LEARNING ACTION–CONSEQUENCE REPRESENTATIONS FOR VISION-LANGUAGE-ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing future-oriented vision-language-action (VLA) methods typically predict actions and future observations in parallel from the same context, without explicitly modeling how actions lead to subsequent changes in the environment. Consequently, a model may learn to anticipate what will happen next largely from scene context, without learning what will happen as a consequence of a particular action. To address this limitation, we propose a framework for learning action–consequence representations (ACE) from future supervision. ACE models future states as consequences of actions and contrasts demonstrated actions with alternative actions under the same context and observed consequence, enabling future representations to distinguish different actions and their corresponding environmental changes. ACE further emphasizes action-relevant changes by weighting interaction regions and uses task-terminal states to constrain the task relevance of consequence representations. We evaluate different forms of future supervision on LIBERO and RoboCasa under a unified training budget. Under a unified training and evaluation protocol, ACE improves the average success rate on LIBERO by 6.4 percentage points over a matched VLA baseline. On RoboCasa, ACE outperforms VLA and FOCA baselines by 5.0 and 8.0 percentage points, respectively. These results suggest that modeling how actions lead to subsequent environmental changes enables future supervision to translate more effectively into control performance than independent future prediction.

1 INTRODUCTION

Vision-language-action (VLA) models learn to map visual observations and language instructions to robot actions, providing a general interface for acquiring diverse manipulation skills (Kim et al., 2024; Black et al., 2024). Beyond the demonstrated actions themselves, robot trajectories also record the environmental changes that follow them: every executed action is paired with an observed transition of the surrounding world. These future observations therefore provide a natural source of supervision for learning representations that may better support control (Hafner et al., 2020). Motivated by this idea, existing future-oriented VLA methods exploit future information through future observation prediction (Wu et al., 2024; Zhang et al., 2026b), visual subgoal generation (Zhao et al., 2025), or latent future alignment (Zheng et al., 2025). However, *predicting the future is not necessarily the same as understanding the consequence of an action*. In expert trajectories, the current scene, task instruction, and trajectory progress can already strongly constrain what happens next. For example, given a closed drawer and an instruction to open it, an open-drawer future may be highly predictable without requiring the model to distinguish which candidate robot motion actually produces that transition. A model can therefore learn *what is likely to happen next* while remaining insensitive to *what will happen as a consequence of a particular action*. This distinction is particularly important for control: choosing between candidate actions requires reasoning about how their consequences differ (Assran et al., 2025), rather than merely anticipating a plausible continuation of the demonstrated trajectory. We therefore argue that the value of future supervision depends not only on whether the future can be predicted accurately, but on whether future representations explicitly capture the dependency between actions and the environmental changes that follow them.

As shown in Fig. 1, we formulate this objective as an *action–consequence representation learning* problem and introduce **ACE**, a framework for learning **A**ction–**C**onsequence r**E**presentations from future supervision. ACE conditions future representations on both the current context and the

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.

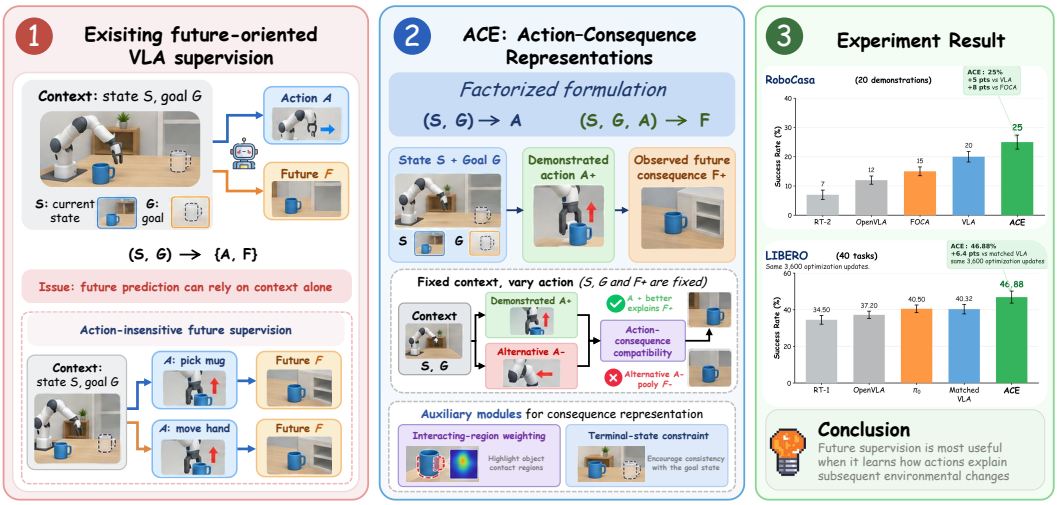


Figure 1: Unlike prior VLA methods that predict actions and futures in parallel, ACE treats the future as a consequence of the chosen action and learns action-sensitive future representations, leading to improved manipulation performance on LIBERO and RoboCasa under the same training budget.

demonstrated action sequence, and contrasts the demonstrated action with alternative actions while holding the context and observed future consequence fixed. The demonstrated action is encouraged to produce a consequence representation that is more compatible with the observed future than those induced by alternative actions. This within-context comparison is important: because every candidate shares the same visual observation, language instruction, and future target, contextual regularities alone cannot resolve which action is most consistent with the observed consequence. A predictor that ignores its action input therefore cannot satisfy the contrastive objective simply by learning the most likely future under the scene and task context, a form of shortcut learning (Geirhos et al., 2020). Instead, the representation must capture information that distinguishes candidate actions through their compatibility with the subsequent environmental change. Importantly, ACE requires only the observed future of the demonstrated trajectory and does not require executing alternative actions or obtaining their counterfactual ground-truth outcomes.

Action sensitivity alone, however, is not sufficient for learning useful consequence representations. First, manipulation-induced changes often occupy only a small portion of the visual scene, whereas most pixels may remain unchanged across consecutive observations. A future-learning objective dominated by static scene content may therefore underemphasize precisely the regions that reveal the effect of an action. Second, not every observable change is equally relevant to accomplishing the instructed task. ACE addresses these two issues by emphasizing interaction regions when learning future representations and by using task-terminal states to constrain the semantic relevance of the predicted consequences. These components complement action contrast in a natural way: action contrast identifies *which action* is consistent with the future, interaction-weighted supervision emphasizes *what changed* because of the interaction, and terminal-state supervision encourages the representation to preserve *whether that change matters* for the task outcome. Together, ACE explicitly connects three quantities that standard future prediction may only weakly couple: the *action taken*, the *environmental change it produces*, and the *task outcome that change contributes to*. In this way, ACE turns future supervision from an auxiliary objective for predicting what comes next into a structured learning signal for shaping action-sensitive control representations.

We evaluate ACE on LIBERO (Liu et al., 2023) and RoboCasa (Nasiriany et al., 2024) under unified training budgets and matched evaluation protocols, allowing us to compare different forms of future supervision under controlled conditions. On LIBERO, ACE improves the average success rate by 6.4 percentage points over a matched VLA baseline. On RoboCasa, ACE outperforms the VLA and FOCA baselines by 5.0 and 8.0 percentage points, respectively. Across both benchmarks, the results support a consistent conclusion: future supervision becomes more effective for control when the learned future representation is explicitly grounded in the action that precedes it. More broadly, these findings suggest that modeling how actions lead to subsequent environmental changes provides

a stronger supervisory signal for policy learning than treating future prediction as an independent auxiliary task.

Our contributions are threefold:

- ***Action–consequence formulation.*** We identify a fundamental mismatch in future-supervised VLA learning: predicting the demonstrated future need not require understanding which action produced it. We formulate future supervision instead as learning representations that explicitly connect actions to their observed consequences.
- ***ACE.*** We introduce an action–consequence learning framework that uses within-context action contrast to enforce action sensitivity, interaction-weighted supervision to emphasize manipulation-induced changes, and terminal-state supervision to preserve task relevance.
- ***Empirical evidence.*** Across LIBERO and RoboCasa, ACE improves control over VLA and future-oriented baselines, indicating that action–consequence modeling provides a more effective form of future supervision for robot control than independent future prediction.

2 Related Work

VLA adaptation. Pretrained policies transfer visual and language representations to robot control (Team et al., 2024; Qu et al., 2025; Kim et al., 2025; Zhang et al., 2025a; Ding et al., 2026; Zhang et al., 2025b; Brohan et al., 2023; Jiang et al., 2023). Object-centric adaptation and value-guided selection further address few-shot transfer (Li et al., 2025b; Xu et al., 2026). We hold the policy and update budget fixed to study auxiliary supervision across demonstration budgets.

Future supervision and world modeling. Visual or latent future prediction can support policy learning (Nguyen et al., 2026; Chi et al., 2025; Chan et al., 2026; Du et al., 2023; Li et al., 2025a). Action-world models extend this direction through autoregressive prediction (Yang et al., 2024; Cen et al., 2025), latent dynamics (Hafner et al., 2019; Hansen et al., 2024; Chen et al., 2026), and unified embodied representations (Zhang et al., 2026c; Lu et al., 2026). Behavior tracing (Shi et al., 2026), embodied perception and spatial reasoning (Shan et al., 2026; Qi et al., 2026), and world-model evaluation (Fan et al., 2026; Zou et al., 2026; Zhang et al., 2026a) motivate diagnostics beyond task success. ACE studies within-context action contrast; parallel prediction names our control, not all future-oriented methods.

Counterfactual supervision. Counterfactual augmentation modifies environment factors or trajectories (Urpí et al., 2024; Ameperosa et al., 2025; Shin et al., 2026; Glossop et al., 2026). ACE instead uses alternative actions as discriminative candidates without their ground-truth outcomes. Separate same-state interventions evaluate agreement with executed consequences.

3 Methodology

As shown in Figure 2, ACE organizes auxiliary supervision around three operations: selecting interaction-relevant future targets, distinguishing action consequences, and transferring the resulting gradients into shared policy representations. The policy update budget and trainable backbone configuration are fixed across the native VLA and ACE comparisons.

3.1 Native Policy and Training-Only Consequence Head

Let o_t, s_t, and q denote camera observations, proprioception, and the instruction. The policy retains the native π_0 flow-matching action head (Lipman et al., 2022). Its context tokens are processed without additional future or summary tokens. We pool the valid final context tokens to obtain h_t and attach a separate auxiliary predictor during training. Low-rank adapters (Hu et al., 2021) in the language backbone are shared by the policy and auxiliary objectives; the visual teacher is frozen. Auxiliary gradients thus shape control representations without extra inference-time tokens. The action expert and its adaptation settings are identical to the corresponding settings of the shared-adapter VLA control.

