DIWA: DECISION-INFLUENTIAL WORLD ABSTRACTION FOR VLA-WAM POLICIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Predicting future observations can improve robot decision making, but a controller need not model every visible change with equal fidelity. We introduce Decision-Influential World Abstraction (DIWA), a framework that allocates future modeling capacity according to its effect on action generation. A lightweight estimator ranks object–time queries before an action-conditioned world decoder expands them. Its supervision measures changes in action responses, value estimates, and task progress under latent interventions, with shared randomness for stochastic action heads. A complementary regret geometry organizes representations by the relative returns of aligned candidate action rules. During inference, only selected future queries are decoded, avoiding dense future generation. Across LIBERO, RoboTwin, RoboCasa, and real-robot manipulation, DIWA achieves 73.5% aggregate success, improving over DreamVLA by 5.6 percentage points. The out-of-distribution evaluation yields 75.8% success, a 10.6-point improvement. Mean inference latency is 89 ms, compared with 213 ms for dense imagination and 241 ms for WorldVLA. Three-seed simulation results, 300 physical trials per method, budget sweeps, and intervention diagnostics distinguish the roles of influence supervision, regret geometry, and selective computation. These results support using decision sensitivity to determine which future information a robot policy should compute.

1 INTRODUCTION

Vision-language-action (VLA) models connect visual perception and language understanding to robot control, enabling broadly pretrained representations to be adapted to manipulation tasks (Zitkovich et al., 2023; Kim et al., 2025; Ghosh et al., 2024). However, understanding the current scene does not always determine the next action. An apparently similar gripper pose can require closing, retracting, or maintaining contact depending on object state and task progress. Anticipating action consequences is therefore useful, particularly around contact transitions and in tasks with several dependent stages.

World models offer a route to such anticipation. Video-conditioned policies and action world models predict future observations, while DreamVLA predicts compact dynamic, spatial, and semantic information (Wu et al., 2024; Zhou et al., 2024; Cen et al., 2025; Zhang et al., 2025). Nevertheless, predictive detail and decision utility are different objectives. A model may spend substantial computation on a moving background without improving grasp selection. Conversely, a small change in whether an object is supported or already grasped can substantially change the appropriate action. The computational question is therefore not only how to compress a future representation, but also which parts of that representation deserve to be computed for the current decision.

We study this question through *decision influence*: the sensitivity of a policy's action response and decision heads to a specified intervention on an imagined variable. We propose DIWA, which represents candidate futures as object–time queries and estimates their influence from the current context and a proposed action chunk. Crucially, scoring precedes future decoding. A selector then passes a subset of the queries to an action-conditioned world model. The policy combines the resulting imagined tokens with current-state context to produce the action chunk.

The influence estimator is trained using dense reference futures and sampled latent interventions. Replacing one token with a mask or a same-task, cross-episode token provides a reference–intervention pair. Changes in local action moments, predicted value, and predicted progress provide supervision for a cheaper estimator that does not need to generate those interventions at test time. For diffusion

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.

and flow heads, the paired responses use the same noise and time samples, reducing variation unrelated to the intervention. This construction measures dependence inside the learned policy; it does not, on its own, identify a causal effect in the physical environment.

Selection also needs an appropriate representation. Two states can look different yet favor the same actions, while visually similar states can reverse action preferences. We therefore regularize latent distances using *regret vectors* computed from measured returns of a consistently indexed candidate bank. Regret removes state-dependent value offsets and retains the relative cost of choosing each candidate. It complements influence supervision: influence determines where to allocate computation, and regret geometry constrains the decision information that the representation should preserve.

Our contributions are threefold. First, we develop an object–time selection mechanism that predicts influence before future decoding and supports a state-dependent compute budget. Second, we couple intervention-based influence supervision with measured-return regret geometry and characterize the resulting finite-candidate notion of decision equivalence. Third, we evaluate success, robustness, and inference cost on complementary manipulation settings. DIWA improves aggregate success over DreamVLA by 5.6 points while reducing mean latency by 58.2% relative to its dense-imagination variant. Independent intervention diagnostics further connect the learned influence ranking and latent geometry to measured decision consequences.

2 Related work

VLA policies and action generation. RT-2 transfers visual-language knowledge to action generation; OpenVLA and Octo provide open generalist robot policies (Zitkovich et al., 2023; Kim et al., 2025; Ghosh et al., 2024). Diffusion Policy and π_0 use denoising and flow-based action generation, respectively (Chi et al., 2023; Black et al., 2025). DIWA selects the future latent information supplied to an action head. Paired response probes accommodate different generators, although the experiments do not establish equal benefits across action-head families.

Predictive models for control. GR-1 combines future-image and action prediction, RoboDreamer learns compositional video world models, and WorldVLA unifies image and action generation (Wu et al., 2024; Zhou et al., 2024; Cen et al., 2025). DreamVLA predicts compact world knowledge and supplies our backbone (Zhang et al., 2025). MuZero and TD-MPC2 further motivate control models that avoid observation reconstruction (Schrittwieser et al., 2020; Hansen et al., 2024). We study allocation within a predictive policy, selecting object–time queries before their future representations are decoded.

Decision-relevant abstraction and selective computation. Value equivalence and bisimulation motivate representations that preserve decision-relevant information (Grimm et al., 2020; Zhang et al., 2021); our finite-candidate regret construction provides a narrower guarantee. Slot Attention supplies object representations, and DynamicViT learns visual token sparsification (Locatello et al., 2020; Rao et al., 2021). Recent VLA methods address action-aware efficiency directly: SpecPrune-VLA combines attention history with an action-aware pruning controller; LightVLA learns differentiable visual-token selection; and VLA-Pruner combines semantic relevance with temporally estimated action relevance (Wang et al., 2026; Jiang et al., 2025; Liu et al., 2026). These approaches prune observed visual tokens. DIWA selects unexpanded future queries, supervising their priority through latent interventions and their representation through candidate-return regret. The distinction concerns the selected computation and its supervision; we do not claim the first action-aware pruning method or an empirical advantage over these untested alternatives.

