DREAMIF: INTENTION-GUIDED SPARSE DYNAMICS PREDICTION FOR LONG-HORIZON VISION-LANGUAGE-ACTION MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Leveraging robust semantic representations of pre-trained VLMs, Vision-Language-Action (VLA) models build end-to-end multimodal perception-to-action mappings to boost semantic alignment and task generalization for embodied manipulation. Nevertheless, existing VLA paradigms lack task-oriented dynamics inference, making them vulnerable to irrelevant background redundancy and cumulative errors in long-horizon manipulation. This work proposes Dream Intention Flow (DreamIF), a novel intention-guided embodied foundation model that enables VLA to focus on intention-relevant core cues and mitigate environmental redundancy and long-horizon errors. We design an intention-driven dynamics inference mechanism: optical-flow-guided motion saliency locates Intention-Associated Dynamic Visual Regions (IADVR) to filter redundant backgrounds, capture physical transition rules, and extract action intentions. The model synchronously predicts multi-frame future IADVR features and embeds these temporal priors into the native action representations, strengthening temporal modeling and shifting control from reactive to predictive. Evaluated on the comprehensive LIBERO-PLUS benchmark under matched training data, DreamIF attains 85.7% on average, and extending its method to a $\pi_{0.5}$ backbone lifts that backbone's average success rate by 15.0 points. Notably, on the long-horizon LIBERO-PLUS Long suite it achieves an 85.5% success rate, 8.9 points above the matched baseline, and on eight real-world robotic tasks involving environmental noise and long-horizon manipulation it raises the average success rate by 15.6 points. Experimental results validate that DreamIF greatly enhances the ability to model task-oriented dynamics and remedies the inherent limitations of conventional VLA models.

1 INTRODUCTION

Vision-Language-Action (VLA) models leverage pre-trained Vision-Language Models (VLMs) to map visual observations and language instructions end-to-end to the robot action space (Karamcheti et al., 2024; Steiner et al., 2024; Li et al., 2025), and their robust multimodal comprehension makes them excel at general-purpose manipulation (Lisondra et al., 2026; Liang et al., 2025). Representative works include OpenVLA-OFT, which accelerates inference via parallel decoding without sacrificing action precision (Kim et al., 2025a), π_0 and π_0-FAST, which adopt flow matching over continuous action chunks (Black et al., 2025b; Pertsch et al., 2025), and lightweight adapters that bring the paradigm to small backbones (Wang et al., 2026; Wen et al., 2025; Shukor et al., 2025). Despite promising results, these models follow a reactive control paradigm (see Fig. 1(a)): they generate instant actions solely from the current observation, lacking anticipation for the dynamics inference that follows action execution. This makes them fragile on long-horizon tasks, where small deviations accumulate and cause task failure through compounding errors (Ross et al., 2011).

To endow robots with the ability to infer their physical environment, researchers have proposed World-Model-driven VLA (WVLA) frameworks (Ha & Schmidhuber, 2018; Huang et al., 2021; 2026). WorldVLA predicts the next timestep's visual information alongside action generation to augment action representations with underlying physical laws (Cen et al., 2025b) (see Fig. 1(b)), and UniVLA instead models inter-frame dynamics in a latent space (Bu et al., 2025). Both require processing holistic image data, whose static background and task-irrelevant redundancy dilute the

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

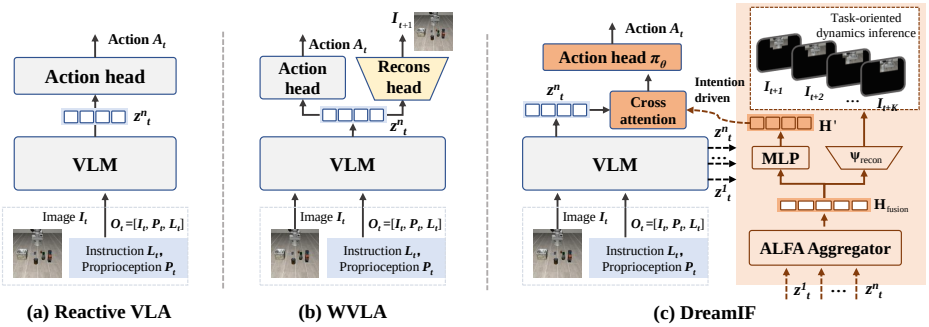

Figure 1: **(a) Reactive VLA:** Directly maps observation to instantaneous action. **(b) WVLA:** Integrates visual state prediction with action generation. **(c) DreamIF (Ours):** It aggregates heterogeneous VLM features to infer manipulation intentions, which further guides deductive control.

physical features pertinent to actions. ReconVLA addresses this with an auxiliary Gaze Region that re-allocates visual attention using diffusion models (Song et al., 2026; Peebles & Xie, 2023), but it predicts only single-frame visual states and its Gaze Region sits on the static centre of the target object, ignoring the dense-contact dynamics that underpin task progression.

To address these limitations, we propose **Dream I**ntention **F**low (DreamIF), a VLA framework integrating intention-guided inference with dynamic visual attention. In cognitive psychology, intention mediates between internal mental states and external actions and directs attention during perception and execution (Ajzen, 1991; Memelink & Hommel, 2013); for embodied manipulation we operationalise it as the temporal dynamics inference governing task progression. We further introduce the Intention-Associated Dynamic Visual Region (IADVR): an optical-flow-guided motion-saliency perception (Karaev et al., 2025) that localises the dynamic regions strictly pertinent to the current action intention (the end-effector and the objects it interacts with) and filters out redundant static background. To strengthen intention reasoning on long-horizon tasks, we design a multi-frame future prediction mechanism: the model synchronously predicts multiple future frames while generating actions, compelling the policy to capture the overarching intention of the task (Fig. 1(c)). Finally, during action decoding, a cross-attention mechanism embeds these temporal predictive features into the native action representations, anchoring the action sequence to predictable physical trajectories. One design commitment follows from defining the target by motion: the supervision is produced *once, offline, during dataset preprocessing* and consumed only as a training target, so at deployment the tracker is never loaded. The primary contributions of this paper are summarised as follows.

- **Design of the Intention-Associated Dynamic Visual Region (IADVR):** the dynamic regions highly correlated with the task intention (the end-effector and the objects it displaces or deforms) are automatically localised through optical-flow-guided motion saliency perception, which filters out static background redundancy and captures the interactions that carry the intention.
- **Construction of an Intention-Driven Manipulation Framework:** To address the accumulation of long-horizon errors, we introduce an intention-driven multi-frame reasoning mechanism for IADVR, enabling the model to synchronously learn task-oriented dynamics during action prediction. By explicitly embedding these temporal inference features into native action representations, the framework converts reactive control into intention-guided deductive control.
- **Extensive Experimental Validation:** The efficacy of our DreamIF is rigorously validated on the LIBERO-PLUS benchmark. DreamIF attains an 85.5% success rate on long-horizon tasks, 8.9 points above the baseline, and averages 80.6% over eight real-world tasks. Extending the method to a $\pi_{0.5}$ backbone lifts that backbone's average success rate by 15.0 points. This validates that the proposed method effectively improves generalization and long-horizon inference ability.

2 Related Work

2.1 Direct Perception–Action Mapping: Instantaneous Action Generation

Vision-Language-Action (VLA) models aim to establish end-to-end mappings from multimodal observations to robotic actions (Bjorck et al., 2025; Li et al., 2024; Lin et al., 2025; Wang et al., 2026). RT-1 (Brohan et al., 2023), RT-2 (Zitkovich et al., 2023), PaLM-E (Driess et al., 2023)

and OpenVLA (Kim et al., 2025b) established transformer-based instruction following and moved large-scale VLM priors into control; the systems that followed bought precision and efficiency rather than anticipation: OpenVLA-OFT added parallel decoding and action chunking (Kim et al., 2025a), π_0 used flow matching (Lipman et al., 2023) over continuous action chunks (Black et al., 2025b; Pertsch et al., 2025), and lightweight adapters brought the paradigm to small backbones (Wen et al., 2025; Shukor et al., 2025; Hung et al., 2025). Despite their impressive performance, these models conceptualise manipulation as a reactive "perception-action" mapping; neglecting the learning of dynamics transitions, they cannot rectify execution deviations autonomously and are therefore vulnerable on long-horizon tasks (Ross et al., 2011).

2.2 World Model-Driven VLA: Incorporating Future State Prediction

World-model-based robotic control captures underlying dynamics features to boost control efficacy (Huang et al., 2026; 2021). SuSIE (Black et al., 2024), GAIA-1 (Hu et al., 2023) and UniSim (Yang et al., 2023) adopt a two-stage pipeline that synthesises future frames, keypoints or goal images and then conditions the action on them. WorldVLA (Cen et al., 2025b) unifies action generation and future-frame prediction in one sequence, learning transition laws to strengthen action representations, while UniVLA (Bu et al., 2025) models temporal transitions in a task-centric latent space and VLA-JEPA (Sun et al., 2026) predicts leakage-free future latents instead of pixels; 3D-VLA (Zhen et al., 2024) extends prediction to depth, RynnVLA-002 (Cen et al., 2025a) and DreamVLA (Zhang et al., 2025) to a generative world model, and GR-1 (Wu et al., 2024) and GR-2 (Cheang et al., 2024) to large-scale video pre-training. Despite these advances, the target stays dense or globally entangled, so background redundancy enters the objective, and single-step prediction disrupts the causal dependencies between multi-step actions and long-horizon transitions (Hu et al., 2025; Baker et al., 2022; Tian et al., 2025; Ye et al., 2025). MemoryVLA (Shi et al., 2026) and HoloBrain-0 (Lin et al., 2026) instead retain more of the *past*, a complementary line: it widens the window of past observations, whereas DreamIF predicts the consequences of the action it is about to take. DreamIF differs on two axes: the prediction target is a motion-defined sparse region, and the horizon is multi-frame within a single non-autoregressive forward pass.

3 Preliminaries

3.1 Vision-Language-Action (VLA) Formulation

General robotic manipulation tasks can be formulated as a conditional action sequence generation problem. At each discrete decision time step t, the policy receives a multimodal observation tuple $O_t = \{I_t, P_t, L\}$, where I_t denotes the visual observation, P_t represents the robot's proprioceptive state, and L specifies the natural language instruction. In representative paradigms such as OpenVLA-OFT Kim et al. (2025a), a pre-trained Vision-Language Model (VLM) is first employed to extract the last-layer hidden state representation, denoted as $z_t^n = \text{VLM}(O_t)$. This feature z_t^n is subsequently mapped to a sequence of continuous control vectors in the action space, defined as $A_t = \{a_t, a_{t+1}, \ldots, a_{t+k-1}\}$, where k corresponds to the action chunking horizon.

