly better on at least one. Figure 3 illustrates dominance and nondominance among candidate prefixes in the predicted objective space, with command roughness held fixed.

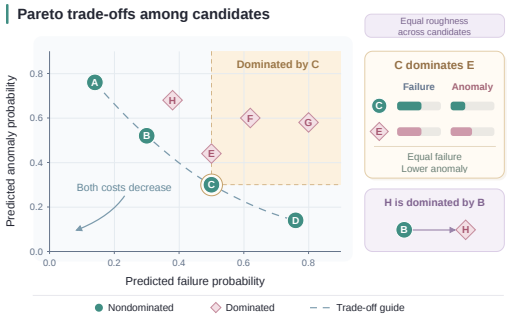


Figure 3: Pareto dominance in the predicted objective space with equal command roughness. A–D are nondominated, whereas E–H are dominated. The dashed curve connects A–D as a visual guide, and the shaded region indicates the area dominated by C.

2.3 Pareto-consistent augmented Tchebycheff selection

To systematically navigate objective trade-offs, we construct our selection rule via an augmented Tchebycheff scalarization (Lin et al., 2024). Unlike weighted sums, Tchebycheff-based scalarizations can recover unsupported Pareto-optimal solutions on non-convex fronts. For our augmented score, positive weights, scales, and augmentation ensure that exact minimizers are nondominated within the candidate set. Let $w \in \mathbb{R}^3_{>0}$ satisfy $\sum_i w_i = 1$, representing a strictly positive preference vector. Let $c_i > 0$ be fixed objective scales, and let z_i be a fixed lower reference point. We define the scalarized score as:

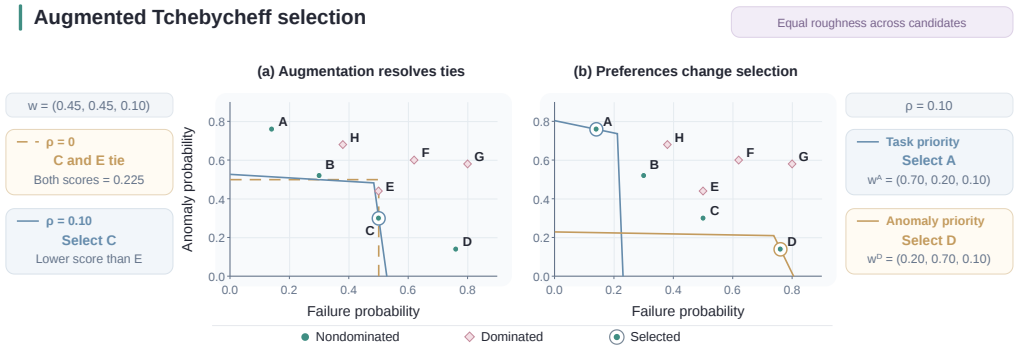


Figure 4: Augmented Tchebycheff selection with equal command roughness. (a) Augmentation resolves the score tie between C and E and selects C. (b) Different preference weights lead to the selection of A or D. Lines represent score level sets.

$$G_\rho(\mathbf{f}; w) = \max_{i\in\{1,2,3\}} \left\{ w_i \frac{f_i - z_i}{c_i} \right\} + \rho \sum_{i=1}^{3} w_i \frac{f_i - z_i}{c_i}, \qquad \rho > 0. \tag{7}$$

We set the reference point $z = (0, 0, 0)$ and $c_1 = c_2 = 1$, and fix a positive roughness scale c_3 derived from the training split. Scales are held constant across candidate sets and data partitions. The maximum term heavily isolates and penalizes the worst-case weighted objective deviation, while the strictly positive augmentation term ρ ensures the score remains monotonically responsive to local improvements in any coordinate. Preference weights determine trade-offs. They do not impose a hard success or anomaly constraint. Figure 4 illustrates how augmentation and preference weights affect candidate selection under Equation 7.

Proposition 1 (Candidate-set Pareto consistency). *Let $\mathcal{C}$ be a nonempty finite candidate set with objective vectors $\mathbf{f}$. If all w_i, c_i, ρ are strictly positive, every exact minimizer of G_ρ over $\mathcal{C}$ is mathematically guaranteed to be Pareto nondominated with respect to $\mathbf{f}$ within $\mathcal{C}$.* The proof is provided in the Appendix B.2.

This foundational property applies to the objective vectors used for scoring. While it does not establish global action-space optimality, it guarantees non-dominance within the predicted candidate subset. Crucially, it renders explicit Pareto pruning algorithms unnecessary for the exact minimization in Equation 7.

2.4 Dual-objective optimization and supervisory compatibility

To operationalize this selection criterion effectively, we design a dual-objective training paradigm. For $\ell_{\mathrm{BCE}}(p, y) = -y \log p - (1 - y) \log(1 - p)$, we anchor the probability heads to empirical event frequencies via:

$$\mathcal{L}_{\mathrm{prob}} = \frac{1}{NK} \sum_{n=1}^{N} \sum_{k=1}^{K} \left[\ell_{\mathrm{BCE}}(\hat{p}_{s,nk}, \bar{S}_{nk}) + \ell_{\mathrm{BCE}}(\hat{p}_{d,nk}, \bar{D}_{nk}) \right]. \tag{8}$$

Because BCE is affine in its target, applying empirical proportions as soft labels is mathematically equivalent to averaging over the individual binary outcomes of the branch rollouts:

$$\ell_{\mathrm{BCE}} \left(p, \frac{1}{R} \sum_{r=1}^{R} y_r \right) = \frac{1}{R} \sum_{r=1}^{R} \ell_{\mathrm{BCE}}(p, y_r). \tag{9}$$

To align candidate comparisons with the selection criterion, we introduce soft preference supervision. Let $\mathcal{W} = \{w^{(1)}, \ldots, w^{(L)}\}$ be a fixed, finite set of strictly positive preference vectors. For each preference, we compute the empirical target scores and the predicted scores:

$$g^*_{nk\ell} = G_\rho(\bar{\mathbf{f}}_{nk}; w^{(\ell)}), \qquad \hat{g}_{nk\ell} = G_\rho(\hat{\mathbf{f}}_{nk}; w^{(\ell)}). \tag{10}$$

The superscript $*$ denotes a finite-sample empirical target, not an oracle value. We first estimate expected objectives and then scalarize. For a random rollout-cost vector $\mathbf{Y}$, generally $G_\rho(\mathbb{E}[\mathbf{Y}]; w) \neq \mathbb{E}[G_\rho(\mathbf{Y}; w)]$, where our objective is the former, estimated by a finite-sample plug-in score.

For candidates $k < j$ from the same context, we derive soft preference targets and their corresponding predictions:

$$q^*_{nkj\ell} = \sigma\left(\frac{g^*_{nj\ell} - g^*_{nk\ell}}{\tau_{\text{pair}}}\right), \qquad \hat{q}_{nkj\ell} = \sigma\left(\frac{\hat{g}_{nj\ell} - \hat{g}_{nk\ell}}{\tau_{\text{pair}}}\right), \tag{11}$$

where $\tau_{\text{pair}} > 0$. A lower cost for k yields $q^*_{nkj\ell} > 1/2$, and equal scores yield $q^*_{nkj\ell} = 1/2$. These targets encode the ordinal preference strength induced by scalarized cost differences. They are neither success probabilities nor statistical confidence levels. Following the pairwise logistic cross-entropy formulation of (Burges et al., 2005), we minimize:

$$\mathcal{L}_{\text{pref}} = \frac{1}{NL\binom{K}{2}} \sum_{n=1}^{N} \sum_{\ell=1}^{L} \sum_{k<j} \ell_{\text{BCE}}(\hat{q}_{nkj\ell}, q^*_{nkj\ell}). \tag{12}$$

The unified objective is

$$\min_{\psi} \ \mathcal{L}(\psi), \qquad \mathcal{L}(\psi) = \mathcal{L}_{\text{prob}} + \lambda_{\text{pref}} \mathcal{L}_{\text{pref}}, \qquad \lambda_{\text{pref}} \geq 0. \tag{13}$$

The loss weight λ_{pref} is distinct from the selection preference w. Gradients flow exclusively through the predicted probabilities and predicted scores. Empirical probabilities, target scores, soft labels, command roughness, scales, reference points, and preference vectors are fixed. BCE is implemented from logits for numerical stability, and the maximum uses a valid subgradient at ties.

Proposition 2 (Theoretical compatibility of dual-objective supervision). *Suppose the predicted event probabilities exactly match the empirical proportions for every training context n and candidate k (i.e., $\hat{p}_{s,nk} = \bar{S}_{nk}$ and $\hat{p}_{d,nk} = \bar{D}_{nk}$). Then, for any candidate pair $k < j$ and strictly positive preference vector $w^{(\ell)}$, the predicted soft preference exactly matches its empirical target: $\hat{q}_{nkj\ell} = q^*_{nkj\ell}$. Furthermore, if candidate k dominates candidate j in the empirical objective space, then $q^*_{nkj\ell} > 1/2$ for all valid preferences.* The proof is provided in the Appendix B.3.

This result establishes compatibility of the empirical supervision targets in probability space. The soft preference supervision structurally reinforces rankings consistent with the scalarized scores without demanding arbitrary margin distortions once empirical cost differences are accurately captured. We note that this establishes target compatibility rather than statistical consistency or out-of-distribution generalization, as finite rollout noise and limited model capacity remain potential sources of estimation error. Notably, this formulation structurally bypasses the need for any ad-hoc, standalone Pareto loss.

2.5 Post-hoc probability calibration and preference selection

Errors in predicted event probabilities can distort scalarized candidate scores. Therefore, after training, we freeze V_ψ and perform post-hoc temperature scaling (Guo et al., 2017) on an independent calibration split. For $v \in \{s, d\}$, we define $\bar{y}_{s,nk} = \bar{S}_{nk}$ and $\bar{y}_{d,nk} = \bar{D}_{nk}$, and solve:

$$T^*_v \in \arg\min_{T>0} \sum_{(n,k)\in\mathcal{D}_{\text{cal}}} \ell_{\text{BCE}}(\sigma(u_{v,nk}/T), \bar{y}_{v,nk}), \qquad \tilde{p}_v = \sigma(u_v/T^*_v). \tag{14}$$

Temperature scaling adjusts probability confidence empirically. Temperature scaling does not guarantee perfect calibration, particularly under distribution shift. The head temperatures are separate

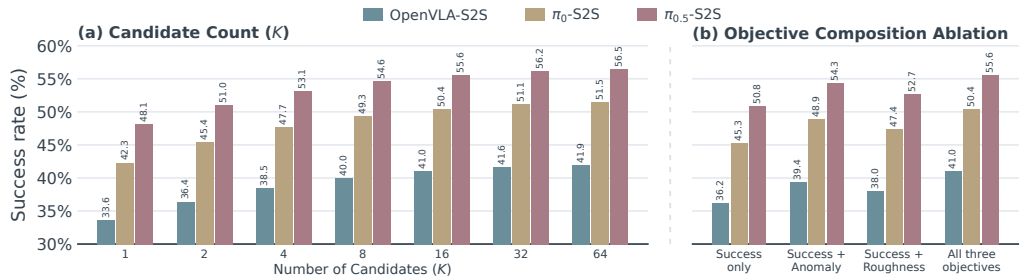


Figure 5: Candidate scaling and objective composition ablation of S2S-VLA. (a) Task success rate as the number of sampled action candidates (K) increases from 1 to 64. (b) Task success rate when selection uses success alone, success with anomaly avoidance, success with roughness reduction, or all three objectives. To limit inference latency, we set $(K = 16)$ for main experiments.

from τ_{pair}. Because independent head-specific calibration can subtly alter the combined candidate ordering, final deployment preferences must be validated using the calibrated controller.

Let π_w denote the complete receding-horizon controller under preference w. On a separate validation partition, we select:

$$w^* \in \arg\max_{w \in \mathcal{W}} \widehat{\mathrm{SR}}_{\mathrm{val}}(\pi_w), \qquad \widehat{\mathrm{SR}}_{\mathrm{val}}(\pi_w) = \frac{1}{M} \sum_{m=1}^{M} S_m^{\mathrm{val}}(w). \tag{15}$$

Here, M is the number of validation episodes, and $S_m^{\mathrm{val}}(w) \in \{0, 1\}$ indicates whether controller π_w completes the task within the prescribed budget in validation episode m. Preferences use the same episode set and candidate-generation budget, and all choices are fixed before testing. Anomaly and roughness are auxiliary selection objectives whose true usefulness is assessed directly through measured closed-loop task performance.