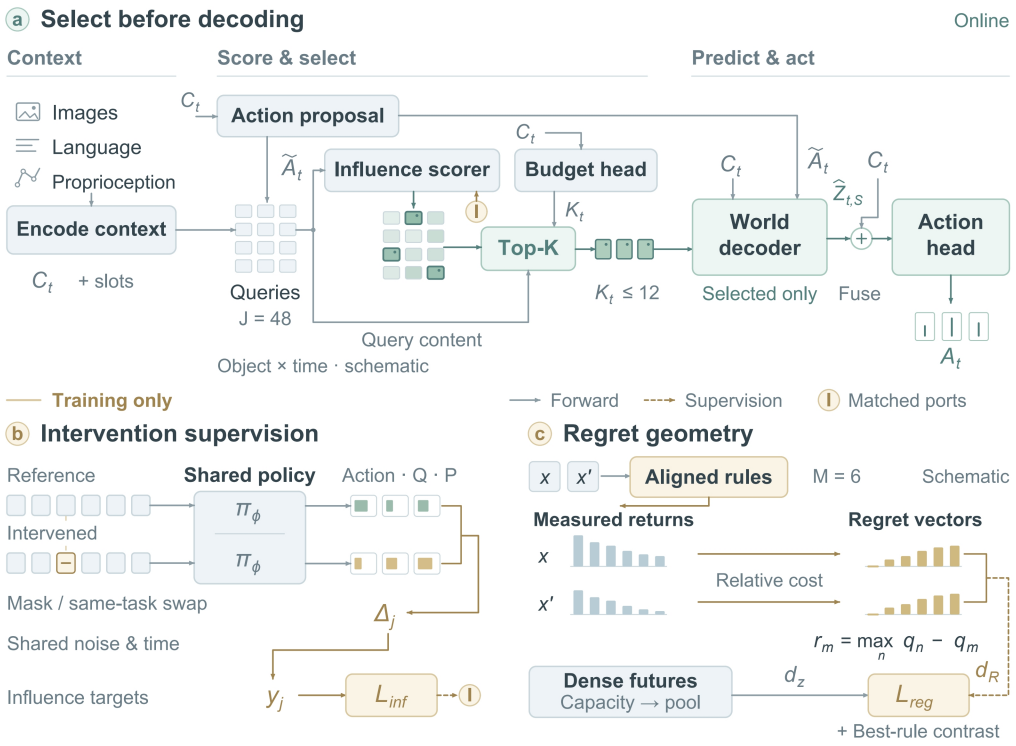


Figure 1: DIWA overview. **(a)** Context C_t and an action proposal form $J = 48$ object–time queries; influence scores and the budget head select $K_t \le 12$ for world decoding and action generation. **(b)** Reference and intervened futures share policy weights, noise, and time. Changes in actions, value Q, and progress P produce influence targets y_j; matching "I" ports connect this supervision to the scorer. **(c)** Measured returns under $M = 6$ aligned rules define regret vectors, which supervise pooled future geometry together with best-rule contrast. Solid arrows show forward computation; dashed gold arrows show supervision. Token grids and bars are schematic.

3 Decision-influential world abstraction

Figure 1 connects the online policy with its two training signals. Panel (a) selects object–time queries before future decoding; panel (b) trains the influence scorer from paired latent interventions; panel (c) organizes future representations using measured candidate-action regret. The supervision branches shape what the online policy predicts and preserves, without requiring intervention probes or candidate-return collection during inference.

3.1 Structured context and action-conditioned futures

Let $x_t = (o_{\le t}, p_{\le t}, \ell)$ denote observation history, proprioception, and a language instruction. The policy predicts an action chunk $A_t = (a_t, \dots, a_{t+L-1}) \in \mathbb{R}^{L \times D}$. The encoder produces context C_t and N object slots $u_t^{1:N} \in \mathbb{R}^{N \times d}$ using Slot Attention and temporal Sinkhorn matching. Matching aligns observations within a trajectory, not object identities across episodes. Segmentation features and point tracks can provide additional inputs.

A small proposal head predicts $\widetilde{A}_t = g(C_t)$. For future offset $k \in \{1, \dots, H\}$ and slot $i \in \{1, \dots, N\}$, define

$$q_t^{k,i} = u_t^i + e_i + e_k + \bar{c}_t + v(\widetilde{A}_t), \qquad \ell_t^{k,i} = G_\omega(q_t^{k,i}), \tag{1}$$

where e_i, e_k are learned slot and time embeddings, $\bar{c}_t$ is normalized pooled context, and v projects and pools the proposed actions. There are $J = HN$ candidate queries. Throughout, a single index j

can represent the pair (k, i). The scorer G_ω is a lightweight multilayer perceptron; evaluating it does not require a predicted future token.

Given selected indices S_t, a Transformer decoder expands only their queries:

$$\widehat{Z}_{t,S_t} = F_\psi(q_{t,S_t}, [C_t, v(\widetilde{A}_t)]), \qquad A_t \sim \pi_\phi(\cdot \mid C_t, \widehat{Z}_{t,S_t}). \tag{2}$$

Learned action queries attend to current context and selected futures, refining the preliminary proposal into the executed action chunk. Regret supervision uses a separate training-time candidate bank.

3.2 Learning influence from latent interventions

During training, we decode all candidate futures and obtain a dense reference policy response. For a sampled index j, we replace the corresponding token while keeping the current context and other future tokens fixed. Replacement alternates between a learned mask token and a token from another episode of the same task when an eligible partner exists. Partner selection prefers similar contexts with different demonstrated actions. A mixture of high-scoring and randomly sampled indices provides both targeted supervision and coverage of tokens that the current scorer might overlook.

Let (μ, v) and $(\mu^{(j)}, v^{(j)})$ be the local action-response means and elementwise variances before and after intervention. For stochastic action heads these quantities are estimated using paired denoising or flow probes, sharing noise and sampled time. They are sensitivity surrogates, not exact moments of the complete action distribution. Define

$$\Delta_j^\pi = \mathrm{RMS}(\mu - \mu^{(j)}) + \beta_v \,\mathrm{RMS}(v - v^{(j)}), \tag{3}$$

where RMS averages over chunk positions and action coordinates. Let b and $b^{(j)}$ be the corresponding pooled action-fusion features. Twin critics supply $Q(b, A) = \min_{r\in\{1,2\}} Q_r(b, A)$, and a progress head supplies $P(b)$. The total intervention response is

$$\Delta_j = \alpha_\pi \Delta_j^\pi + \alpha_Q \left| Q(b, \mu) - Q(b^{(j)}, \mu^{(j)}) \right| + \alpha_P \left| P(b) - P(b^{(j)}) \right|. \tag{4}$$

The value term reflects a joint change in the decision representation and action response; it should not be interpreted as a fixed-action treatment effect.

We map nonnegative responses into bounded supervision using a running scale s_Δ:

$$y_j = \mathrm{sg}[1 - \exp(-\Delta_j / s_\Delta)], \qquad \mathcal{L}_{\mathrm{inf}} = \mathbb{E}_{j\in\mathcal{I}_t} \,\mathrm{Huber}\left(\sigma(\ell_t^j), y_j\right), \tag{5}$$

where $\mathcal{I}_t$ contains sampled interventions and sg stops gradients through the target. An exponential moving average of batch responses sets the scale. Defaults use four interventions and four stochastic probes per reference response.

These latent interventions can produce physically unrealizable or out-of-distribution states. Same-task pairing limits semantic mismatch without eliminating it. Influence therefore guides computational priority; it does not identify environmental causal effects.

3.3 Regret geometry from measured candidate returns

To preserve decision-relevant distinctions, let $\{b_m\}_{m=1}^M$ be candidate action-generation rules with stable meanings across paired states. Candidate m produces $A_m(x)$, and its measured discounted return is $q_m(x)$. A candidate can be state dependent, but the index m must refer to the same rule, such as the demonstrated action, a no-op, or a signed Cartesian perturbation. An arbitrary reordering would invalidate comparisons between regret coordinates.

Define the candidate-relative regret vector and distance

$$r_m(x) = \max_n q_n(x) - q_m(x), \qquad d_R(x, x') = \sum_{m=1}^{M} |r_m(x) - r_m(x')|. \tag{6}$$

Candidate returns are external supervision, distinct from policy-generated value estimates. The LIBERO collector executes each candidate from the same restored simulator state for a short action chunk, using the environment reward and measured predicate progress. This bank defines a local

decision comparison rather than a globally optimal continuous-control value. Appendix B specifies the labels and their collection boundaries.

Let $z(x)$ pool the capacity-adjusted future tokens, and let $d_z(x, x') = 1 - \cos(z(x), z(x'))$. For same-task, cross-episode pairs with valid returns, the geometry loss is

$$\mathcal{L}_{\text{reg}} = \mathbb{E}_{x,x'} \operatorname{Huber}\big(d_z(x, x'), \operatorname{sg}[d_R(x, x')]\big). \tag{7}$$

An auxiliary contrastive term distinguishes equal and unequal best-candidate indices:

$$\mathcal{L}_{\text{con}} = \mathbb{E}_{x,x'}\left[\mathbf{1}_{m^*=m'^*} \operatorname{softplus}(d_z) + \mathbf{1}_{m^*\neq m'^*} \operatorname{softplus}(\delta - d_z)\right], \tag{8}$$

where $m^* = \arg\max_m q_m(x)$ and δ is a margin. Ties follow a fixed candidate ordering. Equal best indices provide coarse supervision: such pairs need not imply equal regret vectors, and this term can compete with full-vector matching. Because $d_z \in [0, 2]$ while d_R is unnormalized, Eq. (7) is a scale-dependent regularizer, not an exact metric embedding (Appendix C).

