

PAVE: PREDICTIVE ALIGNMENT AND VALUE-GUIDED EVOLUTION FOR WORLD-ACTION POLICIES

Botong Zhao^{1,2,3} Fang Yu^{1,3} Tim Yu² Senhua Zhu² Xinyuan Chen³ Yue Lu^{1,*}

¹Multi-Dimensional Information Processing Laboratory, East China Normal University

²EBKernel ³Shanghai Artificial Intelligence Laboratory

*Corresponding author

ABSTRACT

Direct vision-language-action policies generate continuous robot actions efficiently, but standard behavior cloning leaves two complementary gaps: their representations are not explicitly required to describe how the scene evolves over multiple time scales, and deployment trajectories of unequal quality are often reused without separating useful dynamics from undesirable behavior. We introduce PAVE, a direct world-action policy that combines outcome-agnostic predictive learning with outcome-aware policy improvement. PAVE first retains a local fixed-offset JEPA objective and adds trajectory-relative multi-horizon transition alignment at 25%, 50%, 75%, and 100% of the remaining episode. These training-only targets require the current policy representation to preserve both local physical changes and longer-range task progress, without supplying explicit future tokens to the action head. PAVE then trains an independent distributional value critic on cumulative deployment trajectories, computes action-chunk-aligned N -step advantages, and converts them into positive, negative, or null text conditions for a flow-matching actor. Thus, every valid trajectory can teach what physically happened, while the actor is deployed only under the condition associated with relatively better actions. The multi-horizon predictor and critic are removed from online execution, preserving direct action generation from the current observation, language instruction, and proprioception. Across the three simulation benchmarks, PAVE achieves the strongest overall performance while preserving the direct actor’s online execution path.

1 INTRODUCTION

Vision-language-action (VLA) models map visual observations and language instructions directly to robot actions, offering a scalable interface for generalist manipulation. Recent systems use large pretrained vision-language backbones and expressive continuous action decoders, including diffusion policies and flow-matching action experts (Chi et al., 2023; Kim et al., 2024; Black et al., 2024; Physical Intelligence et al., 2025b). Their direct execution path is attractive for closed-loop control, but the dominant action-imitation objective only asks a policy to reproduce demonstrated actions. It does not explicitly require the internal representation to encode what environmental transition an action is associated with, especially over time scales longer than one action chunk.

World-action models address this limitation by introducing future prediction. Pixel-generative models can imagine future observations, whereas latent world models predict compact state representations and can support planning (Assran et al., 2025). More recent methods move predictive supervision inside a direct policy. JEPA-VLA augments VLA perception with video-predictive embeddings, VLA-JEPA learns leakage-free future latent prediction, Fast-WAM removes future generation at deployment, and JEPA-WAM jointly shapes action representations with current-future joint embeddings (Miao et al., 2026; Sun et al., 2026; Yuan et al., 2026; Lin et al., 2026). These methods establish that future-aware training can benefit control without necessarily generating RGB futures online. However, a single fixed future target captures only one temporal scale: it may describe local motion but under-specify intermediate progress and trajectory-level consequences.

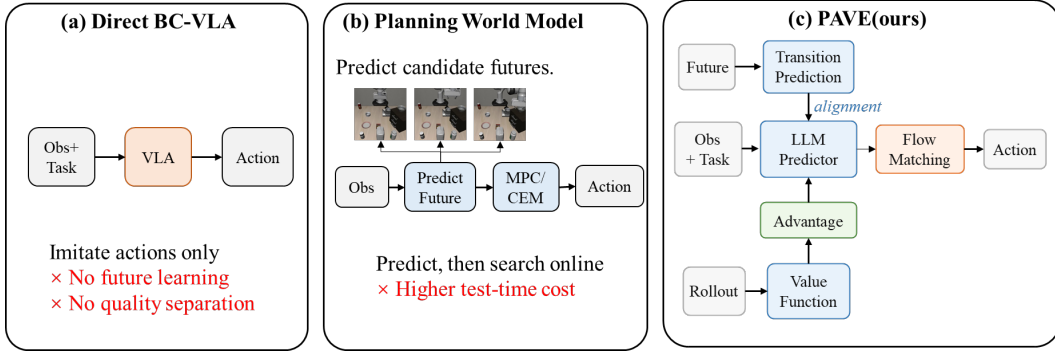


Figure 1: Conceptual positioning of PAVE. Predictive correctness and behavioral desirability are complementary, while both auxiliary modules remain training-time or offline components.

A separate line of work improves VLA policies from deployment experience. In particular, $\pi_{0.6}^*$ introduces RECAP, where a value model converts demonstrations, rollouts, and corrections into advantage conditions for policy extraction (Physical Intelligence et al., 2025a). This addresses the fact that autonomous deployment produces successful, failed, inefficient, and recovery behaviors. Nevertheless, value-guided relabeling alone does not explicitly improve the policy’s multi-timescale transition representation. Conversely, future prediction alone is behavior-agnostic: a predictor can accurately represent the consequences of a poor action without teaching the actor to avoid it.

This distinction motivates a simple principle: *learn transitions from every valid trajectory, but execute actions from the better ones*. We therefore propose Predictive Alignment and Value-Guided Evolution (PAVE). PAVE retains JEPA-WAM’s local fixed-offset transition target and adds trajectory-relative future anchors at 25%, 50%, 75%, and 100% of the remaining episode. The local objective provides a consistent short-range transition signal, while the relative anchors expose the representation to near-, mid-, and long-range progress despite variable episode lengths. These anchors are supervision targets rather than explicit subgoals: they never enter the action expert. Predicting what happens is still insufficient to decide whether an action is desirable, so PAVE independently trains a distributional state-value critic on cumulative rollout data and computes N -step advantages over the same horizon as the actor’s action chunk. The resulting positive, negative, and null labels are concatenated with the task text. This separation allows valid failed trajectories to contribute physical transition information while preventing low-quality actions from being imitated indiscriminately. At deployment, the actor uses only the positive condition; neither the future-target encoder, multi-horizon predictor, nor critic participates online.

Our contributions are threefold:

- We introduce **trajectory-relative multi-horizon transition alignment**, which combines a stable local JEPA target with normalized future anchors that span the remaining trajectory.
- We formulate a **predictive-preferential decomposition**: all valid trajectories supervise what physically occurred, while a distributional critic separates relatively better and worse action chunks through advantage-conditioned behavior learning.
- We develop a **self-improving direct policy** in which both the multi-horizon predictor and value critic are restricted to training or offline annotation, avoiding online future rollout, candidate-action evaluation, and MPC search.