2.6 Rollout-free receding-horizon execution and computational complexity

At deployment, the frozen VLA proposes K candidates. Let $\mathcal{C}_t$ contain the prefixes that pass the platform's existing action-validity rules. For every valid prefix, the controller computes the calibrated cost vector:

$$\tilde{\mathbf{f}}_t(B) = \big(1 - \tilde{p}_s(x_t, B),\ \tilde{p}_d(x_t, B),\ s(B, a_{t-1})\big), \qquad B_t^* \in \arg\min_{B \in \mathcal{C}_t} G_\rho(\tilde{\mathbf{f}}_t(B); w^*). \tag{16}$$

Ties are broken by a fixed candidate order. The controller executes B_t^*, obtains a new observation, and repeats until completion or timeout. If no valid candidate exists, it invokes the platform's predefined stop behavior, with incomplete episodes counting as unsuccessful. No branch rollouts or pairwise comparisons are required online.

Crucially, S2S-VLA ensures computationally efficient test-time scaling. Although collecting N training contexts requires NKR branch rollouts, this represents a strictly offline amortized cost. At deployment, the method completely eliminates the need for test-time online rollouts or dynamics simulation (unlike planning methods that explicitly roll out a dynamics model (Hansen et al., 2024)). Beyond candidate generation, online selection requires K evaluator passes, which can be batched, followed by an $\mathcal{O}(K)$ scan over the scalarized scores. Closed-loop success must be measured directly because repeated reselection dynamically changes the continuation policy relative to the one used for supervision. The pseudocode is provided in the Appendix A.5.

3 Experiments

Setup. Detailed descriptions of the tasks, datasets, experimental setup, hyperparameters and more results are provided in the Appendix A.

Method	PnP CTC1	PnP CTC2	PnP CTS1	PnP STC1	PnP CTS2	PnP STC2	PnP CTM	PnP MTC	CM SU	CM SV	CB PS	TSS
OpenVLA	6.0	18.0	20.0	8.0	2.0	6.0	14.0	2.0	16.0	22.0	48.0	54.0
OpenVLA-S2S	**12.0**	**20.0**	**32.0**	**16.0**	**8.0**	**12.0**	**20.0**	**8.0**	**24.0**	**30.0**	**56.0**	**64.0**
π_0	24.0	6.8	26.8	15.6	11.2	12.8	6.8	7.6	18.4	33.6	74.0	69.6
π_0-S2S	**32.4**	**14.8**	**34.8**	**24.8**	**19.2**	**22.0**	**14.8**	**15.6**	**27.2**	**43.6**	**80.8**	**78.8**
$\pi_{0.5}$	32.4	8.8	27.6	31.2	20.4	28.8	14.4	18.0	28.4	49.2	67.6	70.8
$\pi_{0.5}$-S2S	**40.0**	**16.8**	**35.6**	**39.6**	**28.4**	**37.2**	**22.4**	**26.0**	**36.8**	**58.0**	**74.8**	**80.4**
Avg. SR (%)	**OSD**	**CSD**	**ODD**	**CDD**	**OD**	**CD**	**TNSF**	**TFSF**	**TNS**	**TFS**	**TNM**	**TFM**
33.6	46.0	56.0	28.0	50.0	42.0	80.0	52.0	50.0	32.0	22.0	62.0	70.0
41.0	**54.0**	**66.0**	**36.0**	**58.0**	**52.0**	**88.0**	**60.0**	**58.0**	**40.0**	**26.0**	**70.0**	**74.0**
42.3	40.8	82.0	55.2	69.2	66.4	96.0	43.6	86.0	43.2	6.0	59.6	60.0
50.4	**50.4**	**88.8**	**65.6**	**78.4**	**76.4**	**98.4**	**54.0**	**92.4**	**52.8**	**11.2**	**66.8**	**65.6**
48.1	44.8	74.0	62.8	64.8	68.8	92.4	58.8	80.8	51.6	13.2	72.0	72.8
55.6	**53.6**	**81.2**	**71.6**	**72.8**	**77.2**	**96.8**	**67.2**	**86.8**	**59.6**	**17.6**	**77.2**	**76.8**

Table 1: RoboCasa Simulation Benchmark Results. Applying our S2S-VLA method leads to substantial improvements in success rates across different VLA backbones. See Appendix A.2.1 for detailed sub-task definitions.

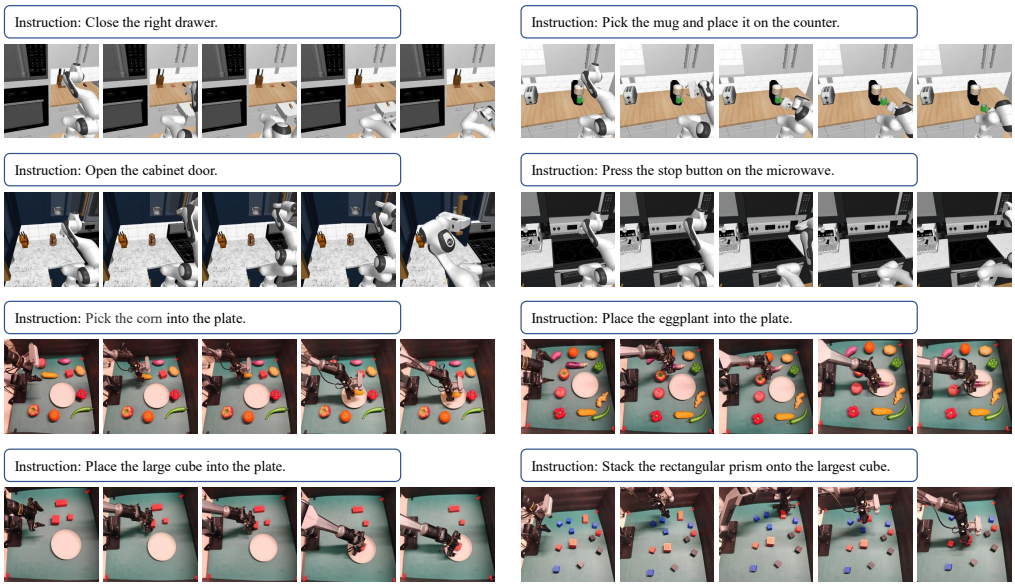


Figure 6: Visualization of our S2S-VLA in both simulation and real-world environments. We use $\pi_{0.5}$ for the test.

Main results. We evaluate OpenVLA (Kim et al., 2025), π_0 (Black et al., 2025), and $\pi_{0.5}$ (Physical Intelligence et al., 2025) with and without S2S-VLA on the 24 reported RoboCasa (Nasiriany et al., 2024) tasks (Table 1), LIBERO-Plus (Fei et al., 2025) (Appendix A.1.2) and RLBench (James et al., 2020) (Appendix A.1.1). On RoboCasa, S2S-VLA improves average success from 33.6% to 41.0% for OpenVLA, 42.3% to 50.4% for π_0, and 48.1% to 55.6% for $\pi_{0.5}$.

Analysis. Increasing K improves success across all three backbones, while gains beyond $K = 16$ are modest (Figure 5 (a)). Using all three selection objectives outperforms success-only selection for each backbone (Figure 5 (b)). Figure 6 shows representative manipulation trajectories. The transfer heatmaps in Figure 7 show that matched training and evaluation domains achieve the highest success in each row, while performance remains measurable across backbones and benchmarks (Tested on $\pi_{0.5}$). In real-world experiments, S2S-VLA improves success by 10–20 percentage points across the reported backbones and evaluation categories (Figure 8 (a)). The reported inference latency tested on OpenVLA is 0.35 s per step at $K = 16$, compared with 0.65 s at $K = 32$ and 1.30 s at $K = 64$ (Figure 8 (b)). Together with the diminishing success gains, these measurements motivate our choice of $K = 16$. More videos can be found in the supplementary material.

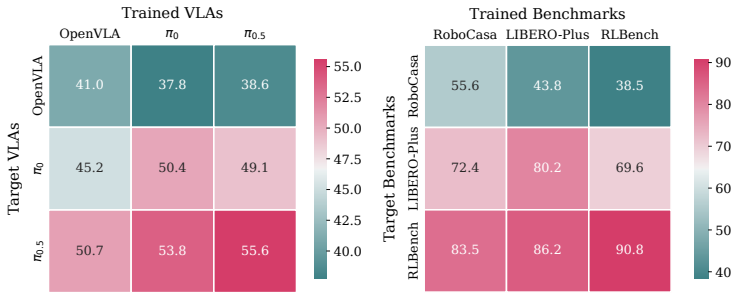


Figure 7: Transfer of S2S-VLA across backbones and benchmarks. (a) Cross-backbone transfer among OpenVLA, π_0, and $\pi_{0.5}$. (b) Cross-benchmark transfer among RoboCasa, LIBERO-Plus, and RLBench.

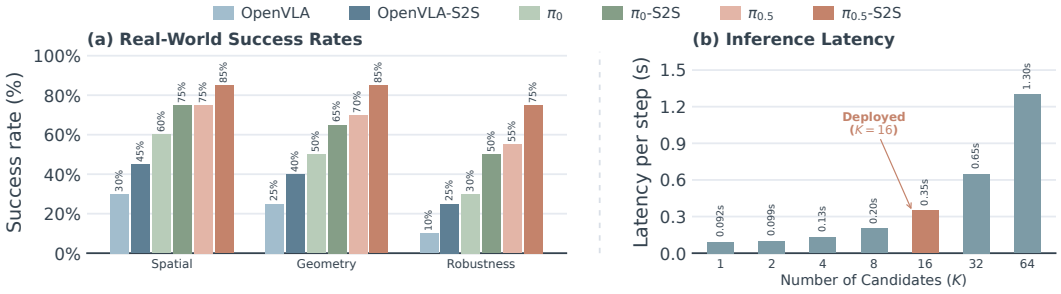


Figure 8: Real-world performance and inference cost. (a) Real-world results with and without S2S-VLA across three VLA backbones. (b) Inference latency versus candidate count K.

4 Related Work

Vision-language-action models (Zhong et al., 2025), including RT-2 (Zitkovich et al., 2023), OpenVLA (Kim et al., 2025), π_0 (Black et al., 2025), and $\pi_{0.5}$ (Physical Intelligence et al., 2025), connect visual-language representations with robot control. ACT (Zhao et al., 2023) and Diffusion Policy (Chi et al., 2023) develop action-sequence policies for visuomotor manipulation. Complementary work explores value-guided candidate selection in V-GPS (Nakamoto et al., 2025) and Hume (Song et al., 2025), geometric action-chunk evaluation in VGAS (Xu et al., 2026), and learned verification in RoboMonkey (Kwok et al., 2025) and CoVer-VLA (Kwok et al., 2026). S2S-VLA builds on this generation–selection paradigm by explicitly considering task failure probability, local anomaly probability, and command roughness. Its augmented Tchebycheff rule draws on reference-point scalarization (Wierzbicki, 1980) and nonlinear multi-objective optimization (Lin et al., 2024), ensuring nondominance within the evaluated candidate set in the predicted objective space. The evaluator connects probabilistic pairwise ranking (Burges et al., 2005) with the principle of training predictions for downstream decisions (Donti et al., 2017), using empirical branch outcomes to construct compatible probability and scalarization-derived preference targets. Post-hoc temperature scaling (Guo et al., 2017) adjusts predictive confidence before validation-based preference selection.

5 Conclusion

We introduced S2S-VLA, which reformulates test-time candidate arbitration for frozen vision-language-action policies as a finite-set multi-objective optimization problem under receding-horizon control. To guide candidate selection, we map candidate prefixes into a predefined cost space comprising marginal event probabilities and an analytic command roughness metric. We resolve objective trade-offs via an augmented Tchebycheff scalarization and prove that its exact minimizers are nondominated within the evaluated candidate set in the predicted objective space. This selection criterion is operationalized by a theoretically compatible dual-objective training paradigm that aligns pointwise probability anchoring with pairwise ordinal preference learning. Following post-hoc temperature scaling, S2S-VLA deterministically minimizes the scalarized score, yielding a mathematically consistent and rollout-free arbitration mechanism.

