ASYNCMOT: INTERACTION-GUIDED ASYNCHRONOUS MIXTURE OF TRANSFORMERS FOR GENERALIZED AND ROBUST ROBOTIC MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Robotic manipulation involves interaction contexts that persist across control steps but change at irregular, interaction-dependent moments. Yet existing policies typically couple interaction reasoning to control frequency or predefined task structure, creating a mismatch between when robots act and when they need to reason. We introduce **AsyncMoT**, an interaction-guided asynchronous Mixture of Transformers that makes *when to reason* an explicit policy decision. Given a high-level task instruction, AsyncMoT maintains a structured interaction state capturing the current stage, target object, and action-relevant region, and selectively refreshes it as the interaction evolves. Between updates, cached Key-Value features preserve interaction guidance across control steps, while the action expert continuously adapts motor commands to current visual and proprioceptive observations. This decoupling further enables interaction supervision to be shared across heterogeneous robot tasks and human videos, without requiring human-to-robot action correspondence. AsyncMoT achieves average success rates of **98.1%** on LIBERO and **88.9%** on LIBERO-Plus, **66.3%** on RoboCasa-24 and **35.9%** on RoboCasa365, demonstrating strong performance across long-horizon, distribution-shifted, and compositional manipulation settings.

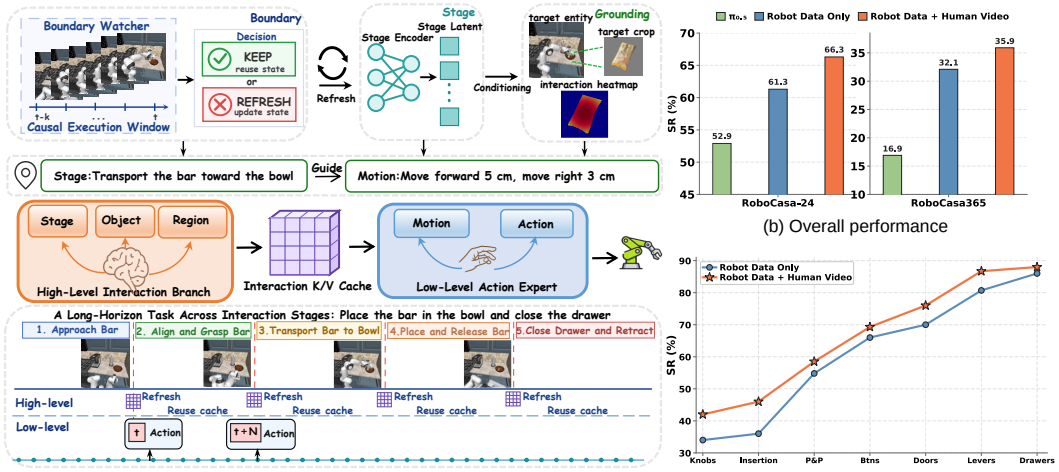
1 INTRODUCTION

Long-horizon robotic manipulation unfolds through a sequence of physical interactions, each centered on a particular entity (Pan et al., 2025; Zhu et al., 2025) and a spatial region (Huang et al., 2025a; Yuan et al., 2024) relevant to the ongoing action. While low-level observations and actions continuously evolve during execution, the underlying interaction often remains stable across many control steps (Zhang et al., 2025; Zou et al., 2025; Yan et al., 2026). Yet such interaction context is persistent rather than static, as the interaction evolves, the previously retained context can become outdated and misguide subsequent actions. This raises a central question, how should a manipulation policy *maintain interaction context* and determine when to *update it*?

Existing manipulation policies incorporate different interaction structures. Some decompose behavior into temporal segments or skills to organize demonstrations for policy learning (Chen et al., 2025b; Sanchez et al., 2026), while others introduce intermediate representations, *i.e.*, affordances, semantic subgoals, or high-level trajectories, to guide downstream action generation (Li et al., 2025c; Zhang et al., 2026; Kong et al., 2026). However, they primarily structure *what* task-level information should be abstracted for the controller, while temporal validity is typically implicit or tied to predefined task decompositions. We instead argue that interaction context should follow the temporal dynamics of physical interaction, rather than by control frequency or predefined task structure.

Crucially, physical interaction does not evolve on a fixed clock. Consider a robot approaching a drawer handle, its observations, poses, and motor commands change continuously, while the underlying interaction context may remain stable across many control steps. Whereas, physical events such as contact or target displacement can abruptly invalidate the retained context, without necessarily corresponding to a change in high-level behavior. Interaction reasoning should therefore be governed by the interaction itself. This motivates an *interaction-guided asynchronous* formulation that selectively updates interaction context while maintaining continuous low-level control.

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.



(a) Overview of AsyncMoT (b) Overall performance (c) Performance from human videos

Figure 1: (a) AsyncMoT maintains a persistent interaction state for long-horizon control and refreshes it only when physical execution makes the retained context outdated, while low-level control proceeds continuously. (b) AsyncMoT substantially improves performance on RoboCasa-24 (Nasiriany et al., 2024) and RoboCasa365 (Nasiriany et al., 2026). (c) Human-video supervision consistently improves performance, demonstrating interaction-level knowledge transfer across embodiments.

To this end, we introduce **AsyncMoT**, an *interaction-guided asynchronous Mixture of Transformers* that makes *when to reason* an explicit decision of the policy. At its core is a lightweight **Boundary Watcher** that continuously makes a binary KEEP/REFRESH decision by comparing recent execution with the retained interaction context. KEEP directly reuses the interaction-level key-value memory, whereas REFRESH activates the interaction Transformer to update it, allowing expensive interaction reasoning to be invoked by physical interaction rather than a predefined computational schedule. The retained memory encodes complementary **Stage**, **Object**, and **Region** representations, respectively capturing the current interaction semantics, interacting entity, and action-relevant spatial grounding, and is directly attended to by the action expert at every control step. In this way, AsyncMoT couples persistent interaction memory with continuously updated visuomotor features, turning MoT from a mixture of specialized computations into a mixture that is also *asynchronously activated* according to the evolving interaction.

The resulting interaction-guided asynchronous policy exhibits strong generalization and robustness across diverse manipulation settings. Across interaction-rich manipulation skills, our AsyncMoT reaches 66.3% on RoboCasa-24 (Nasiriany et al., 2024) (vs. 52.9% for $\pi_{0.5}$ (Black et al., 2025a)); when being tested over composed longer horizons, it achieves 35.9% on RoboCasa365 (Nasiriany et al., 2026) (vs. 21.9% for GR00T N1.6 (NVIDIA GEAR Team, 2025)). The gains extend to controlled distribution shifts on LIBERO-Plus (Fei et al., 2026) (88.9%) and history-dependent manipulation on RoboMME (Dai et al., 2026) (28.13%), showing robustness to changing observations and execution context. Moreover, ablations further support the benefit of asynchronous interaction timing: interaction-driven updates achieve 98.1% on LIBERO (Liu et al., 2023), outperforming per-step reconstruction by 6.3% and fixed-rate updates by 7.2%-7.8%, showing that *when* interaction reasoning is invoked matters beyond simply *how often*. Finally, heterogeneous human videos provide interaction supervision without human-to-robot action correspondence, further improving RoboCasa-24 (Nasiriany et al., 2024) and RoboCasa365 (Nasiriany et al., 2026) by 5.0% and 3.8%. Our main contributions are summarized as:

- We introduce AsyncMoT, an interaction-guided asynchronous Mixture of Transformers that makes *when to reason* an explicit policy decision, while keeping continuous action generation.
- We develop a persistent interaction memory capturing Stage, Object, and Region, with decision-gated KEEP/REFRESH updates and reusable KV states for interaction-conditioned control.
- We demonstrate strong generalization and robustness across diverse, compositional, distribution-shifted, and history-dependent tasks, with further boosts from human-video supervision.