2 RELATED WORK

Vision-language-action policies. Generalist VLA policies adapt pretrained visual-language representations to robot control. OpenVLA provides an open-source autoregressive VLA, while π_0 uses flow matching to model continuous action chunks and $\pi_{0.5}$ expands open-world generalization through heterogeneous co-training (Kim et al., 2024; Black et al., 2024; Physical Intelligence et al., 2025b). Diffusion Policy demonstrates the strength of conditional generative action models without a language foundation model (Chi et al., 2023). These methods are direct at inference, but their

standard action losses do not explicitly organize the representation around multiple future time scales or mixed-quality deployment experience.

Predictive representations and world-action models. V-JEPA 2 and V-JEPA 2.1 learn temporally predictive visual representations that are suitable for physical understanding and planning (Assran et al., 2025; Mur-Labadia et al., 2026). DINOv2 emphasizes high-quality semantic and dense visual features, whereas LingBot-Vision studies boundary-centric pretraining for metric spatial perception (Oquab et al., 2023; Fu et al., 2026). In robot policies, JEPA-VLA, VLA-JEPA, Being-H0.7, Fast-WAM, StageWAM, and JEPA-WAM transfer future information into deployable policies through different latent interfaces (Miao et al., 2026; Sun et al., 2026; Luo et al., 2026; Yuan et al., 2026; Liu et al., 2026; Lin et al., 2026). PAVE differs in two respects. First, it predicts a set of trajectory-relative current-future transition embeddings rather than one fixed offset, one inferred semantic stage, or action-conditioned candidate rollouts. Second, the future branch is paired with value-guided experience relabeling, so predictive representation learning and behavior selection are trained as distinct objectives.

Experience-driven policy improvement. RECAP improves VLAs with a distributional value function and advantage-conditioned policy extraction (Physical Intelligence et al., 2025a). Other recent approaches directly optimize preferences or redirect failed actions, including FlowPRO and RedFlow (Wu et al., 2026; Yan et al., 2026). PAVE follows the critic-driven conditional-policy route because it can annotate unpaired successful and failed rollouts at action-chunk resolution. Its contribution is not a new policy-gradient objective; rather, it connects offline value-based relabeling to a multi-horizon world-action representation while keeping the online actor direct.

3 PROPOSED METHOD

Given expert demonstrations $\mathcal{D}_{\text{demo}}$ and deployment trajectories $\mathcal{D}_k$ collected in round k , our goal is to learn a direct continuous policy $\pi_{\theta}(a_{t:t+H-1} \mid o_t, l, q_t, c_t)$. Here, o_t contains the current global and wrist observations, l is the task instruction, q_t is proprioception, and $c_t \in \{\text{positive, negative, null}\}$ is a behavior-quality text condition. Figure 2 summarizes the three parts of PAVE. First, the actor learns actions jointly with local and multi-horizon current-future transition targets. Second, an independent distributional critic converts cumulative deployment trajectories into N -step advantage labels. Third, the labeled deployment data and expert demonstrations retrain a flow-matching actor from the same base initialization; the improved actor then collects the next round. Only the current-observation actor is used online.

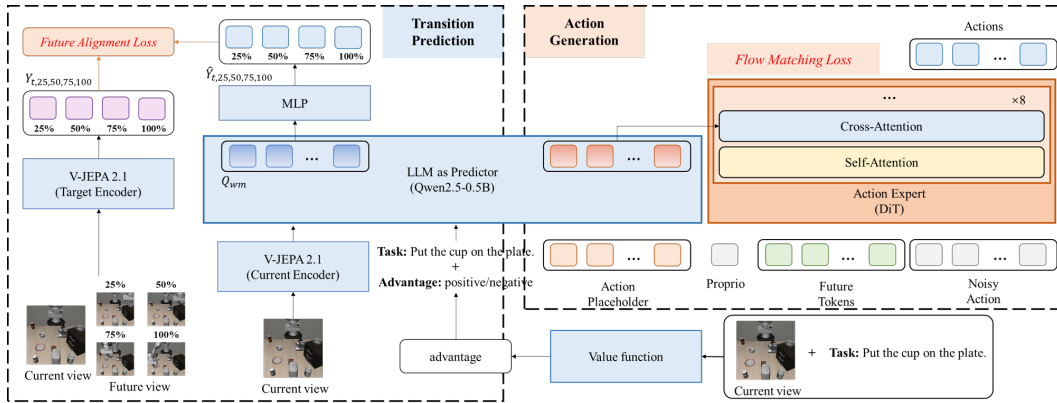


Figure 2: Overview of PAVE. Multi-horizon current-future alignment supplies training-only predictive supervision, while the offline value branch converts accumulated trajectories into advantage text that is concatenated with the task. The deployed path retains only the current-observation predictor and flow-matching action expert.

3.1 MULTI-HORIZON PREDICTIVE WORLD-ACTION LEARNING

Direct actor. The current observation contains two 384×384 camera views. A frozen V-JEPA 2.1 ViT-L/16 encoder produces 24×24 patch tokens per view, yielding $N_v = 1152$ visual tokens with dimension $d_v = 1024$. A frozen projector maps them to the $d_l = 896$ hidden space of Qwen2.5-0.5B (Mur-Labadia et al., 2026; Yang et al., 2024). The model sequence consists of the first text token, all visual tokens, the remaining task-plus-advantage text, and 64 learned action placeholders. Let

$$(V_t, M_t) = Q_\theta([w_0; P(E(o_t)); w_{1:L}; c_t; q_{1:64}^a]), \quad (1)$$

where $V_t \in \mathbb{R}^{1152 \times 896}$ denotes hidden states at visual positions and $M_t \in \mathbb{R}^{64 \times 896}$ denotes the placeholder states supplied to the action expert. The frozen Qwen base is adapted with LoRA, while the flow action head is trainable (Hu et al., 2022).