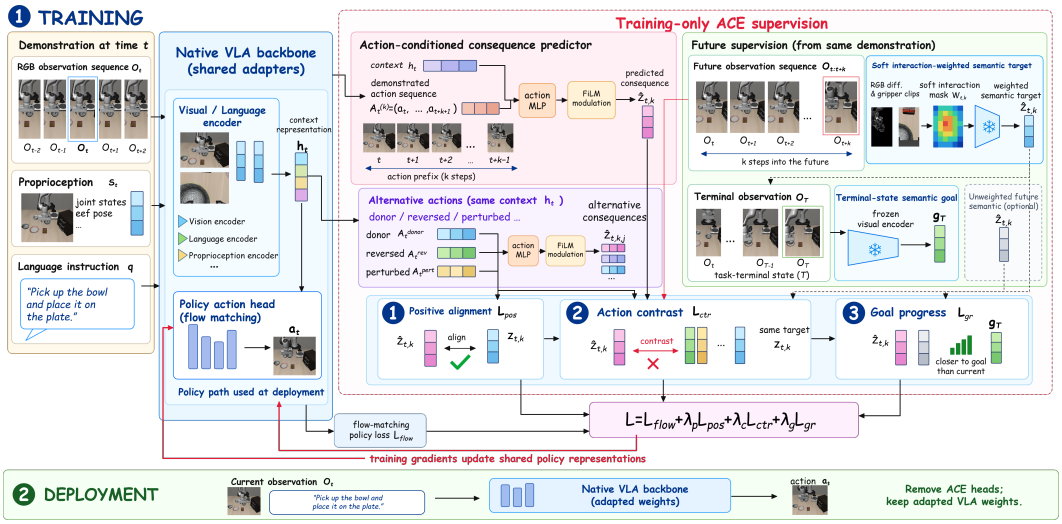


Figure 2: **ACE overview.** Demonstrated and alternative actions are contrasted against interaction-weighted future targets; terminal observations supervise task relevance. Shared adapters transfer auxiliary gradients to the native policy. ACE-specific heads are removed at deployment.

We use a fixed offset $k = h$ per model, with h selected from $\{5, 10, 20, 30, 40, 50\}$. The training sample includes o_{t+h} and the terminal observation o_T from the same demonstration. The indicator $m_{t,k} = \mathbf{1}[t + k \leq T],\ k \in \mathcal{H}$ masks targets beyond the episode boundary. We retain the index $k \in \mathcal{H}$ below to cover both the main singleton setting $\mathcal{H} = \{h\}$ and the joint-horizon comparison. Future and terminal observations supply training targets; deployment uses the current policy inputs.

3.2 Soft Interaction-Weighted Semantic Targets

For each camera, we compute a smoothed, peak-normalized RGB difference $D_{t,k} \in [0,1]^{H\times W}$. The aperture proxy is the mean absolute magnitude of the two finger positions, $b_t = \frac{1}{2}(|s_t^{\text{finger1}}| + |s_t^{\text{finger2}}|)$; using magnitudes avoids cancellation between oppositely signed joints. We define scalar aperture-change and closure gates as

$$c_{t,k} = \sigma\left(\frac{|b_{t+k} - b_t| - 10^{-2}}{10^{-2}}\right), \qquad r_t = \sigma\left(\frac{2.0\times10^{-2} - b_t}{2.0\times10^{-2}}\right). \tag{1}$$

These two gates encode aperture change and closure cues. A Gaussian $S_{t,k}$ with width 2.5×10^{-1} in normalized image coordinates is centered at the motion-weighted image centroid. The soft interaction weights are

$$W_{t,k} = \sigma\left(\frac{D_{t,k} + 0.5c_{t,k}D_{t,k} + 2.5\times10^{-1}r_tS_{t,k} - 3.5\times10^{-1}}{0.1}\right). \tag{2}$$

We use a uniform mask when the maximum smoothed, unnormalized frame difference is below 10^{-3}. This fallback avoids amplifying numerical noise in static scenes. The interaction-weighted target construction uses neither oracle object masks nor object bounding-box crops.

Let $p_{t+k,v,n} \in \mathbb{R}^{1152}$ be patch n from camera v, obtained from the final hidden states of the frozen SigLIP visual encoder (Zhai et al., 2023). We average-pool $W_{t,k,v}$ onto the patch grid and form

$$\begin{aligned} \text{Norm}(u) &= \frac{u}{\|u\|_2}, \qquad \|u\|_2 > 0, \\ z_{t,k} &= \text{Norm}\left[\frac{1}{V}\sum_{v=1}^{V}\text{Norm}\left(\frac{\sum_n w_{t,k,v,n}p_{t+k,v,n}}{\sum_n w_{t,k,v,n} + \epsilon}\right)\right]. \end{aligned} \tag{3}$$

All teacher outputs are stop-gradient targets. No randomly initialized target projection is used. We also compute unweighted semantic vectors, related to visual value and reward representation

learning (Ma et al., 2023b;a; Wang et al., 2026) $\bar{z}_{t,k} = \mathrm{Norm}(\mathrm{Mean}_{v,n}\, p_{t+k,v,n})$ and $g_T = \mathrm{Norm}(\mathrm{Mean}_{v,n}\, p_{T,v,n})$ for goal progress supervision. Teacher images are unaugmented and resized to 224×224; native policy image augmentation remains separate.

3.3 Action-Conditioned Consequences and Contrast

For horizon k, only actions preceding that horizon are supplied to the predictor: $A_t^{(k)} = [\mathbf{a}_t, \ldots, \mathbf{a}_{t+k-1}, \mathbf{0}, \ldots, \mathbf{0}] \in \mathbb{R}^{50 \times d_a}$ is padded to a fixed capacity of 50 actions for every horizon setting. Later actions therefore cannot leak into earlier predictions. An action MLP generates FiLM (Perez et al., 2017) parameters and a learned horizon embedding e_k specifies the offset:

$$\begin{aligned} (\gamma_{t,k}, \beta_{t,k}) &= F_\phi(E_\phi(A_t^{(k)})), \\ \widehat{z}_{t,k} &= \mathrm{Norm}\big(P_\phi\big(C_\phi(h_t) \odot (\mathbf{1} + \gamma_{t,k}) + \beta_{t,k} + e_k\big)\big), \\ C_\phi(h_t), \gamma_{t,k}, \beta_{t,k}, e_k \in \mathbb{R}^d, &\qquad P_\phi : \mathbb{R}^d \to \mathbb{R}^{1152}. \end{aligned} \tag{4}$$

The final FiLM layer is zero-initialized. Its output dimension and the teacher target dimension are both fixed independently of task identity. The positive objective is a masked cosine loss,

$$\begin{aligned} \mathcal{V} &= \{(t,k) : k \in \mathcal{H},\ m_{t,k} = 1\}, \\ \mathcal{L}_{\mathrm{pos}} &= \frac{\sum_{(t,k)\in\mathcal{V}} (1 - \widehat{z}_{t,k}^\top z_{t,k})}{|\mathcal{V}| + \epsilon}, \qquad \epsilon > 0. \end{aligned} \tag{5}$$

We generate synthetic alternatives using batch action donors, temporal reversal, and Gaussian perturbations. An alternative whose relevant action prefix equals the expert prefix is excluded. Holding h_t fixed, the same head produces $\widehat{z}^{-}_{t,k,j}$. With normalized representations and a learned inverse temperature $0 < s \le 100$, we use (Oord et al., 2018)

$$\begin{aligned} \ell_{t,k} &= -\log \frac{\exp(s\, z_{t,k}^\top \widehat{z}_{t,k})}{\exp(s\, z_{t,k}^\top \widehat{z}_{t,k}) + \sum_{j\in\mathcal{N}_{t,k}} \exp(s\, z_{t,k}^\top \widehat{z}^{-}_{t,k,j})}, \\ \mathcal{L}_{\mathrm{cf}} &= \frac{\sum_{(t,k)\in\mathcal{I}} \ell_{t,k}}{\max\{1, |\mathcal{I}|\}}, \qquad \mathcal{I} = \{(t,k) \in \mathcal{V} : |\mathcal{N}_{t,k}| \ge 1\}. \end{aligned} \tag{6}$$

where $\mathcal{I}$ includes valid horizons with alternatives indexed by $\mathcal{N}_{t,k}$. Held-out policy-reference and time-reversed actions, together with simulator execution, evaluate the transfer of this discrimination.

3.4 Goal Progress at a Fixed Horizon

We use the demonstrated terminal image as a semantic goal reference and define

$$y_{t,k} = \frac{1 + \bar{z}_{t,k}^\top g_T}{2}, \qquad \widehat{y}_{t,k} = \sigma(R_\omega[\widehat{z}_{t,k} \,\|\, g_T \,\|\, e_k]). \tag{7}$$

The target measures semantic proximity to the demonstrated terminal scene. At the selected offset, the progress objective is

$$\begin{aligned} \mathcal{V}_h &= \{t : m_{t,h} = 1\}, \\ \mathcal{L}_{\mathrm{gr}} &= \frac{\sum_{t\in\mathcal{V}_h} (\widehat{y}_{t,h} - y_{t,h})^2}{\max\{1, |\mathcal{V}_h|\}}. \end{aligned} \tag{8}$$

This couples predicted consequences to the task endpoint. Joint-horizon training can add a teacher-gated ordering penalty between valid offsets; the singleton method uses only progress regression.

3.5 Optimization and Deployment

The implemented objective is

$$\begin{aligned} \mathcal{L} &= \mathcal{L}_{\mathrm{flow}} + \lambda_{\mathrm{p}}\mathcal{L}_{\mathrm{pos}} + \lambda_{\mathrm{c}}\mathcal{L}_{\mathrm{cf}} + \lambda_{\mathrm{g}}\mathcal{L}_{\mathrm{gr}}, \\ (\lambda_{\mathrm{p}}, \lambda_{\mathrm{c}}, \lambda_{\mathrm{g}}) &= (0.1, 1, 0.1) \qquad \text{(reported LIBERO settings)}. \end{aligned} \tag{9}$$

At deployment, we remove ACE heads and retain the adapted VLA weights and normalization statistics. Shared adapters remain, and the exported policy uses the native token sequence and action-generation graph. Training supplies extra supervision; deployment uses the native policy.