Proposition 1 (Finite-candidate decision equivalence)**.** *For an aligned candidate bank, $r(x) = r(x')$ if and only if there exists a scalar c such that $q_m(x) = q_m(x') + c$ for every m. Moreover, if $\|r(x) - r(x')\|_\infty \le \epsilon$ and $m' \in \arg\max_m q_m(x')$, then $r_{m'}(x) \le \epsilon$.*

Appendix C proves the proposition by subtracting returns from each state's maximum. Regret preserves candidate choice while removing value offsets. The guarantee concerns the measured bank and does not follow automatically from the sampled geometry objective.

3.4 Sparse allocation, capacity, and training

Adaptive query budget. Let $\rho_{\max}$ be the configured ceiling and $\rho_{\min} > 0$ a floor. A context-conditioned budget head gives

$$\rho_t = \rho_{\max} - \sigma(B_\eta(\bar{c}_t))(\rho_{\max} - \rho_{\min}), \quad K_t = \max(1, \lceil J\rho_t \rceil), \quad S_t = \operatorname{TopK}(\ell_t, K_t). \tag{9}$$

The implementation has a positive minimum budget, so it adapts the amount of imagination without providing a zero-imagination branch. Rounding can make K_t/J slightly larger than a continuous ceiling unless $J\rho_{\max}$ is integral. With $H = 3$, $N = 16$, and $\rho_{\max} = 0.25$, at most 12 of 48 queries are selected per state.

The budget objective aligns the mean sigmoid influence with ρ_t, aligns ρ_t with the detached fraction of scores above a threshold, penalizes ceiling violations, and encourages low gate entropy. Hard Top-K selection itself is discrete; the influence and budget heads receive learning signals through these auxiliary objectives. Selected queries are packed before the world decoder. With variable-length batches, padding up to the largest selected count can still incur computation, so token ratios alone do not establish latency speedups.

Prediction and capacity allocation. An exponential-moving-average tokenizer provides temporally aligned future targets $z^{+,j}$. On valid targets, prediction uses an influence-weighted cosine loss

$$\mathcal{L}_{\text{fut}} = \mathbb{E}_t \sum_j w_t^j \big(1 - \cos(\widehat{z}_t^j, \operatorname{sg}[z_t^{+,j}])\big), \qquad w_t^j \propto \exp\big(\operatorname{sg}[\ell_t^j]/\tau\big). \tag{10}$$

Weights are renormalized over available targets. In the dense training reference, selected tokens retain full capacity, unselected tokens pass through a bottleneck, and very low-scoring tokens additionally stop policy gradients. The action-producing branch consumes selected tokens. At inference, unselected queries never enter the future decoder; the training-time dense reference and intervention passes are absent.

Optimization. The main imitation loss is augmented with proposal, future, influence, budget, critic, progress, regret, and contrastive terms:

$$\mathcal{L} = \mathcal{L}_{\text{act}} + \sum_{u \in \{\text{prop,fut,inf,bud,Q,prog,reg,con}\}} \lambda_u \mathcal{L}_u. \tag{11}$$

Proposal supervision uses continuous-action regression and gripper classification. Critics combine temporal-difference supervision with measured candidate-return regression, and progress uses measured task predicates. Training first establishes action prediction and future targets, then activates decision supervision and tighter budgets. Teacher forcing of demonstration action chunks decays toward conditioning on learned proposals. Exact reference settings and auxiliary objectives appear in Appendix A.

Algorithm 1 DIWA at one control step

Require: Observation history $o_{\le t}$, proprioception $p_{\le t}$, instruction ℓ

1: Encode current context C_t and temporally aligned object slots $u_t^{1:N}$.
2: Predict proposal chunk $\widetilde{A}_t = g(C_t)$ and construct HN queries.
3: Score queries using G_ω and obtain budget K_t using Eq. (9).
4: Gather q_{t,S_t} for $S_t = \mathrm{TopK}(\ell_t, K_t)$ before world decoding.
5: Decode selected futures $\widehat{Z}_{t,S_t} = F_\psi(q_{t,S_t}, [C_t, v(\widetilde{A}_t)])$.
6: Fuse C_t and $\widehat{Z}_{t,S_t}$, then generate the refined action chunk.
7: Execute the configured action prefix and repeat at the next observation.

4 EXPERIMENTS

We evaluate whether decision-guided abstraction improves task completion and robustness, whether selective decoding lowers inference cost, and whether the learned scores and geometry track measured decision consequences.

4.1 EVALUATION SETTINGS

Tasks and statistical units. We evaluate LIBERO (Liu et al., 2023), RoboTwin (Mu et al., 2025), RoboCasa (Nasiriany et al., 2024), and six real-robot manipulation tasks. Simulation experiments use training seeds 42, 43, and 44. LIBERO includes Spatial, Object, Goal, and Long (`libero_10`), each with ten tasks and 50 initial states per task per seed: 6,000 episodes per method. RoboTwin and RoboCasa each use 20 selected tasks and 50 episodes per task per seed: 3,000 episodes per method per platform. The physical evaluation uses the seed-42 checkpoint, with 50 trials on each of six tasks, totaling 300 trials per method. Success requires completion of all task-specific goals.

We average tasks within each platform and then across training seeds; the four-platform aggregate weights the platform means equally. Simulation uncertainty is the sample standard deviation (SD) across three training seeds. Physical trials characterize one checkpoint, so we do not attach a training-seed SD to physical scores or to the mixed four-platform aggregate. Appendix D gives seed scores, SD, standard errors, and physical success counts.

Baselines and implementation. We evaluate Octo, OpenVLA, GR-1, RoboDreamer, WorldVLA, and DreamVLA in this study. DreamVLA serves as the backbone for DIWA and its variants, whereas external baselines retain their architectural and pretraining differences. DIWA also uses candidate-return and progress supervision, so the DreamVLA comparison measures the complete system rather than the isolated effect of query selection. DIWA has 16 slots, three future offsets, three-step action chunks, and a 6.25%–25% adaptive budget. Appendix A specifies the AdamW reference recipe and staged optimization.

Robustness and timing. The OOD evaluation equally averages appearance changes, task-irrelevant background motion, and contact counterfactuals. Appearance changes affect object or container texture, color, and material; background motion introduces distractors outside the task interaction. Contact counterfactuals change contact state or the appropriate action in visually similar scenes. Episode success and counterfactual (CF) decision accuracy are distinct outcomes. OOD and standard results use different populations, so their ratio is not a robustness-retention metric. Latency includes perception, future computation, and action decoding. Appendix E specifies metric definitions and the available comparison scope.

4.2 TASK SUCCESS ACROSS PLATFORMS AND TASK SUITES

DIWA achieves 73.5% four-platform success, compared with DreamVLA's 67.9% (Table 1). Improvements on LIBERO, RoboTwin, RoboCasa, and the physical platform are 4.8, 5.6, 5.3, and 6.7 percentage points, respectively. DIWA has the highest mean on every platform among the evaluated methods. The simulation gains exceed the observed variation across the three training seeds, although these summaries alone do not constitute a significance test.

Table 1: Task success (%) under our evaluation protocol. Simulation columns report mean $\pm$ sample SD across three training seeds. Real-robot results use the seed-42 checkpoint and 300 trials per method. The final column weights the four platform means equally.