The actor predicts a normalized action chunk $a \in \mathbb{R}^{H \times 7}$ with $H = 20$. Given Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$ and flow time τ , we form

$$x_\tau = (1 - \tau)\epsilon + \tau a, \quad u^*(x_\tau, \tau) = a - \epsilon. \quad (2)$$

The conditional action expert predicts the velocity field $u_\theta(x_\tau, \tau, M_t, q_t)$, with the flow-matching objective (Lipman et al., 2023)

$$\mathcal{L}_{\text{FM}} = \frac{\sum m^a \odot \|u_\theta(x_\tau, \tau, M_t, q_t) - (a - \epsilon)\|_2^2}{\sum m^a}, \quad (3)$$

where m^a excludes padded or provenance-inconsistent action chunks.

Paired transition targets. Write $o_t = (I_t^g, I_t^w)$ for the global and wrist images. For each view, E_{pair} stacks the current and future images along the temporal axis and encodes the resulting two-frame clip with the frozen V-JEPA 2.1 target encoder. A two-frame clip forms one temporal tubelet and therefore retains the 24×24 spatial grid for each view. The global-view and wrist-view grids are concatenated in a fixed order without spatial or global pooling:

$$E_{\text{pair}}(o_t, o_h) = \text{Concat}_{v \in \{g, w\}} E_{\text{tgt}}(\text{Stack}_{\text{time}}(I_t^v, I_h^v)) \in \mathbb{R}^{1152 \times 1024}. \quad (4)$$

Accordingly, P_δ and P_{MH} map the 1152×896 Qwen visual states to 1152×1024 target tokens. All cosine losses below are evaluated token-wise and averaged over the 1,152 tokens and valid pairs.

Local transition alignment. We retain the original fixed-offset JEPA target to preserve stable local supervision. For episode endpoint T and offset $\delta = 31$,

$$h_t^\delta = \min(t + \delta, T), \quad Y_t^\delta = \text{sg}[E_{\text{pair}}(o_t, o_{h_t^\delta})]. \quad (5)$$

A token-wise predictor P_δ maps V_t to $\hat{Y}_t^\delta$, and the valid-pair cosine loss is

$$\mathcal{L}_{\text{local}} = \frac{\sum_t m_t^\delta [1 - \cos(\hat{Y}_t^\delta, Y_t^\delta)]}{\sum_t m_t^\delta}. \quad (6)$$

Trajectory-relative multi-horizon alignment. A fixed offset covers only one temporal scale. We instead define normalized future fractions

$$\rho = \{0.25, 0.50, 0.75, 1.00\}, \quad (7)$$

and select the k -th target index by

$$h_{t,k} = \text{clip}(\text{round}[t + \rho_k(T - t)], t, T). \quad (8)$$

For each k , we reuse Equation 4 to construct the joint transition target

$$Y_{t,k} = \text{sg}[E_{\text{pair}}(o_t, o_{h_{t,k}})]. \quad (9)$$

A shared MLP and learned horizon embedding $e_k \in \mathbb{R}^{896}$, broadcast across all visual-token positions, predict

$$\hat{Y}_{t,k} = P_{\text{MH}}(V_t + e_k), \quad (10)$$

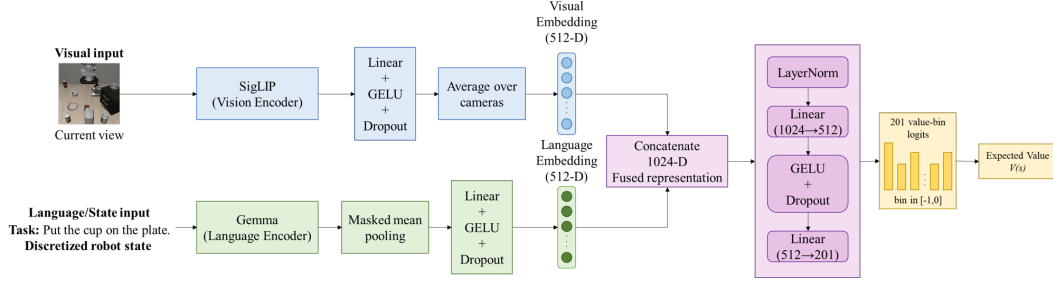


Figure 3: Architecture of the distributional value function. Multimodal state features are fused and mapped to 201 value-bin logits, whose expectation yields $V(s)$.

The corresponding masked cosine objective is

$$\mathcal{L}_{\text{MH}} = \frac{\sum_{t,k} m_{t,k} \left[1 - \cos(\hat{Y}_{t,k}, Y_{t,k}) \right]}{\sum_{t,k} m_{t,k}}. \quad (11)$$

The mask $m_{t,k}$ requires the target to remain within one episode and not cross a policy-version, action-source, intervention, or censored-outcome boundary. Importantly, $\hat{Y}_{t,k}$ is never passed to the action expert. The multi-horizon branch shapes the shared Qwen LoRA representation during training and is skipped entirely at inference.

The actor objective is

$$\boxed{\mathcal{L}_{\text{actor}} = \mathcal{L}_{\text{FM}} + \lambda_{\text{local}} \mathcal{L}_{\text{local}} + \lambda_{\text{MH}} \mathcal{L}_{\text{MH}}}, \quad (12)$$

with $\lambda_{\text{local}} = \lambda_{\text{MH}} = 0.5$ in the default configuration.

3.2 DISTRIBUTIONAL VALUE-GUIDED EXPERIENCE ANNOTATION

The predictive branch answers what transition occurred, but not whether the underlying action chunk was preferable. We therefore train a separate state-value critic $V_\phi(s_t)$ with $s_t = (o_t, l, q_t)$. The critic reuses frozen V-JEPA, projector, and Qwen modules, then appends eight proprioceptive tokens and a learned value-query token. A lightweight trainable head outputs 201 logits over equally spaced supports $z_i \in [-1, 0]$. It receives no candidate action and therefore estimates $V(s_t)$ rather than $Q(s_t, a_t)$. As shown in Figure 3, the value function maps multimodal state features to a 201-bin distribution and takes its expectation as $V(s)$.