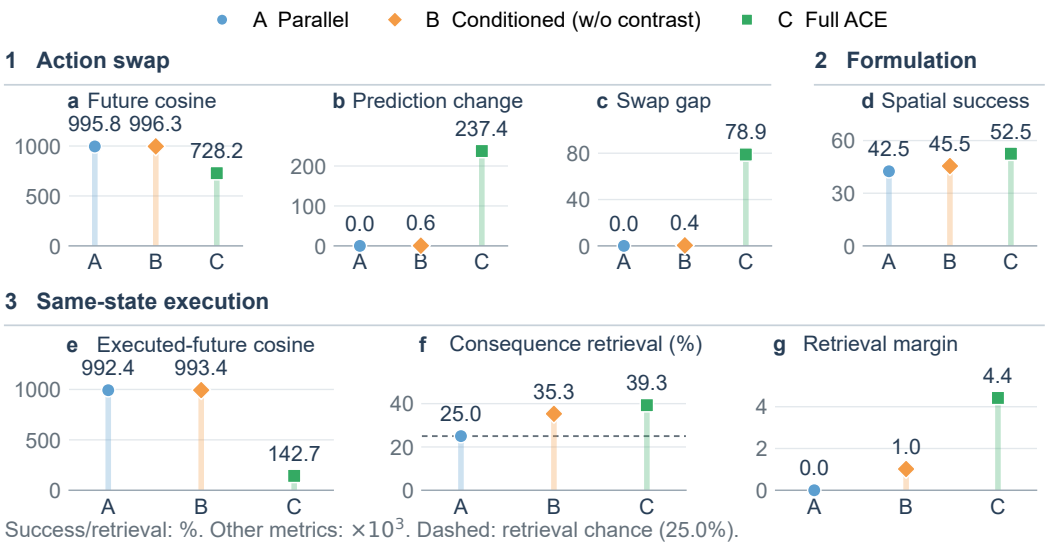


Figure 3: **Action dependence and executed-consequence fidelity.** A: Parallel; B: Conditioned without contrast; C: Full ACE. Action swaps use 295 held-out contexts with three donors each; control uses 200 Spatial rollouts/model; same-state execution uses 89 valid states across nine tasks and four actions/state. Cosines, prediction changes, swap gaps, and matching margins are scaled by 10^3; control and retrieval are percentages. ACE increases action sensitivity, control success, and relative outcome matching while reducing absolute feature agreement.

4 Analysis of Action–Consequence Representations

We test action dependence, its relationship to control, and correspondence to executed outcomes in this order. A (Parallel) uses a zero action input, B (Conditioned) uses the demonstrated prefix without contrast, and C (Full ACE) adds contrast. They share the backbone, rank-8 language adapters, action adaptation, allocated head size, targets, progress objective, $h = 30$, 20 demonstrations/task, and 3,600 updates. A's effective action-dependent capacity differs by construction. Figure 3 summarizes the results; protocols and tabulated values appear in Appendix C.

4.1 Action Swap Test

Holding $X = (o_t, s_t, q)$ fixed, we replace A with same-task, phase-matched donor prefixes $\widetilde{A}$ selected without future features or model scores. For normalized predictions $p = f(X, A)$, $\widetilde{p} = f(X, \widetilde{A})$ and the demonstrated target z, define

$$C_{\text{future}} = \mathbb{E}[p^\top z], \quad D_{\text{pred}} = \mathbb{E}[1 - p^\top \widetilde{p}], \quad \Delta_{\text{swap}} = \mathbb{E}[p^\top z - \widetilde{p}^\top z]. \tag{10}$$

The donor protocol compares action alternatives under similar current conditions. Eligible prefixes come from separate episodes of the same task and match normalized phase and gripper class; selection uses current state and action information only. The three nearest eligible donors by state satisfy the prescribed state- and action-distance thresholds. Five of 300 proposed contexts lack three eligible donor episodes, leaving 295 contexts from 100 episodes shared across models. The persistence reference pools current-image teacher patches with the target's current-to-future mask. It therefore uses target-derived pooling weights and serves as a diagnostic reference rather than an online predictor that could infer the same mask from the current frame alone.

Donors are averaged within context, then episode and task. On the displayed $\times 10^3$ scale, A and B achieve future cosines of 995.8 and 996.3, exceeding the persistence reference of 988.4, yet their swap gaps are only 0.0 and 0.4. Thus strong feature agreement can coexist with weak action dependence. ACE raises prediction change to 237.4 and the swap gap to 78.9, but lowers future cosine to 728.2. These metrics establish action discrimination, not accuracy on unexecuted donor outcomes.

4.2 Parallel versus Action-Conditioned Future

A implements $(S, G) \to F$, B implements $(S, G, A) \to F$, and C adds contrast; (S, G) denotes current state and instruction. Conditioning increases observed Spatial success from 42.5% to 45.5%; contrast raises it to 52.5%. The C–B comparison isolates the contrast setting, giving a 7.0-point control gain and a 78.4 increase in swap gap on the displayed scale. Lower cosine means the result supports stronger action dependence and control success, not improved future-feature fidelity.

Within-context contrast provides a constraint absent from positive-only prediction. Candidates share the context and target, so any shared additive context-dependent score cancels from the loss. An action-invariant predictor incurs $\log(J + 1)$ for J valid negatives, irrespective of target alignment (Appendix B). Reducing this loss requires discriminating actions within the same context, although distinct actions can have identical physical outcomes. Same-state execution tests whether that discrimination corresponds to observed consequences under matched simulator states.

4.3 Same-State Execution

An independent VLA reference prefix and three matched demonstration alternatives execute for 30 steps from the same restored simulator and controller state. A repeated reference checks restoration; all models use the same 89 valid states and candidate-independent pooling masks. For predictions p_i and executed outcomes z_j^{exec},

$$M_{ij} = p_i^\top z_j^{\text{exec}}, \quad C_{\text{exec}} = \mathbb{E}_i[M_{ii}], \quad R_{\text{margin}} = \mathbb{E}_i\left[M_{ii} - \frac{1}{3}\sum_{j \neq i} M_{ij}\right]. \tag{11}$$

Retrieval chooses the row maximum, with fractional tie credit and a 25.0% chance reference. A has high absolute cosine but chance retrieval. B and C reach 35.3% and 39.3% retrieval, with matching margins of 1.0 and 4.4 on the displayed scale. C's executed-future cosine falls to 142.7, versus B's 993.4. Predictions contain candidate-specific outcome information but do not establish uniformly accurate alternative-future prediction. Pooling and outcome-distinct sensitivity analyses are in Appendix C.5. These tests distinguish action sensitivity from complete counterfactual dynamics.

Restoration checks fix the denominator before scoring. Of 100 attempted states, 89 pass: all ten top-drawer states fail restoration, and one stove state exceeds the pixel tolerance. Exclusions are model independent; identical or unsuccessful candidate outcomes among valid states remain included. Within each state, all queries share a symmetric mask averaged from motion across the four outcomes. Under uniform pooling and on the outcome-distinct subset, ACE retains higher retrieval and margins but lower absolute cosine (Table 1). Relative outcome matching remains distinct from accurate prediction of individual executed futures for each candidate action.

Table 1: Executed-consequence robustness to pooling and outcome separation. Metrics are task-macro averages. Uniform pooling uses all 89 valid states; the outcome-distinct subset uses local pooling on 85 states. B is action-conditioned; C adds contrast (ACE).

Evaluation	Model	Cosine ($\times 10^3$)	Retrieval (%)	Margin ($\times 10^3$)
Uniform, all 89	B	988.0	31.7	0.5
	C	152.6	37.9	2.0
Local, distinct 85	B	993.3	35.3	1.0
	C	141.3	39.5	4.4

5 Experiments

5.1 Experimental Setup

Benchmarks and training. We jointly train on LIBERO's Spatial, Object, Goal, and Long suites (40 tasks) using nested subsets of 50, 20, or 5 demonstrations/task. RoboCasa uses five MimicGen (Mandlekar et al., 2023) household tasks with 100, 20, or 5 demonstrations/task. Success averages tasks equally, with 20 initial states/task: 800 LIBERO and 100 RoboCasa rollouts/setting.

Table 2: LIBERO success (%) across demonstration budgets: joint 40-task training, 3,600 updates, 200 rollouts/suite. Left: 50-demo suite results and their Avg. Right: suite means by budget and their Avg. Averages use equal weights and unrounded values. Bold marks the best result per column.

	50 demos/task					Mean by demos/task			
Method	Spatial	Object	Goal	Long	Avg.	50	20	5	Avg.
VLA	51.5	45.0	40.5	25.0	40.5	40.5	40.8	**40.4**	40.5
FOCA	46.0	48.0	20.0	17.5	32.9	32.9	41.3	34.6	36.3
LoRA (rank 8)	54.5	51.0	**48.5**	**29.5**	45.9	45.9	42.8	39.9	42.8
ACE, $h = 30$	**65.0**	**56.5**	45.0	21.0	**46.9**	**46.9**	**45.4**	39.3	**43.8**

Runs use 3,600 updates on four A800 GPUs, microbatch 13, five accumulation steps, batch 260. Policies generate 50-action chunks and replan every ten actions. ACE uses seven negatives and $h = 30$, selected in a Spatial-only development sweep. We evaluate final checkpoints; equal updates do not imply equal training time.

Comparisons. All baselines use our pipeline. Original VLA, the additional LoRA run, and ACE use native π_0, language adapters of rank 8 with $\alpha = 16$, action adapters of rank 16 with $\alpha = 32$, and the configured trainable dense action-expert weights. VLA and LoRA remain separate runs. FOCA uses explicit future supervision with weight 0.1 and a different trainable parameter count. Appendix D.1 details the adaptation audit and complete suite-level results.

5.2 Control under a Fixed Update Budget

At 50 demonstrations/task, ACE reaches 46.9% mean success, compared with 40.5% for original VLA and 45.9% for LoRA (Table 2). Gains are 6.4 and 1.0 points, computed before rounding. ACE leads Spatial, Object, and the mean among these methods; LoRA leads Goal and Long. The small mean difference and suite-dependent ordering limit claims of general superiority.

5.3 Demonstration Coverage and Household Manipulation

Rankings vary with data coverage (Table 2). At 20 LIBERO demonstrations/task, ACE exceeds VLA by 4.6 points and LoRA by 2.6 points; at five, it trails both. The gains at 20 and 50 demonstrations therefore do not extend uniformly to the smallest budget. Appendix D.1 reports each suite separately for all three budgets, clarifying where the aggregate gains and regressions arise.

On RoboCasa at 20 demonstrations/task, ACE improves over VLA and FOCA by 5.0 and 8.0 points (Table 3). At 100 demonstrations it trails VLA by 6.0 points; at five it trails both references. These five-task results support a benefit at the intermediate budget, not a uniform transfer advantage.

Table 3: RoboCasa success (%) for ACE and the original VLA and FOCA references. Each method uses 100 rollouts per budget across five tasks; task-level results appear in Figure 6.