2 Related Work

Temporal Abstraction and Asynchronous VLA Policies. Robot policies use temporal abstraction to reduce the frequency of high-level decisions, through action chunking (Zhao et al., 2023) or hierarchical control (Belkhale et al., 2024; Zhang et al., 2025). Recent VLA models further decouple reasoning and action computation using Mixture-of-Transformers (MoT), fast–slow pathways, or heterogeneous-frequency experts (Liang et al., 2025; Huang et al., 2025b; Liu et al., 2026; Cai et al., 2026; Lu et al., 2026; Yu et al., 2026). Other approaches asynchronously reuse slower semantic processing or trigger explicit reasoning at selected task-progress-dependent keyframes (Zou et al., 2025; Yan et al., 2026; Li et al., 2026c). AsyncMoT instead determines when to refresh the interaction state from the evolving physical interaction, rather than from a fixed frequency or predefined task-progress structure.

Structured Representations for Robot Interaction. Intermediate representations bridge high-level task understanding and low-level robot execution. Prior work uses language or compositional abstractions (Hancock et al., 2026; Li et al., 2026b; Tang et al., 2026), grounds task-relevant objects and regions (Huang et al., 2025a; Pan et al., 2025), or predicts affordances and action-oriented representations before control (Li et al., 2025a; Zhong et al., 2026; Tan et al., 2026). AsyncMoT additionally models when a persistent interaction representation should be refreshed, separating interaction content from its update timing and allowing it to persist across multiple motor steps.

Human-Video and Cross-Embodiment Learning. Human videos provide scalable manipulation experience but differ from robot demonstrations in embodiment and action space. Prior work learns transferable or latent actions from videos (Chen et al., 2025a; Bu et al., 2025), aligns human and robot behaviors in shared latent spaces (Luo et al., 2026; Chen et al., 2026), or learns embodiment-invariant representations and cross-embodiment policies (Jiang et al., 2026; Zheng et al., 2026; Zha et al., 2026). AsyncMoT instead uses human videos to supervise interaction-level structure shared across embodiments, while keeping executable low-level control robot-specific.

3 Method

Overview. AsyncMoT models interaction context and motor execution at different temporal scales. While visual observations, proprioceptive states, and motor actions evolve at every control step, interaction context can persist across multiple steps and is refreshed only when ongoing physical execution makes it outdated. A Boundary Watcher determines when to refresh this context, while an interaction branch represents it through complementary Stage, Object, and Region features that are cached and reused by the action expert between refreshes (Figure 2). Let t_k denote the k-th interaction refresh, at which the interaction branch produces $\mathcal{I}_k = (S_k, O_k, R_k)$. This state remains active until the next refresh while low-level perception and action generation continue at every step. Importantly, k indexes refresh events rather than semantic Stages, so consecutive refreshes may preserve the same Stage semantics while re-grounding the interaction context.

3.1 Persistent Interaction State and Asynchronous Refresh

At each refresh time t_k, the interaction branch processes the current observation o_{t_k} together with the task instruction l and constructs the interaction state used by subsequent control steps. Stage captures the semantic context of the ongoing interaction, Object identifies the involved entity, and Region grounds its action-relevant spatial support. Together, they specify the content of the retained interaction state, while the Boundary Watcher determines when the retained state should be refreshed.

Stage captures the semantic phase or intent of the ongoing interaction as a continuous representation. For each boundary-delimited interval, its textual Stage description is encoded into a semantic target $S_k^\star$, while a learnable Stage query attends to the multimodal features to predict $\hat{S}_k$. Stage also provides semantic context for the downstream Object and Region representations and for the movement supervision used by the action expert. Because Stage describes interaction semantics rather than refresh timing, the same Stage may persist across consecutive refresh intervals.

Object identifies the entity involved in the current interaction. Its target $O_k^\star$ is constructed from the annotated interaction object while suppressing unrelated background. Object queries attend to the current visual features and are conditioned on Stage, so that object identification is tied to the current interaction semantics rather than visual saliency alone.

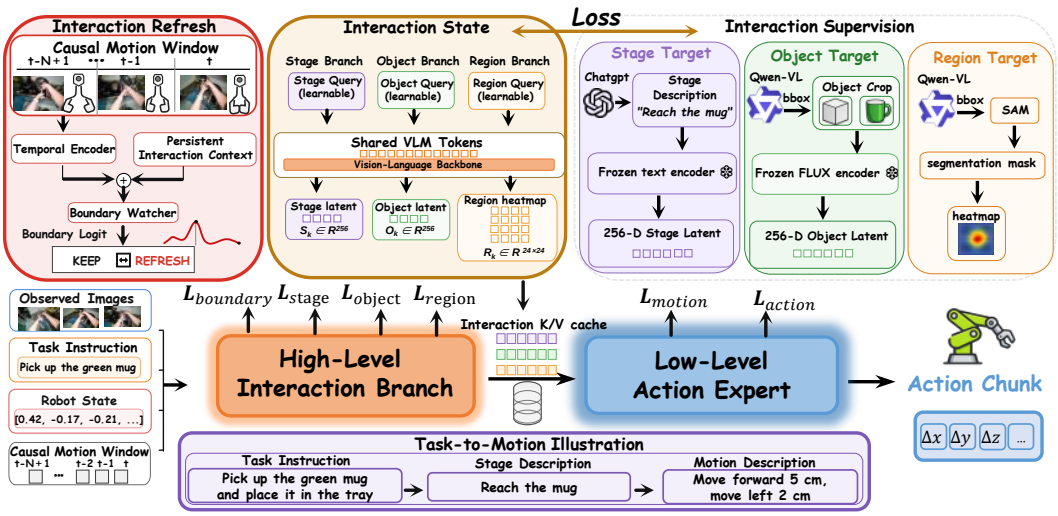


Figure 2: Overview of AsyncMoT. A lightweight Boundary Watcher determines when the retained interaction state should be refreshed. At each refresh, the interaction branch extracts Stage, Object, and Region representations and updates the corresponding K/V cache. Between refreshes, the action expert reuses the cached interaction features while continuously generating actions from the latest visual and proprioceptive observations.

Region specifies the action-relevant spatial support of the current interaction. Rather than reducing this support to a point or bounding box, we represent it as a continuous heatmap $R_k^\star$. Region queries attend to visual features and are conditioned jointly on Stage and Object. This allows the spatial grounding to adapt as physical execution evolves, including when the high-level *Stage* is steady. We supervise the three representations with

$$\mathcal{L}_{\text{stage}} = \left\| \text{norm}(\hat{S}_k) - \text{norm}(S_k^\star) \right\|_2^2, \quad \mathcal{L}_{\text{object}} = \text{MSE}(\hat{O}_k, O_k^\star),$$
$$\mathcal{L}_{\text{region}} = \mathcal{L}_{\text{bal-BCE}}(\hat{R}_k, R_k^\star) + 0.5\,\mathcal{L}_{\text{Dice}}(\hat{R}_k, R_k^\star). \tag{1}$$

Boundary-driven refresh. The interaction state is not recomputed at every control step. Instead, the Boundary Watcher determines whether the retained interaction context remains valid for the current physical execution. It combines a temporal execution context u_t, which summarizes the latest N control steps, with a persistent interaction context c_t derived from the most recently refreshed Stage, Object, and Region representations.