Method	LIBERO	RoboTwin	RoboCasa	Real robot	Macro
Octo	73.2 ± 0.70	55.6 ± 0.92	47.2 ± 0.72	50.0	56.5
OpenVLA	76.0 ± 0.56	59.8 ± 0.92	50.2 ± 0.72	56.0	60.5
GR-1	77.4 ± 0.56	61.6 ± 0.92	53.4 ± 0.72	58.0	62.6
RoboDreamer	78.2 ± 0.66	64.0 ± 0.80	56.2 ± 0.72	60.0	64.6
WorldVLA	79.0 ± 0.56	65.8 ± 0.80	57.6 ± 0.92	62.0	66.1
DreamVLA	79.6 ± 0.66	67.6 ± 0.80	59.1 ± 0.81	65.3	67.9
DIWA	$\mathbf{84.4 \pm 0.56}$	$\mathbf{73.2 \pm 0.80}$	$\mathbf{64.4 \pm 0.80}$	**72.0**	**73.5**

Table 2: LIBERO suite success (%). Each suite contains ten tasks with equal evaluation counts. Long denotes `libero_10`. Values average seeds 42, 43, and 44.

Method	Spatial	Object	Goal	Long	Mean
Octo	78.4	79.0	74.0	61.4	73.2
OpenVLA	81.6	82.4	76.8	63.2	76.0
GR-1	83.0	83.6	78.6	64.4	77.4
RoboDreamer	83.6	84.4	79.6	65.2	78.2
WorldVLA	84.2	85.0	80.2	66.6	79.0
DreamVLA	84.8	85.4	80.6	67.6	79.6
DIWA	88.6	89.4	86.0	73.6	84.4

The LIBERO breakdown localizes part of the improvement (Table 2). Relative to DreamVLA, gains are 3.8 points on Spatial, 4.0 on Object, 5.4 on Goal, and 6.0 on Long. The larger gains on Goal and Long are consistent with retaining information about target relations and multi-stage action consequences. They do not by themselves isolate the contribution of either training objective.

On the physical robot, DIWA succeeds in 216 of 300 trials, compared with 196 for DreamVLA. The resulting 20 additional successes correspond to a 6.7-point improvement. The largest task-level gains occur for insertion (52% to 62%) and multi-stage contact (56% to 64%); DIWA improves all six task categories. Appendix D reports the complete counts. These results establish the behavior of the evaluated seed-42 checkpoints, while cross-training-seed physical variation remains outside this evaluation.

4.3 Robustness and counterfactual decisions

DIWA obtains 75.8% OOD success, versus 65.2% for DreamVLA (Table 3). Across training seeds, these means have SDs of 0.80 and 0.92 points. Gains are 10.4 points for appearance changes, 9.8 for background motion, and 11.6 for contact counterfactuals. Improving both visual-shift tolerance and contact discrimination is consistent with the intended abstraction: ignore variation that does not alter the task while preserving distinctions that change the appropriate action. The contact success rate of 76.4% measures completed episodes, whereas DIWA's 81.6% CF accuracy measures decision correctness.

4.4 Inference efficiency and budget sensitivity

DIWA requires 89 ms per inference update, versus 213 ms for dense DIWA: a 58.2% reduction and a 2.39$\times$ speedup. WorldVLA and RoboDreamer require 241 and 318 ms, respectively (Appendix D). These cross-architecture comparisons contextualize pipeline cost; the dense DIWA comparison more directly tests selective future expansion. The adaptive model retains 23.7% of future queries on average. The resulting 76.3% query reduction exceeds the latency reduction because perception, scoring, and action generation still incur computation.

Figure 2 distinguishes fixed query counts from an adaptive ceiling. Fixed 50% decoding uses 24 queries and attains 74.1% at 132 ms. Fixed 25% uses 12 queries and attains 73.6% at 91 ms, compared

Table 3: Success under visual shifts and contact counterfactuals (%). Contact episode success differs from the CF decision accuracy in Table 4.

Condition	DreamVLA	DIWA	Gain (points)
Appearance	65.8	76.2	+10.4
Background motion	65.0	74.8	+9.8
Contact counterfactual	64.8	76.4	+11.6
Equal-weight mean	65.2	75.8	+10.6

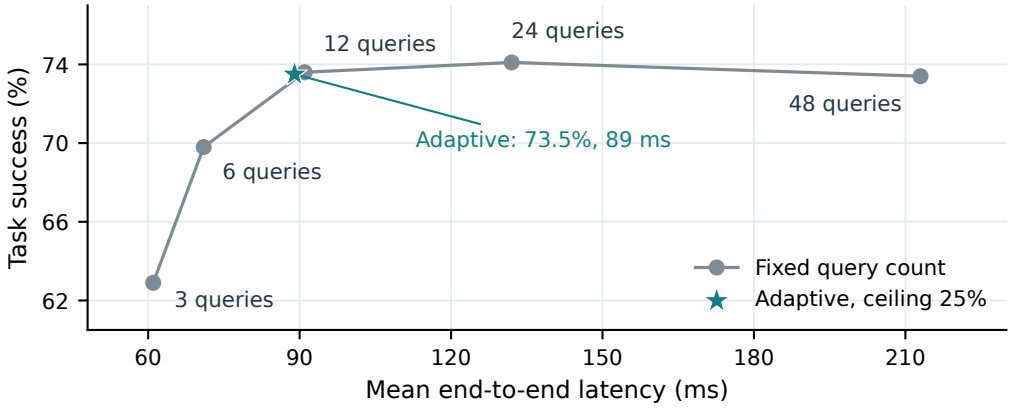


Figure 2: Measured success–latency tradeoff. Gray points fix the decoder-query count out of 48 candidates; the teal star is the adaptive main setting with a 25% ceiling. Fixed 12-query decoding yields 73.6% success at 91 ms; adaptive decoding yields 73.5% at 89 ms. Lines connect measured fixed-budget settings as a visual guide.

with 73.4% at 213 ms for all 48 queries. Reducing the budget to six or three queries lowers latency to 71 or 61 ms but reduces success to 69.8% or 62.9%. Thus, moderate sparsity preserves performance, whereas aggressive pruning removes useful information. The adaptive setting offers a measured operating point close to fixed 25%; the 2 ms difference is not evidence of a statistically established advantage. Complete values appear in Table 10.

4.5 Ablations and mechanism diagnostics

Influence and selection. The no-influence variant yields standard and OOD success rates lower by 5.3 and 9.7 points respectively (Table 4). Uniform Top-K selection yields success rates lower by 2.9 and 6.9 points respectively, and a CF accuracy of 74.3%.. These comparisons concern evaluated variants; attributing their differences solely to ranking requires matched retained counts and training settings. The 142 ms latency of the no-influence variant does not identify the computational cost of the scorer. Dense DIWA achieves similar success and CF accuracy at 213 ms, supporting the measured efficiency of the sparse system. Appendix E separates variant semantics from implementation switches.

Intervention and regret supervision. Removing cross-episode swaps lowers CF accuracy by 8.8 points and OOD success by 5.6 points. Removing regret geometry lowers standard success by 4.2 points and CF accuracy by 11.0 points. Their 88 and 89 ms latencies are close to the full model's 89 ms. The point estimates favor both training signals, without establishing statistical significance or physical validity for every latent substitution.

Do the learned quantities track decision consequences? On independent test states, influence scores correlate with intervention-induced action changes at Spearman $\rho = 0.71$, versus 0.48 without swaps (Table 5). Paired probes hold action noise and time fixed. Sweeping all 48 interventions

Table 4: Evaluated DIWA variants. Standard success is the four-platform macro-average; OOD success and CF decision accuracy are separate metrics. Rates are percentages; latency is in milliseconds. These point estimates do not establish equal retained counts across variants.