Method	100 demos/task	20 demos/task	5 demos/task
VLA	32.0	20.0	6.0
FOCA	15.0	17.0	10.0
ACE, $h = 30$	26.0	25.0	3.0

5.4 Component Ablations

Removing action conditioning, contrast, or interaction weighting reduces the four-suite mean from 46.9% to 43.6%, 44.1%, or 44.8% (Table 4). Removing goal progress changes the mean by 0.5 points but reduces Spatial by 9.5, showing task-dependent effects. FiLM removal also makes contrast non-discriminative; the A/B/C analysis isolates conditioning from contrast. The added RoboCasa comparison tests whether component effects transfer across environments.

Table 4: ACE ablations, success (%), $h = 30$. LIBERO: joint 40-task training, 50 demos/task, 200 rollouts/suite. RoboCasa: 100 demos/task, 100 rollouts over five tasks.

Variant	Spatial	Object	Goal	Long	Avg.	RoboCasa
Full ACE	65.0	56.5	45.0	21.0	**46.9**	26.0
Without action conditioning	53.5	51.0	49.5	20.5	43.6	27.0
Without action contrast	55.5	53.0	47.0	21.0	44.1	22.0
Without goal progress	55.5	54.5	52.5	23.0	46.4	25.0
Uniform spatial weights	61.0	46.5	53.5	18.0	44.8	23.0

RoboCasa shares the contrast and weighting effects: removing them reduces success by 4.0 and 3.0 points. Removing conditioning instead improves success by 1.0 point. Goal-progress effects also vary: its removal improves LIBERO Goal from 45.0% to 52.5% and Long from 21.0% to 23.0%, but lowers RoboCasa from 26.0% to 25.0%. These results support task-dependent contributions rather than a uniform benefit from every auxiliary component across task families.

5.5 Design Choices and Deployment Cost

Interaction weighting raises Spatial success from 61.0% to 65.0%, while reference cosine falls from 819.8 to 770.8 on the $\times 10^3$ scale. This compact diagnostic uses a common automatic target for both models, so better control need not imply higher reference-feature agreement. Its test episodes differ from the matched A/B/C contexts, and the cosine values are separate diagnostics (Appendix D.3).

Table 5: Horizon diagnostics, success (%). Spatial-only development selects $h = 30$; RoboCasa uses 100 demos/task and 100 rollouts/model. Joint supervises $\{5, 10, 20\}$. These are configuration comparisons within each row, with horizon-dependent target coverage.

Horizon h	5	10	20	30	50	Joint
Spatial development	64.0	60.5	65.0	70.0	61.5	56.5
RoboCasa	25.0	26.0	26.0	26.0	30.0	31.0

The Spatial development maximum occurs at $h = 30$ (Table 5), whereas RoboCasa favors joint supervision among the tested settings. Spatial valid-target coverage falls from 96.0% at $h = 5$ to 59.8% at $h = 50$; the scan therefore does not isolate temporal distance. We retain the Spatial-selected $h = 30$ on RoboCasa, without retuning the horizon on household evaluation outcomes. Singleton heads share a 50-step input capacity. Joint supervision improves RoboCasa by 5.0 points relative to $h = 30$, suggesting that horizon choices can vary across environments.

At fixed $h = 30$, reducing training negatives from seven to three lowers Spatial success from 65.0% to 51.5%. Progress weights 0, 3.0×10^{-2}, and 0.1 yield 55.5%, 61.0%, and 65.0%. Two-candidate pairwise ranking stays at 100.0% for all settings, showing that this diagnostic does not resolve their control differences across these one-factor changes in training (Appendix D.5).

The removable head adds 4,958,850 parameters, 0.9% of VLA's trainable count. Export preserves outputs on 20 paired inputs with identical action noise. On an A800 at batch 1, ACE's p50/p95 latency is 300.2/303.6 ms, versus VLA's 297.5/300.5 ms, with 7.3 GiB peak memory for both. Measurements use ten denoising steps and 1,000 synchronized queries after 30 warmups; normalization and tokenization are included, data fetch and upload excluded (Table 18).

6 Conclusion

ACE organizes future supervision around action dependence, interaction regions, and task relevance. Matched analyses connect this supervision to higher action sensitivity and control success, while revealing weaker absolute executed-feature fidelity. Benchmark gains depend on task and demonstration coverage. ACE thus provides a useful training signal for native-policy adaptation, while the contrastive objective does not establish a complete counterfactual world model.

AI USE STATEMENT

Generative AI was used solely for language polishing and improving the clarity and readability of the manuscript. It was not used for idea generation, method development, experimental design, data analysis, or result interpretation. The authors take full responsibility for the content of the paper.

REPRODUCIBILITY STATEMENT

Editable sources are included. Experiment configurations, data subsets, rollout records, and provenance accompany the runs. Singleton horizon scans use identical head capacity.

REFERENCES

Ezra Ameperosa, Jeremy A Collins, Mrinal Jain, and Animesh Garg. Rocoda: Counterfactual data augmentation for data-efficient robot learning from demonstrations. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 13250–13256. IEEE, 2025.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

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

Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, et al. Worldvla: Towards autoregressive action world model. *arXiv preprint arXiv:2506.21539*, 2025.

Nga Teng Chan, Yi Zhang, Yechi Liu, Renwen Cui, Fanhu Zeng, Zeyuan Ding, Xiancong Ren, Zhang Zhang, Qifeng Chen, Jian Liu, et al. Robo-cortex: A self-evolving embodied agent via dual-grain cognitive memory and autonomous knowledge induction. *arXiv preprint arXiv:2605.18729*, 2026.

Jialei Chen, Kai Wang, Kang Chen, Shuaihang Chen, Feng Gao, Wenhao Tang, Zhiyuan Li, Weilin Liu, Zhuyu Yao, Boxun Li, et al. Lawam: Latent world action models for efficient dynamics-aware robot policies. *arXiv preprint arXiv:2606.15768*, 2026.

Xiaowei Chi, Peidong Jia, Chun-Kai Fan, Xiaozhu Ju, Weishi Mi, Kevin Zhang, Zhiyuan Qin, Wanxin Tian, Kuangzhi Ge, Hao Li, et al. Wow: Towards a world omniscient world model through embodied interaction. *arXiv preprint arXiv:2509.22642*, 2025.

Zeyuan Ding, Wenhai Liu, Yang Xu, Jiayu Hu, Yinda Chen, Yi Zhang, Yong Dai, Jian Tang, and Xiaozhu Ju. Pelican-vla 0.5: Attending before acting benefits generalization. *arXiv preprint arXiv:2607.06655*, 2026.

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 9156–9172, 2023. doi: 10.52202/075280-0403. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1d5b9233ad716a43be5c0d3023cb82d0-Abstract-Conference.html.

Chun-Kai Fan, Xiaowei Chi, Xiaozhu Ju, Hao Li, Yong Bao, Yu-Kai Wang, Lizhang Chen, Zhiyuan Jiang, Kuangzhi Ge, Ying Li, et al. Wow, wo, val! a comprehensive embodied world model evaluation turing test. *arXiv preprint arXiv:2601.04137*, 2026.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Catherine Glossop, William Chen, Arjun Bhorkar, Dhruv Shah, and Sergey Levine. Cast: Counterfactual labels improve instruction following in vision-language-action models. *IEEE Robotics and Automation Letters*, 2026.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pp. 2555–2565. PMLR, 2019.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1lOTC4tDS.

Nick Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, volume 2024, pp. 47376–47405, 2024.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. VIMA: Robot manipulation with multimodal prompts. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 14975–15022. PMLR, 2023. URL https://proceedings.mlr.press/v202/jiang23b.html.

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

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

Peiyan Li, Hongtao Wu, Yan Huang, Chilam Cheang, Liang Wang, and Tao Kong. Gr-mg: Leveraging partially-annotated data via multi-modal goal-conditioned policy. *IEEE Robotics and Automation Letters*, 10(2):1912–1919, 2025a.

Puhao Li, Yingying Wu, Ziheng Xi, Wanlin Li, Yuzhe Huang, Zhiyuan Zhang, Yinghan Chen, Jianan Wang, Song-Chun Zhu, Tengyu Liu, et al. Controlvla: Few-shot object-centric adaptation for pre-trained vision-language-action models. *arXiv preprint arXiv:2506.16211*, 2025b.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

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

Jinpeng Lu, Dexu Zhu, Haoyuan Shi, Linghan Cai, Guo Tang, Yinda Chen, Jie Cao, Duyu Tang, Yi Zhang, Yong Dai, et al. Current world models lack a persistent state core. *arXiv preprint arXiv:2606.20545*, 2026.

Yecheng Jason Ma, Vikash Kumar, Amy Zhang, Osbert Bastani, and Dinesh Jayaraman. Liv: Language-image representations and rewards for robotic control. In *International Conference on Machine Learning*, pp. 23301–23320. PMLR, 2023a.

Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. VIP: Towards universal visual reward and representation via value-implicit pre-training. In *The Eleventh International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=YJ7o2wetJ2.

Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. *arXiv preprint arXiv:2310.17596*, 2023.

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. *arXiv preprint arXiv:2406.02523*, 2024.

Duc Minh Nguyen, Nghiem Tuong Diep, Binh Gia Nguyen, Trong-Bao Ho, Doanh Le, Tan Q Nguyen, Thien-Loc Ha, Nhiem Tran, Bao Thach, Nhat X Tran, et al. Foca: Future-oriented conditioning for data-efficient vision-language-action adaptation. *arXiv preprint arXiv:2606.20867*, 2026.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

E Perez, F Strub, H De Vries, and V Dumoulin. Visual reasoning with a general conditioning layer, courville. In *In Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.

Junbo Qi, Yi Zhang, Hanchu Ni, Che Liu, Zhimin Yao, Ruilin Yang, Xiancong Ren, Liangjian Wen, Wei Ge, Yuya Ieiri, et al. E-vic: Reasoning beyond text via embodied visual chain for spatial intelligence. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 40283–40303, 2026.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*, 2025.

Haozhe Shan, Xiancong Ren, Han Dong, Haoyuan Shi, Yingji Zhang, Jiayu Hu, Yi Zhang, Yong Dai, Bin Shen, Lizhen Qu, et al. Epic-bench: A perception-centric benchmark for fine-grained embodied visual grounding in vision-language models. *arXiv preprint arXiv:2605.17070*, 2026.

Haoyuan Shi, Xiancong Ren, Yingji Zhang, Qinfan Zhang, Jiayu Hu, Haozhe Shan, Han Dong, Jinpeng Lu, Yinda Chen, Yi Zhang, et al. Vla-trace: Diagnosing vision-language-action models through representation and behavior tracing. *arXiv preprint arXiv:2605.30117*, 2026.

Woo Chul Shin, Zhenyang Chen, Alfred Cueva, Nadun Ranawaka Arachchige, Yingyan Celine Lin, Benjamin Joffe, Shreyas Kousik, and Danfei Xu. Static in, dynamic out: Counterfactual action augmentation for moving object manipulation. *arXiv preprint arXiv:2607.27890*, 2026.