The optimisation objective is to minimise the discrepancy between the policy's predicted actions $\pi_\theta(z_t^N)$ and the expert demonstration trajectories A_t. For continuous action spaces, this is typically achieved by optimising an L_1 regression loss over the data distribution,

$$\mathcal{L}_{\text{action}} = \mathbb{E}_{(O_t, A_t)\sim\mathcal{D}}\big[\|\pi_\theta(z_t^N) - A_t\|_1\big], \tag{1}$$

where $\mathcal{D}$ denotes the dataset, and π_θ represents the action decoding policy parameterised by θ.

3.2 World Model-Driven VLA Formulation

To enhance the dynamics inference capabilities of VLA models, WorldVLA (Cen et al., 2025b) extends the conventional singular action mapping paradigm to explicitly internalise dynamics transitions, so that simulating the consequences of executed actions provides temporal consistency constraints for policy decision-making. The framework consists of two synergistic branches, a future state prediction branch and an action generation branch. The former models the environment dynamics, predicting the future state after action execution conditioned on the current observation and action: $\hat{I}_{t+1} = f_\phi(z_t^N)$,

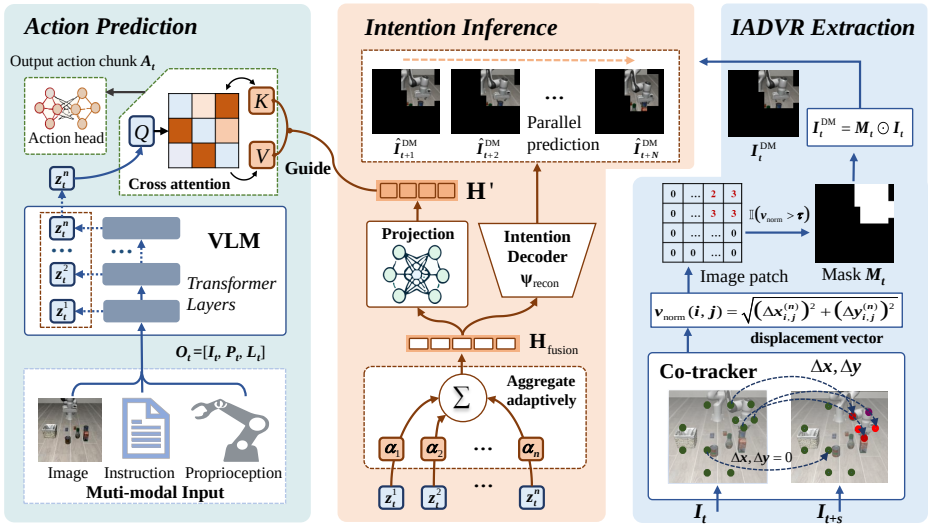

Figure 2: **Overview of DreamIF.** The **IADVR Extraction** module localizes dynamic interaction regions via optical-flow-guided motion saliency perception. The **Intention Inference** leverages multi-granularity VLM features to synchronously predict action intention. The **Action Prediction** explicitly embeds intention into the native action representations, realizing intention-guided control.

where $z_t^N = \mathrm{VLM}(O_t, A_t)$, and f_ϕ represents the state prediction model parameterised by ϕ. The optimisation objective for this branch is defined as the image reconstruction loss:

$$\mathcal{L}_{\text{recon}} = \mathbb{E}_{(O_t, A_t)\sim\mathcal{D}}\left[\|f_\phi(z_t^N) - I_{t+1}\|_2^2\right]. \tag{2}$$

Building upon this, a joint optimisation strategy is employed to train the framework, with the total loss function formulated as $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{action}} + \alpha\mathcal{L}_{\text{recon}}$, where α serves as a weighting coefficient. Despite endowing the policy with dynamics inference capabilities, current paradigms predict full-frame visual observations, introducing substantial spatial redundancy, and their myopic single-step predictions fail to capture the underlying task intention of human demonstrations (Hu et al., 2025; Tian et al., 2025), leaving a critical lack of foresight on complex, long-horizon manipulation tasks.

4 Methodology

Our methodology comprises three components. Sec. 4.1 introduces Intention-Associated Dynamic Visual Region (IADVR) extraction, which uses optical-flow-guided motion saliency to identify intention-relevant interaction regions while filtering redundant backgrounds. Sec. 4.2 presents Adaptive Layer-wise Feature Aggregation (ALFA), which integrates multi-granularity semantic and structural VLM features for dynamic region modeling and captures intention-aware temporal transitions. Sec. 4.3 embeds these predictive features into native action representations to dynamically calibrate action generation. Together, these components shift the framework (Fig. 2).

4.1 Intention-Associated Dynamic Visual Region Extraction

To compel the model to focus on visual regions relevant to the manipulation intention and suppress task-irrelevant redundant interference, we propose the IADVR, which is defined as the image areas undergoing physical interactions (e.g., displacement or deformation) associated with the current action intention. These areas require focused attention from the model, whereas static backgrounds (e.g., desks, walls) and task-irrelevant objects are actively filtered out. Specifically, we extract this region via an optical-flow-guided motion saliency perception mechanism.

Given two RGB images $I_t \in \mathbb{R}^{H\times W\times 3}$ and $I_{t+s} \in \mathbb{R}^{H\times W\times 3}$ separated by s frames, we first construct a regular grid on the image plane with a stride of S_{patch} pixels to align with the patch granularity of the visual encoder. This generates C query points, where $C = (H/S_{\text{patch}}) \times (W/S_{\text{patch}})$. The initial timestamp for each query point is set to t. Subsequently, we employ a pre-trained Co-Tracker model (Karaev et al., 2024) to extract dense trajectories for these query points, outputting

their predicted coordinates at frame $t+s$. For the c-th query point located at spatial grid position (i,j), its displacement vectors along the x-axis and the y-axis are denoted as $\Delta x_{i,j}^{(c)} = x_{t+s}^{(c)} - x_t^{(c)}$ and $\Delta y_{i,j}^{(c)} = y_{t+s}^{(c)} - y_t^{(c)}$, respectively, where $(x_t^{(c)}, y_t^{(c)})$ and $(x_{t+s}^{(c)}, y_{t+s}^{(c)})$ represent the coordinates of the point at frames t and $t+s$. Consequently, we evaluate the motion intensity of this point using the L_2 norm of the displacement vector:

$$v_{\text{norm}}(i,j) = \sqrt{\left(\Delta x_{i,j}^{(c)}\right)^2 + \left(\Delta y_{i,j}^{(c)}\right)^2}. \tag{3}$$

Here, $v_{\text{norm}}(i,j)$ represents the pixel displacement distance, quantifying the intensity of physical motion in this local area across the two adjacent frames. Then, a binary mask M_t is generated by applying a dynamic threshold τ, formally defined using an indicator function:

$$M_t(i,j) = \mathbb{I}\big(v_{\text{norm}}(i,j) > \tau\big), \tag{4}$$

where $\mathbb{I}(\cdot)$ outputs 1 if the condition is met and 0 otherwise. Finally, the IADVR is extracted by performing an element-wise multiplication between the binary mask M_t and the original visual observation I_t, formulated as $I_t^{\text{DM}} = M_t \odot I_t$.

The threshold τ can be adaptively adjusted based on the specific task scenario (see Fig. 5(a) for details). To preserve the visual integrity of the robotic end-effector and interacting objects, and to prevent region discontinuity caused by motion edge fragmentation, we apply a morphological dilation operation to M_t (Appendix B). The aforementioned mechanism effectively suppresses interference from task-irrelevant information, steering the focus towards interactive regions undergoing displacement or deformation, which serves as a crucial prerequisite for intention prediction in our approach.

4.2 Intention Inference via Adaptive Feature Aggregation

To construct an embodied policy equipped with task-oriented dynamics inference capabilities, we propose a multi-step future frame synchronous prediction architecture termed the Adaptive Layer-wise Feature Aggregator (ALFA). At its core, this architecture employs a dual-stage mechanism, combining dynamic region prediction with multi-frame temporal intention modeling, enabling the model to focus on critical interaction dynamics and provide foresight for long-horizon manipulation.

Dynamic interaction region prediction via adaptive feature aggregation. While existing VLA models predominantly rely on the final-layer output of the LLM, recent studies indicate that intermediate layers encapsulate richer spatial topological information, whereas deeper layers are inherently biased towards task semantics (Wang et al., 2026). Motivated by this, we construct a heterogeneous feature aggregation space to fuse shallow visual structural cues with deep abstract semantics. Suppose the backbone network consists of a total of N transformer blocks, and the features extracted from each layer are denoted as z_t^n. The aggregated features $\mathbf{H}_{\text{fusion}}$ are obtained via adaptive weighted summation using cross-layer gated linear units,where $w_n \in \mathbb{R}$ is a *learnable scalar* associated with layer n (not a feature vector) and the softmax is taken over the N scalars.

$$\mathbf{H}_{\text{fusion}} = \sum_{n=1}^{N} \alpha_n z_t^n, \qquad \alpha_n = \frac{\exp(w_n)}{\sum_{j=1}^{N} \exp(w_j)}, \tag{5}$$

Multi-frame parallel prediction for explicit temporal intention modeling. Relying on single-frame dynamic region features is insufficient to capture the temporal coherence of task intentions and the long-horizon dynamic trends of objects post-interaction. Consequently, we introduce a parallel prediction module Ψ_{recon}, which synchronously decodes the visual dynamic regions for the future K frames conditioned on $\mathbf{H}_{\text{fusion}}$:

$$\left[\hat{I}_{t+1}^{\text{DM}}, \ldots, \hat{I}_{t+K}^{\text{DM}}\right] = \Psi_{\text{recon}}\big(\mathbf{H}_{\text{fusion}}\big), \tag{6}$$

During training, Ψ_{recon} is optimized using a sparse visual reconstruction loss, which is exclusively applied to the intention-associated dynamic regions masked by M_{t+n}:

$$\mathcal{L}_{\text{intent}} = \sum_{k=1}^{K} \left\| \hat{I}_{t+k}^{\text{DM}} - M_{t+k} \odot I_{t+k} \right\|_2^2. \tag{7}$$

If the prediction horizon K is too small, the model fails to capture long-horizon dynamic trends; conversely, an excessively large K introduces compounding rollout errors. We empirically analyse the impact of this inference depth on overall model performance in Sec. 5.4.