For an episode of length L , nonterminal rewards are $r_t = -1$. The terminal reward is 0 for success and $-C_{\text{fail}}$ for a valid failure. With fixed task horizon M_{task} and $C_{\text{fail}} = M_{\text{task}}$, the undiscounted return and normalized target are

$$G_t = \sum_{i=t}^{L-1} r_i, \quad y_t = \frac{G_t}{2M_{\text{task}}} \in [-1, 0]. \quad (13)$$

The scalar y_t is linearly projected to a two-hot categorical target p_t^* on the 201 supports. The distributional objective (Bellemare et al., 2017) is

$$\mathcal{L}_{\text{value}} = - \sum_{i=0}^{200} p_{t,i}^* \log p_\phi(z_i | s_t), \quad (14)$$

and the scalar prediction is the distribution expectation

$$V_\phi(s_t) = \sum_{i=0}^{200} p_\phi(z_i | s_t) z_i. \quad (15)$$

We align policy labels with the actor horizon by setting $N = H = 20$. Let $\bar{r}_t = r_t / (2M_{\text{task}})$ and $h_t = \min(N, L - t)$. The N -step target and advantage are

$$\widehat{G}_t^{(N)} = \sum_{j=0}^{h_t-1} \bar{r}_{t+j} + \mathbf{1}[t+N < L]V_\phi(s_{t+N}), \quad (16)$$

$$A_t^{(N)} = \widehat{G}_t^{(N)} - V_\phi(s_t). \quad (17)$$

Only complete, provenance-consistent 20-step chunks are eligible. Within each task, chunks are ranked by $A_t^{(N)}$; the top 30% receive the positive label and the remainder receive the negative label. Valid human corrections, when available, are included in the positive set. The label is relative to the current cumulative data, so positive does not simply mean episode success: a recovery segment in a failed episode may be positive, while an inefficient segment in a successful episode may be negative.

3.3 ADVANTAGE-CONDITIONED POLICY EVOLUTION (ACP)

The advantage is injected as text rather than as a scalar feature. The prompt contains the task instruction followed by either “Advantage: positive”, “Advantage: negative”, or no advantage phrase for the null condition. Each labeled condition is dropped to null with probability 0.3 during training. Expert demonstrations are assigned positive labels; deployment data receive critic-derived labels.

At round k , the cumulative experience pool is

$$\mathcal{D}_{\leq k} = \bigcup_{j=1}^k \mathcal{D}_j. \quad (18)$$

We train a fresh critic on all states in $\mathcal{D}_{\leq k}$ and use that critic to score the same cumulative pool. Thus, both value terms in Equation 17 are in-sample critic predictions rather than cross-fitted estimates. We then annotate all eligible chunks and materialize a labeled evolution dataset $\widetilde{\mathcal{D}}_{\leq k}$. The actor is initialized from the same fixed base checkpoint π_{base} at every round and optimized on

$$\mathcal{D}_{\text{train}}^k = \mathcal{D}_{\text{demo}} \cup \widetilde{\mathcal{D}}_{\leq k}, \quad (19)$$

using a 3:1 demonstration-to-evolution sampling ratio and the actor loss in Equation 12. Reinitializing from one base model reduces confounding from sequential weight drift: successive actors differ primarily because their cumulative data and advantage labels differ. We use three rounds, collect 96 episodes per round, train each critic for 8,000 updates, and train each actor for 30,000 updates.

During deployment, the policy fixes $c_t = \text{positive}$ and directly samples a 20-step action chunk with four Euler integration steps. The paired-future encoder, local and multi-horizon predictors, critic, and advantage computation are not executed online. Thus, PAVE changes training supervision and data selection without introducing test-time candidate actions or search.

4 EXPERIMENTS

We organize the evaluation around four questions: (1) Does PAVE improve standard manipulation success? (2) Does multi-horizon predictive alignment improve robustness under visual and spatial shifts? (3) Does value-guided evolution improve a policy across deployment rounds? (4) Are these gains obtained without adding online planning cost? Unless otherwise stated, success rates with error bars are reported as mean $\pm$ standard deviation over three independent seeds. Compared methods use the same fixed task and initial-state manifests.

4.1 EXPERIMENTAL SETUP

Benchmarks. LIBERO contains four suites that emphasize spatial, object, goal, and long-horizon transfer (Liu et al., 2023). LIBERO-Plus applies seven controlled perturbations: camera viewpoint, robot initial state, language instruction, lighting, background texture, sensor noise, and object layout (Fei et al., 2025). RoboTwin 2.0 evaluates 50 bimanual tasks with structured randomization over clutter, lighting, background, tabletop height, and language (Chen et al., 2025). We additionally provide a qualitative real-robot table-wiping demonstration illustrating approach, contact, coverage, and completion. This sequence is not treated as a quantitative benchmark.

Table 1: Default training and evaluation configuration used throughout the experiments.

Component	Configuration
Visual inputs	Global + wrist RGB, 384×384 each
Visual encoder	Frozen V-JEPA 2.1 ViT-L/16; 576 tokens per view
Language predictor	Qwen2.5-0.5B; hidden size 896; 64 action placeholders
Adaptation	LoRA rank 32, $\alpha = 64$, dropout 0.1
Action expert	16-layer flow-matching DiT; action chunk 20×7
Flow sampling	4 Euler steps; execution horizon 20
Local JEPA target	Fixed frame offset $\delta = 31$
Multi-horizon targets	Remaining-trajectory fractions $\{0.25, 0.50, 0.75, 1.00\}$
Loss weights	$\lambda_{\text{local}} = 0.5$, $\lambda_{\text{MH}} = 0.5$
Critic	Token-fusion distributional $V(s)$; 201 bins on $[-1, 0]$
Advantage labels	$N = 20$; task-wise top 30% positive; label dropout 0.3
Evolution schedule	3 rounds; 96 new episodes/round; demos:evolution = 3:1
Optimization	Actor 30,000 updates/round; critic 8,000 updates/round
Default batches	Actor 128 global; token-fusion critic 240 global
Reporting	Three independent seeds; mean $\pm$ standard deviation

Table 2: LIBERO success rate (%). Values are mean $\pm$ standard deviation over three seeds.