AI USE STATEMENT

In this work, we used generative AI tools to assist with manuscript writing and language polishing. We have reviewed all AI-assisted content and take responsibility for the final content of this work, including any text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This research adheres to the ICLR ethical guidelines and upholds the principles of responsible research. We ensure that no personally identifiable, sensitive, or harmful data were used. Our experiments did not involve human subjects or vulnerable groups. We have considered the potential societal impacts of our methods, including the risk of misuse, and believe that our contributions primarily advance scientific understanding and do not pose foreseeable harm.

REPRODUCIBILITY STATEMENT

We follow the reproducibility guidelines outlined in the ICLR author guidelines. We will open-source the code, configuration files, and scripts needed to reproduce our results, including those for dataset construction, model training, and evaluation, on platforms such as GitHub and Hugging Face as soon as possible.

REFERENCES

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey Gritsenko, Neil Houlsby, Manoj Kumar, Keran Rong, Julian Eisenschlos, Rishabh Kabra, Matthias Bauer, Matko Bošnjak, Xi Chen, Matthias Minderer, Paul Voigtlaender, Ioana Bica, Ivana Balazevic, Joan Puigcerver, Pinelopi Papalampidi, Olivier Henaff, Xi Xiong, Radu Soricut, Jeremiah Harmsen, and Xiaohua Zhai. Paligemma: A versatile 3b vlm for transfer, 2024. URL https://arxiv.org/abs/2407.07726.

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

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine Learning*, 2005. URL https://www.microsoft.com/en-us/research/wp-content/uploads/2005/08/icml_ranking.pdf.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin CM Burchfiel, and Shuran Song. Diffusion Policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems*, 2023. doi: 10.15607/RSS.2023.XIX.026. URL https://roboticsproceedings.org/rss19/p026.html.

Pengxiang Ding, Han Zhao, Wenjie Zhang, Wenxuan Song, Min Zhang, Siteng Huang, Ningxi Yang, and Donglin Wang. Quar-vla: Vision-language-action model for quadruped robots, 2025. URL https://arxiv.org/abs/2312.14457.

Priya L. Donti, Brandon Amos, and J. Zico Kolter. Task-based end-to-end model learning in stochastic optimization. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3fc2c60b5782f641f76bcefc39fb2392-Abstract.html.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. Libero-plus: In-depth robustness analysis of vision-language-action models, 2025. URL https://arxiv.org/abs/2510.13626.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330. PMLR, 2017. URL https://proceedings.mlr.press/v70/guo17a.html.

Nicklas Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control, 2024. URL https://arxiv.org/abs/2310.16828.

Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J. Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 2020.

Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic vlms: Investigating the design space of visually-conditioned language models, 2024. URL https://arxiv.org/abs/2402.07865.

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. PMLR, 2025. URL https://proceedings.mlr.press/v270/kim25c.html.

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

Jacky Kwok, Xilun Zhang, Mengdi Xu, Yuejiang Liu, Azalia Mirhoseini, Chelsea Finn, and Marco Pavone. Scaling verification can be more effective than scaling policy learning for vision-language-action alignment, 2026. URL https://arxiv.org/abs/2602.12281.

Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, Xiaofan Wang, Bei Liu, Jianlong Fu, Jianmin Bao, Dong Chen, Yuanchun Shi, Jiaolong Yang, and Baining Guo. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation, 2024. URL https://arxiv.org/abs/2411.19650.

Xi Lin, Xiaoyuan Zhang, Zhiyuan Yang, Fei Liu, Zhenkun Wang, and Qingfu Zhang. Smooth Tchebycheff scalarization for multi-objective optimization. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 30479–30509. PMLR, 2024. URL https://proceedings.mlr.press/v235/lin24y.html.

Jiaming Liu, Hao Chen, Pengju An, Zhuoyang Liu, Renrui Zhang, Chenyang Gu, Xiaoqi Li, Ziyu Guo, Sixiang Chen, Mengzhen Liu, Chengkai Hou, Mengdi Zhao, KC alex Zhou, Pheng-Ann Heng, and Shanghang Zhang. Hybridvla: Collaborative diffusion and autoregression in a unified vision-language-action model, 2025. URL https://arxiv.org/abs/2503.10631.

Mitsuhiko Nakamoto, Oier Mees, Aviral Kumar, and Sergey Levine. Steering your generalists: Improving robotic foundation models via value guidance. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 4996–5013. PMLR, 2025. URL https://proceedings.mlr.press/v270/nakamoto25a.html.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots, 2024. URL https://arxiv.org/abs/2406.02523.

NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. Gr00t n1: An open foundation model for generalist humanoid robots, 2025. URL https://arxiv.org/abs/2503.14734.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models, 2025. URL https://arxiv.org/abs/2501.09747.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization, 2025. URL https://arxiv.org/abs/2504.16054.

Haoming Song, Delin Qu, Yuanqi Yao, Qizhi Chen, Qi Lv, Yiwen Tang, Modi Shi, Guanghui Ren, Maoqing Yao, Bin Zhao, Dong Wang, and Xuelong Li. Hume: Introducing system-2 thinking in visual-language-action model, 2025. URL https://arxiv.org/abs/2505.21432.

Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, Steven Bohez, Konstantinos Bousmalis, Anthony Brohan, Thomas Buschmann, Arunkumar Byravan, Serkan Cabi, Ken Caluwaerts, Federico Casarini, Oscar Chang, Jose Enrique Chen, Xi Chen, Hao-Tien Lewis Chiang, Krzysztof Choromanski, David D'Ambrosio, Sudeep Dasari, Todor Davchev, Coline Devin, Norman Di Palo, Tianli Ding, Adil Dostmohamed, Danny Driess, Yilun Du, Debidatta Dwibedi, Michael Elabd, Claudio Fantacci, Cody Fong, Erik Frey, Chuyuan Fu, Marissa Giustina, Keerthana Gopalakrishnan, Laura Graesser, Leonard Hasenclever, Nicolas Heess, Brandon Hernaez, Alexander Herzog, R. Alex Hofer, Jan Humplik, Atil Iscen, Mithun George Jacob, Deepali Jain, Ryan Julian, Dmitry Kalashnikov, M. Emre Karagozler, Stefani Karp, Chase Kew, Jerad Kirkland, Sean Kirmani, Yuheng Kuang, Thomas Lampe, Antoine Laurens, Isabel Leal, Alex X. Lee, Tsang-Wei Edward Lee, Jacky Liang, Yixin Lin, Sharath Maddineni, Anirudha Majumdar, Assaf Hurwitz Michaely, Robert Moreno, Michael Neunert, Francesco Nori, Carolina Parada, Emilio Parisotto, Peter Pastor, Acorn Pooley, Kanishka Rao, Krista Reymann, Dorsa Sadigh, Stefano Saliceti, Pannag Sanketi, Pierre Sermanet, Dhruv Shah, Mohit Sharma, Kathryn Shea, Charles Shu, Vikas Sindhwani, Sumeet Singh, Radu Soricut, Jost Tobias Springenberg, Rachel Sterneck, Razvan Surdulescu, Jie Tan, Jonathan Tompson, Vincent Vanhoucke, Jake Varley, Grace Vesom, Giulia Vezzani, Oriol Vinyals, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Fei Xia, Ted Xiao, Annie Xie, Jinyu Xie, Peng Xu, Sichun Xu, Ying Xu, Zhuo Xu, Yuxiang Yang, Rui Yao, Sergey Yaroshenko, Wenhao Yu, Wentao Yuan, Jingwei Zhang, Tingnan Zhang, Allan Zhou, and Yuxiang Zhou. Gemini robotics: Bringing ai into the physical world, 2025. URL https://arxiv.org/abs/2503.20020.

Andrzej P. Wierzbicki. The use of reference objectives in multiobjective optimization. In Günter Fandel and Tomas Gal (eds.), *Multiple Criteria Decision Making Theory and Application*, volume 177 of *Lecture Notes in Economics and Mathematical Systems*, pp. 468–486. Springer, 1980. doi: 10.1007/978-3-642-48782-8_32. URL https://link.springer.com/chapter/10.1007/978-3-642-48782-8_32.

Changhua Xu, En Yu, Junyu Xuan, and Jie Lu. Vgas: Value-guided action-chunk selection for few-shot vision-language-action adaptation, 2026. URL https://arxiv.org/abs/2602.07399.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Proceedings of Robotics: Science and Systems*,

2023. doi: 10.15607/RSS.2023.XIX.016. URL https://roboticsproceedings.org/rss19/p016.html.

Yifan Zhong, Fengshuo Bai, Shaofei Cai, Xuchuan Huang, Zhang Chen, Xiaowei Zhang, Yuanfei Wang, Shaoyang Guo, Tianrui Guan, Ka Nam Lui, Zhiquan Qi, Yitao Liang, Yuanpei Chen, and Yaodong Yang. A survey on vision-language-action models: An action tokenization perspective, 2025. URL https://arxiv.org/abs/2507.01925.

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. PMLR, 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

Method	ASR ↑	AR ↓	CJ	DS	IP	MG	OD	PC1	PW	PB
OpenVLA	59.8	6.00	68.0	86.4	17.6	80.0	68.8	12.0	68.0	92.0
OpenVLA-S2S	**67.4**	**4.94**	**78.4**	**92.0**	**32.8**	**88.0**	**77.6**	**20.8**	**78.4**	**96.0**
π_0	78.2	3.92	89.6	94.4	56.0	89.6	86.4	34.4	88.0	98.4
π_0-S2S	**85.6**	**2.17**	**95.2**	**98.4**	**72.0**	**96.0**	**96.0**	**50.4**	**94.4**	**100.0**
$\pi_{0.5}$	84.2	2.86	97.6	98.4	74.4	96.0	92.0	48.0	92.0	**100.0**
$\pi_{0.5}$-S2S	**90.8**	**1.11**	**100.0**	**100.0**	**85.6**	**100.0**	**99.2**	**65.6**	**96.0**	**100.0**
Method	**PC2**	**PD**	**PS**	**SB1**	**SB2**	**SS**	**SB3**	**SC**	**SD**	**TT**
OpenVLA	42.4	80.0	84.0	44.0	81.6	24.0	36.0	36.0	77.6	78.4
OpenVLA-S2S	**48.0**	**84.0**	**88.0**	**52.0**	**86.4**	**30.4**	**42.4**	**44.0**	**82.4**	**92.0**
π_0	58.4	93.6	92.0	76.0	94.4	41.6	64.0	66.4	89.6	95.2
π_0-S2S	**66.4**	**98.4**	**96.0**	**84.0**	**98.4**	**53.6**	**72.0**	**76.0**	**96.0**	**97.6**
$\pi_{0.5}$	65.6	96.0	92.0	82.4	96.0	52.0	72.0	76.0	94.4	91.2
$\pi_{0.5}$-S2S	**73.6**	**100.0**	**98.4**	**88.0**	**100.0**	**64.0**	**82.4**	**84.8**	**100.0**	**96.8**

Table 2: Performance comparison across 18 RLBench tasks. ASR denotes average success rate (%), and AR denotes average rank among the six compared methods, with average ranks assigned to ties. Bold indicates the better result within each backbone–S2S pair. Task abbreviations are defined in Table 5.

A More Information

A.1 Additional Simulation Results

This section provides additional results on RLBench and LIBERO-Plus, together with a task-level discussion of the RoboCasa results reported in Table 1. We compare OpenVLA, π_0, and $\pi_{0.5}$ with their corresponding S2S-VLA variants. Throughout the results, success rates are reported as percentages, and improvements refer to absolute percentage-point differences. The suffix *-S2S* denotes a backbone equipped with our candidate evaluation and action selection procedure.

A.1.1 RLBench Results

Table 2 reports success rates on 18 RLBench tasks. S2S-VLA increases the average success rate of OpenVLA from 59.8% to 67.4%, π_0 from 78.2% to 85.6%, and $\pi_{0.5}$ from 84.2% to 90.8%, corresponding to improvements of 7.6, 7.4, and 6.6 percentage points, respectively. The corresponding average ranks improve from 6.00 to 4.94, from 3.92 to 2.17, and from 2.86 to 1.11. These results show that candidate selection benefits all three backbones despite their different baseline performance levels.