4.3 Intention-Guided Action Execution

To further bolster the temporal predictive modeling of action generation, we explicitly embed temporal prediction features into the native action representations. Specifically, we project the fused representation $\mathbf{H}_{\text{fusion}}$ into the same latent space as the action features using a Multi-Layer Perceptron (MLP), yielding $\mathbf{H}' = \text{MLP}(\mathbf{H}_{\text{fusion}})$. This feature subsequently guides action generation via a cross-attention mechanism: at each of the N_a residual attention blocks, the current observation feature queries the intention features,

$$h_{\text{attn}}^{(n)} = \text{Softmax}\left(\frac{(z_t^{(n)}\mathbf{W}^Q)(\mathbf{H}'\mathbf{W}^K)^\top}{\sqrt{d_k}}\right)(\mathbf{H}'\mathbf{W}^V), \tag{8}$$

$$\hat{z}_t^{(n)} = z_t^{(n)} + \text{FFN}\big(h_{\text{attn}}^{(n)}\big). \tag{9}$$

After the last block the hidden state is normalised and projected to the action space,

$$\hat{A}_t = \text{MLP}\big(\text{LayerNorm}(\hat{z}_t^{(N_a)})\big), \qquad \mathcal{L}_{\text{action}} = \big\|\hat{A}_t - A_t\big\|_1. \tag{10}$$

We employ a joint optimization strategy, formulating the total loss as $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{action}} + \alpha(t)\,\mathcal{L}_{\text{intent}}$. This module adopts intention features $\mathbf{H}'$ as physical priors to dynamically calibrate the original observation feature z_t^n. Specifically, the model constructs query vectors from z_t^n, and obtains key and value vectors from $\mathbf{H}'$. Attention weights calculated by query-key similarity explicitly quantify the importance of diverse spatio-temporal regions in intention features for action prediction. High weights are allocated to intention-relevant dynamics, including post-grasp object displacement and contact-induced deformation, while task-irrelevant background motions and prediction artifacts are effectively suppressed. Finally, attention-enhanced value features are fused with current observation representations and input into the action decoder to produce continuous action sequences.

5 Experiments

5.1 Experimental Setup and Metrics

Experimental setup. We evaluate DreamIF on the LIBERO-PLUS benchmark (Fei et al., 2025) to strictly assess its foundational performance and out-of-distribution (OOD) robustness; LIBERO-PLUS extends LIBERO (Liu et al., 2023) with seven perturbation families across four task suites. Furthermore, We further evaluate DreamIF in the **Real-world** using a 6-DoF AgileX Piper robot across 8 complex tasks, evaluating robust grasping, spatial generalization, and long-horizon manipulation. Details are provided in Appendices D and F.

Metrics. Following standard protocols in embodied intelligence, we explicitly adopt the average Success Rate (SR) as the primary evaluation metric. A trial is strictly considered successful only when the robot correctly executes all operational steps guided by the natural language instruction, culminating in an environmental state that aligns with the anticipated goal.

5.2 Simulation Results

In the in-distribution setting, DreamIF attains an average success rate of 85.7%, 3.7 points above the corresponding baseline. The gains concentrate on changes in object placement and robot initialization. This suggests that predicting regions affected by the next action helps the policy adapt to changes in *what* is manipulated and *where*. **In the out-of-distribution setting**, DreamIF reaches 65.3%, 6.2 points above the matched baseline, with the largest margins on the long-horizon suite and on the families that change the camera viewpoint or the initial pose. The reason is that the prediction target is defined by motion rather than by appearance: after a change of viewpoint or pose, it still points at the future dynamic region, which assists the model in predicting its actions.

Extending DreamIF to the $\pi_{0.5}$ backbone yields 90.7% in-distribution and 87.3% out-of-distribution success rates. Taken together, the in-distribution and out-of-distribution results indicate that grounding the prediction target in motion rather than in appearance is what produces the improvement; DreamIF outperforms the matched baseline in both settings and gains the most on long-horizon tasks and under viewpoint or pose perturbations, showing that it combines stronger task-oriented dynamics inference with robustness under distribution shift.

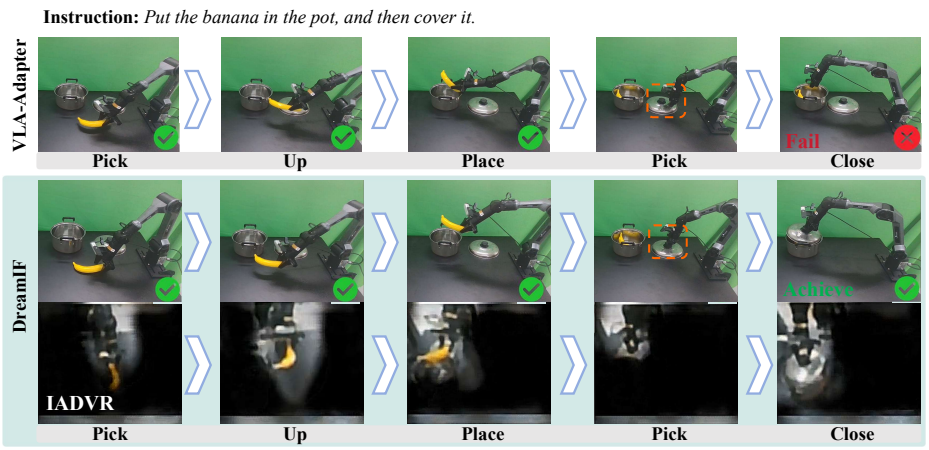

Figure 3: **Stage-level comparison on the pot-with-lid composite (Task 3).** The five columns are the pick, lift, place, lid-pick and close stages. Top: the reactive VLA-Adapter baseline. Bottom: DreamIF, with the IADVR supervision drawn above the frames it executes.

Method	Camera	Robot	Language	Light	Background	Noise	Layout	Average
In-distribution: train and test on LIBERO-PLUS								
OpenVLA-OFT (Kim et al., 2025a)	92.8	30.3	85.8	94.9	93.9	89.3	77.6	79.6
VLA-Adapter (Wang et al., 2026)	94.3	46.0	80.4	95.4	95.7	93.3	77.7	82.0
π_0 (Black et al., 2025b)	89.7	65.9	85.6	90.4	91.3	85.7	74.6	83.3
$\pi_{0.5}$ (SFT, frozen) (Black et al., 2025a)	70.3	41.7	81.1	**97.3**	94.6	71.8	84.9	75.7
ACoT-VLA (Zhong et al., 2026)	**96.6**	70.4	79.7	95.1	97.1	95.9	85.0	88.0
DreamIF (ours)	94.8	50.7	**88.0**	96.6	97.1	**97.1**	85.6	85.7
DreamIF+$\pi_{0.5}$ (ours)	93.7	**80.1**	84.8	96.5	**97.6**	96.5	**88.8**	**90.7**
Out-of-distribution: train on LIBERO, test on LIBERO-PLUS								
WorldVLA (Cen et al., 2025b)	0.1	27.9	41.6	43.7	17.1	10.9	39.0	25.0
UniVLA (Bu et al., 2025)	1.8	46.2	69.6	69.0	81.0	21.2	31.9	42.9
π_0 (Black et al., 2025b)	13.8	6.0	58.8	85.0	81.4	79.0	68.9	53.6
VLA-Adapter (Wang et al., 2026)	37.4	37.8	74.5	70.3	75.6	58.9	71.1	59.0
OpenVLA-OFT (Kim et al., 2025a)	56.4	31.9	79.5	88.7	93.3	75.8	74.2	69.6
$\pi_{0.5}$ (zero-shot) (Black et al., 2025a)	**75.8**	79.4	83.3	95.5	95.0	89.6	87.0	85.7
ACoT-VLA (Zhong et al., 2026)	72.6	**82.6**	87.5	**97.7**	96.5	87.8	88.1	86.6
DreamIF (ours)	50.1	49.0	82.2	73.7	75.3	63.9	71.2	65.3
DreamIF+$\pi_{0.5}$ (ours)	73.1	81.4	**88.4**	96.5	**98.0**	**90.9**	**88.6**	**87.3**

Table 1: **Success rates (%) on LIBERO-PLUS.**The upper and lower blocks report in-distribution and out-of-distribution results, respectively.VLA-Adapter and DreamIF are trained by us under the corresponding protocols;other results are taken from their original sources (Appendix D.3).

5.3 Real-world Results

Table 2 compares DreamIF with VLA-Adapter and VLA-JEPA under the same demonstrations and trial protocol. DreamIF matches or outperforms VLA-Adapter on all eight tasks, achieving an average success rate of 80.6%, 15.6 points above VLA-Adapter and higher than VLA-JEPA. Fig. 5(c) shows the region the model infers on the real arm. DreamIF's mechanistic advantages grow with the difficulty of the task. On foundational manipulation(Tasks 1-3), intention anticipation constrains the end-effector trajectory and mitigates the execution jitter and contact deviations that plague the baseline, while in clutter the IADVR mechanism filters out background distractors and prevents attention drift, a difficulty that is purely perceptual rather than dynamic.

DreamIF achieves its largest gains in long-horizon manipulation, especially on the two articulated composites, where later sub-goals depend on clean completion of earlier ones. In the representative Task 3 episode in Fig. 3, both policies complete the first three stages, but DreamIF then grasps the lid and closes the pot while the baseline stops. Because the lid is large, unoccluded, and stationary, the challenge is not recognition but conditioning the next action on a workspace reshaped by previous actions. By predicting the region the pending action will move, DreamIF identifies the lid as the next interaction target, enabling the grasp and closing actions.