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

Núria Armengol Urpí, Marco Bagatella, Marin Vlastelica, and Georg Martius. Causal action influence aware counterfactual data augmentation. *arXiv preprint arXiv:2405.18917*, 2024.

Yidong Wang, Yan Zhan, Ziteng Feng, Zhenyu Cui, Ziyi Zhou, Renzhao Liang, Jiaxuan Zhu, Zilei Yang, Yiran Zhao, Zhongkuan Mao, et al. Trustroboreward: Preference-ordered isotonic score editing for multi-paradigm robot reward models. *arXiv preprint arXiv:2608.08491*, 2026.

Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. In *International Conference on Learning Representations*, volume 2024, pp. 10641–10662, 2024.

Changhua Xu, En Yu, Junyu Xuan, and Jie Lu. Vgas: value-guided action-chunk selection for few-shot vision-language-action adaptation. *arXiv preprint arXiv:2602.07399*, 2026.

Sherry Yang, Yilun Du, Seyed Ghasemipour, Jonathan Tompson, Leslie Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *International Conference on Learning Representations*, pp. 45210–45234, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/c4d66eae503694424123b93ac0fbaf17-Abstract-Conference.html.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11941–11952. IEEE, 2023.

Shuai Zhang, Jiayu Hu, Zijie Chen, Zeyuan Ding, Yi Zhang, Yingji Zhang, Ziyi Zhou, Junwei Liao, Shengjie Zhou, Yong Dai, et al. A2eval: Agentic and automated evaluation for embodied brain. *arXiv preprint arXiv:2602.01640*, 2026a.

Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, Xinqiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, He Wang, Zhizheng Zhang, et al. Dreamvla: a vision-language-action model dreamed with comprehensive world knowledge. *Advances in Neural Information Processing Systems*, 38: 24195–24228, 2026b.

Yi Zhang, Che Liu, Xiancong Ren, Hanchu Ni, Shuai Zhang, Zeyuan Ding, Jiayu Hu, Hanzhe Shan, Zhenwei Niu, Zhaoyang Liu, et al. Pelican-vl 1.0: A foundation brain model for embodied intelligence. *arXiv preprint arXiv:2511.00108*, 2025a.

Yi Zhang, Che Liu, Xiancong Ren, Hanchu Ni, Yingji Zhang, Shuai Zhang, Zeyuan Ding, Jiayu Hu, Haozhe Shan, Junbo Qi, et al. Bridging vlms and embodied intelligence with deliberate practice policy optimization. *arXiv preprint arXiv:2511.16602*, 2025b.

Yi Zhang, Yinda Chen, Che Liu, Zeyuan Ding, Jin Xu, Shilong Zou, Junwei Liao, Jiayu Hu, Xiancong Ren, Xiaopeng Zhang, et al. Pelican-unify 1.0: A unified embodied intelligence model for understanding, reasoning, imagination and action. *arXiv preprint arXiv:2605.15153*, 2026c.

Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1702–1713. IEEE, 2025.

Ruijie Zheng, Jing Wang, Scott Reed, Johan Bjorck, Yu Fang, Fengyuan Hu, Joel Jang, Kaushil Kundalia, Zongyu Lin, Loic Magne, et al. Flare: Robot learning with implicit world modeling. *arXiv preprint arXiv:2505.15659*, 2025.

Shilong Zou, Shilin Zhang, Yingji Zhang, Yuhang Huang, Yi Zhang, Zeyuan Ding, Han Dong, Junwei Liao, Yong Dai, Jian Tang, et al. Pelican-sim 1.0: A general world model simulator for embodied intelligence. *arXiv preprint arXiv:2609.12036*, 2026.

APPENDIX: SUPPLEMENTARY EXPERIMENTS AND ANALYSIS

A IMPLEMENTATION AND EVALUATION RECORD

Main results use the revised implementation and matched native-policy control. Initialization checks require all pretrained backbone tensors to load after explicit namespace conversion. Gradient accumulation and learning-rate steps are counted per optimizer update. Future targets and terminal references are restricted to the sample's own episode; out-of-range horizons are masked.

A.1 SHARED TRAINING CONFIGURATION

Table 6 records the shared settings for the primary control comparisons. The budget counts optimizer updates after gradient accumulation. The resulting 936,000 sample presentations include repeated sampling and do not imply an equal number of epochs across demonstration budgets. Task balancing gives each task equal expected sampling probability rather than enforcing equal task counts in every batch.

Table 6: Shared configuration of the primary training runs. The configured scheduler extends beyond the actual training budget; its final learning rate is not the learning rate reached at update 3,600.

Item	Setting
Base model	Native π_0.
Tokenizer	PaliGemma tokenizer paired with the base model.
Training seed	42.
Update budget	3,600 optimizer updates, rather than microbatches.
Parallel training	Four NVIDIA A800-SXM4-80GB GPUs per independent run.
Effective batch	13 examples/GPU $\times$ 4 GPUs $\times$ 5 accumulation steps $=$ 260 examples/update; 936,000 sample presentations in total.
Sampler	Task-balanced sampling with per-frame weights inversely proportional to the number of frames in the corresponding task.
Optimizer	AdamW; learning rate 10^{-4}, $\beta = (0.9, 0.95)$, $\epsilon = 10^{-8}$, weight decay 10^{-10}, and gradient clipping norm 10.
Scheduler	Cosine decay with 1,000 warmup steps, 240,000 configured decay steps, and final decay learning rate 2.5×10^{-6}. The decay length is not reset to 3,600.
Checkpoint selection	Final checkpoint after 3,600 optimizer updates; intermediate checkpoints are not selected by test success.
Data loading	Eight data-loading workers.
Precision controls	Automatic mixed precision is disabled in both training and the accelerator configuration. Internal weight dtype conversions remain implementation-dependent; this does not imply full-model FP32 execution.
Visual inputs	Two views for LIBERO and three for RoboCasa, with 224×224 inputs. Teacher preprocessing and native π_0 image settings are recorded per run.
Action interface	Native flow-matching action head; chunk size 50; execute ten actions before replanning during evaluation.
Logging and saving	Training statistics recorded every ten updates; final checkpoint saved at update 3,600. The in-training evaluation interval is 20,000 updates; reported results use independent final evaluation.

A.2 TRAINABLE PARAMETER ALLOCATION

For the primary VLA–ACE comparison, shared language-backbone adapters use rank 8 and scale parameter $\alpha = 16$; the action-expert adapters use rank 16 and $\alpha = 32$. The action expert also retains its configured trainable dense weights.

The primary comparison uses the same language-backbone adapter rank, action adaptation, observation processing, training examples, and update budget for ACE and its native VLA control. Historical experiments that did not fully load the pretrained initialization are excluded. Oracle-mask

and automatic-mask results must remain distinct. Run records link each result to its configuration and evaluation coverage.

Table 7 specifies the adaptation scope. The language-adapter rank alone does not describe this configuration: action-side adapters have a different rank, and selected dense action-expert weights and the state projection also remain trainable.

Table 7: Parameter adaptation and audited trainable counts during training. These counts include configured dense trainable weights and are not adapter-only or total model parameter counts.

Component	Setting
Language adapters	LoRA rank 8, $\alpha = 16$; query, key, value, output, gate, up, and down projections.
Action adapters	LoRA rank 16, $\alpha = 32$; `q_proj` and `v_proj`; dropout 0.
Visual encoder	Frozen.
Language trunk	Frozen apart from its adapters.
Other trainable weights	Configuration-specified dense action-expert weights and the state projection.
Trainable parameters	VLA: 548,989,984; FOCA: 648,860,961; ACE: 553,948,834.

Cross-horizon optimization. If g_h is the shared-parameter gradient of the auxiliary objective at horizon h, a small joint update changes that objective by

$$\begin{aligned} g_k &= \nabla_\theta \mathcal{L}_k(\theta), \qquad g_{\text{flow}} = \nabla_\theta \mathcal{L}_{\text{flow}}(\theta), \\ \Delta\mathcal{L}_h &= \mathcal{L}_h(\theta + \Delta\theta) - \mathcal{L}_h(\theta) = g_h^\top \Delta\theta + O(\|\Delta\theta\|_2^2), \qquad \mathcal{L}_h \in C^2, \\ \Delta\theta &= -\eta \left(g_{\text{flow}} + \sum_{k \in \mathcal{H}} \alpha_k g_k \right) \quad \text{(gradient-descent step; fixed } \alpha_k \geq 0\text{)}, \\ \Delta\mathcal{L}_h &= -\eta g_h^\top g_{\text{flow}} - \eta \alpha_h \|g_h\|_2^2 - \eta \sum_{\substack{k \in \mathcal{H} \\ k \neq h}} \alpha_k g_h^\top g_k + O(\eta^2). \end{aligned} \tag{12}$$

This identity characterizes a possible optimization effect rather than an empirical gradient-alignment result.

B Scene-Prior Decomposition and Contrastive Signal

B.1 Scene-Prior Solutions under the Demonstration Objective

Under squared prediction error, the population-optimal context-only and action-conditioned predictors are $\mu_X = \mathbb{E}[Z_h \mid X]$ and $\mu_{X,A} = \mathbb{E}[Z_h \mid X, A]$. Their optimal risks satisfy

$$\begin{aligned} \mathbb{E}\|Z_h\|_2^2 < \infty, \qquad U \in \{X, (X, A)\}, \\ R_U^* := \inf_{f \text{ measurable}} \mathbb{E}[\|Z_h - f(U)\|_2^2], \\ R_X^* - R_{X,A}^* = \mathbb{E}[\|\mu_{X,A} - \mu_X\|_2^2]. \end{aligned} \tag{13}$$

The identity follows by decomposing $Z_h - \mu_X$ into the conditional residual $Z_h - \mu_{X,A}$ and the conditional-mean difference; the cross term has zero expectation. Actions offer additional prediction benefit only to the extent that they change the expected future after conditioning on the scene.