Variant	Standard	OOD	CF accuracy	Latency
DIWA	73.5	75.8	81.6	89
Without influence estimator	68.2	66.1	71.9	142
Without cross-episode swaps	71.4	70.2	72.8	88
Without regret geometry	69.3	68.2	70.6	89
Dense imagination	73.4	75.5	81.5	213
Uniform Top-K	70.6	68.9	74.3	88

Table 5: Mechanism diagnostics on independent test states or state pairs, reported as point estimates. Correlations use Spearman's ρ; Top-12 recall compares the predicted ranking with interventions on all 48 candidates.

Diagnostic	DIWA	Control	Control variant
Influence–action change ρ	0.71	0.48	Without swaps
Top-12 recall (%)	83	65	Without swaps
Latent–regret distance ρ	0.68	0.29	Without regret

supplies the reference ranking; Top-12 recall is 83%, versus 65%. For same-task, cross-episode pairs with independently measured returns, latent–regret distance correlation is 0.68, versus 0.29 without regret geometry. These point estimates support an association with the training targets. They do not establish metric calibration or quantify uncertainty without the underlying state and pair records.

5 Limitations and Discussion

Influence depends on the action model and critics, so a mistaken policy may ignore an important physical variable. Single-token interventions miss interactions and redundancy, while learned slots and cross-episode substitutions need not correspond to physical objects or realizable states. The diagnostics measure model sensitivity, not identified environmental causal effects.

Regret supervision requires outcomes for aligned candidate actions and adds data-collection cost. A six-rule bank and short branches may omit useful actions or delayed effects; binary progress can also leave candidates tied away from completion. Physical candidate collection is separate from success evaluation. The geometry loss has a bounded prediction range and can conflict with the coarser best-index objective, so the benefit is empirical rather than an exact embedding guarantee.

Dense training and sparse inference use different query interactions, and padded batches may reduce savings. The minimum budget precludes skipping imagination entirely. Three training seeds and one physical checkpoint leave uncertainty about broader variation and action-head transfer. Comparisons with recent VLA pruning methods, training-cost measurements, and complete run-level timing and diagnostic records remain outside the available evidence (Appendix E).

6 Conclusion

DIWA scores object–time queries before future decoding, learns computational priorities from latent interventions, and regularizes representations with measured candidate-action regret. Across three simulation platforms and physical manipulation, it improves task success and OOD performance while reducing latency relative to dense imagination. Budget sweeps identify a useful sparsity range, and independent intervention diagnostics support the intended influence ranking and regret geometry. The results connect predictive computation to robot decisions and motivate interaction-aware selection and longer-horizon supervision.

AI USE STATEMENT

Generative AI was used for language polishing to improve the grammar, clarity, and readability of the manuscript.

REPRODUCIBILITY STATEMENT

Equations (1)–(11) and Algorithm 1 specify the method. Appendix A gives the reference implementation and optimization; Appendix B defines supervision; and Appendix C proves the finite-candidate guarantee. Appendix D reports simulation seeds, physical success counts, and statistical conventions. The accompanying manuscript source and numerical records support regeneration of the paper's tables and figures. This reproduces the reported summaries; reproducing policy training and evaluation additionally requires the experiment records discussed in Appendix E.

REFERENCES

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. doi: 10.15607/RSS.2025.XXI.010. URL https://www.roboticsproceedings.org/rss21/p010.html.

Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, Deli Zhao, and Hao Chen. WorldVLA: Towards autoregressive action world model. *arXiv preprint arXiv:2506.21539*, 2025. URL https://arxiv.org/abs/2506.21539v1. Version 1, 26 June 2025.

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.

Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Quan Vuong, Ted Xiao, Pannag R Sanketi, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, 2024. doi: 10.15607/RSS.2024.XX.090. URL https://www.roboticsproceedings.org/rss20/p090.html.

Christopher Grimm, Andre Barreto, Satinder Singh, and David Silver. The value equivalence principle for model-based reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5541–5552. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/3bb585ea00014b0e3ebe4c6dd165a358-Abstract.html.

Nick Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, pp. 47376–47405, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/cf73d57b6dcda32b293df7c2d5341f49-Abstract-Conference.html.

Titong Jiang, Xuefeng Jiang, Yuan Ma, Xin Wen, Bailin Li, Kun Zhan, Peng Jia, Yahui Liu, Sheng Sun, and Xianpeng Lang. The better you learn, the smarter you prune: Towards efficient vision-language-action models via differentiable token pruning. *arXiv preprint arXiv:2509.12594*, 2025. URL https://arxiv.org/abs/2509.12594v2. Version 2, 21 September 2025.

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.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 44776–44791. Curran Associates, Inc., 2023. doi: 10.52202/075280-1939. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/8c3c666820ea055a77726d66fc7d447f-Abstract-Datasets_and_Benchmarks.html.

Ziyan Liu, Yeqiu Chen, Hongyi Cai, Tao Lin, Shuo Yang, Zheng Liu, and Bo Zhao. Bridging the semantic-action gap in visual token pruning for efficient VLA inference. *arXiv preprint arXiv:2511.16449*, 2026. URL https://arxiv.org/abs/2511.16449v5. Version 5, 26 May 2026.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems*, volume 33, pp. 11525–11538. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/8511df98c02ab60aea1b2356c013bc0f-Abstract.html.

Yao Mu, Tianxing Chen, Zanxin Chen, Shijia Peng, Zhiqian Lan, Zeyu Gao, Zhixuan Liang, Qiaojun Yu, Yude Zou, Mingkun Xu, Lunkai Lin, Zhiqiang Xie, Mingyu Ding, and Ping Luo. RoboTwin: Dual-arm robot benchmark with generative digital twins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27649–27660, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Mu_RoboTwin_Dual-Arm_Robot_Benchmark_with_Generative_Digital_Twins_CVPR_2025_paper.html.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. RoboCasa: Large-scale simulation of household tasks for generalist robots. In *Proceedings of Robotics: Science and Systems*, 2024. doi: 10.15607/RSS.2024.XX.050. URL https://www.roboticsproceedings.org/rss20/p050.html.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. DynamicViT: Efficient vision transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems*, volume 34, pp. 13937–13949. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/hash/747d3443e319a22747fbb873e8b2f9f2-Abstract.html.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588: 604–609, 2020. doi: 10.1038/s41586-020-03051-4. URL https://doi.org/10.1038/s41586-020-03051-4.

Hanzhen Wang, Jiaming Xu, Yushun Xiang, Jiayi Pan, Yongkang Zhou, Yong-Lu Li, and Guohao Dai. SpecPrune-VLA: Accelerating vision-language-action models via action-aware self-speculative pruning. *arXiv preprint arXiv:2509.05614*, 2026. URL https://arxiv.org/abs/2509.05614v3. Version 3, 24 May 2026. Accepted at ICML 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*, pp. 10641–10662, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/2c37c5bcef24b9541550261dcd63261b-Abstract-Conference.html.

Amy Zhang, Rowan McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/pdf/603fb2b2d3c728fdb6a0d1f23a60748227ff46c7.pdf.

Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, XinQiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, He Wang, Zhizheng Zhang, Li Yi, Wenjun Zeng, and Xin Jin. DreamVLA: A vision-language-action model dreamed with comprehensive world knowledge. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 24195–24228. Curran Associates, Inc., 2025. doi: 10.52202/085713-0818. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/22d4f952efa13970f0b1ffb22170d416-Abstract-Conference.html.