The temporal context is produced by a lightweight temporal encoder over recent observations and robot execution states, while the persistent context remains fixed between consecutive refreshes. The Boundary Watcher predicts a boundary logit z_t^B and is trained with

$$\mathcal{L}_{\text{boundary}} = \text{BCEWithLogits}\left(z_t^B, b_t; w_+\right), \tag{2}$$

where $b_t \in \{0, 1\}$ is the Boundary target and w_+ compensates for sparse refresh events. At inference, KEEP reuses the retained interaction state, whereas REFRESH reruns the interaction branch on the latest observation and updates the interaction cache. Thus, a Boundary represents a change in the temporal validity of the retained context rather than an explicit Stage transition.

3.2 Interaction-Conditioned Action Generation

Persistent interaction guidance. The action expert continuously processes the latest visual and proprioceptive observations while reusing the interaction features produced at the most recent refresh. Let $\tau(t)$ denote the latest refresh before control step t. At each step, current execution features are concatenated with the cached interaction features, $K_t = [K_t^{\text{act}}; K_{\tau(t)}^{\text{int}}]$ and $V_t = [V_t^{\text{act}}; V_{\tau(t)}^{\text{int}}]$. Here, $(K_t^{\text{act}}, V_t^{\text{act}})$ are recomputed from the current execution state at every control step, while $(K_{\tau(t)}^{\text{int}}, V_{\tau(t)}^{\text{int}})$ are reused until the next interaction refresh. This separates persistent interaction guidance from continuously changing execution features, allowing the policy to retain relevant interaction context without sacrificing responsiveness to current observations.

Movement direction and displacement supervision. We further supervise the action representation with movement direction and displacement derived from each future action trajectory. For an action chunk $a_{t:t+H}$, we summarize its translational and rotational motion into a structured target $m_{t:t+H}$ describing the corresponding movement directions and displacements. The current Stage provides semantic context for this supervision, associating the motion structure of the future trajectory with the interaction being executed. We optimize

$$\mathcal{L}_{\text{motion}} = -\sum_{j} \log p_\theta \left(m_{t:t+H,j} \mid o_t, l, S_{\tau(t)}, m_{t:t+H,<j} \right). \tag{3}$$

This objective provides auxiliary supervision on the action representation, while the original continuous-action objective $\mathcal{L}_{\text{action}}$ remains responsible for executable motor generation. Stage conditions the semantic structure of this auxiliary target but does not determine when the retained interaction state is refreshed.

Further implementation details of the asynchronous refresh and action-conditioning mechanisms, together with a notation reference (Table 6), are provided in Appendix A.

3.3 Two-Stage Training

AsyncMoT adopts a two-stage training strategy that first establishes the structured interaction representation and then couples it with asynchronous refresh and robot control. In Stage-1, we use robot trajectories from DROID (Khazatsky et al., 2024) and RH20T (Fang et al., 2024), together with human interaction data from HOI4D (Liu et al., 2022) and Open-AoE (Li et al., 2026d). In Stage-2, we retain DROID, RH20T, and HOI4D, while excluding Open-AoE after the learned interaction representation is coupled with our single-manipulator action formulation. Both stages maintain a Robot:Human sampling ratio of 2:1.

Stage-1: Interaction representation learning. We first train the interaction branch to establish Stage, Object, and Region representations without simultaneously learning refresh timing or embodiment-specific motor execution. Since these targets describe interaction-level structure rather than robot actions, this stage can naturally leverage heterogeneous robot and human interaction data. Boundary supervision is intentionally excluded: Stage-1 learns what interaction information should be represented, whereas Boundary supervision is introduced in Stage-2 to learn when the retained representation should be refreshed. The objective is

$$\mathcal{L}_{\text{S1}} = \lambda_S \mathcal{L}_{\text{stage}} + \lambda_O \mathcal{L}_{\text{object}} + \lambda_R \mathcal{L}_{\text{region}}. \tag{4}$$

Stage-1 therefore establishes the structured interaction representation before coupling it to asynchronous refresh and robot control.

Stage-2: Interaction-conditioned action learning. We then couple the learned interaction representation with robot action generation and introduce Boundary supervision to learn when the retained state should be refreshed during execution. For robot trajectories, we additionally supervise movement direction and displacement under the current Stage, while retaining the original continuous-action objective for executable motor generation:

$$\mathcal{L}_{\text{S2}} = \mathcal{L}_{\text{action}} + \lambda_M \mathcal{L}_{\text{motion}} + \lambda_S \mathcal{L}_{\text{stage}} + \lambda_O \mathcal{L}_{\text{object}} + \lambda_R \mathcal{L}_{\text{region}} + \lambda_B \mathcal{L}_{\text{boundary}}. \tag{5}$$

Human-video samples retain interaction-level supervision but omit the action-specific objectives. Stage-2 therefore learns when the retained interaction context should be refreshed and how that context should guide embodiment-specific motor execution.

Automatic interaction annotation. We derive Boundary, Stage, Object, and Region supervision for robot trajectories from execution cues and multimodal models. Boundary targets are determined from physical execution independently of semantic Stage labels. A text LLM assigns Stage semantics to each boundary-delimited interval, while a VLM identifies the interacting Object and grounds its action-relevant Region. Adjacent intervals are not required to have different Stage descriptions, so refresh supervision does not require a Stage change. Human videos use their existing interaction-level supervision. Details are provided in Appendix B.

4 Experiments

Benchmarks. We evaluate AsyncMoT using five simulation benchmarks, each testing a distinct property of interaction-centric control: LIBERO (Liu et al., 2023) on standard long-horizon manipulation;

Table 1: Evaluation results on **LIBERO** and **LIBERO-Plus**. Success rates are reported in %. Orig. and Plus denote the reported overall scores on LIBERO and LIBERO-Plus, respectively. Parentheses indicate the absolute drop from Orig. to Plus in percentage points. Blue highlights denote the top three available results within each performance column: first, second, and third.

Method	Size	Orig.	Plus	Cam.	Robot	Lang.	Light	BG	Noise	Layout
$\pi_{0.5}$ (Black et al., 2025a)	3.3B	96.9	85.7 (↓11.2)	75.4	77.5	85.6	96.9	94.6	89.7	85.7
OpenVLA-OFT (Kim et al., 2025a)	7B	96.8	69.6 (↓27.2)	56.4	31.9	79.5	88.7	93.3	75.8	74.2
RIPT-VLA (Tan et al., 2025)	7B	97.5	68.4 (↓29.1)	55.2	31.2	77.6	88.4	91.6	73.5	74.2
π_0 (Black et al., 2025b)	3.3B	94.2	53.6 (↓40.6)	13.8	6.0	58.8	85.0	81.4	79.0	68.9
π_0-FAST (Pertsch et al., 2025)	3.3B	85.5	61.6 (↓23.9)	65.1	21.6	61.0	73.2	73.3	74.4	68.8
UniVLA (Bu et al., 2025)	8.5B	95.2	43.9 (↓51.3)	1.8	46.2	69.6	69.0	81.0	21.2	31.9
MemoryVLA (Shi et al., 2026)	—	96.5	70.2 (↓26.3)	42.7	44.9	84.4	92.8	95.0	62.1	84.7
LAP (Zha et al., 2026)	—	96.3	75.2 (↓21.1)	65.2	52.4	84.3	93.4	91.7	74.4	75.3
NORA (Hung et al., 2025)	3B	87.9	39.0 (↓48.9)	2.2	37.0	65.1	45.7	58.6	12.8	62.1
OpenVLA (Kim et al., 2025b)	7B	76.5	15.6 (↓60.9)	0.8	3.5	23.0	8.1	34.8	15.2	28.5
$\pi_{0.5}$+Spatial Forcing (Li et al., 2026a)	3.3B	94.0	25.7 (↓68.3)	0.1	0.3	26.8	66.0	45.9	0.1	59.8
Cosmos-Policy (Kim et al., 2026)	2B	98.5	82.4 (↓16.1)	73.4	63.3	89.3	98.9	83.5	89.3	84.0
Fast-WAM (Yuan et al., 2026)	6B	97.6	50.0 (↓47.6)	16.4	44.5	68.9	78.2	53.7	37.7	60.7
WorldVLA (Cen et al., 2025)	7B	79.1	25.0 (↓54.1)	0.1	27.9	41.6	43.7	17.1	11.0	38.0
AsyncMoT (Ours)	—	**98.1**	**88.9** (↓9.2)	**88.2**	**77.4**	**90.6**	**98.4**	**97.3**	**87.4**	**88.2**

LIBERO-Plus (Fei et al., 2026) on robustness under distribution-shifted observations; the 24-task RoboCasa benchmark (Nasiriany et al., 2024) on diverse manipulation skills; RoboCasa365 (Nasiriany et al., 2026) on task recomposition; and RoboMME (Dai et al., 2026) on history-dependent control. We follow the standard protocol to report task success rate (%). Detailed configurations and evaluation protocols are provided in Appendix C.