The improvements extend across the task set. OpenVLA-S2S and π_0-S2S improve over their respective backbones on all 18 tasks. $\pi_{0.5}$-S2S improves on 17 tasks and retains the baseline's 100.0% success rate on Push Buttons. Several tasks exhibit particularly substantial gains. On Insert Peg, OpenVLA improves from 17.6% to 32.8%, while π_0 improves from 56.0% to 72.0%. On Place Cups, $\pi_{0.5}$ improves from 48.0% to 65.6%, and on Sort Shape it improves from 52.0% to 64.0%. These results indicate that the benefit is not restricted to tasks that already have high baseline success rates.

Metrics and table notation. ASR denotes the average success rate across the 18 tasks, with each task contributing equally. AR denotes the average of the per-task ranks among the six compared methods. Methods are ranked by descending success rate, and ties receive the average of the positions they occupy. Higher ASR and lower AR indicate better performance. Bold entries mark the better result within each backbone–S2S pair; both entries are bold when they are tied. Task abbreviations are defined in Table 5.

A.1.2 LIBERO-Plus Results

Figure 9 reports success rates across seven perturbation dimensions and four task suites. S2S-VLA increases the overall success rate of OpenVLA from 15.7% to 30.6%, π_0 from 68.3% to 76.1%, and $\pi_{0.5}$ from 72.8% to 80.2%, corresponding to improvements of 14.9, 7.8, and 7.4 percentage

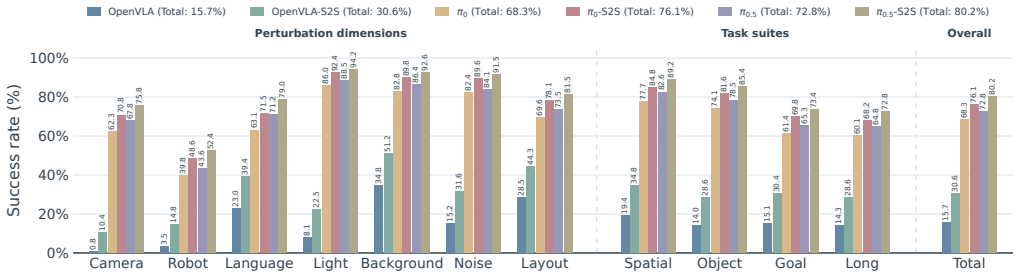


Figure 9: Success rates (%) on LIBERO-Plus across seven perturbation dimensions and four task suites. S2S-VLA improves performance over each corresponding backbone in every reported category. Total denotes the unweighted average across Spatial, Object, Goal, and Long.

points, respectively. All three S2S-VLA variants improve over their corresponding backbones in every reported perturbation dimension and task suite.

Performance under perturbations. Across the seven perturbation dimensions, the improvements range from 9.6 to 16.4 percentage points for OpenVLA, from 6.4 to 8.8 for π_0, and from 5.7 to 8.8 for $\pi_{0.5}$. OpenVLA achieves its largest absolute gains under language, background, and noise perturbations, improving from 23.0% to 39.4%, from 34.8% to 51.2%, and from 15.2% to 31.6%, respectively. Improvements also occur under conditions where its baseline performance is particularly low: camera perturbation performance increases from 0.8% to 10.4%, and robot initial-state perturbation performance increases from 3.5% to 14.8%.

The stronger backbones also benefit across all perturbation dimensions. Under robot initial-state perturbations, π_0 improves from 39.8% to 48.6%, while $\pi_{0.5}$ improves from 43.6% to 52.4%, giving an 8.8-percentage-point gain for both models. Under layout perturbations, their success rates increase from 69.6% to 78.1% and from 73.5% to 81.5%, respectively. These results show that the gains extend across changes to visual input, language instructions, scene arrangement, and robot initialization.

Performance across task suites. S2S-VLA improves performance on all four task suites: Spatial, Object, Goal, and Long. For OpenVLA, the respective success rates increase from 19.4%, 14.0%, 15.1%, and 14.3% to 34.8%, 28.6%, 30.4%, and 28.6%. For π_0, they increase from 77.7%, 74.1%, 61.4%, and 60.1% to 84.8%, 81.6%, 69.8%, and 68.2%. For $\pi_{0.5}$, they increase from 82.6%, 78.5%, 65.3%, and 64.8% to 89.2%, 85.4%, 73.4%, and 72.8%. In particular, the Long suite improves by 14.3, 8.1, and 8.0 percentage points for the three backbones, showing that the benefits also extend to longer-horizon task execution.

Together, these results support the effectiveness of S2S-VLA under the evaluated perturbations and across different task suites. The improvements are obtained by evaluating and selecting candidate action prefixes while keeping the underlying VLA backbone frozen.

Metrics and figure notation. All values denote success rates in percent. Camera, Robot, Language, Light, Background, Noise, and Layout refer to perturbations of camera viewpoints, robot initial states, language instructions, lighting conditions, background textures, sensor observations, and object layouts, respectively. Spatial, Object, Goal, and Long denote the four task suites. Total reports their unweighted mean, rounded to one decimal place. The perturbation dimensions and task suites provide different breakdowns of performance and are not averaged together to obtain Total.

A.1.3 RoboCasa Results

Table 1 reports results on 24 RoboCasa manipulation tasks. S2S-VLA increases the average success rate from 33.6% to 41.0% for OpenVLA, from 42.3% to 50.4% for π_0, and from 48.1% to 55.6% for $\pi_{0.5}$. The corresponding improvements are 7.4, 8.1, and 7.5 percentage points. Each S2S-VLA variant improves over its backbone on all 24 reported tasks.

The gains cover object transport, articulated-object manipulation, and appliance operation. For example, OpenVLA improves from 20.0% to 32.0% on counter-to-sink pick-and-place. On opening a

double-door cabinet, π_0 improves from 55.2% to 65.6%. For $\pi_{0.5}$, serving a coffee mug improves from 49.2% to 58.0%, while turning the sink spout improves from 70.8% to 80.4%. These examples illustrate that the improvements are distributed across different manipulation skills.

Metrics and table layout. SR denotes success rate, and Avg. SR denotes the unweighted average success rate across all 24 tasks. The upper and lower blocks of Table 1 contain 12 task columns each. Rows in the lower block follow the same method order as those in the upper block, and the first column of the lower block reports the overall average. Bold entries indicate the better result within each backbone–S2S pair. The complete task abbreviations are provided in Table 3.

A.2 Benchmark Descriptions and Task Definitions

We use three simulation benchmarks to examine complementary aspects of VLA performance. RoboCasa provides everyday manipulation tasks in kitchen environments, LIBERO-Plus evaluates robustness under controlled perturbations, and RLBench covers a diverse collection of tabletop manipulation skills. The task counts reported below refer to the evaluated subsets where applicable.

A.2.1 RoboCasa

RoboCasa (Nasiriany et al., 2024) is a simulation framework for everyday robot manipulation, with an emphasis on diverse kitchen environments. Its scenes contain interactable furniture and appliances, including cabinets, drawers, sinks, stoves, microwaves, and coffee machines. The benchmark supports both atomic skills and composite activities, allowing policies to be evaluated on different forms of object and appliance interaction.

Our reported evaluation covers 24 atomic manipulation tasks. These include eight pick-and-place tasks, six door or drawer tasks, two coffee-mug handling tasks, and eight tasks involving buttons, faucets, the sink spout, or stove controls. The set therefore covers both transporting movable objects and changing the state of articulated fixtures or appliances.

Task abbreviations. Table 3 maps the abbreviated column headers in Table 1 to their task meanings. PnP denotes pick-and-place. CM SU and CM SV refer to coffee-mug setup and serving, respectively, while CB PS refers to pressing the coffee-machine button. The numbered suffixes distinguish task names with otherwise ambiguous abbreviations; they do not represent difficulty levels. In particular, CTS1 and STC1 involve the sink, whereas CTS2 and STC2 involve the stove.

The coffee-related tasks correspond to `CoffeeSetupMug`, `CoffeeServeMug`, and `CoffeePressButton` in the benchmark implementation. For the pick-and-place tasks, the destination determines the required placement interaction. For example, counter-to-cabinet and counter-to-microwave tasks place an object inside the corresponding fixture, whereas counter-to-stove places it in a pot or pan on the stove.

A.2.2 LIBERO-Plus

LIBERO-Plus (Fei et al., 2025) extends LIBERO with controlled perturbations for evaluating the robustness of vision-language-action models. The released benchmark contains 10,030 task instances spanning seven perturbation dimensions. Its purpose is to examine whether task execution remains successful when conditions differ from the nominal setting.

The seven dimensions cover changes to scene geometry, visual appearance, language input, and robot initialization. Table 4 summarizes their meanings. This separation helps distinguish sensitivity to geometric changes, such as camera displacement, from sensitivity to appearance changes, such as lighting or texture variation.

Task-suite terminology. The LIBERO suite names distinguish different task groupings. *Spatial* emphasizes spatial relationships, *Object* emphasizes object identity, and *Goal* emphasizes the target outcome. *Long*, commonly used for LIBERO-10, denotes the longer-horizon task suite. These suite names describe the underlying manipulation tasks, whereas the perturbation dimensions describe changes applied to their execution conditions.

Reporting terminology. SR denotes success rate, and Avg. or Average denotes the aggregate result over the evaluation categories specified in the corresponding plot or table. An Original or Unper-

Abbreviation	Task meaning
PnP CTC1	Pick-and-place from counter to cabinet
PnP CTC2	Pick-and-place from cabinet to counter
PnP CTS1	Pick-and-place from counter to sink
PnP STC1	Pick-and-place from sink to counter
PnP CTS2	Pick-and-place from counter to stove
PnP STC2	Pick-and-place from stove to counter
PnP CTM	Pick-and-place from counter to microwave
PnP MTC	Pick-and-place from microwave to counter
CM SU	Set up the coffee mug at the coffee machine
CM SV	Serve the coffee mug by moving it to the counter
CB PS	Press the coffee-machine button
TSS	Turn the sink spout
OSD	Open a single door
CSD	Close a single door
ODD	Open double doors
CDD	Close double doors
OD	Open a drawer
CD	Close a drawer
TNSF	Turn on the sink faucet
TFSF	Turn off the sink faucet
TNS	Turn on the stove
TFS	Turn off the stove
TNM	Turn on the microwave
TFM	Turn off the microwave

Table 3: Definitions of the 24 RoboCasa task abbreviations used in Table 1. The ordering follows the task columns in the upper block and then the lower block of the results table.

Perturbation dimension	Description
Object layout	Changes to object placement or the arrangement of distracting objects.
Camera viewpoint	Changes to camera position, orientation, or field of view.
Robot initial state	Changes to the manipulator's initial configuration.
Language instruction	Rewording of the instruction used to specify the task.
Lighting	Changes to illumination, including intensity, direction, color, or shadows.
Background texture	Changes to the visual appearance of scene surfaces.
Sensor noise	Image corruption or degradation affecting visual observations.

Table 4: The seven perturbation dimensions of LIBERO-Plus.

turbed setting, when reported, denotes evaluation without the additional LIBERO-Plus perturbation. The benchmark-wide task count describes the released LIBERO-Plus collection; the results in Figure 9 correspond to the evaluation settings reported in that figure.

A.2.3 RLBench

RLBench (James et al., 2020) is a robot manipulation benchmark built using CoppeliaSim and PyRep. It provides language-described tasks, visual observations, robot state information, and task-specific success conditions. Its task collection supports the study of imitation learning, reinforcement learning, and multitask robot control.

Our evaluation uses 18 manipulation tasks covering object placement, insertion, stacking, articulated-object interaction, and tool use. These tasks require different combinations of target identification, spatial alignment, grasping, and sequential execution. For example, Insert Peg and

Abbreviation	Task	Abbreviation	Task
CJ	Close Jar	PD	Put in Drawer
DS	Drag Stick	PS	Put in Safe
IP	Insert Peg	SB1	Screw Bulb
MG	Meat off Grill	SB2	Slide Block
OD	Open Drawer	SS	Sort Shape
PC1	Place Cups	SB3	Stack Blocks
PW	Place Wine	SC	Stack Cups
PB	Push Buttons	SD	Sweep to Dustpan
PC2	Put in Cupboard	TT	Turn Tap

Table 5: Definitions of the 18 RLBench task abbreviations used in Table 2.