Method	Spatial	Object	Goal	Long	Avg.
Diffusion Policy (Chi et al., 2023)	81.5 ± 1.6	84.0 ± 1.4	80.2 ± 1.8	71.4 ± 2.1	79.3 ± 1.2
$\pi_{0.5}$ (Physical Intelligence et al., 2025b)	93.8 ± 0.9	94.6 ± 0.8	93.0 ± 1.0	89.8 ± 1.3	92.8 ± 0.7
$\pi_{0.6}^*$ (Physical Intelligence et al., 2025a)	95.8 ± 0.7	95.4 ± 0.8	94.9 ± 0.8	92.4 ± 1.1	94.6 ± 0.6
JEPA-VLA (Miao et al., 2026)	92.6 ± 1.1	93.8 ± 1.0	92.1 ± 1.2	87.2 ± 1.5	91.4 ± 0.8
VLA-JEPA (Sun et al., 2026)	93.4 ± 1.0	94.1 ± 0.9	93.2 ± 1.0	89.5 ± 1.3	92.5 ± 0.7
Fast-WAM (Yuan et al., 2026)	94.9 ± 0.9	95.0 ± 0.8	94.0 ± 0.9	91.1 ± 1.2	93.8 ± 0.6
JEPA-WAM (Lin et al., 2026)	95.5 ± 0.8	94.8 ± 0.9	94.1 ± 0.9	91.6 ± 1.1	94.0 ± 0.6
PAVE (ours)	97.8 ± 0.5	97.0 ± 0.6	96.6 ± 0.6	95.0 ± 0.8	96.6 ± 0.4

Table 3: LIBERO-Plus success rate (%) across seven perturbation dimensions. Values are mean $\pm$ standard deviation over three seeds.

Method	Camera	Robot	Language	Light	Background	Noise	Layout	Avg.
Diffusion Policy	43.5 ± 2.2	46.2 ± 2.1	50.1 ± 2.0	47.0 ± 2.3	39.8 ± 2.6	35.7 ± 2.8	37.2 ± 2.5	42.8 ± 1.8
$\pi_{0.5}$	60.2 ± 1.7	61.8 ± 1.6	66.4 ± 1.5	63.1 ± 1.7	58.3 ± 1.9	54.7 ± 2.1	57.0 ± 1.8	60.2 ± 1.3
$\pi_{0.6}^*$	67.1 ± 1.4	70.0 ± 1.3	73.8 ± 1.2	70.4 ± 1.4	68.0 ± 1.6	64.2 ± 1.7	66.1 ± 1.5	68.5 ± 1.0
JEPA-VLA	58.9 ± 1.8	60.5 ± 1.7	65.2 ± 1.6	62.7 ± 1.8	57.6 ± 2.0	53.8 ± 2.2	55.1 ± 2.0	59.1 ± 1.4
VLA-JEPA	62.4 ± 1.6	64.8 ± 1.5	68.6 ± 1.4	66.1 ± 1.6	61.2 ± 1.8	57.9 ± 1.9	61.4 ± 1.7	63.2 ± 1.2
Fast-WAM	64.9 ± 1.5	67.3 ± 1.4	70.5 ± 1.3	68.4 ± 1.5	64.0 ± 1.7	60.8 ± 1.8	64.7 ± 1.6	65.8 ± 1.1
JEPA-WAM	66.0 ± 1.4	68.2 ± 1.4	71.2 ± 1.3	69.5 ± 1.4	65.8 ± 1.6	62.1 ± 1.7	67.6 ± 1.5	67.2 ± 1.0
PAVE (ours)	71.8 ± 1.1	74.1 ± 1.0	78.0 ± 0.9	75.3 ± 1.1	73.7 ± 1.2	69.5 ± 1.3	69.8 ± 1.2	73.2 ± 0.8

Baselines and protocol. We compare with Diffusion Policy, $\pi_{0.5}$, $\pi_{0.6}^*$, JEPA-VLA, VLA-JEPA, Fast-WAM, and JEPA-WAM (Chi et al., 2023; Physical Intelligence et al., 2025b;a; Miao et al., 2026; Sun et al., 2026; Yuan et al., 2026; Lin et al., 2026). All baseline results in Tables 2–4 were reproduced by us using the same demonstrations, observation views, action normalization, task manifests, and evaluation budget; the citations identify the source methods rather than quoted numerical results.

4.2 MAIN BENCHMARK RESULTS

Compared with JEPA-WAM, PAVE improves average success by 2.6 points on LIBERO and long-horizon success by 3.4 points. Its average is also 2.0 points above $\pi_{0.6}^*$.

On LIBERO-Plus, PAVE improves average success by 6.0 points over JEPA-WAM and by 4.7 points over $\pi_{0.6}^*$. Relative to JEPA-WAM, the gains are 7.9 points under background changes and 7.4 points under sensor noise.

Table 4: RoboTwin 2.0 bimanual task success rate (%). Values are mean $\pm$ standard deviation over three seeds.

Method	Clean	Clutter	Light	Background	Height	Language	Avg.
Diffusion Policy	42.1 $\pm$ 2.8	29.8 $\pm$ 3.1	34.2 $\pm$ 2.9	30.5 $\pm$ 3.2	27.6 $\pm$ 3.0	26.5 $\pm$ 3.3	31.8 $\pm$ 2.1
$\pi_{0.5}$	57.8 $\pm$ 2.3	44.3 $\pm$ 2.5	49.0 $\pm$ 2.3	45.1 $\pm$ 2.6	42.8 $\pm$ 2.5	41.1 $\pm$ 2.7	46.7 $\pm$ 1.8
$\pi_{0.6}^*$	65.2 $\pm$ 1.9	53.1 $\pm$ 2.1	57.3 $\pm$ 2.0	54.0 $\pm$ 2.2	51.5 $\pm$ 2.1	50.0 $\pm$ 2.3	55.2 $\pm$ 1.5
JEPA-VLA	55.1 $\pm$ 2.4	42.3 $\pm$ 2.6	47.0 $\pm$ 2.4	43.1 $\pm$ 2.7	40.2 $\pm$ 2.6	39.1 $\pm$ 2.8	44.5 $\pm$ 1.9
VLA-JEPA	59.4 $\pm$ 2.2	46.1 $\pm$ 2.4	50.5 $\pm$ 2.2	47.0 $\pm$ 2.5	44.6 $\pm$ 2.4	42.2 $\pm$ 2.6	48.3 $\pm$ 1.7
Fast-WAM	62.0 $\pm$ 2.1	49.0 $\pm$ 2.3	53.5 $\pm$ 2.1	50.1 $\pm$ 2.4	47.0 $\pm$ 2.3	45.6 $\pm$ 2.5	51.2 $\pm$ 1.6
JEPA-WAM	64.0 $\pm$ 2.0	51.4 $\pm$ 2.2	55.4 $\pm$ 2.0	52.0 $\pm$ 2.3	49.8 $\pm$ 2.2	47.8 $\pm$ 2.4	53.4 $\pm$ 1.5
PAVE (ours)	69.3 $\pm$ 1.6	58.4 $\pm$ 1.8	62.0 $\pm$ 1.7	59.5 $\pm$ 1.9	56.3 $\pm$ 1.8	54.5 $\pm$ 2.0	60.0 $\pm$ 1.2