Baselines. We compare against representative visuomotor and VLA policies, including OpenVLA (Kim et al., 2025b), OpenVLA-OFT (Kim et al., 2025a), π_0 (Black et al., 2025b), $\pi_{0.5}$ (Black et al., 2025a), GR00T variants (NVIDIA et al., 2025), LAP (Zha et al., 2026), and the memory-augmented SAM2Act+ (Fang et al., 2025). Since method availability differs across benchmark protocols, each result table reports the applicable comparison set.

Implementation Details. We use PaliGemma-3B (Beyer et al., 2024) as the vision-language backbone and follow the two-stage training procedure described in Section 3.3. For downstream evaluation, we adapt the policy using the corresponding benchmark training demonstrations while keeping the AsyncMoT architecture and training recipe fixed across benchmarks.

4.1 Main Results

We organize the main evaluation around three questions: whether interaction modeling supports sustained control across long horizons and diverse manipulation skills, whether the learned interaction structure remains useful under observation changes and task recomposition, and whether persistent interaction context benefits settings in which successful control depends on earlier execution. Together, these benchmarks probe the persistence, robustness, and reuse of interaction representations from complementary perspectives.

Sustained control across horizons and skills. LIBERO and RoboCasa-24 test two complementary aspects of interaction-centric control. On LIBERO, the benefit is most informative on the Long split, where successful execution requires maintaining coherent interaction context over an extended horizon; the shorter suites are already close to saturation for strong VLA baselines. RoboCasa-24 probes a different dimension of generality: the gains extend across all seven manipulation categories rather than concentrating on a particular skill. Together, these results suggest that the structured interaction representation provides a reusable control context across both temporal scale and skill diversity, rather than serving as a task-specific intermediate representation.

Accelerating downstream adaptation. The fine-tuning behavior on RoboCasa-24 provides additional evidence that the learned interaction structure is reusable rather than benchmark-specific. AsyncMoT reaches its high-performance regime substantially earlier than $\pi_{0.5}$, and human-video supervision further accelerates adaptation (Fig. 3). This suggests that downstream learning does not

Table 2: Category-wise results on the 24-task RoboCasa benchmark (Nasiriany et al., 2024) with success rate (%). AsyncMoT outperforms prior methods. Furthermore, involvement of heterogeneous human videos yields further performance gains of 5.0%.

Method	P&P	Doors	Drawers	Knobs	Levers	Buttons	Insertion	Avg.
SIR (Mattes et al., 2026)	0.13	30.25	46.25	16.50	48.50	21.83	4.75	19.5
BC-Transformer (Wei et al., 2026)	3.75	39.50	61.00	18.00	47.33	60.00	11.00	28.8
GWM (Lu et al., 2025)	14.75	47.75	68.00	34.00	56.00	70.00	26.00	39.2
UVA (Li et al., 2025b)	-	-	-	-	-	-	-	50.0
π_0 (Black et al., 2025b)	24.0	45.0	78.0	29.0	57.3	57.3	0.0	38.8
OpenVLA-OFT (Kim et al., 2025a)	21.8	45.7	59.0	8.0	36.0	56.0	27.0	33.1
GeoPredict (Qian et al., 2026)	22.7	69.1	87.2	36.6	79.3	78.4	38.8	52.4
$\pi_{0.5}$ (Black et al., 2025a)	54.3	51.0	75.0	28.0	79.3	60.0	4.0	52.9
LAP (Zha et al., 2026)	38.3	54.5	75.0	19.0	63.3	58.7	15.0	46.2
Ours	**54.8**	**70.0**	**86.0**	**34.0**	**80.7**	**66.0**	**36.0**	**61.3**
Ours+Human Videos	**58.5**	**76.0**	**88.0**	**42.0**	**86.7**	**69.3**	**46.0**	**66.3**

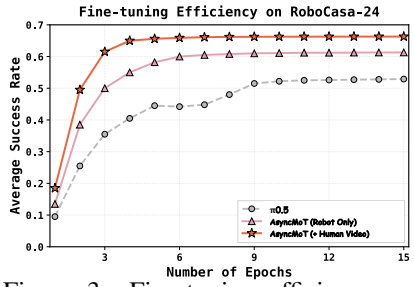


Figure 3: Fine-tuning efficiency on RoboCasa-24 (Nasiriany et al., 2024). AsyncMoT requires substantially fewer fine-tuning epochs, and benefits from human-video supervision.

need to reconstruct interaction structure entirely from robot demonstrations; instead, adaptation can focus on grounding an already learned interaction representation into embodiment-specific control.

Distribution shift and task recomposition. LIBERO-Plus and RoboCasa365 probe two fundamentally different forms of change. LIBERO-Plus perturbs viewpoint, initialization, appearance, noise, and layout, which can substantially alter the current observation without necessarily changing the underlying interaction. The consistent robustness across these shifts suggests that persistent interaction context provides a stable reference against transient visual variation, rather than forcing high-level interaction information to be reconstructed from every new observation. **RoboCasa365** poses the complementary challenge: interaction structure must be reused when familiar behaviors are recomposed into longer and previously unseen task configurations. The advantage of AsyncMoT is especially pronounced on the composite splits, supporting the view that the learned interaction representation captures reusable structure rather than memorizing a fixed task sequence. Taken together, LIBERO-Plus and RoboCasa365 show two desirable properties of the same representation: it can remain stable under changes that should not invalidate the ongoing interaction, while still supporting reuse when interaction structure must be recomposed.

Table 3: Results on RoboCasa365. Success rate (%) is reported for atomic and seen/unseen compositions of skills.

Method	Atomic-Seen	Compos.-Seen	Compos.-Unseen	Overall
π_0	36.3	5.2	0.7	15.0
π_0-FAST	51.7	8.0	1.8	21.7
$\pi_{0.5}$	39.6	7.1	1.2	16.9
GR00T N1.5 (NVIDIA et al., 2025)	50.7	14.8	2.7	23.9
GR00T N1.6 (NVIDIA GEAR Team, 2025)	51.1	9.4	1.7	21.9
RLDX-1 (Zha et al., 2026)	67.3	19.0	5.6	32.1
LAP (Zha et al., 2026)	38.0	4.5	0.6	15.3
Ours	**63.7**	**30.4**	**10.1**	**35.9**

History-dependent control. RoboMME tests a qualitatively different requirement: information needed for the current decision may depend on events from earlier execution and cannot always be recovered from the current observation alone. The consistent gains across its memory-oriented tasks therefore provide direct evidence for the value of persistent interaction context. This result complements LIBERO-Plus from the opposite direction: LIBERO-Plus asks whether retained context can remain stable despite changes in the current observation, whereas RoboMME asks whether information from earlier execution remains available when current control depends on it. Together, they support persistence as a functional property of the interaction representation rather than merely a computational mechanism for avoiding recomputation.