Siyuan Zhou, Yilun Du, Jiaben Chen, Yandong Li, Dit-Yan Yeung, and Chuang Gan. RoboDreamer: Learning compositional world models for robot imagination. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 61885–61896. PMLR, 2024. URL https://proceedings.mlr.press/v235/zhou24f.html.

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.

A Reference implementation and optimization

We implement DIWA as an optional DreamVLA path with an object tokenizer, proposal and influence heads, an adaptive budget, an action-conditioned world decoder, policy fusion, twin critics, and a progress predictor. Target critics and the target object encoder use exponential moving averages. Table 6 specifies the reference LIBERO recipe and evaluation seeds. Its script defaults describe the implementation; they do not establish the hardware or configuration of every reported run.

Table 6: Reference architecture and optimization settings, with the evaluation seeds. The 8-device launch and its effective batch are script defaults; actual hardware requires the corresponding run record.

Setting	Value
Hidden width / attention heads	1024 / 16
Object slots N / future offsets H	16 / 3
World-decoder layers / policy-fusion layers	2 / 2
Action-chunk length / policy sequence length	3 / 7
Observation window including future targets	10
Slot Attention iterations	3
Maximum / minimum query-budget ratio	0.25 / 0.0625
Influence temperature τ / gate threshold	0.25 / 0.5
Intervened tokens / stochastic probes	4 / 4
Candidate rules M / critic discount	6 / 0.99
Regret contrastive margin / EMA update rate	0.5 / 0.005
Action-variance coefficient β_v	0.1
Influence weights $(\alpha_\pi, \alpha_Q, \alpha_P)$	$(1, 1, 1)$
Learning rate / weight decay	10^{-4} / 10^{-4}
Epochs / scheduler warm-up epochs	40 / 5
Per-device batch size / gradient accumulation	16 / 4
Default training devices / effective batch	8 / 512
Policy / visual-module precision	FP32 / BF16
Training seeds used in simulation	42, 43, 44
Physical checkpoint seed / evaluation-script default	42 / 66

Loss weights. The weights for proposal, future, influence, budget, critic, progress, regret, and contrastive learning are $(0.1, 0.1, 0.1, 0.001, 0.1, 0.05, 0.01, 0.01)$. The action imitation objective is inherited from the configured action head. The proposal head uses smooth-L_1 regression for continuous coordinates and binary cross-entropy for gripper coordinates, with a 0.01 relative gripper weight. Action normalization and gripper conversion must match the data adapter; the LIBERO adapter maps native $-1/+1$ gripper values to the policy's $0/1$ convention.

Staging. Proposal and future prediction are active from initialization. Critic and progress objectives start at step 1,000; counterfactual influence and budget objectives also start at step 1,000; regret and contrastive objectives start at step 2,000. The budget schedule has a 1,000-step warm-up followed by 10,000 annealing steps. The teacher-forcing schedule spans 5,000 steps. The reference command uses FP32 policy computation with BF16 for the vision module. These settings describe the source recipe, rather than an unspecified hyperparameter search.

Budget objective. Writing $p_j = \sigma(\ell_j)$ and $\bar{p} = J^{-1} \sum_j p_j$, the detached target allocation is

$$\widehat{\rho} = \mathrm{clip}\left(J^{-1} \sum_j \mathbf{1}[\mathrm{sg}[p_j] \geq \kappa], \rho_{\min}, \rho_{\max} \right). \tag{12}$$

The auxiliary objective is

$$\mathcal{L}_{\mathrm{bud}} = \mathbb{E}\left[|\bar{p} - \rho_t| + |\rho_t - \widehat{\rho}| + [\rho_t - \rho_{\max}]_+ \right] + \lambda_{\mathrm{ent}} \mathbb{E}_j h(p_j), \tag{13}$$

where $h(p) = -p \log p - (1-p) \log(1-p)$, $\kappa = 0.5$, and $\lambda_{\text{ent}} = 0.01$. The ceiling term is redundant when the bounded head is used exactly as specified, but is included in the implementation. The initial budget-head bias is -4, placing initial adaptive budgets near their ceiling.

Training reference capacity. For a dense future token z_j, the reference branch retains z_j when it is selected, otherwise applies a learned bottleneck $B(z_j)$. If an unselected token's sigmoid score is below $\kappa/2$, the resulting bottleneck feature is detached. This capacity adjustment affects the dense reference and pooled regret representation. The online branch gathers only selected queries and does not generate or pool all low-influence future tokens.

Computational scope. For a Transformer world decoder with context length T and width d, replacing J future queries by K changes its attention terms from $O(J^2d + JTd)$ to $O(K^2d + KTd)$ and its tokenwise projections from $O(Jd^2)$ to $O(Kd^2)$. Visual encoding, query scoring, and action generation remain. The complexity comparison concerns the future decoder; measured end-to-end latency is the relevant control-system quantity. Dense decoding during training also means that online savings should not be read as training-cost savings.

B Measured supervision and candidate semantics

The strict supervision interface requires measured reward, episode termination, task progress, candidate action chunks, and candidate returns. For an episode of length T_e, their shapes are $[T_e]$, $[T_e]$, $[T_e]$, $[T_e, M, L, D]$, and $[T_e, M]$. Missing or malformed fields fail validation. Each return must correspond to its accompanying state and action chunk, with common candidate semantics, normalization, discount, horizon, and shaping across paired states. This interface validates data structure; it does not establish how labels were collected on a particular platform.

LIBERO branch measurement. The collector restores the recorded simulator state before each candidate. Candidate 0 replays the demonstration chunk, candidate 1 uses no continuous motion while retaining the gripper command, and candidates 2–5 add $+x$, $-x$, $+y$, and $-y$ Cartesian perturbations to the demonstration, respectively. With six candidates, the finite rule bank does not cover every Cartesian direction or every possible robot action. The default perturbation magnitude is 0.15 in the normalized action coordinates. With $n_m \le L$ executed steps, the return is

$$q_m(x) = \sum_{s=0}^{n_m-1} \gamma^s R_s + \lambda_p \gamma^{n_m} \max(0, P_{\text{end}} - P_{\text{start}}), \tag{14}$$

where $\gamma = 0.99$ and the collector's default $\lambda_p = 1$. The reference collector obtains P from the environment success check, so its measured progress is binary, $P \in \{0, 1\}$; it does not implement intermediate quarter-stage labels. Execution stops on episode termination. This is a short-branch shaped return, not an independently established infinite-horizon Q^*.

Critic objectives and interpretation. For valid consecutive sequence states, twin critics use a target

$$y_t^{\text{TD}} = R_t + \gamma(1 - d_t) \min_{r \in \{1,2\}} \overline{Q}_r(b_{t+1}, \widetilde{A}_{t+1}). \tag{15}$$

Smooth-L_1 losses fit both online critics to this target and to supplied candidate-return labels. A sigmoid progress head is trained with binary cross-entropy on measured progress in $[0, 1]$. Short-branch candidate targets and bootstrapped TD targets can have different horizons and shaping conventions. Thus, the critic is a practical auxiliary sensitivity model; a single calibrated value-function interpretation requires the data-collection horizon and reward conventions to be aligned. The regret loss uses the supplied candidate returns directly and does not require interpreting the learned critic as ground-truth return.

Offline and physical data. Recorded trajectories reveal outcomes only for executed actions. Offline adapters mark unavailable counterfactual returns rather than treating them as zero. Strict training requires external candidate supervision; a non-strict critic-derived fallback has different semantics and is not a measured-return substitute. Simulator state restoration specifies the LIBERO branch

procedure. RoboTwin, RoboCasa, and physical candidate-label collection cannot be inferred from their task-success counts; platform-specific reset, repetition, and return records are required to reproduce that supervision. Additional branch interactions also belong in the training-data budget when comparing against imitation-only baselines.