Two properties of manipulation demonstrations can make this benefit small. First, expert behavior concentrates action choices around a narrow range at each scene and task phase. In the limiting case $A = \pi(X)$, conditioning on A adds no information on the demonstration distribution. Second, pooled visual features combine localized interaction with persistent scene content. In the simplified model $Z_h^{\text{lin}} = B(X) + \rho D_h$ (before normalization), predictable content $B(X)$ cancels from the

conditional-mean difference, yielding

$$\begin{aligned} \mathbb{E}\|B(X)\|_2^2 + \mathbb{E}\|D_h\|_2^2 < \infty, \qquad \rho \in \mathbb{R} \text{ fixed}, \\ R^*_{U,\mathrm{lin}} := \inf_{f \text{ measurable}} \mathbb{E}[\|Z_h^{\mathrm{lin}} - f(U)\|_2^2], \qquad U \in \{X, (X,A)\}, \\ R^*_{X,\mathrm{lin}} - R^*_{X,A,\mathrm{lin}} = \rho^2 \mathbb{E}\Big[\|\mathbb{E}[D_h \mid X, A] - \mathbb{E}[D_h \mid X]\|_2^2\Big]. \end{aligned} \tag{14}$$

A small interaction contribution ρ weakens the predictive benefit of actions. These are conditions under which the objective admits a strong scene-only solution; the tests below examine whether training reaches one. For unit-normalized targets and cosine loss, the optimal context-only prediction points along $\mathbb{E}[Z_h \mid X]$ whenever this mean is nonzero, so the same scene regularity also benefits the positive cosine objective.

$$\begin{aligned} \|Z_h\|_2 = 1 \text{ a.s.}, \qquad \mu_U = \mathbb{E}[Z_h \mid U], \qquad U \in \{X, (X,A)\}, \\ R^*_{U,\cos} := \inf_{\substack{f \text{ measurable} \\ \|f(U)\|_2 = 1 \text{ a.s.}}} \mathbb{E}[1 - f(U)^\top Z_h] = 1 - \mathbb{E}\|\mu_U\|_2, \\ R^*_{X,\cos} - R^*_{X,A,\cos} = \mathbb{E}\|\mu_{X,A}\|_2 - \mathbb{E}\|\mu_X\|_2 \geq 0, \\ \|u - v\|_2^2 = 2(1 - u^\top v), \qquad \|u\|_2 = \|v\|_2 = 1. \end{aligned}$$

B.2 Implications for the Training Signal

The preceding argument motivates two complementary interventions in the objective. Interaction weighting increases the contribution of manipulated regions to the target. Within-context action contrast changes what the predictor must distinguish. For scores $s_0, \ldots, s_J$ ($J = |\mathcal{N}_{t,k}|$), the loss is $\ell = -s_0 + \log \sum_{j=0}^{J} \exp(s_j)$. An additive scene-dependent score shared by every candidate cancels, and

$$\begin{aligned} \frac{\partial \ell}{\partial s_j} = \frac{\exp(s_j)}{\sum_{i=0}^{J} \exp(s_i)} - \mathbf{1}[j = 0], \qquad j = 0, \ldots, J, \\ \ell(s_0 + b, \ldots, s_J + b) = \ell(s_0, \ldots, s_J), \qquad b \in \mathbb{R}. \end{aligned} \tag{15}$$

The ranking objective thus rewards candidate-dependent information. An action-invariant predictor has loss $\log(J+1)$ and pairwise accuracy $1/2$, regardless of its positive-future accuracy. This gives a concrete reference for the validation protocol.

C Protocols and Additional Results for the Matched Analysis

C.1 Training Controls and Data Separation

All three analysis models use the LIBERO40 20-demo subset, containing 800 training episodes, with 3,600 updates, effective batch 260, and $h = 30$. Their language-adapter rank and scale are 8 and 16; action-adapter rank and scale are 16 and 32. The auxiliary teacher, interaction weighting, progress target, optimizer, and scheduler are shared. A/B use loss weights $(0.1, 0, 0.1)$ and C uses $(0.1, 1, 0.1)$. A fixes the action input to zero while retaining the allocated predictor architecture. A/B were retrained; C reuses the audited 20-demo checkpoint. C's control report uses the native export of the same final checkpoint, with auxiliary heads removed and adapters retained.

The swap test reserves demonstration episodes outside the training subset. It samples approximately quarter-, half-, and three-quarter-trajectory contexts from ten episodes per Spatial task. The donor pool contains 30 reserved episodes per task. This is an episode-level separation; historical normalization statistics are reused, so it should not be interpreted as a fully inductive preprocessing split. The execution experiment instead uses benchmark initial-state indices 20–29 and policy-generated contexts. Those indices do not certify unseen training geometry. Closed-loop control uses initial-state indices 0–19, 20 rollouts per task, and the Spatial limit of 220 control steps after ten stabilization steps. The three protocols therefore have different sampling units and must not be pooled.

C.2 Matched Donors and Persistence Reference

Donor selection uses only current state and action information. Scales are estimated from the 20 training episodes per task. Eligible prefixes come from another episode of the same task at a matched normalized phase and gripper class, with normalized action distance in $[1.0 \times 10^{-2}, 3]$ and normalized state distance at most 3. Orientation matching uses an $\mathrm{SO}(3)$ geodesic distance. The three nearest eligible donors by state are selected. Each donor is separated from the demonstrated prefix by the required action-distance threshold and comes from a distinct episode. Five of 300 proposed contexts have fewer than three eligible donor episodes, leaving 295 contexts from 100 episodes. Every model uses these same contexts and donors.

For each context, the demonstrated-action prediction is compared against three donor-action predictions using the single demonstrated future. Thus Δ_{swap} measures preference for the demonstrated action under that target, not the quality of an unobserved donor future. The persistence reference pools current-image teacher patches using the same current-to-future mask used for the target. It is a diagnostic reference with access to target-derived pooling weights, not an online predictor with a mask inferred only from the current frame. A's maximum prediction difference is checked below 1.0×10^{-5}; its reported sensitivity near 1.0×10^{-9} is floating-point error. The matched-donor test reports feature agreement, prediction change, and the reference-future gap; the two-candidate ranking pilot below is kept separate.

C.3 Branch Execution, Restoration, and Coverage

An independent rank-8 native VLA trained with 20 demonstrations per task generates a 30-step preparatory prefix using three replans. At the resulting state, it supplies the reference chunk; three distinct demonstration-donor chunks near phase 2.5×10^{-1} provide alternatives. These are policy-reference interventions, not expert demonstration actions at an expert state. Actions are clipped to $[-1, 1]$, and the identical clipped arrays are supplied to the predictor and simulator. Across retained states, the fraction of clipped scalars in the 30-step candidate prefixes ranges from zero to 3.5%.

Each branch reconstructs its context using the same reset, benchmark initial state, ten stabilization steps, and exact replay of the preparatory prefix. We compare simulator state, proprioception, selected body/camera/geometry arrays, and rendered observations before scoring. State, proprioception, and model-array tolerances are 1.0×10^{-6}; the maximum pixel tolerance is one intensity level. Candidate execution order is randomized. Repeating the reference prefix provides a separate noise check. Among valid states, maximum simulator restoration and repeat-execution discrepancies are zero and maximum pixel discrepancies are one. Each candidate then executes 30 steps open-loop without replanning.

Of 100 attempted states, 89 pass these checks. All ten states for the top-drawer task fail restoration, and one stove-task state fails the pixel tolerance (error 2 rather than at most 1). The primary execution result therefore averages nine represented tasks, with 356 unique candidate branches and 89 repeated reference branches. These exclusions are model independent and applied before prediction scoring. They limit task coverage; they are not unsuccessful control outcomes that were removed to raise success rates. Identical or unsuccessful candidate outcomes among valid states remain in the primary analysis.

C.4 Pooling and Aggregation

The local execution mask is the symmetric average of motion-based masks from all four executed outcomes. The same mask pools every outcome for every prediction, so target construction does not depend on query identity. Uniform pooling is an additional check. Retrieval ties use absolute tolerance 1.0×10^{-6} and zero relative tolerance, with credit divided by the number of tied maxima. The diagonal cosine and matching margin average over the four action queries within a state.

For swap metrics, we first average contexts within each episode, then average episodes within each task, and finally average tasks equally. Execution metrics use the same hierarchy with one context per initial-state episode. Paired differences first subtract metrics for common context or state identifiers and then use the same aggregation. All comparisons describe the evaluated checkpoints on the observed tasks.

C.5 Pooling and Outcome-Distinct Sensitivity

The primary analysis retains every valid state. A secondary subset requires each pair of executed target features to have cosine distance greater than $\max(1.0 \times 10^{-3}, 10\epsilon_{\text{repeat}})$, where ϵ_{repeat} is that state's repeated-reference feature noise. This rule retains 85 of 89 states for every model. It is a robustness filter, not an independently validated semantic threshold or a replacement for the primary denominator.

Table 8: Executed-consequence sensitivity checks. The primary local analysis uses a common symmetric mask; uniform pooling removes that weighting. The outcome-distinct subset retains 85/89 states and uses local pooling. Metrics are task-macro averages, with display scaling indicated in the headers.

Evaluation	Model	Cosine ($\times 10^3$)	Retrieval (%)	Margin ($\times 10^3$)
Uniform, all 89	A	987.0	25.0	0.0
	B	988.0	31.7	0.5
	C	152.6	37.9	2.0
Local, distinct 85	A	992.4	25.0	0.0
	B	993.3	35.3	1.0
	C	141.3	39.5	4.4

Table 9: Paired C–B differences on common records. Retrieval differences are percentage points (pp); other columns report scaled cosine or cosine-margin differences. Differences are computed before rounding.

Metric	Display unit	C–B
Swap: future cosine	$\times 10^3$	-268.2
Swap: prediction change	$\times 10^3$	$+236.9$
Swap: reference-future gap	$\times 10^3$	$+78.4$
Execution: local cosine	$\times 10^3$	-850.7
Execution: local retrieval	pp	$+4.0$
Execution: local margin	$\times 10^3$	$+3.4$
Execution: uniform retrieval	pp	$+6.2$
Execution: uniform margin	$\times 10^3$	$+1.5$

Uniform-pooling retrieval also favors C over B, but this secondary result does not replace the primary local-mask comparison. In both pooling schemes, C has much lower absolute cosine. Frozen visual-feature retrieval also does not identify which physical variable was predicted: robot-pose changes may contribute alongside object motion. The metrics quantify visual-feature retrieval rather than object-centric state error, articulated-joint prediction error, or calibrated counterfactual success. These distinctions prevent relative retrieval from being interpreted as a complete dynamics evaluation.

C.6 Task-Level Control and Execution Coverage

Table 10: Task-level audit on LIBERO-Spatial. Task IDs follow benchmark order. Control uses 20 rollouts per task; execution uses a separate set of initial states. Task 4 has no valid execution states after restoration checks. No task-level retrieval value is imputed for it.

	Control successes			Valid states	Local retrieval (%)		
Task	A	B	C	N	A	B	C
0	8/20	9/20	12/20	10	25.0	45.0	37.5
1	12/20	15/20	12/20	10	25.0	37.5	37.5
2	9/20	10/20	15/20	10	25.0	32.5	37.5
3	10/20	8/20	13/20	10	25.0	25.0	45.0
4	11/20	11/20	8/20	0	—	—	—
5	0/20	0/20	0/20	10	25.0	37.5	40.0
6	15/20	13/20	16/20	10	25.0	37.5	55.0
7	11/20	14/20	15/20	9	25.0	27.8	36.1
8	7/20	4/20	8/20	10	25.0	35.0	27.5
9	2/20	7/20	6/20	10	25.0	40.0	37.5

C.7 Tabulated Results for the Three Core Analyses

Table 11: Action-swap results on 295 held-out contexts with three donors per context. Larger D_{pred} indicates greater prediction change; larger Δ_{swap} indicates a stronger preference for the demonstrated action against the shared future. A is action-invariant by construction, with numerical sensitivity rounded to zero. Scaling in the column headers is for display and does not turn cosine into a success rate.