Table 4: Results on RoboMME with task success rate (%).

Method	Counting	Permanence	Reference	Imitation	Avg.
$\pi_{0.5}$ (Black et al., 2025a)	28.78	17.00	17.17	8.78	17.93 ± 0.61
SAM2Act+ (Fang et al., 2025)	35.33	26.00	16.83	7.33	21.37 ± 1.82
LAP (Zha et al., 2026)	21.17	11.33	10.17	5.83	12.13 ± 0.72
Ours	**35.83**	**29.83**	**31.67**	**15.17**	28.13 ± 0.31

Taken together, the main results support three complementary properties of the interaction-centric formulation: interaction structure remains useful across long horizons and diverse skills, persists through observation changes while supporting task recomposition, and carries information that

Table 5: Ablation study on LIBERO. Q1 asks whether *when* to refresh matters beyond update frequency, Q2 examines what interaction information is important for control, and Q3 examines how interaction representation and control should be learned. Each row changes only the named design choice.

	Configuration	Spatial	Object	Goal	Long	Avg.
Q1	Initial-only	72.4	75.2	70.8	49.6	67.0
	Fixed Refresh (1 Hz)	92.6	94.0	92.2	84.8	90.9
	Fixed Refresh (2 Hz)	92.0	93.4	91.8	84.0	90.3
	Every-step Reconstruction	93.4	94.6	93.2	86.0	91.8
	Boundary-driven Refresh (Ours)	**98.8**	**99.2**	**98.6**	**95.8**	**98.1**
Q2	Frozen Upper VLM	95.4	96.4	94.2	88.4	93.6
	w/o Stage	95.6	96.6	95.0	88.0	93.8
	w/o Object	97.0	97.4	96.8	92.6	96.0
	w/o Region	94.8	96.4	95.4	89.8	94.1
	w/o Motion Supervision	97.0	97.8	96.4	91.2	95.6
	Full (Ours)	**98.8**	**99.2**	**98.6**	**95.8**	**98.1**
Q3	w/o Stage-1	93.8	95.2	94.4	85.6	92.3
	w/o Stage-2	88.6	90.8	89.8	75.8	86.3
	Full Two-Stage Training (Ours)	**98.8**	**99.2**	**98.6**	**95.8**	**98.1**

remains useful when control depends on earlier execution. We next use controlled ablations to isolate *when* this context should be refreshed, what information it should retain, how it should be learned, and whether interaction-level supervision transfers across embodiments.

4.2 Ablation Study

We organize the ablations around three questions: Q1 whether *when* interaction context is refreshed matters beyond how often it is updated, Q2 what interaction information is important for control, and Q3 how interaction representation and control should be learned. We further examine cross-embodiment transfer through human-video supervision.

Q1: When to Refresh Matters More Than How Often. The gain comes from choosing *when* to refresh, rather than simply refreshing more frequently. Increasing the fixed update rate does not improve performance, and even every-step reconstruction remains clearly behind boundary-driven refresh, especially on Long. Together with the large drop from retaining only the initial state, this shows that neither static persistence nor frequent recomputation is sufficient. The useful behavior lies between these two extremes: interaction context should persist while it remains valid, but be refreshed when physical execution makes the retained context outdated.

Q2: Long-Horizon Control Requires More Than Object Identity. The largest degradations on Long come from weakening the semantic and spatial structure of the interaction representation, whereas removing Object alone has a noticeably smaller effect. This suggests that long-horizon control is not mainly limited by recognizing *what* object is being manipulated. Instead, the policy must maintain the evolving interaction context—what it is currently doing, where the action should be grounded, and how that context relates to the ongoing motion. The growing gap on Long therefore supports structured interaction context as a mechanism for maintaining control coherence over extended execution.

Q3: Interaction Representations Must Be Grounded in Execution. The two training stages are complementary, but Stage-2 is especially critical. Stage-1 provides a structured interaction representation, while Stage-2 turns that representation into an executable interface by coupling it with refresh timing and motor control. The stronger degradation without Stage-2 shows that learning meaningful Stage/Object/Region features alone is not enough: their value comes from being aligned with when the context should be refreshed and how it should guide physical execution.

Human-Video Supervision Transfers Interaction Knowledge Across Embodiments. Human videos consistently improve both RoboCasa settings, with the benefit remaining evident in multi-phase and compositional tasks. Since these videos do not provide target-robot actions, the improvement cannot come from direct action imitation. Instead, it supports the intended role of the shared

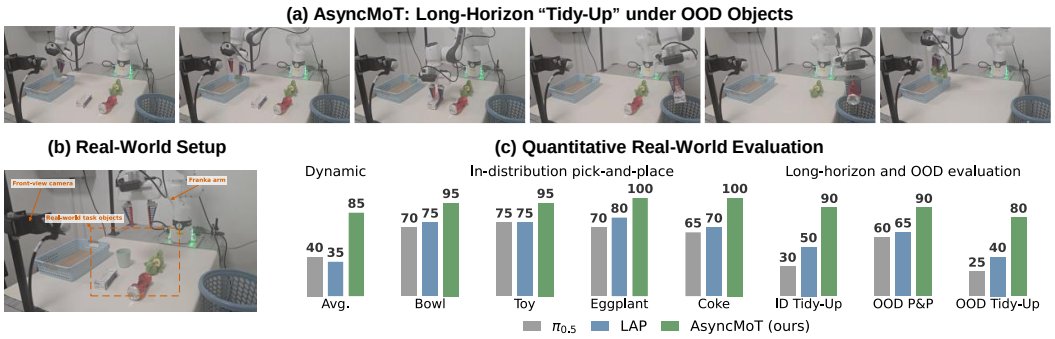


Figure 4: Real-world evaluation of AsyncMoT. (a) Qualitative long-horizon Tidy-Up rollout with OOD objects. (b) Real-world setup with a Franka arm, front-view camera, and manipulation workspace. (c) Quantitative real-world evaluation across dynamic target displacement, in-distribution pick-and-place, and long-horizon/generalization settings. The latter includes Tidy-Up and zero-shot manipulation with unseen objects. Each setting is evaluated over 20 trials.

interaction representation: semantic and spatial interaction structure can transfer across embodiments, while executable motor control remains embodiment-specific. Detailed results are provided in Appendix D.1.

4.3 Real-World Experiments

Real-world evaluation. We deploy AsyncMoT on a physical Franka arm to evaluate real-world robustness, long-horizon execution, and zero-shot object generalization. We consider four in-distribution pick-and-place tasks, a long-horizon Tidy-Up task with distractors, and zero-shot variants using unseen colors and geometries. We further introduce dynamic target displacement during execution. We compare against $\pi_{0.5}$ and LAP under identical conditions, with 20 trials per setting.

Dynamic Target Displacement Tests Interaction-State Refresh. Dynamic target displacement makes previously grounded spatial information stale while leaving the task intent unchanged. As shown in Fig. 4(c), AsyncMoT achieves 85% success and substantially outperforms both baselines. This result supports interaction-driven refresh: retained context should be updated when physical changes invalidate its grounding, even without a change in semantic Stage.

Persistent Interaction Context Benefits Long-Horizon Execution. The advantage of AsyncMoT becomes more pronounced as the execution horizon grows. While all methods perform relatively well on short-horizon pick-and-place, the gap widens substantially on Tidy-Up (Fig. 4(c)). This suggests that persistent interaction context is particularly useful when coherent control must be maintained across extended physical interactions.