Sort Shape require alignment with a target structure, Stack Blocks and Stack Cups require ordered placement, and Sweep to Dustpan requires manipulating objects through a tool.

Task abbreviations. Table 5 defines every abbreviated task name used in Table 2. PC1 and PC2 distinguish Place Cups from Put in Cupboard. Similarly, SB1, SB2, and SB3 distinguish Screw Bulb, Slide Block, and Stack Blocks. These numerical suffixes identify different tasks rather than different variants or difficulty levels of the same task.

Several displayed task names are shortened forms of the benchmark identifiers. Drag Stick corresponds to `reach_and_drag`, Insert Peg to `insert_onto_square_peg`, Screw Bulb to `light_bulb_in`, and Sort Shape to `place_shape_in_shape_sorter`. Put in Cupboard, Put in Drawer, and Put in Safe correspond to placing groceries in a cupboard, an item in a drawer, and money in a safe, respectively.

A.3 Implementation Details and Hyperparameters

Software and Hardware Environment. The implementation is built on Ubuntu 22.04 LTS using Python 3.10 and PyTorch 2.3.0, with GPU acceleration provided by CUDA Toolkit 12.1. To ensure strict reproducibility and minimize discrepancies between training and evaluation phases, the exact same software environment is utilized across offline simulation data collection, evaluator training, and real-world deployment. Configuration states including camera calibration, action normalization statistics, task metadata, and random seeds are carefully versioned and stored with each experiment run.

For computationally intensive offline procedures, specifically the large-scale empirical branch rollouts in the resettable simulator and the dual-objective training of the evaluator, we utilize a dedicated computing cluster equipped with eight NVIDIA RTX 5880 Ada Generation GPUs (48 GB memory per GPU). While offline data collection leverages this substantial parallel capacity, our lightweight S2S-VLA evaluator enables rollout-free and computationally efficient candidate selection during test-time deployment, requiring only a single consumer-grade GPU.

Real-World Physical Setup. Real-world evaluations are conducted using a PiPER robotic arm equipped with a parallel-jaw gripper. The arm features six actuated joints; combined with the binary gripper state, the generative policy produces the standard seven-dimensional action vector defined in our state space (comprising 3D end-effector translation, 3D end-effector rotation, and 1D gripper open–close control). Sampled commands are converted to Cartesian targets and executed via the platform's standard low-level controller. The PiPER arm serves as a compact collaborative platform that provides sufficient reach and joint-level compliance for tabletop pick-and-place manipulation, safely accommodating the repeated physical execution of stochastically generated, and potentially suboptimal, candidate prefixes during evaluation.

Network Architecture and Training Details. To ensure a sufficiently diverse set of action proposals for robust candidate arbitration, we sample candidates from the frozen generative policies using a temperature of $T = 1.0$. As determined by our latency-success analysis in the main text, we sample $K = 16$ candidates per decision step.

To maintain test-time computational efficiency, our lightweight evaluator V_ψ directly reuses the frozen visual-language features e_t extracted by the VLA backbone. These features, along with the candidate prefix and context, are processed by a trainable action encoder and a 3-layer Multilayer Perceptron (MLP) with hidden dimensions of 512 and ReLU activations, which maps to the dual-head probability logits (u_s, u_d). The evaluator is trained using the AdamW optimizer with a learning rate of 3×10^{-4} and a weight decay of 1×10^{-4}. To ensure stable gradient estimates and convergence across our dual-objective formulation, we train the model with a mini-batch size of 256 context-action pairs for 20 epochs.

S2S-VLA Specific Hyperparameters. To operationalize the augmented Tchebycheff scalarization and the dual-objective training paradigm, we define the following key hyperparameters:

- *Empirical Branch Supervision:* During offline data collection, we evaluate each candidate prefix across $R = 15$ independent branch rollouts in the resettable simulator to reliably estimate the unbiased empirical event proportions $\bar{S}_{nk}$ and $\bar{D}_{nk}$.

- *Dual-Objective Optimization:* In our unified loss formulation (Equation 13), we equally balance the pointwise binary cross-entropy loss and the pairwise logistic loss by setting the preference loss weight to $\lambda_{\text{pref}} = 1.0$. The pairwise soft preference targets (Equation 11) are scaled by a temperature $\tau_{\text{pair}} = 0.1$, which provides distinguishable ordinal preference gradients without causing numerical instability or vanishing gradients.

- *Augmented Tchebycheff Scalarization:* For our test-time selection rule (Equation 7), the lower reference point is anchored at the ideal minimum $z = (0, 0, 0)$. Since the failure and anomaly probabilities naturally lie in the $[0, 1]$ interval, their objective scales are fixed at $c_1 = c_2 = 1.0$. The roughness scale c_3 is computed once as the 95^{th} percentile of the roughness values in the offline training split. This definition requires the resulting percentile to be strictly positive. The scale is then held fixed across candidate sets and data partitions. Finally, to mathematically guarantee the Pareto non-dominance of the exact minimizers (Proposition 1) while heavily penalizing the worst-case objective deviation, we set the strictly positive augmentation parameter to $\rho = 0.05$.

A.4 Real-World Experiments

To evaluate S2S-VLA on real-world manipulation tasks, we use an evaluation suite comprising three categories: Spatial, Geometry, and Robustness. Each category includes 50 expert trajectories for training, and each policy is evaluated for 20 trials per category. Success is defined by completion of the instruction-specified task. Qualitative results are presented in Figure 6, and success rates are reported in Figure 8 (a).

Spatial: The robot is instructed to identify a specified object among multiple tabletop objects and place it in a plate. Representative tasks include transferring the corn, eggplant, and apple. The scene contains objects with different shapes, colors, and grasping affordances, requiring the policy to associate the language instruction with the correct target and execute the corresponding grasp-and-place motion.

Geometry: These tasks require the robot to distinguish objects by their geometric properties and perform the requested placement. In one task, the robot must select the large cube from differently sized blocks and place it in the plate. In another, it must identify a rectangular prism and stack it onto the largest cube among multiple blocks. The stacking task additionally requires aligning the grasped object with the supporting surface and releasing it in a stable configuration.

Robustness: This category evaluates task execution in the presence of nearby objects and partial obstruction. A representative task requires picking up a potato near a basket and transferring it to the plate. The surrounding objects constrain the approach and transport motions, requiring the policy to reach the target and complete the transfer while avoiding disruptive interactions with the scene. The qualitative success and failure examples illustrate differences in whether the requested object transfer is completed.

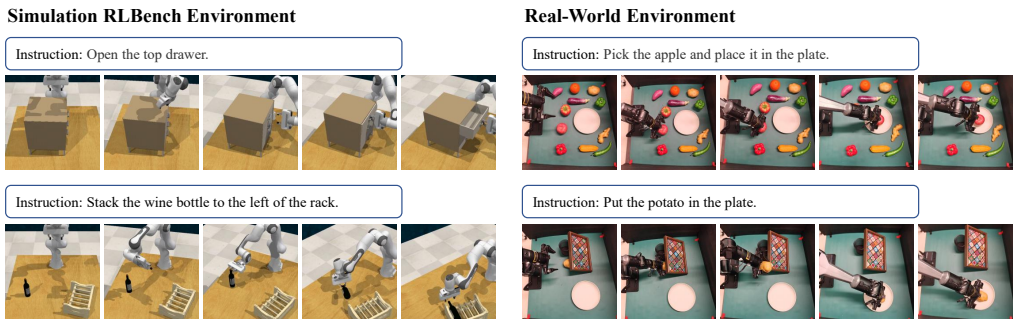


Figure 10: Additional qualitative results of S2S-VLA on simulated and real-world manipulation tasks.

A.5 S2S-VLA Pseudocode

Algorithm 1 summarizes the complete S2S-VLA pipeline. Branch rollouts are used only for offline supervision. During deployment, the controller evaluates candidate prefixes and executes the one with the lowest calibrated scalarized cost.

Empirical targets, command roughness, reference points, objective scales, and preference vectors are treated as constants during evaluator training. BCE losses are implemented from logits, and the maximum in the scalarized score uses a valid subgradient at ties. Online candidate evaluation can be batched and requires neither branch rollouts, pairwise comparisons, nor explicit Pareto pruning.

A.6 Additional Visualizations

Figure 10 presents additional visualization results.

B Proofs and Additional Derivations

This section proves the two propositions in the main text and provides additional derivations concerning supervision, optimization, finite-sample uncertainty, calibration, and candidate selection. Deterministic statements apply to the specified finite candidate set and objective vectors. Statistical statements require the sampling assumptions explicitly stated below.

B.1 Preliminaries and Basic Properties

All three objectives are minimized. For two candidates A and B, write $A \prec B$ when

$$f_i(A) \le f_i(B) \quad \text{for all } i \in \{1,2,3\}, \tag{17}$$

with a strict inequality for at least one coordinate.

Throughout the deterministic analysis, assume that candidate costs are finite, $w_i > 0$, $c_i > 0$, and $\rho > 0$, and that w, c, and z are shared across the candidates being compared. Define

$$\omega_i = \frac{w_i}{c_i} > 0. \tag{18}$$

The scalarized score can then be written as

$$G_\rho(\mathbf{f}; w) = \max_i \omega_i (f_i - z_i) + \rho \sum_{i=1}^{3} \omega_i (f_i - z_i). \tag{19}$$

Positive normalization preserves dominance. Since $c_i > 0$,

$$f_i(A) \le f_i(B) \iff \frac{f_i(A) - z_i}{c_i} \le \frac{f_i(B) - z_i}{c_i}. \tag{20}$$

Strict inequalities are preserved as well. Consequently, the common normalization used in the method does not change the Pareto dominance relation.

The dominance arguments below require a common reference point, but do not require it to be a strict lower bound. The method's choice $z = (0, 0, 0)$ is a valid lower reference because all three costs are nonnegative.

Properties of command roughness. For a nonempty prefix $B = (b_0, \ldots, b_{m-1})$, with $b_{-1} = a_{t-1}$,

$$s(B, a_{t-1}) = \frac{1}{m} \sum_{u=0}^{m-1} \|C(b_u - b_{u-1})\|_2^2 \geq 0. \tag{21}$$

Furthermore,

$$s(B, a_{t-1}) = 0 \quad \Longleftrightarrow \quad Cb_u = Cb_{u-1} \quad \text{for every } u. \tag{22}$$

This follows because a sum of nonnegative terms is zero exactly when every term is zero.

If $\|Cb_u\|_2 \leq U$ for all commands in the prefix and for the preceding command, the triangle inequality gives

$$0 \leq s(B, a_{t-1}) \leq 4U^2. \tag{23}$$

These statements concern the continuous command coordinates retained by C. They impose no restriction on excluded discrete coordinates and do not identify command roughness with realized physical jerk.

B.2 Proof of Proposition 1

Proof of Proposition 1. Suppose $A \prec B$. Define

$$\Delta_i = f_i(B) - f_i(A). \tag{24}$$

Then $\Delta_i \geq 0$ for every i, and $\Delta_j > 0$ for at least one coordinate j.

For each coordinate,

$$\omega_i(f_i(B) - z_i) = \omega_i(f_i(A) - z_i) + \omega_i \Delta_i \geq \omega_i(f_i(A) - z_i). \tag{25}$$

Taking maxima preserves this inequality:

$$\max_i \omega_i(f_i(B) - z_i) \geq \max_i \omega_i(f_i(A) - z_i). \tag{26}$$

Meanwhile, strict positivity of the weights and scales yields

$$\sum_{i=1}^{3} \omega_i \Delta_i > 0. \tag{27}$$

Therefore,

$$\begin{aligned} G_\rho(\mathbf{f}(B); w) - G_\rho(\mathbf{f}(A); w) &= \max_i \omega_i(f_i(B) - z_i) - \max_i \omega_i(f_i(A) - z_i) + \rho \sum_{i=1}^{3} \omega_i \Delta_i \\ &\geq \rho \sum_{i=1}^{3} \frac{w_i}{c_i} \big(f_i(B) - f_i(A)\big) > 0. \end{aligned} \tag{28}$$

Because the candidate set $\mathcal{C}$ is finite and nonempty and all scores are finite, an exact minimizer exists. Suppose an exact minimizer B^* were dominated by some $A \in \mathcal{C}$. Equation equation 28 would imply

$$G_\rho(\mathbf{f}(A); w) < G_\rho(\mathbf{f}(B^*); w), \tag{29}$$

contradicting the definition of B^*. Hence every exact minimizer is Pareto nondominated within $\mathcal{C}$. □

Consequences for ties and Pareto pruning. If two candidates have equal scores, neither can strictly dominate the other. They may have identical objective vectors or represent different objective trade-offs. Thus, selecting any exact minimizer by a fixed tie-breaking rule preserves the conclusion of Proposition 1.