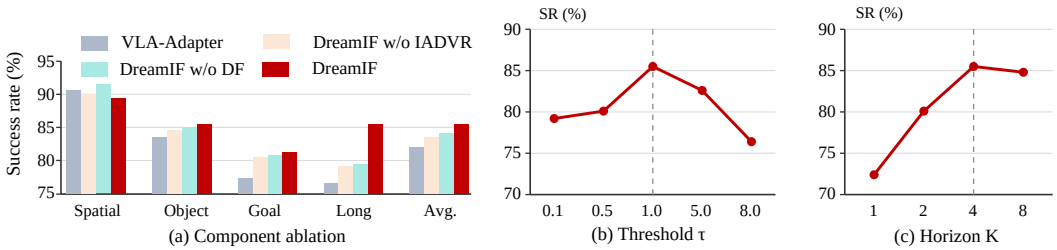

Figure 4: **Component ablation and parameter analysis.** *(a)* Success rate for the four variants. *(b)* Motion threshold τ and *(c)* prediction horizon K, both on the LIBERO-PLUS Long suite.

ID	Task	VLA-Adapter	VLA-JEPA	DreamIF (Ours)
1	Upright mug (fine contact)	60	75	**85**
2	Open cabinet door (articulated)	80	90	**95**
3	Pot with lid (articulated, multi-step)	50	55	**65**
4	Banana into bowl (pose variation)	75	**85**	**85**
5	Eggplant into pot (clutter)	80	**85**	80
6	Pick into cabinet (articulated, multi-step)	55	60	**80**
7	Wipe table (cyclic long-horizon)	65	70	**80**
8	Sequential blocks (colour ordering)	55	50	**75**
	Average	65.0	71.3	**80.6**

Table 2: **Real-world Task success rates (%)** Each method is evaluated over 20 trials per task, with success defined as goal completion without collision within 60 s.

5.4 Ablation Study

To systematically validate the effectiveness of the core components and key parameters of DreamIF, we conduct ablation studies on the LIBERO-PLUS benchmark. Specifically, we investigate: (1) the impact of the IADVR extraction mechanism and its dynamic threshold τ; (2) the influence of the prediction horizon K; and (3) the contribution of the dynamic feedback mechanism.

5.4.1 Ablation Analysis of Core Components

Initially fixing the optimal parameters ($\tau = 1.0$, $K = 4$), we compare the following four variants: **VLA-Adapter:** a purely reactive baseline. **DreamIF w/o IADVR:** employs full-frame image reconstruction for DreamIF. **DreamIF w/o DF:** removes the explicit intention features from DreamIF. **DreamIF:** the complete model, integrating the IADVR extraction with the Dynamic Feedback (DF).

Efficacy of IADVR. As Fig. 4(a) shows, even when relying on full-frame reconstruction, DreamIF w/o IADVR yields a 1.6-point average improvement over the baseline VLA-Adapter, demonstrating the inherent value of dynamics-transition learning: internalising the underlying physical transitions inherently bolsters action representations. The complete model further surpasses DreamIF w/o IADVR, and the improvement is particularly pronounced on the LIBERO-PLUS Long subset, which is characterised by severe background distractors, where the success rate increases by a notable 6.3 points. This advantage stems from the IADVR extraction mechanism, which leverages optical-flow-guided motion saliency perception to actively filter out static backgrounds and task-irrelevant objects; consequently, it compels the model to concentrate its attention on core dynamic regions, such as the end-effector and the interacting objects, thereby effectively mitigating attention drift.

Efficacy of Dynamic Feedback (DF). DreamIF outperforms DreamIF w/o DF across the subsets, with a 6.0-point success-rate gain on long-horizon tasks in Fig. 4(a). Over 250-step open-loop rollouts, DreamIF exhibits lower joint-position errors across all six joints and slower error accumulation than the w/o DF variant (Fig. 7). This indicates that purely implicit multi-task learning alone cannot fully unleash the anticipatory guidance of dynamics prediction for decision making. By explicitly embedding multi-frame future IADVR temporal features into the action representations via cross-attention, DreamIF enables the policy to dynamically anticipate future intentions during action

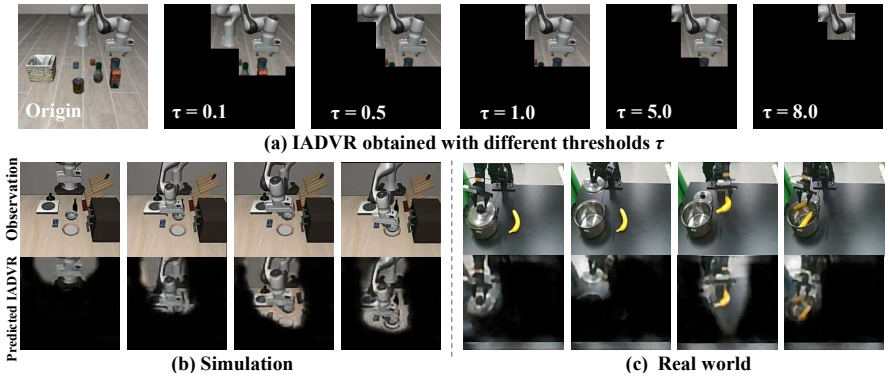

Figure 5: **IADVR extraction and inference.** *(a)* IADVR at different thresholds: as τ increases, the extent of the IADVR narrows step by step. *(b)* Inference in simulation and *(c)* in the real world: the upper row is the observation and the lower row the IADVR the model infers.

generation, thereby maintaining superior trajectory fidelity in open-loop long-horizon rollouts and effectively mitigating compounding errors in long-horizon manipulation.

5.4.2 Analysis of IADVR Dynamic Threshold τ and Prediction Horizon K

Ablation analysis of τ. At $\tau = 0.1$, subtle movements such as sensor noise are misclassified as intention regions, causing the mask to cover nearly the entire frame and lose its filtering efficacy, which yields a success rate of only 79.2%, as shown in Fig. 4(b). At $\tau = 1.0$ the model optimally preserves the complete motion trajectories of the end-effector and the interacting objects while suppressing background noise, achieving a peak success rate of 85.5%. At $\tau = 8.0$ critical motion information is lost and action precision is impaired, dropping the success rate to 76.4%. This inverted U-shaped behaviour arises because an excessively low threshold introduces redundant information that distracts the model's attention, whereas an excessively high threshold removes information the action needs. We therefore adopt $\tau = 1.0$ as the default.

Ablation analysis of the prediction horizon K. As K increases, the success rate first increases and then decreases, highlighting the trade-off between temporal modeling and prediction errors (Fig. 4(c)). At $K = 1$, single-frame prediction captures only instantaneous state transitions, limiting the model's ability to anticipate multi-step operations and yielding a success rate of 72.4%. At $K = 4$, the model successfully learns sufficiently long temporal dependencies, providing effective anticipatory constraints for decision making without letting prediction errors accumulate excessively, reaching the highest success rate of 85.5%. At $K = 8$, compounding errors from long-term predictions begin to dominate and inaccurate future priors inadvertently interfere with current decisions, so the success rate decreases slightly to 84.8%. This demonstrates the benefits of multi-frame prediction, while excessive prediction horizons lead to error accumulation and diminishing returns.

6 Conclusion

To address the compounding errors of existing VLA models in long-horizon manipulation, we propose DreamIF, an intention-guided embodied foundation model. Its core lies in enabling a paradigm shift from reactive control to intention-guided *deductive* control. We first design an Intention-Associated Dynamic Visual Region (IADVR) extraction moudule, which automatically locates dynamic regions highly correlated with task intentions, filtering redundant static backgrounds and task-irrelevant objects. We further build the Adaptive Layer-wise Feature Aggregation (ALFA) architecture to integrate multi-granularity semantic and spatial features from the VLM. Based on this, we conduct parallel multi-frame prediction of IADVRs to explicitly model environment transitions implied by manipulation intentions. Finally, we embed these temporal predictive features into native action representations, realizing dynamic calibration and anticipatory guidance for action generation. Extensive experiments on the perturbation-rich LIBERO-PLUS benchmark show that DreamIF+$\pi_{0.5}$ surpasses existing methods. It reaches an 87.3% success rate on the LIBERO-PLUS, verifying its strong generalization in complex noisy environments.

AI USE STATEMENT

In this work, we used generative AI tools to assist with manuscript polishing and code review, including improving the clarity and readability of the text and identifying potential issues in the implementation. All AI-assisted revisions and code suggestions were carefully reviewed and verified by the authors before being incorporated into the work. The authors take full responsibility for the final manuscript and implementation.

REPRODUCIBILITY STATEMENT

Implementation details, training configurations, and evaluation protocols are provided in the appendix. An anonymized ZIP archive containing the source code for IADVR preprocessing, model training, and evaluation is provided as supplementary material.

REFERENCES

Icek Ajzen. The theory of planned behavior. *Organizational Behavior and Human Decision Processes*, 50:179–211, 1991.

Bowen Baker, Ilge Akkaya, Peter Zhokov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. Video pretraining (vpt): Learning to act by watching unlabeled online videos. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24639–24654, 2022.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pre-trained image-editing diffusion models. In *International Conference on Learning Representations*, 2024.

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

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 *Robotics: Science and Systems*, 2025b.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. In *Robotics: Science and Systems*, 2023.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Learning to act anywhere with task-centric latent actions. In *Robotics: Science and Systems*, 2025.

Jun Cen, Siteng Huang, Yuqian Yuan, Kehan Li, Hangjie Yuan, Chaohui Yu, Bohan Hou, Yuming Jiang, Jiayan Guo, Xin Li, et al. Rynnvla-002: A unified vision-language-action and world model. *arXiv preprint arXiv:2511.17502*, 2025a.

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

Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, et al. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. In *International Conference on Machine Learning*, pp. 8469–8488. PMLR, 2023.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, et al. Libero-plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. Gaia-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023.

Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. In *International Conference on Machine Learning*, pp. 24328–24346. PMLR, 2025.

Changxin Huang, Ronghui Zhang, Meizi Ouyang, Pengxu Wei, Junfan Lin, Jiang Su, and Liang Lin. Deductive reinforcement learning for visual autonomous urban driving navigation. *IEEE Transactions on Neural Networks and Learning Systems*, 32(12):5379–5391, 2021.

Changxin Huang, Lv Tang, Zhaohuan Zhan, Lisha Yu, Runhao Zeng, Zun Liu, Zhengjie Wang, and Jianqiang Li. Unemo: Collaborative visual-language reasoning and navigation via a multimodal world model. In *AAAI Conference on Artificial Intelligence*, pp. 18315–18323, 2026.

Chia-Yu Hung, Qi Sun, Pengfei Hong, Amir Zadeh, Chuan Li, U-Xuan Tan, Navonil Majumder, and Soujanya Poria. Nora: A small open-sourced generalist vision language action model for embodied tasks. *arXiv preprint arXiv:2504.19854*, 2025.

Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. In *European Conference on Computer Vision*, pp. 18–35. Springer, 2024.

Nikita Karaev, Yuri Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6013–6022, 2025.

Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic vlms: Investigating the design space of visually-conditioned language models. In *International Conference on Machine Learning*, 2024.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. In *Robotics: Science and Systems*, 2025a.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, et al. Openvla: An open-source vision-language-action model. In *Conference on Robot Learning*, pp. 2679–2713. PMLR, 2025b.

Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, et al. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv preprint arXiv:2411.19650*, 2024.

Zhiqi Li, Guo Chen, Shilong Liu, Shihao Wang, Vibashan VS, Yishen Ji, Shiyi Lan, Hao Zhang, Yilin Zhao, Subhashree Radhakrishnan, et al. Eagle 2: Building post-training data strategies from scratch for frontier vision-language models. *arXiv preprint arXiv:2501.14818*, 2025.

Wenlong Liang, Rui Zhou, Yang Ma, Bing Zhang, Songlin Li, Yijia Liao, and Ping Kuang. Large model empowered embodied ai: A survey on decision-making and embodied learning. *arXiv preprint arXiv:2508.10399*, 2025.

Kevin Qinghong Lin, Linjie Li, Difei Gao, Zhengyuan Yang, Shiwei Wu, Zechen Bai, Stan Weixian Lei, Lijuan Wang, and Mike Zheng Shou. Showui: One vision-language-action model for gui visual agent. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19498–19508, 2025.

Xuewu Lin, Tianwei Lin, Yun Du, Hongyu Xie, Yiwei Jin, Jiawei Li, Shijie Wu, Qingze Wang, Mengdi Li, Mengao Zhao, et al. Holobrain-0 technical report. *arXiv preprint arXiv:2602.12062*, 2026.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Matthew Lisondra, Beno Benhabib, and Goldie Nejat. Embodied ai with foundation models for mobile service robots: A systematic review. *Robotics*, 2026.

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, 2023.

Jiska Memelink and Bernhard Hommel. Intentional weighting: A basic principle in cognitive control. *Psychological Research*, 77(3):249–259, 2013.

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

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. FAST: Efficient action tokenization for vision-language-action models. In *Robotics: Science and Systems*, 2025.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics*, 2011.

Hao Shi, Bin Xie, Yingru Li, Jiayuan Yang, Hao Dong, et al. Memoryvla: Perceptual-cognitive memory in vision-language-action models for robotic manipulation. In *International Conference on Learning Representations*, 2026.

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, et al. Smolvla: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025.

Wenxuan Song, Ziyang Zhou, Han Zhao, Jiayi Chen, Pengxiang Ding, Haodong Yan, Yuxin Huang, Feilong Tang, Donglin Wang, and Haoang Li. ReconVLA: Reconstructive vision-language-action model as effective robot perceiver. In *AAAI Conference on Artificial Intelligence*, 2026.

A Steiner, AS Pinto, M Tschannen, D Keysers, X Wang, Y Bitton, A Gritsenko, M Minderer, A Sherbondy, S Long, et al. Paligemma 2: A family of versatile vlms for transfer. *arXiv preprint arXiv:2412.03555*, 2024.

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. In *European Conference on Computer Vision*, 2026.

Yang Tian, Sizhe Yang, Jia Zeng, Ping Wang, Dahua Lin, Hao Dong, and Jiangmiao Pang. Predictive inverse dynamics models are scalable learners for robotic manipulation. In *International Conference on Learning Representations*, 2025.

Yihao Wang, Pengxiang Ding, Lingxiao Li, Can Cui, Zirui Ge, Xinyang Tong, Wenxuan Song, Han Zhao, Wei Zhao, Pengxu Hou, et al. Vla-adapter: An effective paradigm for tiny-scale vision-language-action model. In *AAAI Conference on Artificial Intelligence*, pp. 18638–18646, 2026.

Junjie Wen, Yichen Zhu, Jinming Li, Minjie Zhu, Zhibin Tang, Kun Wu, Zhiyuan Xu, Ning Liu, Ran Cheng, Chaomin Shen, et al. TinyVLA: Toward fast, data-efficient vision-language-action models for robotic manipulation. *IEEE Robotics and Automation Letters*, 2025.

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*, 2024.

Ze Yang, Yun Chen, Jingkang Wang, Sivabalan Manivasagam, Wei-Chiu Ma, Anqi Joyce Yang, and Raquel Urtasun. Unisim: A neural closed-loop sensor simulator. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1389–1399, 2023.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu Wei Chao, Yuchen Lin, et al. Latent action pretraining from videos. In *International Conference on Learning Representations*, pp. 90629–90655, 2025.

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.

Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, XinQiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, He Wang, Zhizheng Zhang, et al. Dreamvla: A vision-language-action model dreamed with comprehensive world knowledge. In *Advances in Neural Information Processing Systems*, 2025.

Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3d-vla: A 3d vision-language-action generative world model. In *International Conference on Machine Learning*, pp. 61229–61245, 2024.

Linqing Zhong, Yi Liu, Yifei Wei, Ziyu Xiong, Si Liu, and Guanghui Ren. ACoT-VLA: Action chain-of-thought for vision-language-action models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pp. 2165–2183. PMLR, 2023.

APPENDIX OF DREAMIF

A SYMBOLS AND NETWORK ARCHITECTURE

Table 3 lists every symbol used in the main text, together with its shape or type and the equation in which it first appears. We include it because an earlier version of this work left several symbols under-specified, in particular the aggregation weights of Eq. (5), which are scalars and not layer features.

Table 3: Symbol glossary. Shapes are given for a single sample.

Symbol	Meaning	Type / shape
$O_t = \{I_t, P_t, \ell\}$	observation tuple at step t (image, proprioception, instruction)	n/a
I_t	RGB frame	$\mathbb{R}^{H\times W\times 3}$, $H = W = 224$
$A_t = \{a_t, \dots, a_{t+H_a-1}\}$	expert action chunk	$\mathbb{R}^{H_a\times d_a}$, $H_a = 8$
z_t^n	hidden state of VLM layer n	$\mathbb{R}^d$
N	number of VLM transformer layers	scalar
N_a	number of residual attention blocks in the action head	scalar
w_n	**learnable scalar** weight of layer n	$\mathbb{R}$
α_n	softmax-normalised aggregation weight, Eq. (5)	$\mathbb{R}$, $\sum_n \alpha_n = 1$
$\mathbf{H}_{\text{fusion}}$	layer-aggregated feature	$\mathbb{R}^d$
$\mathbf{H}'$	projection of $\mathbf{H}_{\text{fusion}}$ into action latent space	$\mathbb{R}^{d_k}$
K	intention prediction horizon (number of future frames)	scalar, default 4
S_{patch}	stride of the query grid on the image plane	scalar, default 8
C	number of tracker query points, $C = (H/S_{\text{patch}})(W/S_{\text{patch}}) = 784$	scalar
s	temporal gap between the two frames given to the tracker	scalar
$\Delta x_{i,j}^{(c)}, \Delta y_{i,j}^{(c)}$	displacement of query c at grid cell (i, j) along x / y	$\mathbb{R}$
$v_{\text{norm}}(i, j)$	displacement magnitude, Eq. (2)	$\mathbb{R}_{\geq 0}$
τ	motion-saliency threshold, Eq. (4)	scalar, default 1.0
M_t	binary IADVR mask	$\{0, 1\}^{H\times W}$
I_t^{DM}	masked observation, $M_t \odot I_t$	$\mathbb{R}^{H\times W\times 3}$
Ψ_{recon}	parallel multi-frame prediction head	network
$\hat{I}_{t+k}^{\text{DM}}$	predicted masked frame at $t + k$	$\mathbb{R}^{H\times W\times 3}$
$\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V$	attention projections, Eq. (8)	$\mathbb{R}^{d\times d_k}$, $\mathbb{R}^{d\times d_k}$, $\mathbb{R}^{d\times d_v}$
$h_{\text{attn}}^{(n)}$	attention output at action-head block n	$\mathbb{R}^{d_v}$
$\hat{z}_t^{(n)}$	residual-updated feature, Eq. (9)	$\mathbb{R}^d$
$\hat{A}_t$	decoded action chunk	$\mathbb{R}^{H_a\times d_a}$
$\alpha(t)$	time-decaying weight on the intention loss	scalar
$\mathcal{D}$	expert demonstration dataset	n/a

B DATA PREPROCESSING AND IADVR MASK GENERATION

This section documents the offline stage that produces the motion supervision. It runs once per episode and is never invoked at deployment.

We extract the base and wrist images of each episode and resize both to 224×224. For the temporal pairing we use a sliding window of length 4 and keep the first (t=0) and last (t=3) frames, forming a two-frame clip. On the 224×224 plane we build a regular query grid with stride $S_{\text{patch}} = 8$ px:

$$C = \left(\frac{H}{S_{\text{patch}}}\right) \times \left(\frac{W}{S_{\text{patch}}}\right) = \left(\frac{224}{8}\right) \times \left(\frac{224}{8}\right) = 28 \times 28 = 784. \tag{11}$$

Component	Parameter	Value
Heterogeneous visual encoders		
DINOv2	Pre-trained variant	ViT-L/14 (reg4)
	Input resolution	224×224
	Patch embedding dimension D	1024
SigLIP	Pre-trained variant	ViT-SO400M/14
	Input resolution	224×224
	Patch embedding dimension D	1152
Parallel prediction head Ψ_{recon}		
Residual blocks	Number of blocks	24
	Neurons per block (LLM dim)	896
	Dropout rate	0.0
Output layer	Linear transformation	896×588
Action head Φ_{act}		
MLPResNet	Number of layers (cross-attention)	24
	Number of attention heads	8
	Feed-forward network dimension	896
Action chunking	Chunk size H_a	8

Table 4: Network architecture and hyper-parameters.

All 784 query points are timestamped at $T_{\text{start}} = 0$. We do not store the full trajectory tensor; only the displacement of the last frame relative to the first is retained,

$$\Delta\mathcal{F} = \mathbf{T}_{t=1} - \mathbf{T}_{t=0}, \qquad \Delta\mathcal{F} \in \mathbb{R}^{784\times 2}, \tag{12}$$

which is what Eqs. (2)–(4) consume. **Tracker configuration:** we use the offline variant of Co-Tracker (Karaev et al., 2024) so that it can exploit complete temporal context (Table 5); this is a preprocessing-time convenience only, and the online variant would be equally valid for generating supervision.