Interaction Structure Generalizes to Unseen Objects. AsyncMoT also remains robust under zero-shot object shifts (Fig. 4(a,c)). The advantage persists when unseen objects are combined with long-horizon Tidy-Up, indicating that the learned interaction structure can be re-grounded to novel objects rather than being tied to specific training appearances.

5 Conclusion

We presented AsyncMoT, an interaction-guided asynchronous Mixture of Transformers that decouples interaction-state refresh from continuous motor execution. AsyncMoT maintains a structured Stage–Object–Region state and refreshes it only when physical execution makes the retained context outdated, while cached interaction features provide persistent guidance between refreshes. The same interaction-level representation also supports supervision from heterogeneous human and robot data without requiring action correspondence. Experiments across long-horizon, distribution-shifted, compositional, and real-world settings demonstrate the benefit of deciding not only what interaction context to retain, but also when it should be refreshed.

AI USE STATEMENT

Generative AI tools assisted with experimental interpretation, methodological feedback, manuscript drafting and editing, and literature review. The authors take full responsibility for the final content and all AI-assisted materials.

REPRODUCIBILITY STATEMENT

Section 3 specifies the interaction refresh, action-conditioning, and two-stage training procedures. Section 4 describes the evaluation setup. Appendix A provides implementation details and a notation reference, Appendix B describes the automatic interaction annotation pipeline, and Appendix C provides benchmark-specific configurations and evaluation protocols. We provide anonymous source code in the supplementary materials.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, et al. Qwen3-VL Technical Report. *arXiv preprint arXiv:2511.21631*, 2025.

Suneel Belkhale, Tianli Ding, Ted Xiao, Pierre Sermanet, Quon Vuong, Jonathan Tompson, Yevgen Chebotar, Debidatta Dwibedi, and Dorsa Sadigh. Rt-h: Action hierarchies using language. In *Proceedings of Robotics: Science and Systems*, Delft, The Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.049.

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. PaliGemma: A Versatile 3B VLM for Transfer. *arXiv preprint arXiv:2407.07726*, 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 *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 17–40. PMLR, 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 *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2025b. doi: 10.15607/RSS.2025.XXI.010.

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 *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.014.

Junhao Cai, Zetao Cai, Jiafei Cao, Yilun Chen, Zeyu He, Lei Jiang, Hang Li, Hengjie Li, Yang Li, Yufei Liu, Yanan Lu, Qi Lv, Haoxiang Ma, Jiangmiao Pang, Yu Qiao, Zherui Qiu, Yanqing Shen, Xu Shi, Yang Tian, Bolun Wang, Hanqing Wang, Jiaheng Wang, Tai Wang, Xueyuan Wei, Chao Wu, Yiman Xie, Boyang Xing, Yuqiang Yang, Yuyin Yang, Qiaojun Yu, Feng Yuan, Jia Zeng, Jingjing Zhang, Shenghan Zhang, Shi Zhang, Zhuoma Zhaxi, Bowen Zhou, Yuanzhen Zhou, Yunsong Zhou, Hongrui Zhu, Yangkun Zhu, and Yuchen Zhu. InternVLA-A1: Unifying Understanding, Generation and Action for Robotic Manipulation. *arXiv preprint arXiv:2601.02456*, 2026.

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

Guangyan Chen, Qi Shao, Te Cui, Zichen Zhou, Weixin Mao, Luojie Yang, Meiling Wang, Yi Yang, Hua Chen, and Yufeng Yue. Learning a Unified Latent Action Space from Videos with Action-centric Cycle Consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12871–12880, June 2026.

Hanzhi Chen, Boyang Sun, Anran Zhang, Marc Pollefeys, and Stefan Leutenegger. VidBot: Learning Generalizable 3D Actions from In-the-Wild 2D Human Videos for Zero-Shot Robotic Manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 27661–27672, June 2025a. doi: 10.1109/CVPR52734.2025.02576.

Jingjing Chen, Hongjie Fang, Hao-Shu Fang, and Cewu Lu. Towards effective utilization of mixed-quality demonstrations in robotic manipulation via segment-level selection and optimization. In *2025 IEEE international conference on robotics and automation (ICRA)*, pp. 16884–16891. IEEE, 2025b.

Yinpei Dai, Hongze Fu, Jayjun Lee, Yuejiang Liu, Haoran Zhang, Jianing Yang, Chelsea Finn, Nima Fazeli, and Joyce Chai. Robomme: Benchmarking and understanding memory for robotic generalist policies. *arXiv preprint arXiv:2603.04639*, 2026.

Hao-Shu Fang, Hongjie Fang, Zhenyu Tang, Jirong Liu, Chenxi Wang, Junbo Wang, Haoyi Zhu, and Cewu Lu. RH20T: A Comprehensive Robotic Dataset for Learning Diverse Skills in One-Shot. In *2024 IEEE International Conference on Robotics and Automation*, pp. 653–660, 2024. doi: 10.1109/ICRA57147.2024.10611615.

Haoquan Fang, Markus Grotz, Wilbert Pumacay, Yi Ru Wang, Dieter Fox, Ranjay Krishna, and Jiafei Duan. SAM2Act: Integrating Visual Foundation Model with A Memory Architecture for Robotic Manipulation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 15925–15942. PMLR, 2025.

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: A Progressive Robustness Benchmark for Visual-Language-Action Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 38574–38583, June 2026.

Yicheng Feng, Wanpeng Zhang, Ye Wang, Hao Luo, Haoqi Yuan, Sipeng Zheng, and Zongqing Lu. Spatial-Aware VLA Pretraining through Visual-Physical Alignment from Human Videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 712–723, June 2026.

Asher J. Hancock, Xindi Wu, Lihan Zha, Olga Russakovsky, and Anirudha Majumdar. Actions as Language: Fine-Tuning VLMs into VLAs Without Catastrophic Forgetting. In *International Conference on Learning Representations*, 2026.

Haifeng Huang, Xinyi Chen, Yilun Chen, Hao Li, Xiaoshen Han, Zehan Wang, Tai Wang, Jiangmiao Pang, and Zhou Zhao. RoboGround: Robotic Manipulation with Grounded Vision-Language Priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22540–22550, June 2025a.

Wenhui Huang, Changhe Chen, Han Qi, Chen Lv, Yilun Du, and Heng Yang. MoTVLA: A Vision-Language-Action Model with Unified Fast-Slow Reasoning. *arXiv preprint arXiv:2510.18337*, 2025b.

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

Guangqi Jiang, Yutong Liang, Jianglong Ye, Jia-Yang Huang, Changwei Jing, Rocky Duan, Pieter Abbeel, Xiaolong Wang, and Xueyan Zou. Cross-Hand Latent Representation for Vision-Language-Action Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13496–13507, June 2026.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. DROID: A Large-Scale In-The-Wild Robot Manipulation Dataset. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.120.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-Tuning Vision-Language-Action Models: Optimizing Speed and Success. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2025a. doi: 10.15607/RSS.2025.XXI.017.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P. Foster, Pannag R. Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An Open-Source Vision-Language-Action Model. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713. PMLR, 2025b.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, et al. Cosmos policy: Fine-tuning video models for visuomotor control and planning. *arXiv preprint arXiv:2601.16163*, 2026.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment Anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4015–4026, 2023.

He Kong, Zengjue Chen, Qi Wang, Qianli Xing, Runliang Niu, Peidong Liu, Jiawei Li, Shiqi Wang, and Yi Chang. Beyond flat policies: Hierarchical post-training for embodied agents in robotic manipulation. *arXiv preprint arXiv:2608.05999*, 2026.