Let $\mathcal{P}(\mathcal{C})$ denote the nondominated subset. The preceding proof also establishes

$$\arg\min_{B\in\mathcal{C}} G_\rho(\mathbf{f}(B); w) = \arg\min_{B\in\mathcal{P}(\mathcal{C})} G_\rho(\mathbf{f}(B); w). \tag{30}$$

Explicit Pareto pruning therefore does not change the exact minimizer set. It is unnecessary for obtaining the stated candidate-set guarantee.

Why positive augmentation matters. The maximum term alone need not strictly distinguish a dominated candidate. For example, let

$$\mathbf{f}(A) = (1, 0, 0), \qquad \mathbf{f}(B) = (1, 1, 0), \tag{31}$$

with $z = 0$, $c_i = 1$, and $w_i = 1/3$. Candidate A dominates B, but

$$G_0(\mathbf{f}(A); w) = G_0(\mathbf{f}(B); w) = \frac{1}{3}. \tag{32}$$

For $\rho > 0$, their score difference becomes $\rho/3 > 0$.

Strictly positive weights are also essential to the general guarantee. If a coordinate has zero weight, an improvement restricted to that coordinate can leave the score unchanged.

Recovery of an unsupported trade-off. Consider the finite objective set

$$\mathbf{f}(A) = (0, 1, 0), \qquad \mathbf{f}(B) = (0.6, 0.6, 0), \qquad \mathbf{f}(C) = (1, 0, 0). \tag{33}$$

All three candidates are nondominated. For any strictly positive weighted sum,

$$0.6(w_1 + w_2) > \min\{w_1, w_2\}, \tag{34}$$

so B cannot minimize the weighted sum.

In contrast, with $z = 0$, $c_i = 1$, and $w_i = 1/3$,

$$G_\rho(\mathbf{f}(B); w) = 0.2 + 0.4\rho, \tag{35}$$

whereas

$$G_\rho(\mathbf{f}(A); w) = G_\rho(\mathbf{f}(C); w) = \frac{1}{3} + \frac{\rho}{3}. \tag{36}$$

Thus B is the unique minimizer whenever $0 < \rho < 2$. This example demonstrates that the augmented score can select a nondominated trade-off unavailable to positive weighted sums. It does not imply that a fixed finite preference set recovers every nondominated candidate.

B.3 Proof of Proposition 2

Proof of Proposition 2. Assume

$$\hat{p}_{s,nk} = \bar{S}_{nk}, \qquad \hat{p}_{d,nk} = \bar{D}_{nk} \tag{37}$$

for every training context and candidate. The roughness coordinate s_{nk} is computed from the same prefix and preceding command in both objective vectors. Hence

$$\hat{\mathbf{f}}_{nk} = (1 - \hat{p}_{s,nk}, \hat{p}_{d,nk}, s_{nk}) = (1 - \bar{S}_{nk}, \bar{D}_{nk}, s_{nk}) = \bar{\mathbf{f}}_{nk}. \tag{38}$$

The empirical and predicted scores use the same preference, reference point, scales, and augmentation coefficient. Therefore, for every n, k, ℓ,

$$\hat{g}_{nk\ell} = G_\rho(\hat{\mathbf{f}}_{nk}; w^{(\ell)}) = G_\rho(\bar{\mathbf{f}}_{nk}; w^{(\ell)}) = g^*_{nk\ell}. \tag{39}$$

Applying this equality to both members of each pair yields, for every n, ℓ and $k < j$,

$$\hat{q}_{nkj\ell} = \sigma\left(\frac{\hat{g}_{nj\ell} - \hat{g}_{nk\ell}}{\tau_{\text{pair}}}\right) = \sigma\left(\frac{g^*_{nj\ell} - g^*_{nk\ell}}{\tau_{\text{pair}}}\right) = q^*_{nkj\ell}. \tag{40}$$

For the second claim, suppose candidate k dominates candidate j in the empirical objective space. Define

$$\Delta_i = \bar{f}_{nj,i} - \bar{f}_{nk,i}. \tag{41}$$

Then all Δ_i are nonnegative and at least one is strictly positive. Applying the strict score inequality from the proof of Proposition 1 yields

$$g^*_{nj\ell} - g^*_{nk\ell} \geq \rho \sum_{i=1}^{3} \frac{w_i^{(\ell)}}{c_i} \Delta_i > 0. \tag{42}$$

Since $\tau_{\text{pair}} > 0$ and the sigmoid is strictly increasing,

$$q^*_{nkj\ell} \geq \sigma\left(\frac{\rho}{\tau_{\text{pair}}} \sum_{i=1}^{3} \frac{w_i^{(\ell)}}{c_i} \Delta_i \right) > \frac{1}{2}. \tag{43}$$

This proves the claim for every valid preference. □

Interpretation. The result establishes compatibility of empirical targets in probability space. It does not assume that the empirical frequencies equal their population probabilities. It also does not establish that a finite neural evaluator can fit all targets exactly or that an optimizer will reach such a solution.

B.4 BCE Supervision and Joint-Loss Compatibility

Equivalence of frequency and binary supervision. Let $y_1, \ldots, y_R \in \{0, 1\}$ and $\bar{y} = R^{-1} \sum_r y_r$. For $p \in (0, 1)$,

$$\begin{aligned} \frac{1}{R} \sum_{r=1}^{R} \ell_{\text{BCE}}(p, y_r) &= -\left(\frac{1}{R} \sum_r y_r \right) \log p - \left(1 - \frac{1}{R} \sum_r y_r \right) \log(1 - p) \\ &= -\bar{y} \log p - (1 - \bar{y}) \log(1 - p) = \ell_{\text{BCE}}(p, \bar{y}). \end{aligned} \tag{44}$$

This identity is algebraic and does not require independence among the observed labels.

Separate heads do not assume independent events. For any joint distribution of (S, D), linearity of expectation gives

$$\mathbb{E}\left[\ell_{\text{BCE}}(p'_s, S) + \ell_{\text{BCE}}(p'_d, D) \mid x, B \right] = \ell_{\text{BCE}}(p'_s, p_s(x, B)) + \ell_{\text{BCE}}(p'_d, p_d(x, B)). \tag{45}$$

No factorization of the joint event distribution is used. Thus the two losses supervise marginal probabilities even when task success and local anomalies are dependent.

Entropy decomposition. For $y \in [0, 1]$, define

$$H(y) = -y \log y - (1 - y) \log(1 - y), \tag{46}$$

with $0 \log 0 = 0$, and define the Bernoulli divergence

$$\text{kl}(y \| p) = y \log \frac{y}{p} + (1 - y) \log \frac{1 - y}{1 - p}. \tag{47}$$

Then

$$\ell_{\text{BCE}}(p, y) = H(y) + \text{kl}(y \| p). \tag{48}$$

Since $\text{kl}(y \| p) \geq 0$, BCE is minimized at $p = y$ in probability space.

Define the target-dependent constants

$$H_{\text{prob}} = \frac{1}{NK} \sum_{n,k} \left[H(\bar{S}_{nk}) + H(\bar{D}_{nk}) \right] \tag{49}$$

and

$$H_{\text{pref}} = \frac{1}{NL\binom{K}{2}} \sum_{n,\ell} \sum_{k<j} H(q^*_{nkj\ell}). \tag{50}$$

The complete training objective satisfies the exact decomposition

$$\begin{aligned}\mathcal{L} - H_{\text{prob}} - \lambda_{\text{pref}} H_{\text{pref}} = \frac{1}{NK} \sum_{n,k} \left[\text{kl}(\bar{S}_{nk} \| \hat{p}_{s,nk}) + \text{kl}(\bar{D}_{nk} \| \hat{p}_{d,nk}) \right] \\ + \frac{\lambda_{\text{pref}}}{NL\binom{K}{2}} \sum_{n,\ell} \sum_{k<j} \text{kl}(q^*_{nkj\ell} \| \hat{q}_{nkj\ell}) \geq 0.\end{aligned} \tag{51}$$

By Proposition 2, exact matching of the empirical event probabilities also matches every soft preference target. All divergence terms then vanish simultaneously. Consequently, the probability and preference objectives have a common minimum in unrestricted probability space.

This conclusion concerns target compatibility, not convexity in the network parameters. A shared finite-capacity network may be unable to realize the required outputs.

Boundary targets. If an empirical proportion equals 0 or 1, exact matching requires a boundary probability. A sigmoid with finite logits produces values strictly between 0 and 1. Accordingly, the matching argument is interpreted in the closure of probability space: the corresponding BCE infimum is approached as the logit tends to the appropriate infinity. No finite-parameter attainment is implied.

B.5 Soft Preferences and Optimization Derivatives

For a fixed context and preference, define the predicted pair logit

$$z_{kj} = \frac{\hat{g}_j - \hat{g}_k}{\tau_{\text{pair}}}. \tag{52}$$

Then $\hat{q}_{kj} = \sigma(z_{kj})$. The following properties hold directly:

$$\hat{g}_k < \hat{g}_j \iff \hat{q}_{kj} > \frac{1}{2}, \qquad \hat{g}_k = \hat{g}_j \iff \hat{q}_{kj} = \frac{1}{2}. \tag{53}$$

If the reverse ordered pair is defined by the same rule, $\hat{q}_{jk} = 1 - \hat{q}_{kj}$.

Preference matching determines score differences. The sigmoid is injective. Therefore,

$$\hat{q}_{kj} = q^*_{kj} \quad \iff \quad \hat{g}_j - \hat{g}_k = g^*_j - g^*_k. \tag{54}$$

If all pairs match for a fixed context and preference, there exists a constant b such that

$$\hat{g}_k = g^*_k + b \quad \text{for every } k. \tag{55}$$

To see this, fix one reference candidate and apply the pairwise difference equality to every other candidate. Thus preference supervision constrains relative scores, while probability supervision additionally anchors the individual event probabilities.

Stable BCE and pairwise gradients. BCE can be evaluated from a logit u as

$$\ell_{\text{BCE}}(\sigma(u), y) = \text{softplus}(u) - yu, \qquad \text{softplus}(u) = \log(1 + \exp(u)), \tag{56}$$

using a numerically stable implementation. Its derivative is

$$\frac{\partial \ell_{\text{BCE}}(\sigma(u), y)}{\partial u} = \sigma(u) - y. \tag{57}$$

Holding the empirical preference target fixed therefore gives

$$\frac{\partial \ell_{\text{BCE}}(\hat{q}_{kj}, q^*_{kj})}{\partial \hat{g}_k} = -\frac{\hat{q}_{kj} - q^*_{kj}}{\tau_{\text{pair}}}, \qquad \frac{\partial \ell_{\text{BCE}}(\hat{q}_{kj}, q^*_{kj})}{\partial \hat{g}_j} = \frac{\hat{q}_{kj} - q^*_{kj}}{\tau_{\text{pair}}}. \tag{58}$$

Subgradients of the scalarization. Let

$$I(\mathbf{f}) = \arg\max_i \omega_i (f_i - z_i). \tag{59}$$

Choose coefficients $\alpha_i \geq 0$ satisfying $\sum_i \alpha_i = 1$ and $\alpha_i = 0$ outside $I(\mathbf{f})$. A valid subgradient of G_ρ is

$$v_i = \omega_i(\alpha_i + \rho). \tag{60}$$

Every component is strictly positive because $\rho > 0$.

For a unique maximizing coordinate this is the ordinary gradient. Holding command roughness fixed, the corresponding derivatives through the two probability heads are

$$\frac{\partial \hat{g}}{\partial u_s} = -v_1 \hat{p}_s(1 - \hat{p}_s), \qquad \frac{\partial \hat{g}}{\partial u_d} = v_2 \hat{p}_d(1 - \hat{p}_d). \tag{61}$$

At a maximum tie, the same expressions apply with a valid choice of the subgradient in Eq. equation 60. Targets, roughness, preferences, scales, and reference points remain fixed during these calculations.