C Properties and boundaries of regret geometry

Proof of Proposition 1. Write $v(x) = \max_m q_m(x)$. If $r(x) = r(x')$, then for every m,

$$v(x) - q_m(x) = v(x') - q_m(x'),$$

so $q_m(x) - q_m(x') = v(x) - v(x')$, a constant independent of m. Conversely, if $q_m(x) = q_m(x') + c$ for all m, taking maxima gives $v(x) = v(x') + c$, and subtraction gives equal regrets. For the second assertion, optimality of m' at x' gives $r_{m'}(x') = 0$. Hence

$$0 \le r_{m'}(x) = |r_{m'}(x) - r_{m'}(x')| \le \|r(x) - r(x')\|_\infty \le \epsilon.$$

□

Noisy measurements. Suppose observed candidate returns $\widetilde{q}$ have uniform error at most η. The maximum value also has error at most η, so each regret coordinate has error at most 2η. If measured regret vectors satisfy $\|\widetilde{r}(x) - \widetilde{r}(x')\|_\infty \le \epsilon$ and m' maximizes $\widetilde{q}_m(x')$, then $\widetilde{r}_{m'}(x') = 0$ and

$$r_{m'}(x) \le \widetilde{r}_{m'}(x) + 2\eta \le \epsilon + 2\eta.$$

This elementary stability bound assumes consistent candidate semantics and bounded label error. Neither assumption is guaranteed by visual similarity or a learned critic. It is not an end-to-end policy-performance bound.

Why aggregate influence is not additive. Consider a binary decision $f(z_1, z_2) = z_1 \vee z_2$ at $(1, 1)$ and an intervention replacing a token by 0. Removing either token alone leaves the decision unchanged, but removing both changes it. Single-token influence can therefore underestimate the importance of a jointly useful subset. Similarly, independent ranking cannot generally optimize a non-additive set objective. The minimum budget and task supervision provide practical constraints, but do not prove preservation of all decision-relevant information.

Geometry scale and fit residual. Cosine dissimilarity lies in $[0, 2]$, whereas d_R sums unnormalized return differences. For the unit-transition smooth-L_1 penalty h, every pair therefore obeys

$$\mathrm{Huber}(d_z, d_R) = h(d_z - d_R) \;\ge\; h\big([d_R - 2]_+\big). \tag{16}$$

This lower bound does not imply that all pairwise targets are jointly realizable. For $d_R \ge 3$, the derivative with respect to $d_z \in [0, 2]$ is -1: sufficiently large targets push representations apart without encoding their full magnitude. Reward scaling and candidate count thus change the regularizer. The reported formulation uses raw returns; introducing a normalized target would change the method and require reevaluation.

For an aligned bank, let $e(x, x') = |d_z(x, x') - d_R(x, x')|$. If m' maximizes the candidate return at x', Proposition 1 implies

$$r_{m'}(x) \le \|r(x) - r(x')\|_\infty \le d_R(x, x') \le d_z(x, x') + e(x, x'). \tag{17}$$

Small latent distance controls candidate regret only together with small fit residual. Spearman correlation alone establishes neither condition. The noisy-label bound adds 2η when distances and maximizers use measured returns.

Ties and supervision content. If all candidate returns at a state are equal, its regret vector is zero and the best-rule index is selected by tie order. Short branches with binary progress can yield such states. A geometry diagnostic therefore needs the proportion of nonconstant return vectors, best-index tie frequency, the $d_R > 2$ rate, and residuals on its actual evaluation pairs. These quantities cannot be reconstructed from aggregate rank correlations; no empirical values for them are inferred here.

D COMPLETE EVALUATION RESULTS AND STATISTICAL CONVENTIONS

Table 7 reports each simulation training seed; Table 8 gives suite-level seed scores; Table 9 reports physical success counts. All baseline scores are measurements from this study. The numerical ledger separates measured scores and counts from computed means, uncertainty summaries, and relative improvements.

Aggregation and uncertainty. For training-seed scores s_1, s_2, s_3, we compute $\bar{s} = \frac{1}{3}\sum_i s_i$, sample SD $\sqrt{\sum_i (s_i - \bar{s})^2/2}$, and SEM $= \mathrm{SD}/\sqrt{3}$. These describe training-seed variation, not variation among individual episodes. Simulation evaluates 50 initial states per task per training seed. The evaluation-script seed 66 is a separate randomness setting and is not a fourth training seed. The four-platform macro-average combines three seed-averaged simulation scores with one physical-checkpoint score, so no joint seed-level SD is inferred.

Averages use unrounded platform values before final rounding. In particular, DreamVLA's RoboCasa mean is $(58.2 + 59.2 + 59.8)/3 = 59.0667\%$, and its physical score is $196/300 = 65.3333\%$. Together with LIBERO and RoboTwin, these yield a four-platform macro-average of 67.9%. DIWA's physical gain is $(216 - 196)/300 \times 100 = 6.6667$ points. No statistical significance claim is made from the aggregate counts alone.

For a physical task with s successes in $n = 50$ trials, Table 9 additionally reports a conditional Wilson interval. With $\widehat{p} = s/n$ and $z_{.975} \approx 1.95996$, its endpoints are

$$\frac{\widehat{p} + z_{.975}^2/(2n) \;\pm\; z_{.975}\sqrt{\widehat{p}(1-\widehat{p})/n + z_{.975}^2/(4n^2)}}{1 + z_{.975}^2/n}. \tag{18}$$

This calculation assumes independent Bernoulli trials within each task and conditions on the trained checkpoint. It does not capture variation across training seeds or provide a paired test between methods. We do not pool heterogeneous tasks into a single common-probability binomial interval.

Table 7: Measured simulation scores for each training seed and recomputed statistics. SD is the sample standard deviation across three training seeds; SEM equals $\mathrm{SD}/\sqrt{3}$. Rates are percentages, and SD/SEM are percentage points.

Method	Platform	42	43	44	Mean	SD	SEM
Octo	LIBERO	72.4	73.5	73.7	73.2	0.70	0.40
Octo	RoboTwin	54.6	55.8	56.4	55.6	0.92	0.53
Octo	RoboCasa	46.6	47.0	48.0	47.2	0.72	0.42
OpenVLA	LIBERO	75.4	76.1	76.5	76.0	0.56	0.32
OpenVLA	RoboTwin	59.0	59.6	60.8	59.8	0.92	0.53
OpenVLA	RoboCasa	49.4	50.4	50.8	50.2	0.72	0.42
GR-1	LIBERO	76.8	77.5	77.9	77.4	0.56	0.32
GR-1	RoboTwin	60.8	61.4	62.6	61.6	0.92	0.53
GR-1	RoboCasa	52.6	53.6	54.0	53.4	0.72	0.42
RoboDreamer	LIBERO	77.5	78.3	78.8	78.2	0.66	0.38
RoboDreamer	RoboTwin	63.2	64.0	64.8	64.0	0.80	0.46
RoboDreamer	RoboCasa	55.4	56.4	56.8	56.2	0.72	0.42
WorldVLA	LIBERO	78.4	79.1	79.5	79.0	0.56	0.32
WorldVLA	RoboTwin	65.0	65.8	66.6	65.8	0.80	0.46
WorldVLA	RoboCasa	56.8	57.4	58.6	57.6	0.92	0.53
DreamVLA	LIBERO	78.9	79.7	80.2	79.6	0.66	0.38
DreamVLA	RoboTwin	66.8	67.6	68.4	67.6	0.80	0.46
DreamVLA	RoboCasa	58.2	59.2	59.8	59.1	0.81	0.47
DIWA	LIBERO	83.8	84.5	84.9	84.4	0.56	0.32
DIWA	RoboTwin	72.4	73.2	74.0	73.2	0.80	0.46
DIWA	RoboCasa	63.6	64.4	65.2	64.4	0.80	0.46
DreamVLA	OOD	64.4	65.0	66.2	65.2	0.92	0.53
DIWA	OOD	75.0	75.8	76.6	75.8	0.80	0.46

Budget and pipeline cost. Table 10 records fixed counts separately from the adaptive main setting. Its mean count of approximately 11.38 queries is a conversion of the rounded 23.7% retention

Table 8: Measured LIBERO suite scores by training seed. Averaging the four suite scores within each seed recovers that seed's LIBERO score in Table 7.