Fuhao Li, Wenxuan Song, Han Zhao, Jingbo Wang, Pengxiang Ding, Donglin Wang, Long Zeng, and Haoang Li. Spatial forcing: Implicit spatial representation alignment for vision-language-action model. In *International Conference on Learning Representations*, volume 2026, pp. 132324–132345, 2026a.

Hao Li, Ziqin Wang, Zi-han Ding, Shuai Yang, Yilun Chen, Yang Tian, Xiaolin Hu, Tai Wang, Dahua Lin, Feng Zhao, Si Liu, and Jiangmiao Pang. RoboInter: A Holistic Intermediate Representation Suite Towards Robotic Manipulation. In *International Conference on Learning Representations*, 2026b.

Jinming Li, Yichen Zhu, Zhibin Tang, Junjie Wen, Minjie Zhu, Xiaoyu Liu, Chengmeng Li, Ran Cheng, Yaxin Peng, Yan Peng, and Feifei Feng. CoA-VLA: Improving Vision-Language-Action Models via Visual-Text Chain-of-Affordance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9759–9769, October 2025a.

Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. *arXiv preprint arXiv:2503.00200*, 2025b.

Xiang Li, Ya-Li Li, Yuan Wang, and Shengjin Wang. Trm-vla: Temporal-aware chain-of-thought reasoning and memorization for vision-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10943–10953, June 2026c.

Yi Li, Yuquan Deng, Jesse Zhang, Joel Jang, Marius Memmel, Caelan Reed Garrett, Fabio Ramos, Dieter Fox, Anqi Li, Abhishek Gupta, and Ankit Goyal. HAMSTER: Hierarchical action models for open-world robot manipulation. In *The Thirteenth International Conference on Learning Representations*, 2025c.

Zishuo Li, Bowen Yang, Changtao Miao, Kai Zhu, Hao Chen, Qingze Guan, Zhengxing Wu, Wanke Zhan, Yang Sun, Zhiyi Huang, Zitong Shan, Zhenchao Jin, Jiadong Hong, Taowen Wang, Yushi Feng, You Liu, Yibo Wang, Yifan Yang, Zhaowen Zhou, Man Luo, Hao Cheng, Bo Zhang, Jianshu Li, Jiansheng Cai, Guocai Yao, Jize Zhang, Chenhao Lin, Renjing Xu, Lequan Yu, Chao Shen, Chunhua Shen, and Zhe Li. Open-AoE: An Open Egocentric Manipulation Dataset and Toolchain for Embodied Learning, 2026d.

Weixin Liang, Lili Yu, Liang Luo, Srini Iyer, Ning Dong, Chunting Zhou, Gargi Ghosh, Mike Lewis, Wen-tau Yih, Luke Zettlemoyer, and Xi Victoria Lin. Mixture-of-Transformers: A Sparse and Scalable Architecture for Multi-Modal Foundation Models. *Transactions on Machine Learning Research*, 2025.

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, 2023. Datasets and Benchmarks Track.

Yunze Liu, Yun Liu, Che Jiang, Kangbo Lyu, Weikang Wan, Hao Shen, Boqiang Liang, Zhoujie Fu, He Wang, and Li Yi. HOI4D: A 4D Egocentric Dataset for Category-Level Human-Object Interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21013–21022, June 2022.

Zhuoyang Liu, Jiaming Liu, Hao Chen, Jiale Yu, Ziyu Guo, Chengkai Hou, Chenyang Gu, Xiangju Mi, Renrui Zhang, Kun Wu, Zhengping Che, Jian Tang, Pheng-Ann Heng, and Shanghang Zhang. LaST$_0$: Latent Spatio-Temporal Chain-of-Thought for Robotic Vision-Language-Action Model. In *International Conference on Machine Learning*, 2026.

Guanxing Lu, Baoxiong Jia, Puhao Li, Yixin Chen, Ziwei Wang, Yansong Tang, and Siyuan Huang. Gwm: Towards scalable gaussian world models for robotic manipulation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9263–9274. IEEE, 2025.

Tianyi Lu, Hui Zhang, Zijie Diao, Junke Wang, Shengqi Xu, Xingyao Lin, Guojin Zhong, Ziyi Ye, Peng Wang, Zuxuan Wu, et al. Thinkingvla: Interleaved vision and language reasoning for robotic manipulation. *arXiv preprint arXiv:2606.17937*, 2026.

Hao Luo, Ye Wang, Wanpeng Zhang, Haoqi Yuan, Yicheng Feng, Haiweng Xu, Sipeng Zheng, and Zongqing Lu. Joint-Aligned Latent Action: Towards Scalable VLA Pretraining in the Wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35047–35058, June 2026.

Paul Mattes, Jan Schwab, Jens Bosch, Maximilian Xiling Li, Nils Blank, Minh-Trung Tang, Moritz Haberland, and Rudolf Lioutikov. SIR: Structured Image Representations for Explainable Robot Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 42484–42493, June 2026.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. RoboCasa: Large-Scale Simulation of Household Tasks for Generalist Robots. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.050.

Soroush Nasiriany, Sep Nasiriany, Abhiram Maddukuri, and Yuke Zhu. RoboCasa365: A Large-Scale Simulation Framework for Training and Benchmarking Generalist Robots. In *International Conference on Learning Representations*, 2026.

NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, et al. GR00T N1: An Open Foundation Model for Generalist Humanoid Robots. *arXiv preprint arXiv:2503.14734*, 2025.

NVIDIA GEAR Team. GR00T N1.6: An Improved Open Foundation Model for Generalist Humanoid Robots. https://research.nvidia.com/labs/gear/gr00t-n1_6/, December 2025. NVIDIA Research.

OpenAI. GPT-5.6: Frontier Intelligence that Scales with Your Ambition. https://openai.com/index/gpt-5-6/, July 2026. Product release, July 9, 2026.

Mingjie Pan, Jiyao Zhang, Tianshu Wu, Yinghao Zhao, Wenlong Gao, and Hao Dong. OmniManip: Towards General Robotic Manipulation via Object-Centric Interaction Primitives as Spatial Constraints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17359–17369, June 2025.

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. *arXiv preprint arXiv:2501.09747*, 2025.

Jingjing Qian, Boyao Han, Chen Shi, Lei Xiao, Long Yang, Shaoshuai Shi, and Li Jiang. Geopredict: Leveraging predictive kinematics and 3d gaussian geometry for precise vla manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13529–13539, 2026.

Robert Ramirez Sanchez, Daniel J. Evans, Dylan P. Losey, and Siddarth Jain. Volt: Vision and language trajectory segmentation for faster-than-demonstration policies. *arXiv preprint arXiv:2606.06323*, 2026.

Hao Shi, Bin Xie, Yingfei Liu, Lin Sun, Fengrong Liu, Tiancai Wang, Erjin Zhou, Haoqiang Fan, Xiangyu Zhang, and Gao Huang. Memoryvla: Perceptual-cognitive memory in vision-language-action models for robotic manipulation. In *International Conference on Learning Representations*, volume 2026, pp. 18567–18602, 2026.

Huajie Tan, Peterson Co, Yijie Xu, Shanyu Rong, Yuheng Ji, Cheng Chi, Xiansheng Chen, Zhongxia Zhao, Pengwei Wang, Zhongyuan Wang, and Shanghang Zhang. Action-Sketcher: From Reasoning to Action via Visual Sketches for Robotic Manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22433–22444, June 2026.

Shuhan Tan, Kairan Dou, Yue Zhao, and Philipp Krähenbühl. Interactive post-training for vision-language-action models. *arXiv preprint arXiv:2505.17016*, 2025.

Weiliang Tang, Jialin Gao, Jia-Hui Pan, Gang Wang, Li Erran Li, Yun-Hui Liu, Mingyu Ding, Pheng-Ann Heng, and Chi-Wing Fu. Rethinking Intermediate Representation for VLM-based Robot Manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 29652–29662, June 2026.