Table 5: Configuration of the offline point tracker used for mask generation.

Parameter	Value
Window length	60
Optimization iters	6
Spatial stride	4
Correlation radius	3
Temporal stride	$L_{\text{win}}/2$

Coordinate alignment. All spatial transforms use align-corners bilinear semantics. Input video is first downsampled to the tracker's internal resolution (H_m, W_m); grid queries map as

$$x_m = x_{\text{orig}} \cdot \frac{W_m - 1}{W - 1}, \qquad y_m = y_{\text{orig}} \cdot \frac{H_m - 1}{H - 1}, \tag{13}$$

and tracker outputs are mapped back with the inverse transform. **Pooling to the encoder grid.** $\Delta\mathcal{F}$ lives on a 28×28 grid, whereas DINOv2 at 224×224 with patch size 14 produces a 14×14 token grid. We therefore apply average pooling with kernel 2 and stride 2 before fusion. This pooling aligns the motion features with the encoder tokens; it does not discard the underlying motion estimate, which is computed at full resolution.

Dilation. To avoid fragmenting the end-effector and the manipulated object along motion edges, the binary mask of Eq. (4) is morphologically dilated before use. The dilation radius is fixed at 2 grid cells in all experiments.

C Training and Implementation Details

Hardware and software. PyTorch 2.4 with the HuggingFace Transformers library. Training uses eight GPUs; profiling for deployment is done on a single GPU. Peak training memory is 22 GB; inference at 20 Hz needs 5 GB.

Two-stage schedule. Stage 1 is instruction fine-tuning for base VLA capability; stage 2 is joint optimisation with the prediction and cross-attention branches enabled. We freeze the vision towers and the vast majority of the VLM backbone, adapting the network with LoRA of rank 32, and fully train the projector, the language-model head and the prediction head.

Loss balance. The action term is an L_1 loss. The intention term uses a decaying weight $\alpha(t)$: early in training a larger weight lets the model establish the physical evolution prior from motion saliency; the weight then decays so that the anticipation objective acts as a constraint rather than competing with action precision late in training.

Optimisation. AdamW, weight decay 0.1, cosine decay after 500 linear warmup steps from a peak of 1×10^{-4} to a minimum of 1×10^{-6}.

Table 6: Default hyper-parameters. Both task-level knobs are analysed in Fig. 4(b,c).

Hyper-parameter	**Value**
Optimizer	AdamW
Initial / minimum learning rate	1×10^{-4} / 1×10^{-6}
Weight decay	0.1
LR schedule	cosine decay
Warmup steps	500
Action loss weight	1.0
Intention loss weight $\alpha(t)$	dynamic decay (init. 0.5)
LoRA rank	32
Action chunk H_a	8
Intention horizon K	4 (swept over $\{1, 2, 4, 8\}$)
Motion threshold τ	1.0 (swept over $\{0.1, 0.5, 1.0, 5.0, 8.0\}$)
Input resolution	224×224

Unified evaluation protocol. Every number we report is produced under one evaluation protocol: identical test suites, identical success criteria, 20 trials per real-world task, and the same demonstration budget for the fine-tuned baselines. Each comparison table states the training corpus of its rows, and baselines whose published numbers were produced under a different corpus are reported only in the protocols where their data matches (Sec. 5.1); for the baselines we quote rather than reproduce, the training recipe is the one given in the original publication.

D Simulation Protocol

LIBERO-PLUS (Fei et al., 2025) extends LIBERO (Liu et al., 2023) with seven controlled perturbation families: Camera viewpoints, Robot initial states, Light conditions, Background textures, Language instructions, Sensor noise, and Object layout. Evaluations follow the standard protocol of the benchmark: each suite is rolled out to completion, and a rollout counts as a success only if the goal predicate holds at termination. For LIBERO-PLUS this means every task instance in a suite, once; the four suites contain 2519, 2591, 2518 and 2402 instances respectively, $10{,}030$ in total.

D.1 The standard LIBERO protocol

The third protocol of Sec. 5.1, train on LIBERO and test on LIBERO, is reported here rather than in the main text because the benchmark is close to saturation: six of the seven baselines fall inside a 1.1-point band, so the table serves coverage of the long-horizon literature (memory, latent-prediction, holistic-fusion and light world-action policies) rather than a margin. Four suites of ten tasks are rolled out to completion with 50 trials per task, and every method is evaluated on the identical clean test set. The distribution of the small remaining spread is the informative part: DreamIF improves most on Long, where error accumulation binds, which is the same family that carries the margin under shift (Table 1) and on the real robot (Table 2).

D.2 The two LIBERO-PLUS protocols in the four-suite representation

Table 1 reports the in-distribution and out-of-distribution protocols in the seven-family representation, which is the one that exposes *what* kind of appearance change a method absorbs. The same two

Table 7: **The standard protocol: LIBERO train → LIBERO test.** Four suites, ten tasks each, 50 trials per task; every method is evaluated on the identical test set. The *Average* column is the four-suite mean as reported by each source; the rule separates the two strongest rows from the remaining baselines. Bold marks the largest value in a column, so the saturated Object column goes to Fast-WAM. ACoT-VLA leads on Long and ties DreamIF on Goal in this protocol, and DreamIF leads on Average and Spatial, so the table is reported as coverage of the long-horizon literature rather than as a clean win.

Method	Average	Spatial	Object	Goal	Long
WorldVLA (Cen et al., 2025b)	81.8	87.6	96.2	83.4	60.0
MemoryVLA (Shi et al., 2026)	96.5	98.4	98.4	96.4	93.4
VLA-JEPA (Sun et al., 2026)	97.2	96.2	99.6	97.2	95.8
VLA-Adapter (Wang et al., 2026)	97.3	97.8	99.2	97.2	95.0
HoloBrain-0 (Lin et al., 2026)	97.4	97.2	99.6	97.6	95.2
RynnVLA-002 (Cen et al., 2025a)	97.4	99.0	99.8	96.4	94.4
Fast-WAM (Yuan et al., 2026)	97.6	98.2	**100.0**	97.0	95.2
ACoT-VLA (Zhong et al., 2026)	98.5	98.6	99.0	**99.4**	**97.0**
DreamIF (Ours)	**98.6**	**99.6**	99.4	**99.4**	96.8

protocols in the four-suite representation are given in Table 8: it is the view in which the long-horizon claim is cashed, because the Long suite is a single column rather than one of seven. The two rows we trained under the out-of-distribution protocol come from one evaluation batch, so the comparison between them is exact; the *Average* of those two rows is the macro-average over the four suites, and pooled over all 10,030 instances it is 65.02 against 58.70, i.e. the same margin (+6.3 rather than +6.2 points). For the in-distribution protocol the two rows we trained carry the same suite-level numbers as the component ablation (Table 9), and their four-suite arithmetic means are 82.0 and 85.5.

Table 8: **The two LIBERO-PLUS protocols in the four-suite representation.** *Panel A* is the in-distribution protocol, *Panel B* the out-of-distribution one; the suites are the four LIBERO task families inside the perturbed benchmark. *Average* is the macro-average over the four suites for the rows we ran; for quoted rows it is the value printed by the source, whose pooling convention differs slightly, and the two rows below the rule in Panel A carry the seven-family figure of Table 1 because that row's published suite breakdown is not available. Bold marks the largest value of a column, whichever row it belongs to; both $\pi_{0.5}$ rows are scale references rather than like-for-like comparisons with the rows we trained (Appendix D.3).

Method	Average	Spatial	Object	Goal	Long
Panel A, in-distribution: LIBERO-PLUS train → LIBERO-PLUS test					
VLA-Adapter (Wang et al., 2026)	82.0	90.6	83.6	77.4	76.6
DreamIF (Ours)	85.7	89.5	85.5	81.3	85.5
$\pi_{0.5}$ (SFT, frozen) (Black et al., 2025a)	75.7	n/a	n/a	n/a	n/a
DreamIF+$\pi_{0.5}$	**90.7**	**93.8**	**93.1**	**86.1**	**90.0**
Panel B, out-of-distribution: LIBERO train → LIBERO-PLUS test					
OpenVLA (Kim et al., 2025b)	15.6	19.4	14.0	15.1	14.3
WorldVLA (Cen et al., 2025b)	25.0	32.5	28.6	31.8	8.2
NORA (Hung et al., 2025)	39.0	47.6	34.4	38.8	36.3
UniVLA (Bu et al., 2025)	42.9	55.5	36.7	40.7	39.9
π_0 (Black et al., 2025b)	53.6	60.7	61.4	44.9	48.4
VLA-Adapter (Wang et al., 2026)	59.0	85.1	46.4	55.0	49.7
π_0-FAST (Pertsch et al., 2025)	61.6	74.4	72.7	57.5	43.4
DreamIF (Ours)	65.3	84.4	52.1	64.8	59.6
OpenVLA-OFT (Kim et al., 2025a)	69.6	84.0	66.5	63.0	66.4
$\pi_{0.5}$ (zero-shot) (Black et al., 2025a)	85.7	n/a	n/a	n/a	n/a
DreamIF+$\pi_{0.5}$	**87.3**	**92.0**	**91.1**	**83.2**	**83.0**

D.3 COMPARABILITY OF THE QUOTED ROWS

Published LIBERO-PLUS numbers. Every row of the out-of-distribution block in Table 1 except VLA-Adapter and DreamIF is a number published with the LIBERO-PLUS benchmark, quoted here as tabulated by Zhong et al. (2026), which is also the source of the ACoT-VLA rows in both blocks. Those baselines were trained on the *unaugmented* LIBERO dataset, which is exactly the split the out-of-distribution block reproduces, so they are listed under a labelled protocol instead of being mixed into the matched comparison of the in-distribution block; we do not retrain them, and the two rows we did train ourselves are marked in the caption. Reading them against the in-distribution block is what makes the cost of the shift visible per family, and it is also why that block is not presented as a leaderboard: the multi-billion-parameter systems in it (OpenVLA-OFT, zero-shot $\pi_{0.5}$ and ACoT-VLA) were produced with far larger backbones and data budgets than our 0.5B model, and the in-distribution comparison, where every row sees the same augmented corpus, remains the one our claims rest on. The same caution applies there, where π_0 leads us on Robot initial state and ACoT-VLA is the strongest fine-tuned row overall: both are reported rather than explained away.