Method	Suite	42	43	44	Mean	SD
DIWA	Spatial	87.8	88.8	89.2	88.6	0.72
DIWA	Object	89.0	89.4	89.8	89.4	0.40
DIWA	Goal	85.4	86.2	86.4	86.0	0.53
DIWA	Long	73.0	73.6	74.2	73.6	0.60
DreamVLA	Spatial	84.0	84.8	85.6	84.8	0.80
DreamVLA	Object	84.8	85.6	85.8	85.4	0.53
DreamVLA	Goal	79.8	80.8	81.2	80.6	0.72
DreamVLA	Long	67.0	67.6	68.2	67.6	0.60

Table 9: Physical success counts and rates for the seed-42 checkpoints. Brackets give conditional within-task Wilson 95% intervals for success rates (%), assuming independent Bernoulli trials within a task. They do not measure training-seed variation. The total row reports aggregate rates without imposing a common success probability across tasks.

Task	DreamVLA	Rate [95% CI]	DIWA	Rate [95% CI]
Grasping	40/50	80.0 [67.0, 88.8]	42/50	84.0 [71.5, 91.7]
Precise placement	35/50	70.0 [56.2, 80.9]	38/50	76.0 [62.6, 85.7]
Drawer interaction	33/50	66.0 [52.2, 77.6]	36/50	72.0 [58.3, 82.5]
Insertion	26/50	52.0 [38.5, 65.2]	31/50	62.0 [48.2, 74.1]
Container manipulation	34/50	68.0 [54.2, 79.2]	37/50	74.0 [60.4, 84.1]
Multi-stage contact	28/50	56.0 [42.3, 68.8]	32/50	64.0 [50.1, 75.9]
Total / mean	196/300	65.3	216/300	72.0

rate, not a noninteger per-state allocation. Figure 3 gives the pipeline comparisons. An 89 ms update corresponds arithmetically to 11.2 serial policy updates per second before communication, observation waiting, and physical execution; this does not specify the robot's low-level control rate. Mean latency alone does not determine P95 latency or memory consumption.

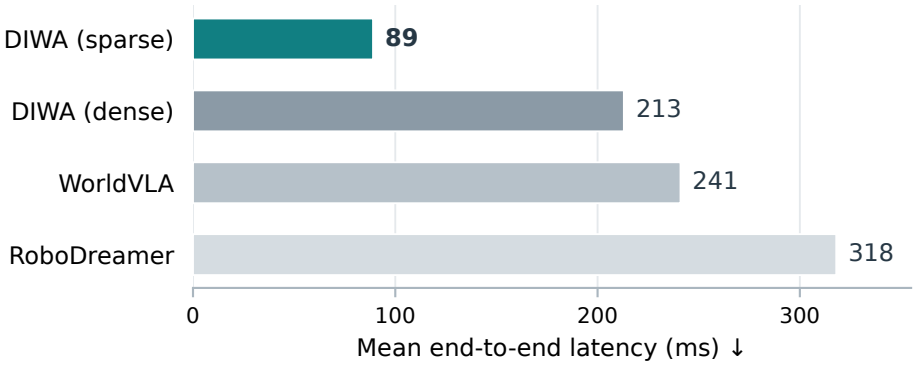


Figure 3: Measured mean inference latency. Adaptive DIWA uses 89 ms, dense DIWA 213 ms, WorldVLA 241 ms, and RoboDreamer 318 ms. Relative reductions are 58.2%, 63.1%, and 72.0%, respectively. Values are point estimates without timing error bars.

Table 10: Measured budget sweep with 48 candidate queries. The first five rows fix the query count; the last row adapts it under a 25% ceiling. The approximate adaptive mean count is derived from the rounded 23.7% retention rate.

Setting	Queries retained	Success (%)	Latency (ms)
Fixed 100%	48	73.4	213
Fixed 50%	24	74.1	132
Fixed 25%	12	73.6	91
Fixed 12.5%	6	69.8	71
Fixed 6.25%	3	62.9	61
Adaptive, ceiling 25%	≈ 11.38 (mean)	73.5	89

E Evaluation definitions and comparison scope

Episode completion, decision-label correctness, intervention ranking, and return geometry have distinct statistical units. Physical success-trial counts do not specify the CF sample size or candidate-return labels.

Counterfactual decision accuracy. For predicted decision labels $\widehat{c}_i$ and reference labels c_i, the evaluator computes

$$\mathrm{Acc}_{\mathrm{CF}} = \frac{1}{N_{\mathrm{dec}}} \sum_{i=1}^{N_{\mathrm{dec}}} \mathbf{1}[\widehat{c}_i = c_i]. \tag{19}$$

N_{dec} counts label entries, not episodes. This compares decision labels rather than continuous action vectors. Reproduction requires the action-to-label mapping, reference labels, and counterfactual scene pairing. Invariant-scene action consistency is a separate metric.

Rank correlations and ties. Spearman's ρ correlates average ranks, including ties. The reference influence evaluator averages within-state correlations and Top-12 recall, $|S_{12}^{\mathrm{pred}} \cap S_{12}^{\mathrm{int}}|/12$. The reference set sweeps all 48 interventions with shared action noise and sampled time. Independent uniform selection has expected recall 25%, a mathematical reference rather than a measured control. The regret evaluator correlates cosine and regret-L_1 distances over the upper triangle of a pairwise matrix. It has no internal task or episode filter: same-task, cross-episode restrictions and subsampling depend on the evaluation data. Constant-rank inputs return zero by convention. Top-K ties require an index rule; pairs sharing an episode are not independent uncertainty replicates.

Run-level reproducibility. The records omit actual GPUs, device counts, precision, batches, resolutions, software versions, sampling steps, timing warm-up/repetitions, and checkpoints. The reference profiler uses synchronized wall-clock timing; its defaults do not identify each measured run. Task IDs, benchmark versions, trajectory splits, clean/OOD pairing, and baseline fine-tuning configurations are also absent. Consequently, timing results describe measured operating points without uncertainty estimates or hardware-independent speed guarantees.

Physical and diagnostic records. The task counts do not specify robot, gripper, cameras, resets, control settings, or candidate-return collection. The aggregate diagnostics omit CF labels and denominators, state/pair counts, sampling rules, action normalization, and individual responses. These summaries do not determine diagnostic uncertainty or label diversity.

Variant semantics. The no-influence variant measures 142 ms. Its reference switch zeros logits but still evaluates the estimator without expanding the budget; this alone cannot explain the latency difference. Equal logits followed by Top-K also do not implement uniform random sampling. Isolating ranking quality requires the actual index rule and matched retained counts. Fixed-count sweeps must disable adaptation or enforce K, rather than only changing its ceiling. The aggregate record leaves retained-count matching and checkpoint/retraining relationships unresolved, precluding attribution to scorer overhead or a statistically established adaptive-budget advantage.