B.6 Statistical Meaning of Branch Supervision

Snapshot-conditional probabilities. Let ξ_n denote the simulator state and other fixed information restored by a snapshot. For either event $v \in \{s, d\}$, define

$$p_v^{\xi_n}(x_n, B_n^{(k)}) = \Pr(Y_v = 1 \mid \xi_n, x_n, B_n^{(k)}), \tag{62}$$

where $Y_s = S$ and $Y_d = D$, and the fixed continuation policy is included for the success event.

Suppose the R branch outcomes for a fixed snapshot and prefix are conditionally independent and identically distributed under the prescribed rollout randomness. Writing $\bar{Y}_v = R^{-1} \sum_r Y_{v,r}$ gives

$$\mathbb{E}[\bar{Y}_v \mid \xi_n, x_n, B_n^{(k)}] = p_v^{\xi_n}, \qquad \mathrm{Var}(\bar{Y}_v \mid \xi_n, x_n, B_n^{(k)}) = \frac{p_v^{\xi_n}(1 - p_v^{\xi_n})}{R} \leq \frac{1}{4R}. \tag{63}$$

The conditional independence assumption is across branch repetitions. It does not require S and D to be independent within the same rollout.

Observable context versus restored state. If x does not determine the complete simulator state, $p_v^{\xi}(x, B)$ need not equal $p_v(x, B)$. Under the collection distribution,

$$p_v(x, B) = \mathbb{E}_{\xi \mid x, B}\left[p_v^{\xi}(x, B)\right]. \tag{64}$$

Thus a branch average is unbiased for the snapshot-conditional probability. It is also unbiased for the observable-context probability after averaging over snapshots drawn from the corresponding conditional collection distribution.

The law of total variance makes the distinction explicit:

$$\mathrm{Var}(\bar{Y}_v \mid x, B) = \frac{1}{R}\mathbb{E}_{\xi \mid x, B}\left[p_v^{\xi}(1 - p_v^{\xi})\right] + \mathrm{Var}_{\xi \mid x, B}(p_v^{\xi}). \tag{65}$$

Increasing the number of repetitions reduces branch randomness but does not remove uncertainty caused by different hidden states sharing the same observable context. When the context determines the relevant state, the second term vanishes.

Concentration for a fixed snapshot and prefix. For conditionally independent Bernoulli outcomes with mean p_v^{ξ}, the bounded-variable exponential-moment inequality gives

$$\mathbb{E}\left[\exp\big(t(Y_{v,r} - p_v^{\xi})\big) \mid \xi, x, B\right] \leq \exp(t^2/8). \tag{66}$$

Combining conditional independence with Markov's inequality,

$$\Pr(\bar{Y}_v - p_v^{\xi} \geq \epsilon \mid \xi, x, B) \leq \inf_{t>0} \exp\left(-tR\epsilon + \frac{Rt^2}{8}\right). \tag{67}$$

The minimum occurs at $t = 4\epsilon$. Applying the same argument to the lower tail yields

$$\Pr\left(|\bar{Y}_v - p_v^{\xi}| \geq \epsilon \mid \xi, x, B\right) \leq 2\exp(-2R\epsilon^2). \tag{68}$$

For N fixed contexts, K candidates per context, and two event heads, a union bound gives

$$\Pr\left(\max_{\substack{n,k\\ v\in\{s,d\}}} |\bar{Y}_{v,nk} - p_v^{\xi_n}(x_n, B_n^{(k)})| \geq \epsilon\right) \leq 4NK\exp(-2R\epsilon^2), \tag{69}$$

where the probability is conditional on the collected snapshots and sampled candidate prefixes.

In particular, with conditional probability at least $1-\delta$, all these deviations are bounded by

$$\epsilon_R = \sqrt{\frac{\log(4NK/\delta)}{2R}}. \tag{70}$$

Independence between different candidates or contexts is not required for the union bound. These bounds concern branch estimation at the collected snapshots; they do not by themselves bound prediction error on unseen contexts.

B.7 Scalarization of Expected Objectives

Fix a snapshot, context, prefix, and preference. Let

$$\mathbf{Y} = (1-S, D, s), \qquad \mathbf{f}^{\xi} = \mathbb{E}[\mathbf{Y} \mid \xi, x, B]. \tag{71}$$

The roughness coordinate is deterministic for the given command sequence.

The maximum of affine functions is convex, and the augmentation term is affine. Hence G_ρ is convex in $\mathbf{f}$. Jensen's inequality gives

$$G_\rho(\mathbf{f}^{\xi}; w) \leq \mathbb{E}[G_\rho(\mathbf{Y}; w) \mid \xi, x, B]. \tag{72}$$

The two quantities need not be equal.

For example, let $\mathbf{Y}$ take values $(1,0,0)$ and $(0,1,0)$ with equal probability, with $z=0$, $c_i=1$, and $w_i = 1/3$. Then

$$G_\rho(\mathbb{E}[\mathbf{Y}]; w) = \frac{1}{6} + \frac{\rho}{3}, \tag{73}$$

whereas

$$\mathbb{E}[G_\rho(\mathbf{Y}; w)] = \frac{1}{3} + \frac{\rho}{3}. \tag{74}$$

Thus averaging rollout scores would supervise a different quantity from scalarizing the estimated expected objectives.

The same convexity argument also implies

$$\mathbb{E}[G_\rho(\bar{\mathbf{f}}; w) \mid \xi, x, B] \geq G_\rho(\mathbf{f}^{\xi}; w). \tag{75}$$

Accordingly, unbiased empirical event frequencies do not make the nonlinear plug-in score unbiased. The method uses this finite-sample plug-in score as its empirical target, rather than treating it as an oracle score.

B.8 Error Propagation and Selection Stability

A deterministic score perturbation bound. For any two finite objective vectors $\mathbf{f}$ and $\mathbf{f}'$, define

$$d_i = |f_i - f_i'|. \tag{76}$$

The inequality

$$|\max_i a_i - \max_i b_i| \leq \max_i |a_i - b_i| \tag{77}$$

and the triangle inequality imply

$$|G_\rho(\mathbf{f}; w) - G_\rho(\mathbf{f}'; w)| \leq \max_i \omega_i d_i + \rho \sum_{i=1}^{3} \omega_i d_i. \tag{78}$$

When the roughness coordinate is shared exactly and the success and anomaly probability errors are at most ϵ_s and ϵ_d, respectively, this becomes

$$|G_\rho(\mathbf{f}; w) - G_\rho(\mathbf{f}'; w)| \le \max\{\omega_1\epsilon_s, \omega_2\epsilon_d\} + \rho(\omega_1\epsilon_s + \omega_2\epsilon_d). \tag{79}$$

Combining Eq. equation 79 with Eq. equation 70 bounds the empirical score error relative to the snapshot-conditional expected cost by

$$[\max\{\omega_1, \omega_2\} + \rho(\omega_1 + \omega_2)]\,\epsilon_R \tag{80}$$

on the corresponding high-probability event. This statement is uniform over a fixed finite preference set if the bracketed factor is replaced by its maximum over that set.

Stability of soft preference targets. Let g_k and $\check{g}_k$ be two sets of scores satisfying

$$|\check{g}_k - g_k| \le \varepsilon_k. \tag{81}$$

Since $\sigma'(u) \le 1/4$,

$$\left|\sigma\left(\frac{\check{g}_j - \check{g}_k}{\tau_{\text{pair}}}\right) - \sigma\left(\frac{g_j - g_k}{\tau_{\text{pair}}}\right)\right| \le \frac{\varepsilon_j + \varepsilon_k}{4\tau_{\text{pair}}}. \tag{82}$$

If every score error is at most ε, the bound is $\varepsilon/(2\tau_{\text{pair}})$. A smaller pair temperature therefore increases sensitivity to score error.

Moreover, if $g_j - g_k > 2\varepsilon$, then

$$\check{g}_j - \check{g}_k \ge g_j - g_k - 2\varepsilon > 0, \tag{83}$$

so the pair ordering is preserved.

Regret of approximate-score selection. Consider the same nonempty finite candidate set under two score functions g and $\check{g}$, with

$$\max_B |\check{g}(B) - g(B)| \le \varepsilon. \tag{84}$$

Let

$$B^\star \in \arg\min_B g(B) \tag{85}$$

and suppose the implemented selection satisfies

$$\check{g}(\check{B}) \le \min_B \check{g}(B) + \eta, \qquad \eta \ge 0. \tag{86}$$

Then

$$0 \le g(\check{B}) - g(B^\star) \le 2\varepsilon + \eta. \tag{87}$$

Proof. By the uniform error bound and approximate minimization,

$$g(\check{B}) \le \check{g}(\check{B}) + \varepsilon \le \check{g}(B^\star) + \eta + \varepsilon \le g(B^\star) + 2\varepsilon + \eta. \tag{88}$$

The lower bound follows from the optimality of $B^\star$. □

For exact minimization, $\eta = 0$. If $B^\star$ is unique and

$$\min_{B \ne B^\star} \big[g(B) - g(B^\star)\big] > 2\varepsilon + \eta, \tag{89}$$

then the same argument guarantees $\check{B} = B^\star$.

Exclusion of sufficiently large dominance errors. Suppose $g(B) = G_\rho(\mathbf{f}(B); w)$ and the preceding uniform score-error assumption holds. If a candidate A dominates the selected candidate $\check{B}$, then Eq. equation 28 and Eq. equation 87 imply

$$\rho \sum_i \omega_i \big(f_i(\check{B}) - f_i(A)\big) \le 2\varepsilon + \eta. \tag{90}$$

Indeed, $g(B^\star) \le g(A)$, so $g(\check{B}) - g(A) \le 2\varepsilon + \eta$.

Thus a selected candidate cannot be dominated by another candidate with a weighted dominance gap exceeding this bound. With nonzero prediction error, however, dominance with arbitrarily small gaps cannot generally be excluded. Proposition 1 remains an exact guarantee for the objective vectors actually used in selection.

B.9 Properties of Temperature Calibration

Convexity in inverse temperature. Fix one head, its calibration logits u_j, and its soft targets y_j. Introduce $\beta = 1/T > 0$. The calibration objective becomes

$$F(\beta) = \sum_j \left[\text{softplus}(\beta u_j) - y_j \beta u_j\right]. \tag{91}$$

Its first and second derivatives are

$$F'(\beta) = \sum_j u_j \big[\sigma(\beta u_j) - y_j\big], \qquad F''(\beta) = \sum_j u_j^2 \sigma(\beta u_j)\big[1 - \sigma(\beta u_j)\big] \geq 0. \tag{92}$$

Therefore calibration is convex in inverse temperature. It is strictly convex at finite β if at least one calibration logit is nonzero.

Existence of an optimizer. Convexity does not ensure that an optimizer exists in the open domain $\beta > 0$. For example, with one observation $u = 1$ and $y = 1$,

$$F(\beta) = \log(1 + \exp(-\beta)), \tag{93}$$

whose infimum is approached only as $\beta \to \infty$. For $u = 1$ and $y = 1/2$, the infimum over $\beta > 0$ is approached as $\beta \to 0$.

Equation equation 14 is understood under the assumption that, for each probability head, the calibration objective attains its minimum at a finite strictly positive temperature. This attainment assumption is additional to convexity and is not guaranteed for arbitrary calibration data.

A sufficient condition for a unique finite positive optimizer in inverse-temperature space is

$$F'(0) < 0, \qquad \lim_{\beta \to \infty} F'(\beta) > 0, \tag{94}$$

where $F'(0)$ denotes the right-hand limit at zero. Under these conditions, at least one calibration logit is nonzero, so F' is continuous and strictly increasing at finite β. The intermediate value theorem therefore gives a unique $\beta^* \in (0, \infty)$ satisfying $F'(\beta^*) = 0$. By convexity, this point is the unique global minimizer, and $T^* = 1/\beta^*$ is finite and strictly positive. If all calibration logits are zero, the objective is constant and any finite positive temperature is a minimizer.