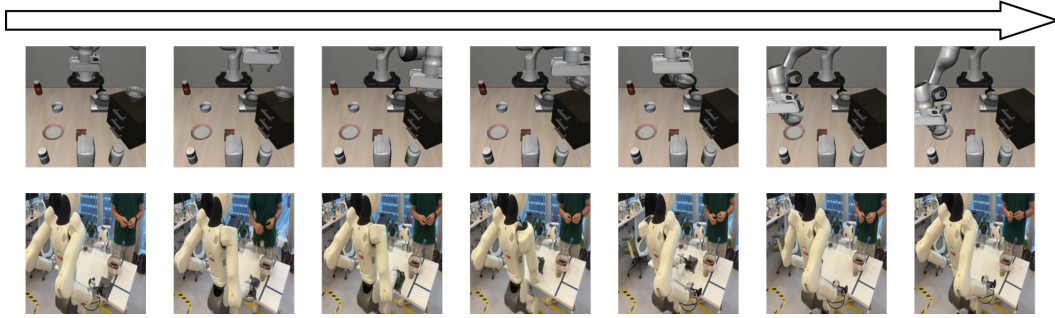


Figure 4: Representative qualitative trajectories in simulation (top) and on a real robot (bottom). Seven frames per row illustrate approach, interaction, and completion; the real-robot sequence is a qualitative demonstration rather than a quantitative evaluation.

Table 5: Online efficiency and model size. Latency is measured at batch size one with two camera views, a 20-step output chunk, four flow steps, CUDA synchronization, and identical hardware.

Method	Online params (B)	Offline-only extras (M)	p50 latency (ms)	p95 latency (ms)	GPU memory (GB)
$\pi_{0.5}$	3.33	0.00	78.6	86.9	15.2
JEPA-WAM	1.29	0.00	62.0	66.6	10.4
PAVE (ours)	1.29	4.63	62.2	66.8	10.4

On RoboTwin 2.0, PAVE improves average success by 6.6 points over JEPA-WAM and by 4.8 points over $\pi_{0.6}^*$. The margins over JEPA-WAM reach 7.0 points under clutter and 7.5 points under background changes.

4.3 QUALITATIVE TRAJECTORY VISUALIZATION

Figure 4 presents example simulation and real-robot trajectories without online future rollout or search. The real-robot row documents a qualitative demonstration; we do not infer a trial-level success rate or robustness from this sequence.

4.4 ONLINE EFFICIENCY

PAVE matches JEPA-WAM’s 1.29B online parameters and 10.4GB memory footprint. Its p50 and p95 latencies are each 0.2ms higher; the 4.63M additional parameters are used only by the training-time predictor and offline critic.

4.5 CORE ABLATIONS

With ACP disabled, the local fixed-offset target improves success over flow matching alone by 8.1 points on LIBERO-Plus and 7.5 points on RoboTwin 2.0. Adding the shared four-horizon objective yields a further 3.5- and 3.4-point improvement, respectively.

Table 6: Effect of transition supervision with ACP evolution disabled for all rows. Success rates are mean $\pm$ standard deviation over three seeds.

Transition supervision	LIBERO-Plus	RoboTwin 2.0
No future transition target	59.1 $\pm$ 1.4	45.9 $\pm$ 2.0
One local fixed-offset target	67.2 $\pm$ 1.0	53.4 $\pm$ 1.5
Local + shared four-horizon targets	70.7 $\pm$ 0.9	56.8 $\pm$ 1.4

Across the three evolution rounds in Table 7, success rises from 67.2 to 73.2 on LIBERO-Plus and from 53.4 to 60.0 on RoboTwin 2.0.

In Table 8, the positive test condition reaches 66.6 average success, compared with 64.2 for the null condition, 47.6 for the negative condition, and 61.7 for unlabeled multi-round behavior cloning.

The round-wise, advantage-condition, encoder, horizon-count, loss, visualization, and value-critic ablations are placed after the bibliography to preserve the nine-page main-text budget.

5 LIMITATIONS AND DISCUSSION

PAVE has several limitations. First, trajectory-relative anchors are temporal progress markers, not semantic subgoals. They may coincide near episode termination and may encode unsuccessful endpoints; their role is representation supervision rather than guaranteed plans. Second, the state-value critic produces chunk-level relative labels and cannot identify the exact failing action inside a chunk. Action-level correction methods such as RedFlow may provide denser supervision (Yan et al., 2026). Third, the top-30% labels use in-sample critic predictions and may therefore be sensitive to critic overfitting, calibration, and data coverage; trajectory-level cross-fitting remains an important extension. Fourth, because every round retrains the actor from one base checkpoint, the method incurs substantial offline computation even though online inference remains efficient. Finally, simulation success and a qualitative real-robot sequence do not establish safety or robustness in unconstrained environments; broader deployment requires conservative termination, intervention logging, and failure auditing.

6 CONCLUSION

We presented PAVE, a self-improving world-action policy that separates two learning questions: what transition occurs after behavior, and which behavior should be preferred. Multi-horizon JEPA alignment extracts local-to-trajectory-scale transition structure from every valid trajectory, while a distributional critic converts cumulative deployment experience into action-chunk-level advantage conditions. Both auxiliary mechanisms are restricted to training or offline annotation, leaving one direct positive-conditioned flow actor online. Across LIBERO, LIBERO-Plus, and RoboTwin 2.0, PAVE improves average success over JEPA-WAM by 2.6, 6.0, and 6.6 points while matching its online model size and memory footprint.

Reproducibility statement. The configuration values needed to reproduce the reported experiments are given in Table 1 and Appendix A. The appendix specifies data isolation, valid action chunks, the baseline protocol, and statistical summaries. Reported success rates use fixed task and initial-state manifests and are presented as mean $\pm$ standard deviation over three independent seeds.

Ethics statement. This work concerns robot manipulation and may introduce physical safety risks if deployed without safeguards. The real-robot sequence in Figure 4 is a qualitative demonstration rather than a safety or robustness evaluation. Broader deployment requires bounded workspaces, speed and force limits, emergency stops, operator supervision, and explicit handling of censored safety terminations. No human-subject or personal-data collection was conducted.