The two $\pi_{0.5}$ SFT rows. In the in-distribution block we list the $\pi_{0.5}$ row exactly as the source tabulates it, with one property that a reader comparing it to the zero-shot row will notice: it is the variant whose *LLM backbone is frozen* during fine-tuning, which is why the same backbone scores 85.7 when transferred zero-shot and 75.7 when adapted to LIBERO-PLUS. The source marks this with a dagger in its own table and reports a matching frozen variant of its own method (84.1) alongside the fully fine-tuned one (88.0), so the freeze is a property of the row and not of the backbone. Because our DreamIF+$\pi_{0.5}$ rows are trained without that freeze, the $+15.0$-point gap between our in-distribution row (90.7) and this reference (75.7) is a same-backbone contrast rather than a like-for-like margin, and the abstract quotes it on that reading; the like-for-like comparison is against the strongest fully fine-tuned row, ACoT-VLA, at $+2.7$ points.

The DreamIF+$\pi_{0.5}$ rows. Both rows mount the same prediction and cross-attention head on the $\pi_{0.5}$ backbone (Black et al., 2025a) and are trained and evaluated by us, so they are the only scale test in the paper in which the head is the variable we changed. They are produced in three steps. *(i) Data regime.* The two rows differ in their training corpus, exactly as the two blocks of Table 1 do: the in-distribution row is trained on the augmented LIBERO-PLUS corpus and the out-of-distribution row on the fixed LIBERO-Original corpus. *(ii) Evaluation checkpoint.* Each protocol is trained once and all four suites are evaluated with the same checkpoint; no checkpoint is selected per suite. The four suites are then pooled over all $10{,}030$ instances. *(iii) Reporting.* The pooled figures are 90.68 (in-distribution) and 87.27 (out-of-distribution); restricting to instances that pass the benchmark's validity check gives 90.67 and 87.26, which is what the per-family and per-suite tables print as 90.7 and 87.3.

Two caveats follow, and they are the reason the rows are labelled a scale reference rather than a controlled comparison in the main text. First, the quoted $\pi_{0.5}$ SFT row carries the frozen backbone discussed above and its per-suite breakdown is not published, so the comparison with it is aggregate-to-aggregate. Second, the out-of-distribution comparisons are against quoted rows produced under a different evaluation pipeline, so our row's $+1.6$-point gap to the quoted zero-shot row, and its $+0.7$-point lead over ACoT-VLA, are orderings rather than measured margins. What the rows do support is what they were added for: the same head keeps helping when the backbone changes, and it does so on the two protocols in which a fully fine-tuned comparison exists.

E COMPONENT ABLATION: PER-SUITE NUMBERS

This table is the per-suite version of Fig. 4(a) in the main text, which plots the same four variants under the same protocol. Pooled over task instances, the complete model is 85.7 against 82.0 for the baseline (Table 1), i.e. the same contrast ($+3.7$ points) under the other convention. Every row is a single training run, so contrasts of a point or less are not resolved by these data.

The differences those numbers support, all computed from the table above and all in points, are the ones quoted in Sec. 5.4:

- complete $-$ baseline: Spatial -1.1, Object $+1.9$, Goal $+3.9$, Long $+8.9$, average $+3.5$;

Table 9: **Component ablation under matched, in-distribution training.** The four variants, per suite; *Avg.* is the arithmetic mean over the four suites. Pooled over task instances the complete model is 85.7 against 82.0 for the baseline (Table 1), the same contrast (+3.7 points) under the other convention. Bold marks the best value of a column; every row is a single training run, so contrasts of a point or less are not resolved.

Variant	Spatial	Object	Goal	Long	Avg.
VLA-Adapter (reactive baseline)	90.6	83.6	77.4	76.6	82.0
DreamIF w/o IADVR (full-frame target)	90.1	84.6	80.5	79.2	83.6
DreamIF w/o DF (no cross-attention)	**91.5**	85.1	80.9	79.5	84.2
DreamIF (complete)	89.5	**85.5**	**81.3**	**85.5**	**85.5**

- w/o IADVR − baseline: Spatial -0.5, Object $+1.0$, Goal $+3.1$, Long $+2.6$, average $+1.6$;
- complete − w/o IADVR: Spatial -0.6, Object $+0.9$, Goal $+0.8$, Long $+6.3$, average $+1.9$;
- complete − w/o DF: Spatial -2.0, Object $+0.4$, Goal $+0.4$, Long $+6.0$, average $+1.3$.

The two knob sweeps are panels (b) and (c) of Fig. 4, and both are read on the Long suite: what each threshold retains of the supervision is shown in Fig. 5(a).

F Real-World Protocol

Platform and asynchronous inference. A 6-DoF AgileX Piper arm with a parallel gripper works on a 150×100 cm table. Two RealSense D415 cameras provide a base view fixed 60 cm in front of the workspace at 45° pitch, and a wrist view that moves with the gripper; both streams are resized and normalised to 224×224. To reconcile model latency with control frequency we use a ZeroMQ-based bidirectional asynchronous bus: the client captures both streams at 30 Hz, preprocesses locally and transmits; the server runs batched inference at 20 Hz and streams back the action chunk together with the predicted physical tensors; the client interpolates the chunk and dispatches 7-dimensional joint commands at a stable 20 Hz.

Fine-tuning and evaluation. Each task is fine-tuned from 50–100 expert demonstrations. Each task is executed 20 times per method. A trial succeeds only if the instructed goal is fully achieved without collision; runs exceeding 60 s or involving unexpected rigid collisions are failures. All three columns of Table 2 follow this protocol exactly, with the same demonstrations, the same trial count and the same success criterion, so the per-task differences are differences between policies and not between evaluation setups. The *Average* row is the arithmetic mean over the eight tasks, and the paired comparison is the one we report: DreamIF is ahead of the reactive baseline on seven of the eight tasks and level on the eighth, and ahead of the latent-predictive baseline on six.

Sub-goal structure and failure modes. Tasks 3 and 6 are the only tasks in the suite that require several sub-goals to be completed in order within one episode. On Task 3 the object must be placed in the pot *before* the lid is grasped, so an error in the first sub-goal is carried into the second; on Task 6 the object has to be carried through the space that the opened door has just reconfigured. Success is recorded per episode, so the numbers in Table 2 do not attribute failures to individual sub-goals: a stage-level breakdown over the whole suite would require instrumenting the trial protocol, and we leave it to future work. Fig. 3 supplies that view for one representative Task 3 episode, where the reactive baseline completes the placement and stops at the lid. What the suite establishes in aggregate is that the two composites are where DreamIF gains the most ground over both baselines, which is what the mechanism predicts.

G Protocol and Result for the Open-Loop Rollout

The open-loop result quoted in Sec. 5.4 is produced as follows. Starting from a single initial observation on the LIBERO-PLUS Long suite, with no further observations and no re-planning, the policy is rolled out for 250 action steps and the resulting joint trajectory is compared against the expert demonstration; the plotted quantity is the mean squared joint-position error per joint, averaged over rollouts. The complete model and the *w/o DF* variant use the same protocol and the same initial observations, so the two curves differ only in whether the predicted intention features are injected into the action head by cross-attention. A 250-step rollout is more than $60\times$ the $K = 4$ prediction



Figure 6: **The real-world task suite.** The eight tasks on the 6-DoF AgileX Piper platform: top row, Tasks 1–4; bottom row, Tasks 5–8, alongside the follower/leader arms and the fixed base camera. Success rates are in Table 2 and rollout sequences for all eight tasks are in Fig. 10.

Table 10: Real-world task definitions and binary success criteria. Tasks 3 and 6 are the articulated multi-step composites: each requires several sub-goals to be completed in order within a single episode, which is what makes them the long-horizon cases of the suite.

ID	Task	Environment setup	Success criterion
1	Upright mug	Mug placed on its side or upside down.	Mug base flush with the table (tilt $< 10^\circ$).
2	Open cabinet door	Arm in front of a closed cabinet.	Door opened $> 45^\circ$ without rigid collision.
3	Pot with lid	Pot and lid separated on the table; object to be placed nearby.	Object placed in the pot, then lid picked and used to cover it.
4	Banana into bowl	Banana randomised, bowl fixed at centre.	Banana rests fully inside the bowl; arm retreats safely.
5	Eggplant into pot	3–5 distractors placed around the pot.	Eggplant inside the pot; no distractor knocked over.
6	Pick into cabinet	Object on the table, closed cabinet beside it.	Object grasped, door opened, object placed inside, door closed.
7	Wipe table	Sponge on the table, target stain marked.	Sponge covers the marked area with ≥ 3 cyclic strokes.
8	Sequential blocks	Red, blue and green blocks provided.	Blocks moved to the target zone in the instructed colour order.

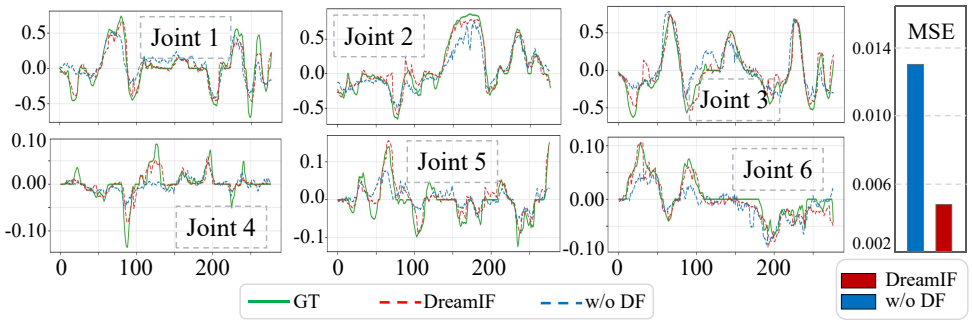

Figure 7: **Long-horizon error behaviour (open-loop rollout, LIBERO-PLUS Long suite).** Per-joint trajectories against the expert (Joint 1–6) and the resulting mean squared joint-position error over a 250-step rollout, conditioned only on the initial observation. Removing the cross-attention binding (w/o DF) raises the error and steepens its growth rate on every joint; the full model holds fidelity over a horizon roughly $60\times$ its $K = 4$ window, which is the empirical answer to whether four steps suffice.

window, which is what makes it a test of whether the short window acts as a persistent constraint rather than as a one-step hint: Fig. 7 shows that it does, with a lower error on all six joints and a markedly slower growth rate once the cross-attention binding is present.