The subsequent Pareto-consistency and selection-stability results require only that the temperatures used for scoring are fixed, finite, and strictly positive; they do not require global optimality of the calibration fit.

Preservation of coordinatewise ordering. For $p \in (0, 1)$, define

$$h_T(p) = \sigma\left(\frac{\text{logit}(p)}{T}\right). \tag{95}$$

For every $T > 0$, this function is strictly increasing. Moreover,

$$1 - h_T(p) = h_T(1 - p). \tag{96}$$

Thus the calibrated failure cost is a strictly increasing transformation of the uncalibrated failure cost, and the calibrated anomaly cost is a strictly increasing transformation of the uncalibrated anomaly cost. Roughness is unchanged.

Because the same temperature for each head is applied to all candidates, it follows that

$$\hat{\mathbf{f}}(A) \prec \hat{\mathbf{f}}(B) \quad \Longleftrightarrow \quad \tilde{\mathbf{f}}(A) \prec \tilde{\mathbf{f}}(B). \tag{97}$$

Hence positive head-specific temperature scaling preserves the predicted Pareto dominance relation and nondominated set.

This does not imply preservation of the scalarized ordering among incomparable candidates. The coordinate transformations are nonlinear and can change objective trade-offs, which motivates selecting the deployment preference after calibration.

A bound on the calibration-induced change. The sigmoid's $1/4$ Lipschitz bound gives

$$|\sigma(u/T) - \sigma(u)| \leq \frac{|u|}{4}\left|\frac{1}{T} - 1\right|. \tag{98}$$

Together with Eq. equation 79, this bounds the change in scalarized scores for fixed logits and temperatures. It bounds the change caused by calibration, not the error relative to the true event probabilities. Temperature scaling alone does not establish perfect calibration or improved task success.

B.10 Finite-Set Validation of Selection Preferences

Condition on the trained evaluator, calibration temperatures, objective scales, and a fixed preference set $\mathcal{W}$ of size L. Assume these are determined independently of the validation episodes.

Let

$$\mathrm{SR}(w) = \mathbb{E}[S_m^{\mathrm{val}}(w)] \tag{99}$$

denote the population success rate of the complete calibrated controller for preference w. Suppose that, for each fixed w, the M validation outcomes are independent Bernoulli observations from the target validation distribution. Different preferences may be evaluated on the same episodes; independence across preferences is not required.

The same bounded-variable argument as in Eq. equation 68, followed by a union bound, gives

$$\Pr\left(\max_{w\in\mathcal{W}} \left|\widehat{\mathrm{SR}}_{\mathrm{val}}(\pi_w) - \mathrm{SR}(w)\right| \geq \epsilon\right) \leq 2L\exp(-2M\epsilon^2). \tag{100}$$

Define

$$\epsilon_M = \sqrt{\frac{\log(2L/\delta)}{2M}}. \tag{101}$$

With probability at least $1-\delta$, the deviation is at most ϵ_M simultaneously for every preference.

Let

$$w^\dagger \in \arg\max_{w\in\mathcal{W}} \mathrm{SR}(w) \tag{102}$$

and let w^* be selected by Eq. equation 15. On the uniform event,

$$\mathrm{SR}(w^\dagger) - \mathrm{SR}(w^*) \leq 2\epsilon_M. \tag{103}$$

Proof. Using empirical optimality of w^*,

$$\mathrm{SR}(w^\dagger) \leq \widehat{\mathrm{SR}}_{\mathrm{val}}(\pi_{w^\dagger}) + \epsilon_M \leq \widehat{\mathrm{SR}}_{\mathrm{val}}(\pi_{w^*}) + \epsilon_M \leq \mathrm{SR}(w^*) + 2\epsilon_M. \tag{104}$$

□

This is a guarantee relative to the best preference in the fixed finite set, under the stated episode-sampling assumptions. It does not compare against all possible preferences, and does not guarantee transfer to a shifted test distribution. Independence of validation episodes must not be replaced by treating correlated snapshots or branches as independent episodes.

B.11 Closed-Loop Interpretation and Computational Cost

Continuation-policy dependence. The supervised success probability concerns execution of a prefix followed by π_{ref}. To make the dependence explicit, let U_B denote completion during prefix execution, let C_B denote reaching a nonterminal state after the prefix, and let X_B^+ be the resulting decision context. With $V^{\pi_{\mathrm{ref}}}$ denoting the continuation success probability,

$$p_s(x,B) = \Pr(U_B \mid x, B) + \mathbb{E}\left[\mathbf{1}_{C_B} V^{\pi_{\mathrm{ref}}}(X_B^+) \mid x, B\right]. \tag{105}$$

A controller that reselects candidates after the prefix generally has a different continuation value V^{π_w}. Replacing $V^{\pi_{\mathrm{ref}}}$ by V^{π_w} need not preserve the expression above.

Accordingly, exact prediction of the supervised probability does not make it the closed-loop success probability of S2S-VLA. Proposition 1 establishes candidate-set Pareto consistency, while improvements in closed-loop success must be measured using the complete controller.

Candidate count and score monotonicity. For fixed context, evaluator, calibration, and scalarization, if two nonempty valid candidate sets satisfy $\mathcal{C} \subseteq \mathcal{C}'$, then

$$\min_{B \in \mathcal{C}'} G_\rho(\tilde{\mathbf{f}}(B); w) \leq \min_{B \in \mathcal{C}} G_\rho(\tilde{\mathbf{f}}(B); w). \tag{106}$$

Adding candidates to a nested set cannot worsen the minimum predicted score. Independently resampled candidate sets need not be nested, and a lower predicted score does not itself guarantee a higher closed-loop success rate.

Budget-respecting execution. Assume integer horizons $1 \leq h \leq H$ and a positive remaining budget d_t. The choice

$$h_t = \min\{h, d_t\} \tag{107}$$

ensures that the proposed execution prefix does not exceed the remaining budget. After execution, the budget is reduced by the number of control steps actually consumed. If each nonterminal execution consumes at least one step, the controller makes at most the initial budget's number of nonterminal execution decisions. An empty valid candidate set invokes the predefined stop behavior.

Offline and online complexity. Collecting labels for N contexts, K candidates, and R repetitions requires NKR branch rollouts. This counts rollouts rather than their simulation time, which also depends on the remaining episode horizons.

For the training preference objective, empirical or predicted scores for all preferences require $\mathcal{O}(NLK)$ scalarizations, and all within-context pairs require $\mathcal{O}(NLK^2)$ comparisons. Fixed empirical targets may be precomputed.

At deployment, let $K_t = |\mathcal{C}_t| \leq K$. Shared context features are computed once per decision. The evaluator performs K_t candidate evaluations, which may be batched. For fixed action dimension, computing roughness costs $\mathcal{O}(K_t h_t)$, while the three-dimensional scalarizations and minimum-score scan cost $\mathcal{O}(K_t)$. With fixed prefix length, selection therefore scales linearly with the number of candidates. Neither branch rollouts nor the quadratic pairwise training comparisons are required online.

Algorithm 1 S2S-VLA: Offline Preparation and Online Execution

Require: Frozen VLA π_0 and feature extractor ϕ_{VLA}; training and calibration contexts with simulator snapshots; separate validation episodes; candidate count $K \geq 2$; chunk horizon H; prefix horizon $h \leq H$; branch repetitions R; positive preference set $\mathcal{W}$; $\rho > 0$, $\tau_{\mathrm{pair}} > 0$, and $\lambda_{\mathrm{pref}} \geq 0$.

Ensure: Evaluator V_ψ, temperatures T_s^*, T_d^*, selected preference w^*, and a receding-horizon controller.

Stage 1: Collect offline branch supervision

Keep related contexts and branches in the same partition, grouped by collection episode or initial condition.
for each training or calibration context $x_n = (o_n, l, a_{n-1}, d_n)$ **do**
    Set $h_n \leftarrow \min\{h, d_n\}$; sample K chunks from π_0 and extract prefixes $\{B_n^{(k)}\}_{k=1}^K$ using Eq. equation 1.
    Compute and cache $e_n \leftarrow \phi_{\mathrm{VLA}}(o_n, l)$.
    for $k = 1, \ldots, K$ **do**
        Compute $s_{nk} \leftarrow s(B_n^{(k)}, a_{n-1})$ using Eq. equation 5.
        for $r = 1, \ldots, R$ **do**
            Restore the context's simulator snapshot and independently resample the prescribed rollout randomness.
            Execute $B_n^{(k)}$, then follow $\pi_{\mathrm{ref}} = \pi_0$ until completion or timeout.
            Record S_{nkr} for completion within budget and D_{nkr} for a harmful event during the prefix.
        end for
        Compute $\bar{S}_{nk}$ and $\bar{D}_{nk}$ using Eq. equation 3; store them with $(e_n, B_n^{(k)}, a_{n-1}, d_n, s_{nk})$.
    end for
end for

Stage 2: Train the shared dual-head evaluator

Fix $z \leftarrow (0, 0, 0)$ and $c_1 = c_2 \leftarrow 1$; derive $c_3 > 0$ from the training split and keep all scales fixed.
Initialize evaluator parameters ψ.
while the evaluator training stopping criterion is not met **do**
    Sample a minibatch of training contexts and their candidates.
    Predict (u_s, u_d) and $(\hat{p}_s, \hat{p}_d)$ using Eq. equation 4.
    Form empirical and predicted cost vectors using Eq. equation 6.
    Compute $g^*_{nk\ell}$ and $\hat{g}_{nk\ell}$ for $w^{(\ell)} \in \mathcal{W}$ using Eq. equation 10.
    Compute $q^*_{nkj\ell}$ and $\hat{q}_{nkj\ell}$ for within-context pairs $k < j$ using Eq. equation 11.
    Update only ψ to minimize $\mathcal{L}_{\mathrm{prob}} + \lambda_{\mathrm{pref}}\mathcal{L}_{\mathrm{pref}}$ using Eqs. equation 8, equation 12, and equation 13.
end while

Stage 3: Calibrate probabilities and select a preference

Freeze ψ and fit separate temperatures $T_s^*, T_d^* > 0$ on the calibration split using Eq. equation 14.
for each $w \in \mathcal{W}$ **do**
    Evaluate the calibrated controller in Stage 4 with input preference w on the same M validation episodes and with the same candidate-generation budget.
    Compute $\widehat{\mathrm{SR}}_{\mathrm{val}}(\pi_w) \leftarrow M^{-1} \sum_{m=1}^M S_m^{\mathrm{val}}(w)$.
end for
Select $w^* \in \arg\max_{w \in \mathcal{W}} \widehat{\mathrm{SR}}_{\mathrm{val}}(\pi_w)$; fix all choices before testing.

Stage 4: Execute the calibrated controller with input preference w

Initialize the episode context $x_t = (o_t, l, a_{t-1}, d_t)$. During validation, use the current preference from Stage 3; during testing, use $w = w^*$.
while the task is incomplete and $d_t > 0$ **do**
    Set $h_t \leftarrow \min\{h, d_t\}$; sample K chunks and extract their length-h_t prefixes.
    Let $\mathcal{C}_t$ contain the prefixes that pass the platform's existing action-validity rules.
    if $\mathcal{C}_t = \varnothing$ **then**
        Invoke the predefined stop behavior; mark the incomplete episode unsuccessful; **break**.
    end if
    Compute $e_t \leftarrow \phi_{\mathrm{VLA}}(o_t, l)$.
    for each $B \in \mathcal{C}_t$ **do**
        $(u_s, u_d) \leftarrow V_\psi(e_t, B, a_{t-1}, d_t)$.
        $\tilde{p}_s \leftarrow \sigma(u_s/T_s^*)$, $\tilde{p}_d \leftarrow \sigma(u_d/T_d^*)$.
        $\tilde{\mathbf{f}}_t(B) \leftarrow (1 - \tilde{p}_s, \tilde{p}_d, s(B, a_{t-1}))$.
        $g_t(B) \leftarrow G_\rho(\tilde{\mathbf{f}}_t(B); w)$.
    end for
    Select $B_t^* \in \arg\min_{B \in \mathcal{C}_t} g_t(B)$, breaking ties by a fixed candidate order.
    Execute B_t^*, stopping early if the task completes or the episode terminates.
    Obtain the new observation, retain the last executed command, and subtract the consumed control steps from the budget.
    Update the decision context and advance t.
end while