REFERENCES

- Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025. doi: 10.48550/arXiv.2506.09985.
- Marc G. Bellemare, Will Dabney, and Remi Munos. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning*, pages 449–458, 2017.
- 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. doi: 10.48550/arXiv.2410.24164.
- Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025. doi: 10.48550/arXiv.2506.18088.
- Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems*, 2023.
- Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. Libero-plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025. doi: 10.48550/arXiv.2510.13626.
- Zelin Fu, Bin Tan, Changjiang Sun, Shaohui Liu, Kecheng Zheng, Yinghao Xu, Xing Zhu, Yujun Shen, and Nan Xue. Vision pretraining for dense spatial perception. *arXiv preprint arXiv:2607.05247*, 2026. doi: 10.48550/arXiv.2607.05247.
- 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. In *International Conference on Learning Representations*, 2022.
- 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. In *Conference on Robot Learning*, 2024. arXiv:2406.09246.
- Yihan Lin, Jiawei He, Shifeng Bao, Chen Zhao, Yang Li, Xiaobo Wang, Yan Wang, Cheng Chi, and Jing Zhang. Jepa-wam: Learning vision-language-action policies with joint-embedding world modeling. *arXiv preprint arXiv:2608.09381*, 2026. doi: 10.48550/arXiv.2608.09381.
- Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. arXiv:2210.02747.
- Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *arXiv preprint arXiv:2306.03310*, 2023. doi: 10.48550/arXiv.2306.03310.
- Xiao Liu, Yuguang Yang, Xi Wang, Kai Jiang, Cheng Chi, Yong Xu, Wenchao Ding, Yilun Chen, and Yan Wang. Stagewam: Joint-embedding stage prediction for world-action models in robot manipulation. *arXiv preprint arXiv:2608.10780*, 2026. doi: 10.48550/arXiv.2608.10780.
- Hao Luo, Wanpeng Zhang, Yicheng Feng, Sipeng Zheng, Hao Xu, Chaoyi Xu, Zhaoyang Xi, Yao Fu, and Zongqing Lu. Being-h0.7: A latent world-action model from egocentric videos. *arXiv preprint arXiv:2605.00078*, 2026. doi: 10.48550/arXiv.2605.00078.
- Shangchen Miao, Ningya Feng, Jialong Wu, Ye Lin, Xu He, Dong Li, and Mingsheng Long. Jepa-vla: Video predictive embedding is needed for vla models. *arXiv preprint arXiv:2602.11832*, 2026. doi: 10.48550/arXiv.2602.11832.

-
- Lorenzo Mur-Labadia, Matthew Muckley, Amir Bar, Mido Assran, Koustuv Sinha, Mike Rabbat, Yann LeCun, Nicolas Ballas, and Adrien Bardes. V-jepa 2.1: Unlocking dense features in video self-supervised learning. *arXiv preprint arXiv:2603.14482*, 2026. doi: 10.48550/arXiv.2603.14482.
- Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. doi: 10.48550/arXiv.2304.07193.
- Physical Intelligence, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Kevin Black, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, et al. $\pi_{0,6}^*$: a vla that learns from experience. *arXiv preprint arXiv:2511.14759*, 2025a. doi: 10.48550/arXiv.2511.14759.
- Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, et al. $\pi_{0,5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025b. doi: 10.48550/arXiv.2504.16054.
- Jingwen Sun, Wenyao Zhang, Zekun Qi, Shaojie Ren, Zezhi Liu, Hanxin Zhu, Guangzhong Sun, Xin Jin, and Zhibo Chen. Vla-jepa: Enhancing vision-language-action model with latent world model. *arXiv preprint arXiv:2602.10098*, 2026. doi: 10.48550/arXiv.2602.10098.
- Yihao Wu, He Zhang, Junbo Tan, Xueqian Wang, and Zhengyou Zhang. Flowpro: Reward-free reinforced fine-tuning of flow-matching vlas via proximalized preference optimization. *arXiv preprint arXiv:2606.05468*, 2026. doi: 10.48550/arXiv.2606.05468.
- Zhengyang Yan, Junhao Li, Fangqi Zhu, Zijun Wang, Quanxin Shou, Yikun Miao, Xiaoyi Pang, Zicong Hong, and Song Guo. Redflow: Redirect failure into action-level corrections for flow-matching vla policy. *arXiv preprint arXiv:2607.27782*, 2026. doi: 10.48550/arXiv.2607.27782.
- An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. doi: 10.48550/arXiv.2412.15115.
- Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-wam: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026. doi: 10.48550/arXiv.2603.16666.

A DETAILED EXPERIMENTAL PROTOCOL

Data isolation. We use non-overlapping episode and task-variant manifests for training demonstrations, evolution rollouts, development diagnostics, and official held-out evaluations. Raw rollout files are kept immutable. Returns, critic predictions, advantages, and labels are stored as derived sidecars with configuration hashes and actor lineage.

Valid action chunks. An anchor contributes to action training or advantage ranking only when all 20 actions are inside the episode, finite on valid dimensions, and generated under a consistent action source and policy version. Endpoint replication is excluded as a real action, and future JEPA pairs remain within the same provenance segment.

Statistics. For each benchmark, we report the mean and standard deviation over three independent seeds. Policies compared within a benchmark use the same fixed task and initial-state manifests.

Fair baseline comparison. All comparative baseline entries were reproduced under our evaluation protocol. When exact original pretrained weights or training data were unavailable, we retained the controlled protocol above and treat the resulting values as reproductions rather than results quoted from the original papers.

B ADDITIONAL ABLATIONS

B.1 POLICY EVOLUTION AND ADVANTAGE CONDITIONING

Table 7: Policy improvement across evolution rounds. Performance values are mean $\pm$ standard deviation over three seeds.

Policy	New episodes	Cumulative episodes	LIBERO-Plus	RoboTwin 2.0
Base actor	0	0	68.1 $\pm$ 1.0	54.5 $\pm$ 1.5
Round 1	96	96	70.5 $\pm$ 0.9	56.4 $\pm$ 1.4
Round 2	96	192	72.1 $\pm$ 0.8	58.6 $\pm$ 1.3
Round 3	96	288	73.2 $\pm$ 0.8	60.0 $\pm$ 1.2

Table 8: Effect of advantage conditioning. Avg. success is the macro-average over LIBERO-Plus and RoboTwin 2.0. Values are mean $\pm$ standard deviation over three seeds.