H QUALITATIVE RESULTS

This section collects the rollouts behind the qualitative statements of Secs. 5.2 and 5.3. Three figures cover the three settings of the evaluation: the seven perturbation families of LIBERO-PLUS (Fig. 8), the four task suites of the benchmark (Fig. 9), and the eight real-world tasks on the AgileX Piper platform (Fig. 10). In every panel the natural-language instruction is printed above the sequence it conditions and the frames run in temporal order from the initial observation towards the goal state. The figures are illustrative of behaviour: the rates they illustrate are the ones tabulated in Table 1 and Table 2, and no quantitative claim in the main text rests on a single sequence.

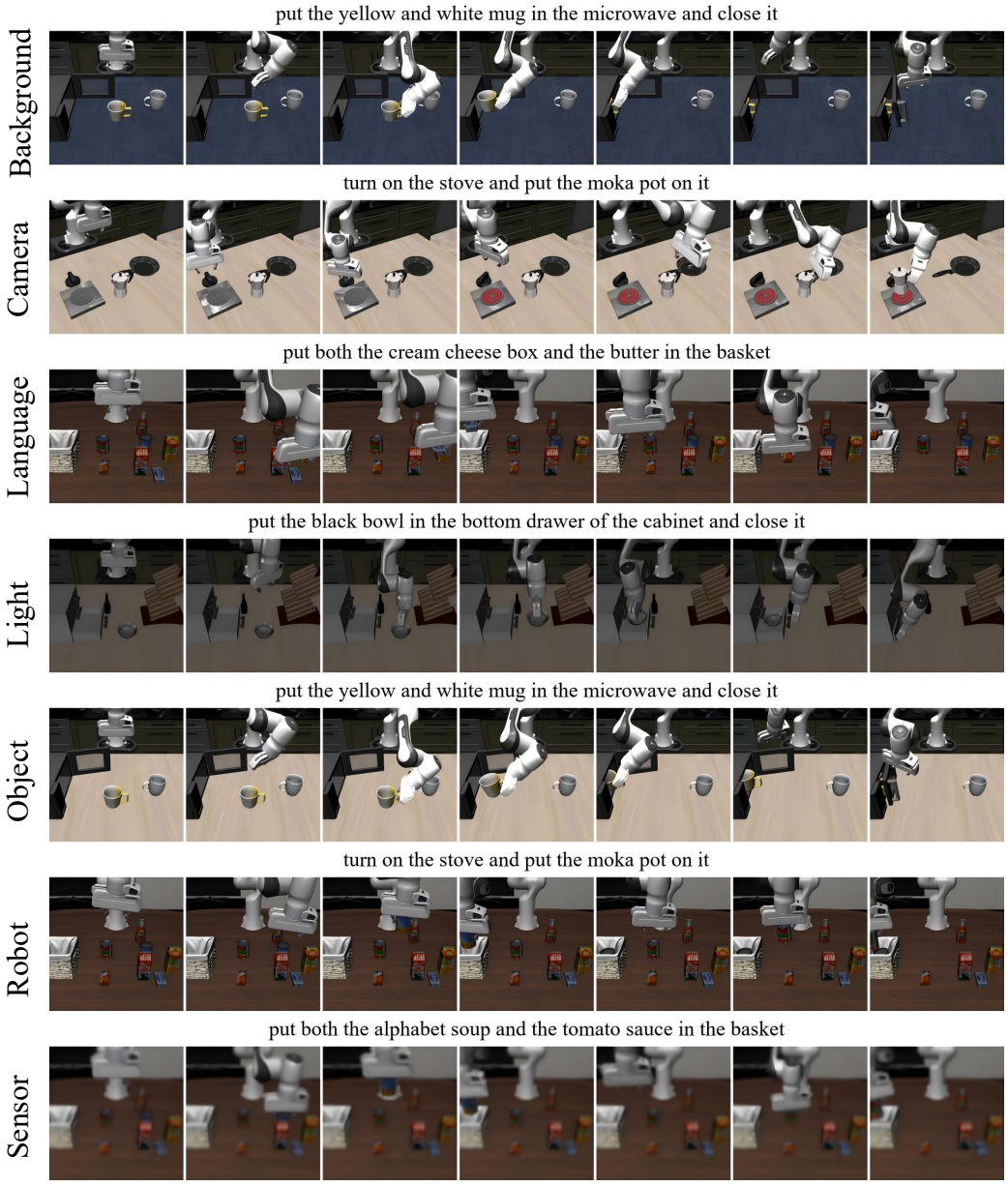

Figure 8: Qualitative rollouts under the seven LIBERO-PLUS perturbations. Each row is one perturbation family: background textures, camera viewpoints, language instructions, light conditions, object layouts, robot initial states, and sensor noise. The instruction is shown above each sequence. The motion-defined target keeps the policy anchored on the interaction region when the surrounding appearance changes.

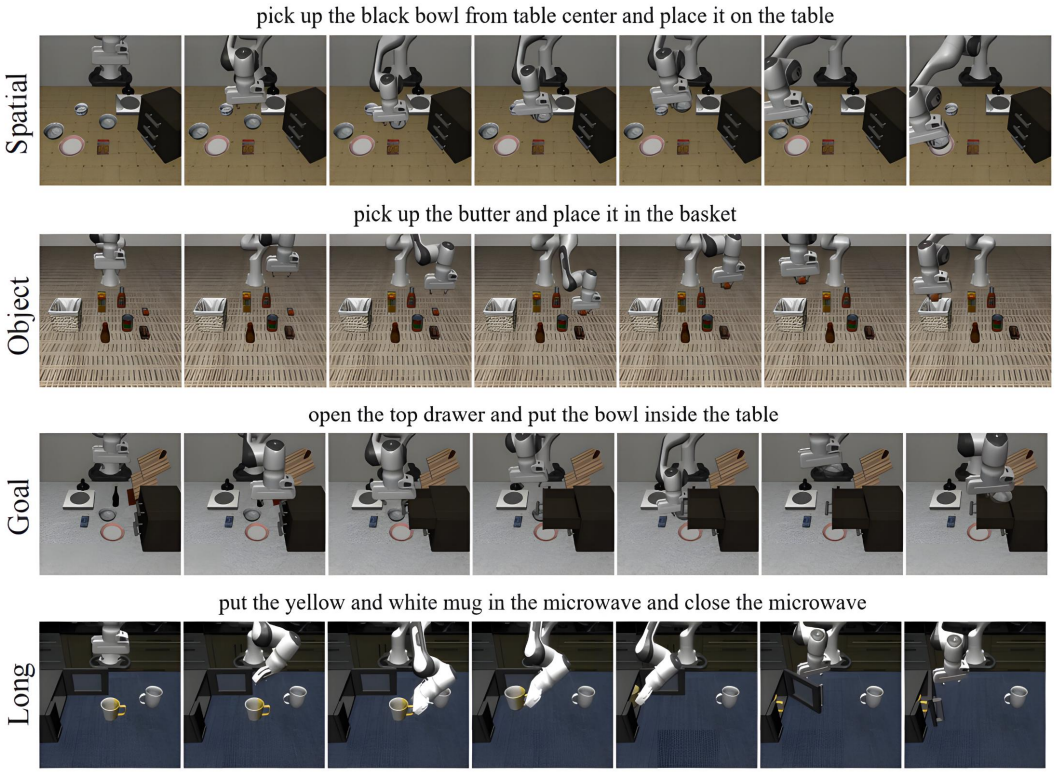

Figure 9: Step-by-step execution on the four LIBERO-PLUS suites (Spatial, Object, Goal, Long), from the initial observation to task completion, with the natural-language instruction shown above each sequence.

The seven rows of Fig. 8 are the seven perturbation families that form the columns of Table 1, one family per row, each with its instruction printed above the sequence. Read against the per-family columns, the rows that change most are the families with the widest margins in Sec. 5.2: over the reactive baseline, DreamIF leads by 7.6 points on Language, 7.9 on Layout, 4.7 on Robot initial state and 3.8 on Sensor noise. What those four share is that they move the object the action operates on and its relation to the scene rather than only the appearance of the image, so a target defined by motion stays on the region the current action is about to change: the instruction is reworded, the object starts elsewhere, the arm begins from a different pose, and the execution still converges on the same interaction region.

Fig. 9 shows step-by-step execution on the four task suites of LIBERO-PLUS — Spatial, Object, Goal and Long — from the initial observation to task completion. The four suites are the ones over which the seven perturbation families are applied, so the two figures cover the benchmark from complementary directions: in Fig. 8 the task is held fixed and the environment changes around it, while here the suite changes and with it the horizon the policy has to plan over. The Long suite is included because it is where the mechanism matters most, gaining **8.9 points** over the reactive baseline in the ablation of Sec. 5.4; the per-suite values of all four variants are tabulated in Appendix E.

Fig. 10 covers the real-world suite of Sec. 5.3 on the 6-DoF AgileX Piper platform, one sequence per task, with the instruction printed above it and the criterion it has to meet in Table 10. The eight tasks are read in three groups: foundational manipulation (Tasks 1–3), grasping under clutter (Tasks 4–5) and long-horizon interaction (Tasks 6–8), the last being where the margin over the reactive baseline is widest, at **20.0 points**. Tasks 3 and 6 are the articulated composites that require several sub-goals to be completed in order within one episode, and Fig. 3 gives the stage-level contrast on one Task 3 episode, where the reactive baseline completes the placement and then stops at the lid. Because success is scored per episode (Appendix F), these sequences show what the policy does while a task is carried out; how often each criterion of Table 10 is met over the 20 trials per task is the *Average* row of Table 2.



Figure 10: Real-world rollouts on the 6-DoF AgileX Piper platform for Tasks 1–8, spanning foundational manipulation (e.g. opening a cabinet) through long-horizon sequential interaction (e.g. wiping a table). Instructions are shown above each sequence; the articulated multi-step composites are Tasks 3 and 6 (Table 10).

Read together, the three figures are the visual record of the protocol rather than a fourth measurement: the seven perturbation families correspond to the columns of Table 1, the four suites to the per-suite values of Appendix E, and the eight tasks to the rows of Table 2. The tables state how often each setting is solved and the sequences show the executions behind them; where a single episode is compared between two policies, that comparison is Fig. 3 in the main text.