Model	C_{future} ($\times 10^3$)	D_{pred} ($\times 10^3$)	Δ_{swap} ($\times 10^3$)
A: Parallel	995.8	0.0	0.0
B: Conditioned	996.3	0.6	0.4
C: Full ACE	728.2	237.4	78.9

Table 12: Formulation comparison at 20 demonstrations per task. All models share the backbone, allocated head architecture, teacher targets, and update budget. Future agreement uses the held-out swap contexts; success uses 200 Spatial rollouts per model. The action and contrast columns identify the controlled changes.

Model	Action input	Contrast	C_{future} ($\times 10^3$)	Success (%)
A: Parallel	No	No	995.8	42.5
B: Conditioned	Yes	No	996.3	45.5
C: Full ACE	Yes	Yes	728.2	52.5

Table 13: Same-state executed-consequence matching on 89 valid states across nine Spatial tasks, with four candidates per state. Eleven attempted states fail restoration checks and are excluded for every model. Retrieval measures relative matching to executed outcomes; C_{exec} measures absolute feature agreement. A gives chance retrieval and zero margin up to numerical precision.

Model	C_{exec} ($\times 10^3$)	Retrieval (%)	R_{margin} ($\times 10^3$)
A: Parallel	992.4	25.0	0.0
B: Conditioned	993.4	35.3	1.0
C: Full ACE	142.7	39.3	4.4

D Extended Experimental Results

The following records complement the main results with suite-level control, design studies, and earlier pilot diagnostics. The 50-demo reversal pilot and the new 20-demo matched analysis use different models and data; their scores are not combined.

D.1 Suite-Level Results and Baseline Provenance

The additional rank-8 LoRA LIBERO run completes all four suites; the complete suite-level results are shown in Figure 4. LoRA is a further action-only run with the same adaptation dimensions as the original VLA reference, not a different rank configuration. The original VLA reference and the additional LoRA run have the same 464 trainable tensor names (ignoring distributed wrappers) and 548,989,984 trainable parameters. Matching these quantities establishes the same adaptation dimensions and parameter count, rather than identical trained policies.

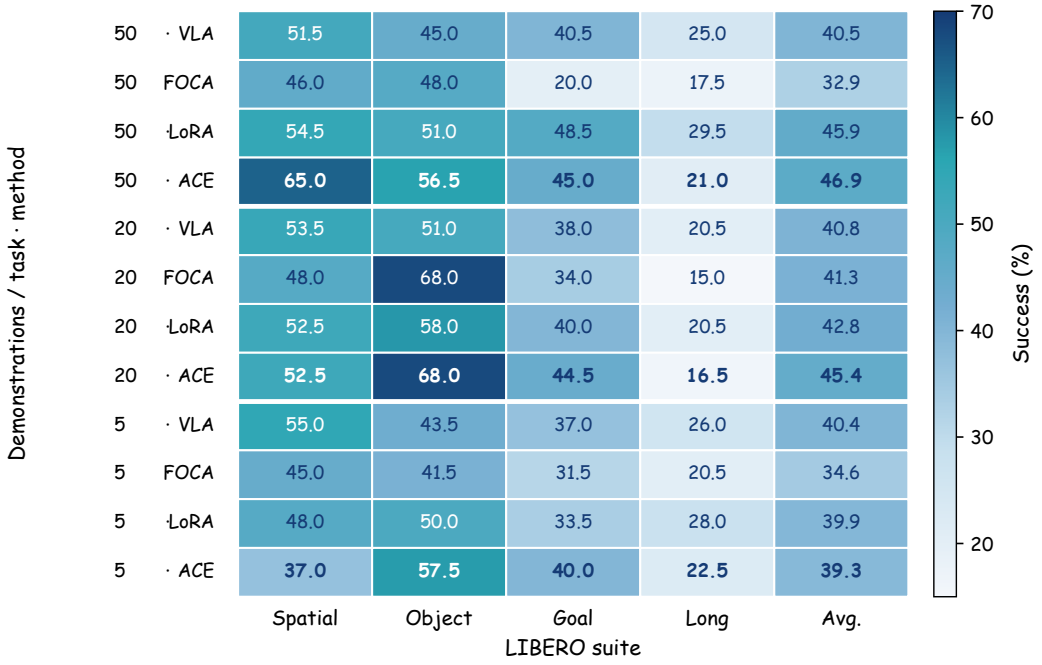


Figure 4: Complete suite-level LIBERO success (%) across demonstration budgets. All models are jointly trained on 40 tasks for 3,600 updates. Every suite contains 200 rollouts; Avg. weights the suites equally. The subsets contain 50, 20, or 5 demonstrations per task.

D.2 Baseline Adaptation Settings

The baseline configurations use the same demonstration budgets, 3,600 updates, and final-checkpoint rule as the main control experiments. LoRA uses language-adapter rank 8 with $\alpha = 16$ and action-adapter rank 16 with $\alpha = 32$.

D.3 Interaction Targets and Horizon Selection

The target-weighting comparison and horizon sweep characterize design choices, rather than supplying additional matched A/B/C evidence; the corresponding diagnostics are shown in Figure 5. Spatial development success is used to choose the horizon; its reported maximum is a selection result, not an independent estimate of generalization. The record does not establish disjoint development and final evaluation initial states. Because horizon-dependent valid-target coverage changes, these scores do not isolate the effect of temporal distance at a fixed set of supervised frames.

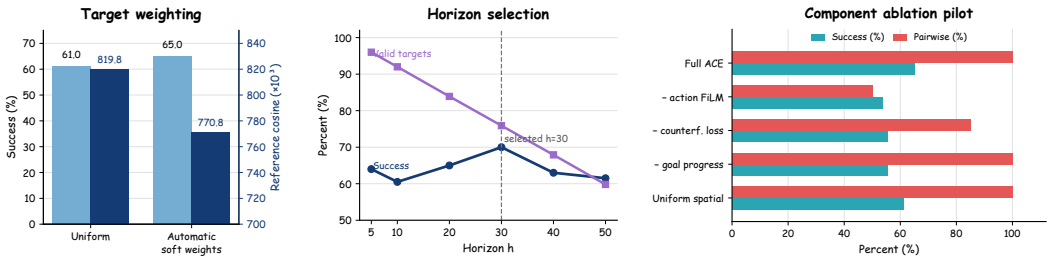


Figure 5: Design-choice diagnostics: target weighting, horizon selection, and component ablation.

Automatic weights improve Spatial control by 4.0 points, while their mean reference cosine is lower than uniform weighting (770.8 versus 819.8 on the $\times 10^3$ scale). Control improvement and reference-feature agreement therefore need not move together.

The Spatial-only development scan selects $h = 30$, with 70.0% success. Joint $\{5, 10, 20\}$ supervision achieves 56.5%. Target coverage also changes with horizon, so this scan measures the complete horizon configuration rather than isolating temporal distance alone. We retain $h = 30$ for main training and RoboCasa transfer; the latter's separate horizon results appear in Table 21.

D.4 Earlier Two-Candidate Pilot: Executed Consequences and Goal Attainment

Table 14: Same-state policy-reference/time-reversal interventions on 60 compact test episodes (120 executed branches); the 40 calibration episodes are excluded. Ordering compares predicted and executed semantic progress, with half credit for ties. The action-invariant control repeats the full-model reference prediction for both branches; it is not a trained probe. Reset error is the maximum restored-state discrepancy (zero for every tested branch).

Predictor	Branches N	Cosine $(\times 10^3) \uparrow$	Ordering (%)	Reset error
Action-invariant control	120	770.7	50.0	0.0
ACE without counterfactual loss	120	993.5	43.3	0.0
Full ACE	120	364.6	51.7	0.0

Table 15: Initial-state progress score versus actual rollout success. Ridge-logistic calibration uses 40 episodes (27 successes); evaluation uses 60 disjoint episodes (33 successes). ECE uses ten equal-width bins. The raw score is not treated as a success probability; a constant probability of 0.5 has Brier score 250.0 on this test set after the same $\times 10^3$ display scaling.

Score / calibration	Test N	AUROC $(\times 10^3)$	Brier $(\times 10^3)$	ECE $(\times 10^3)$
Raw semantic-progress score	60	630.8	N/A	N/A
Calibrated semantic score	60	630.8	253.6	120.0

In this earlier two-candidate pilot, full ACE has lower feature agreement than both controls, while progress ordering is close to chance. These measurements use different data and a different demonstration budget from the matched analysis in Section 4. Calibration and testing are separated by episode. These measurements distinguish candidate discrimination from accurate counterfactual prediction and calibrated success estimation.

D.5 SENSITIVITY AND DEPLOYMENT

Table 16: Trainable parameter counts from the loaded main-model configurations and measured deployment latency (Table 18). ACE latency uses the exported policy.

Method	Trainable parameters	p50 (ms)
VLA, shared adapters	548,989,984	297.5
FOCA	648,860,961	291.4
ACE	553,948,834	300.2

ACE adds 4.96M trainable parameters over VLA (4,958,850 exactly; Table 16). Under the matched training benchmark, it increases mean update time by 19.1%, while peak allocated memory increases by only 0.09 GiB per GPU. After removing the auxiliary modules, the exported policy has a measured median deployment latency of 300.2 ms, compared with 297.5 ms for VLA (Table 18). These measurements distinguish training overhead from the cost of the exported native policy; equal update budgets do not imply equal training time.

The sensitivity comparison varies negative count and progress weight at fixed $h = 30$. Reducing the count from seven to three lowers Spatial success from 65.0% to 51.5%; reducing the progress weight from 0.1 to 3.0×10^{-2} yields 61.0%. Pairwise ranking remains 100.0% for these settings, illustrating the limited resolution of this diagnostic.

Table 17: One-factor sensitivity under the compact LIBERO40/50-demo/$h = 30$ protocol, with Spatial success on 200 rollouts. Diagnostic evaluation always uses two candidates (policy reference and reversal), independently of training negative count J; adjusted top-1 is $100(2a - 1)\%$ $(a \in [0, 1])$.