Evolution setting	Train condition	Test condition	Avg. success
No evolution	Expert only	Null	60.3 $\pm$ 1.0
Unlabeled multi-round BC	None	Null	61.7 $\pm$ 1.1
ACP training	Pos./neg./null	Null	64.2 $\pm$ 0.9
ACP training	Pos./neg./null	Negative	47.6 $\pm$ 1.4
PAVE	Pos./neg./null	Positive	66.6 $\pm$ 0.8

B.2 VISUAL ENCODER CHOICE

Table 9: Effect of the frozen visual encoder. Values are mean $\pm$ standard deviation over three seeds; throughput is training steps/s on the same hardware.

Encoder	LIBERO	LIBERO-Plus	RoboTwin 2.0	Train throughput
DINOv2 (Oquab et al., 2023)	95.4 $\pm$ 0.5	69.4 $\pm$ 1.0	56.2 $\pm$ 1.4	1.31
LingBot-Vision (Fu et al., 2026)	95.8 $\pm$ 0.5	70.6 $\pm$ 0.9	57.4 $\pm$ 1.3	1.14
V-JEPA 2.1 (default)	96.6 $\pm$ 0.4	73.2 $\pm$ 0.8	60.0 $\pm$ 1.2	1.00

V-JEPA 2.1 exceeds DINOv2 by 3.8 points on both LIBERO-Plus and RoboTwin 2.0, and exceeds LingBot-Vision by 2.6 points on both benchmarks. DINOv2 has the highest training throughput at 1.31 steps/s, compared with 1.00 steps/s for V-JEPA 2.1.

Figure 5 complements the quantitative ablation with response maps from DINOv2, V-JEPA 2.1, and LingBot-Vision on the same input images.

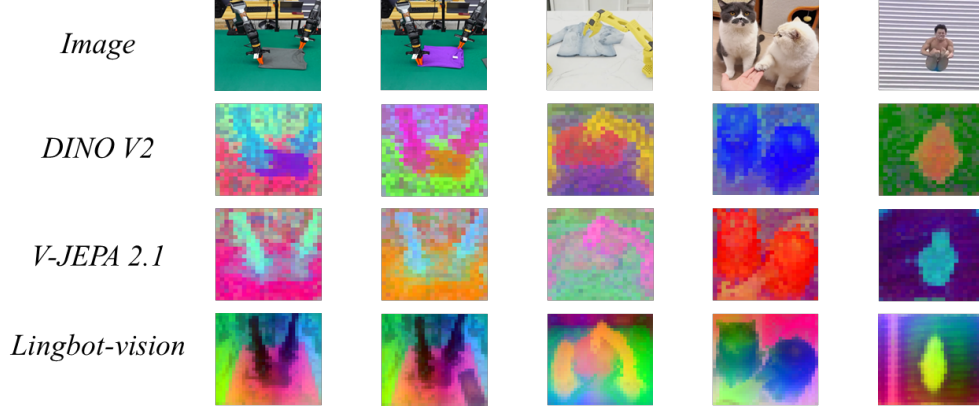


Figure 5: Qualitative comparison of frozen visual encoders on matched input images. From top to bottom, the figure shows the source images followed by response maps from DINOv2, V-JEPA 2.1, and LingBot-Vision.

B.3 NUMBER OF FUTURE ANCHORS

Table 10: Sensitivity to the number and placement of trajectory-relative future anchors. Values are mean $\pm$ standard deviation over three seeds.

K	Fractions	Predictor	LIBERO-Plus	RoboTwin 2.0
1	{1.00}	Shared	67.8 ± 1.1	54.1 ± 1.5
2	{0.50, 1.00}	Shared	70.3 ± 0.9	56.8 ± 1.4
3	{0.33, 0.67, 1.00}	Shared	72.0 ± 0.9	58.7 ± 1.3
4	{0.25, 0.50, 0.75, 1.00}	Shared	73.2 ± 0.8	60.0 ± 1.2
6	Uniform over remaining episode	Shared	72.7 ± 0.8	59.3 ± 1.3
8	Uniform over remaining episode	Shared	71.9 ± 0.9	58.5 ± 1.4

Four anchors achieve 73.2 on LIBERO-Plus and 60.0 on RoboTwin 2.0. Increasing the schedule to six anchors reduces these values to 72.7 and 59.3, while eight anchors reach 71.9 and 58.5.

B.4 LOSS COMPONENTS

Table 11: Contribution of actor losses and advantage-conditioned evolution. Success rates are mean $\pm$ standard deviation over three seeds. ACP is a data-conditioning mechanism rather than a differentiable loss term.

$\mathcal{L}_{\text{FM}}$	$\mathcal{L}_{\text{local}}$	$\mathcal{L}_{\text{MH}}$	ACP evolution	LIBERO-Plus	RoboTwin 2.0
✓				59.1 ± 1.4	45.9 ± 2.0
✓	✓			67.2 ± 1.0	53.4 ± 1.5
✓		✓		66.1 ± 1.1	52.3 ± 1.6
✓	✓	✓		70.7 ± 0.9	56.8 ± 1.4
✓	✓		✓	69.0 ± 1.0	55.2 ± 1.5
✓	✓	✓	✓	73.2 ± 0.8	60.0 ± 1.2

The full configuration reaches 73.2 on LIBERO-Plus and 60.0 on RoboTwin 2.0. Relative to local plus multi-horizon supervision without ACP, the gains are 2.5 and 3.2 points; relative to local supervision with ACP, they are 4.2 and 4.8 points.

B.5 VALUE CRITIC ARCHITECTURE

Table 12: Token-fusion critic versus a $\pi_{0.6}^*$ -style reproduced critic. Final policy success is measured on LIBERO-Plus and reported as mean $\pm$ standard deviation over three seeds.

Critic	Visual/language backbone	Fusion	Trainable params (M) $\downarrow$	Final policy success
$\pi_{0.6}^*$ -style critic	SigLIP + Gemma	pooled concat	682.4	70.4 $\pm$ 1.0
Scalar-regression token critic	Frozen V-JEPA + Qwen	token fusion	0.59	71.1 $\pm$ 0.9
Distributional token-fusion critic (ours)	Frozen V-JEPA + Qwen	patch/proprio/value query	0.69	73.2 $\pm$ 0.8

The distributional token-fusion critic uses 0.69M trainable parameters and reaches 73.2 final LIBERO-Plus success, compared with 71.1 for the scalar-regression token critic (0.59M) and 70.4 for the reproduced $\pi_{0.6}^*$ -style critic (682.4M).