Factor	Setting	Success (%)	Pairwise (%)	Adj. top-1 (%)
Negative count	$J = 3$	51.5	100.0	100.0
	$J = 7$	65.0	100.0	100.0
Progress weight	0	55.5	100.0	100.0
	3.0×10^{-2}	61.0	100.0	100.0
	0.1	65.0	100.0	100.0

Table 18: Deployment on A800, batch 1, ten denoising steps: 30 warmups and 1,000 synchronized action-chunk queries using 20 fixed Spatial frames. Timing includes normalization/tokenization and excludes data fetch/upload; memory is peak allocated CUDA memory. Export matches the pre-export policy exactly on 20 paired inputs with identical initial action noise (maximum absolute error 0).

Inference variant	p50 (ms)	p95 (ms)	Peak GiB
Native VLA, shared adapters	297.5	300.5	7.3
FOCA	291.4	294.5	7.5
ACE before head removal	300.0	303.4	7.3
ACE exported native policy	300.2	303.6	7.3

Removing the auxiliary branch preserves fixed-input action outputs and reduces the exported footprint without changing the native control graph. The p50/p95 measurements show comparable deployment cost to shared-adapter VLA (Table 18). This comparison isolates export behavior from differences in the trained policy weights; checkpoint hashes, probe counts, and maximum action differences accompany the measurements.

D.6 Coverage and Household Manipulation

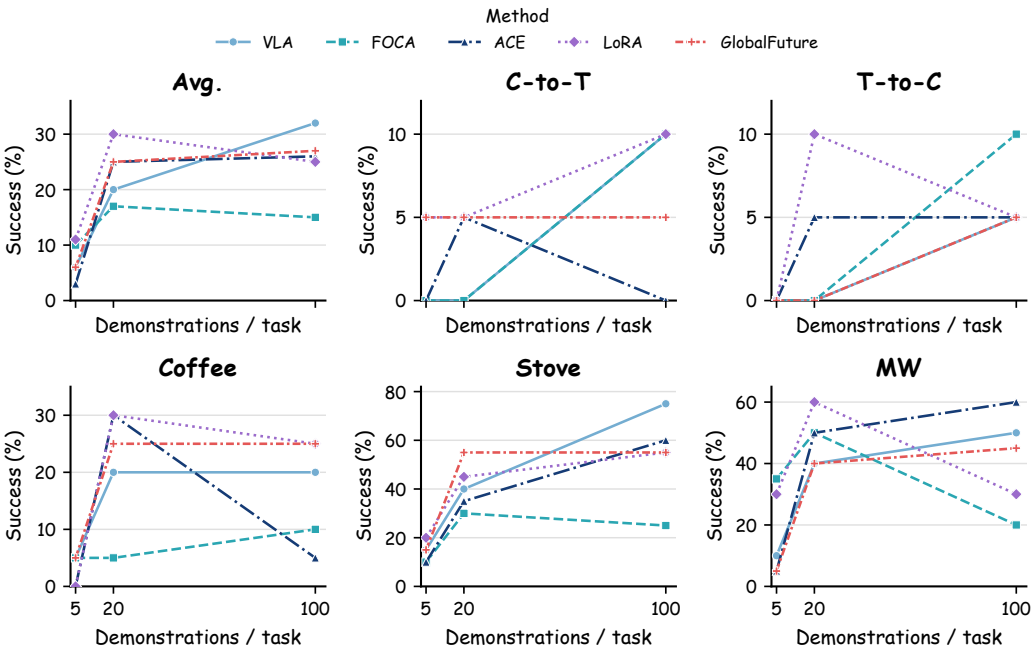


Figure 6: RoboCasa transfer success across demonstration budgets.

The household comparison tests transfer across distinct interaction types.

Table 19: RoboCasa component ablations at 100 demonstrations/task and $h = 30$. Each variant is paired with full ACE on the same 100 task–initial-scene pairs (20 per task). Δ is variant minus full ACE in percentage points.

Variant	Success (%)	Δ
Full ACE	26.0	0.0
Without action conditioning (FiLM off)	27.0	+1.0
Without action contrast	22.0	-4.0
Without goal progress	25.0	-1.0
Uniform spatial weights	23.0	-3.0

Component contributions across environments. RoboCasa exhibits a different component pattern from LIBERO. Removing contrast or spatial weighting reduces the point estimate by 4.0 and 3.0 points, respectively; removing progress changes it by 1.0 point, and disabling action conditioning increases it by 1.0 point. These 100-scene outcomes show that the component effects vary across environments and do not show a uniform control benefit from every component. The primary formulation comparison remains the matched Parallel / Action-Conditioned / Full ACE analysis, which measures action dependence and executed-consequence retrieval alongside control.

Meaning of the action-conditioning ablation. With FiLM disabled, the action MLP is still instantiated but its output is unused by the consequence predictor. The predictor depends on context and horizon only. Consequently, within-context action candidates have identical predictions, and the contrast term carries no discriminative gradient (up to floating-point error). This ablation jointly removes action conditioning and the effective contrast signal; it does not compare FiLM against another action-conditioning mechanism. Parameter tensors remain allocated, but the active parameter set changes. The conditioning-only model in the matched analysis provides the separate contrast comparison.

Table 20: Task-level results for the RoboCasa component ablations in Table 19. All entries use the same 20 initial scenes per task.

Variant	C→T	T→C	Coffee	Stove	MW	Avg.
Full ACE ($h = 30$)	0.0	5.0	5.0	60.0	60.0	26.0
Without action conditioning	5.0	20.0	5.0	50.0	55.0	27.0
Without counterfactual loss	0.0	5.0	10.0	40.0	55.0	22.0
Without goal-progress loss	5.0	5.0	10.0	40.0	65.0	25.0
Uniform spatial weights	15.0	10.0	15.0	50.0	25.0	23.0

Table 21: RoboCasa horizon diagnostics at 100 demonstrations/task. Each model uses 100 rollouts across five tasks. The main horizon remains $h = 30$, selected from LIBERO development; these held-out RoboCasa outcomes are not used to select it.

Horizon	C→T	T→C	Coffee	Stove	MW	Avg.
Full ACE ($h = 30$)	0.0	5.0	5.0	60.0	60.0	26.0
$h = 5$	5.0	10.0	15.0	50.0	45.0	25.0
$h = 10$	0.0	20.0	10.0	55.0	45.0	26.0
$h = 20$	15.0	5.0	15.0	65.0	30.0	26.0
$h = 50$	10.0	15.0	30.0	45.0	50.0	30.0
Joint $\{5, 10, 20\}$	20.0	5.0	15.0	60.0	55.0	31.0

D.7 Supporting Task-Level Qualitative Comparison

Table 22: Task-grounded qualitative analysis of completed full-data LIBERO runs. Entries report successes out of 20 rollouts for the same task. We select the largest ACE–VLA gain in Spatial and Goal, and the largest gain and regression in Object and Long; ties use the lowest task index. Interpretations use task descriptions and aggregate outcomes, without claiming trajectory-video evidence. All models use joint 40-task training; ACE uses $h = 30$.

Task	VLA	FOCA	ACE	Qualitative interpretation
Bowl between plate and ramekin → plate	11/20	9/20	17/20	Improvement with spatially specified object placement.
Orange juice → basket	3/20	13/20	18/20	Large gain for one object within the shared placement goal.
Chocolate pudding → basket	18/20	13/20	11/20	The same goal shows a regression for another object.
Open the middle drawer	2/20	2/20	9/20	Gain on a task centered on an articulated-object state change.
Book → back caddy compartment	0/20	5/20	8/20	Improved placement in a task from the Long suite.
Turn on stove; place moka pot	14/20	12/20	8/20	Regression on a task combining activation and placement.

Task-level records and failures. Task-level success counts support the qualitative comparisons in this appendix. The interpretations are based on task descriptions and aggregate outcomes rather than a video-based failure taxonomy.

D.8 Visual Illustration of Destination and Completion

Figure 7 illustrates the distinction between an observable environmental change and visually verified task completion. In this recorded demonstration, the cup moves into the sink region, but occlusion limits confirmation that it has been released. The example motivates distinguishing task-relevant changes from a complete success judgment. It is a qualitative illustration from an observed trajectory, rather than an evaluation of ACE predictions or an ACE-controlled rollout.

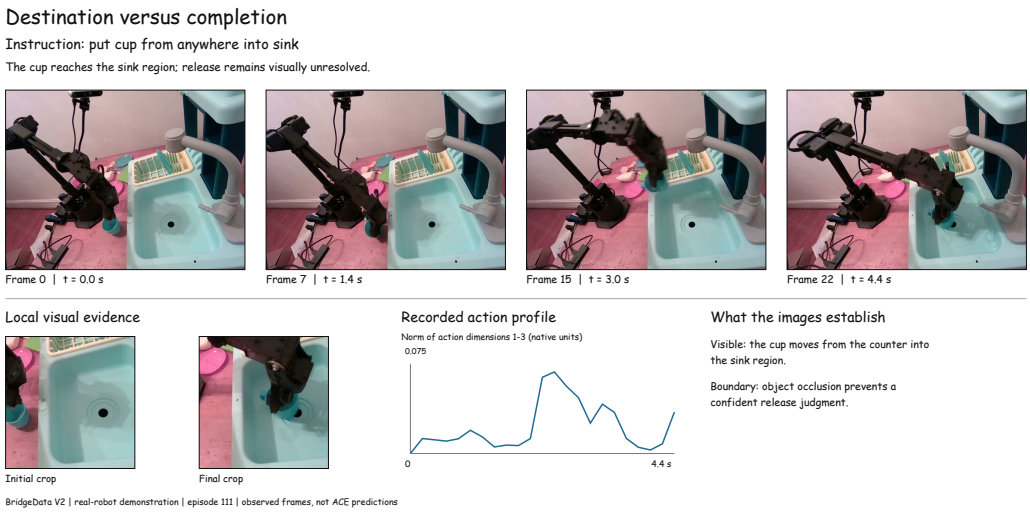


Figure 7: **Destination versus completion.** BridgeData V2 real-robot demonstration, episode 111, with the instruction to put the cup into the sink. Selected frames span 0.0–4.4 s; local crops show the cup moving from the counter into the sink region. The action profile plots the norm of recorded action dimensions 1–3 in native units. Object occlusion prevents a confident visual judgment of release. These are observed demonstration frames, not ACE predictions.

E Limitations and Future Directions

The present evaluation characterizes ACE under a fixed update budget. Longer training would help assess convergence, and broader task and demonstration coverage would clarify when its gains persist. Horizon selection uses Spatial development evaluations; a development split disjoint from all final evaluation states and a common-valid-frame sweep would strengthen future comparisons across horizons.

Interaction cues and terminal-image similarity provide useful but approximate supervision. Alternative actions can share outcomes or be infeasible, and action discrimination alone does not establish intervention-level dynamics. The executed-consequence tests provide an initial assessment of outcome matching; extending these tests and improving absolute future-feature fidelity are natural directions for further study.