Shuqiao Wei, Ying Ni, and Jian Sun. Transformer-based multi-agent traffic simulation for autonomous vehicle testing in shared urban road segments. *Accident Analysis & Prevention*, 232:108550, 2026.

Shenhao Yan, Ge Wang, Qi Liu, Weilin Meng, Jiahao Yang, Chengsi Yao, Fan Feng, Xiaoguang Ma, Yiming Zhao, and Yatong Han. Acting while understanding: Asynchronous semantic-action decoupling for real-time vision-language-action models. *arXiv preprint arXiv:2606.15285*, 2026.

Bin Yu, Shijie Lian, Xiaopeng Lin, Yuliang Wei, Zhaolong Shen, Changti Wu, Yuzhuo Miao, Xinming Wang, Bailing Wang, Cong Huang, and Kai Chen. TwinBrainVLA: Unleashing the Potential of Generalist VLMs for Embodied Tasks via Asymmetric Mixture-of-Transformers. *arXiv preprint arXiv:2601.14133*, 2026.

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.

Wentao Yuan, Jiafei Duan, Valts Blukis, Wilbert Pumacay, Ranjay Krishna, Adithyavairavan Murali, Arsalan Mousavian, and Dieter Fox. Robopoint: A vision-language model for spatial affordance prediction for robotics. *arXiv preprint arXiv:2406.10721*, 2024.

Lihan Zha, Asher J. Hancock, Mingtong Zhang, Tenny Yin, Yixuan Huang, Dhruv Shah, Allen Z. Ren, and Anirudha Majumdar. LAP: Language-Action Pre-Training Enables Zero-Shot Cross-Embodiment Transfer. In *Proceedings of Robotics: Science and Systems*, 2026.

Jianke Zhang, Yanjiang Guo, Xiaoyu Chen, Yen-Jen Wang, Yucheng Hu, Chengming Shi, and Jianyu Chen. HiRT: Enhancing robotic control with hierarchical robot transformers. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 933–946. PMLR, 2025.

Zhilong Zhang, Wenyu Luo, Haonan Wang, Yifei Sheng, Yidi Wang, Hanyuan Guo, Haoxiang Ren, Xinghao Du, Yuhan Che, Tongtong Cao, Lei Yuan, and Yang Yu. Anticipation-vla: Solving long-horizon embodied tasks via anticipation-based subgoal generation. *arXiv preprint arXiv:2605.01772*, 2026.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.016.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, Ya-Qin Zhang, Jiangmiao Pang, Jingjing Liu, Tai Wang, and Xianyuan Zhan. X-VLA: Soft-Prompted Transformer as Scalable Cross-Embodiment Vision-Language-Action Model. In *International Conference on Learning Representations*, 2026.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8152–8162, June 2026.

Minjie Zhu, Yichen Zhu, Jinming Li, Zhongyi Zhou, Junjie Wen, Xiaoyu Liu, Chaomin Shen, Yaxin Peng, and Feifei Feng. Objectvla: End-to-end open-world object manipulation without demonstration. *arXiv preprint arXiv:2502.19250*, 2025.

Teqiang Zou, Hongliang Zeng, Yuxuan Nong, Yifan Li, Kehui Liu, Haotian Yang, Xinyang Ling, Xin Li, and Lianyang Ma. Asynchronous fast-slow vision-language-action policies for whole-body robotic manipulation. *arXiv preprint arXiv:2512.20188*, 2025.

A Additional Method Details

This section provides additional implementation details for the asynchronous interaction refresh and action-conditioning mechanisms of AsyncMoT, followed by a notation reference.

Boundary Watcher. The temporal execution context u_t is computed from a causal window containing the latest N control steps. Each step includes the current visual observation and available robot execution state, and a lightweight Temporal Transformer Encoder aggregates the resulting sequence into u_t. The persistent interaction context c_t is constructed from the Stage, Object, and Region representations produced at the most recent refresh and remains fixed until the next refresh. The Boundary Head jointly processes (u_t, c_t) to predict z_t^B.

Since refresh boundaries are sparse relative to ordinary control steps, positive-class weighting is applied to the Boundary objective. At inference, KEEP preserves the retained interaction representation and its cached key–value features, whereas REFRESH recomputes Stage, Object, and Region from the latest observation and task instruction. A REFRESH does not require a change in Stage semantics; it indicates that the retained interaction context should be recomputed from the current execution state. The Boundary decision affects only the interaction-level state, while visual and proprioceptive features used for action generation continue to be updated at every control step.

Persistent Interaction Cache. At each refresh time t_k, the interaction branch converts the current Stage, Object, and Region representations into interaction-level key–value features $(K_k^{\text{int}}, V_k^{\text{int}})$. These features are reused for all subsequent control steps until the next refresh. In contrast, $(K_t^{\text{act}}, V_t^{\text{act}})$ are recomputed from the latest visual and proprioceptive observations at every step. The attention memory therefore combines persistent interaction features with current execution features, so that interaction-level guidance persists between refreshes while low-level perception and action generation remain conditioned on the latest observations. Consecutive refreshes may preserve the same Stage semantics while updating the retained interaction representation from the latest execution context.

Movement Direction and Displacement Supervision. The executable policy is trained with the original continuous-action objective $\mathcal{L}_{\text{action}}$. In addition, each future action chunk $a_{t:t+H}$ provides an auxiliary motion target $m_{t:t+H}$ constructed from its translational and rotational trajectory. The target summarizes movement directions together with the corresponding displacement magnitudes and is predicted under the semantic context of the current Stage. This provides explicit supervision on the motion structure associated with the ongoing interaction without replacing the original continuous-action target. The auxiliary motion objective is used only for representation learning during training; executable motor commands continue to be generated through the continuous-action policy. Stage provides semantic conditioning for the motion target but does not determine the timing of interaction-state refreshes.

Notation. Table 6 summarizes the notation used throughout the method. The index k denotes interaction-state refresh events rather than semantic Stage transitions; consecutive refreshes may therefore share the same Stage semantics.

B Automatic Interaction Annotation

Existing robot datasets provide visual observations, robot states, and actions, but do not explicitly annotate the interaction structure required by AsyncMoT. We therefore construct an offline annotation pipeline that combines trajectory-derived temporal cues with foundation models for semantic interpretation and visual grounding. Specifically, we use GPT-5.6 (OpenAI, 2026) as the text LLM, Qwen3-VL (Bai et al., 2025) as the VLM, Segment Anything Model (SAM) (Kirillov et al., 2023) for fine-grained segmentation, and the FLUX autoencoder for object representation encoding. The pipeline produces synchronized Boundary, Stage, Object, and Region supervision offline and introduces no additional inference-time computation.

Boundary annotation. We first identify sparse interaction boundaries directly from the robot trajectory. At each timestep, we derive temporal cues from end-effector translation and rotation, their finite differences, dominant motion direction, and gripper state. Gripper transitions, low-motion intervals, and changes in end-effector motion direction jointly provide evidence that the retained interaction context should be refreshed. Boundary candidates supported by these cues are temporally

Table 6: Notation for AsyncMoT. Interaction states are indexed by refresh event k, while low-level execution is indexed by control step t.

Symbol	Meaning
Interaction state and asynchronous refresh	
t	Low-level control step.
t_k	Time of the k-th interaction-state refresh.
$\tau(t)$	Most recent interaction refresh before control step t.
$\mathcal{I}_k$	Structured interaction state produced at refresh k.
S_k, O_k, R_k	Stage, Object, and Region representations at refresh k.
$S_k^\star, \hat{S}_k$	Target and predicted Stage representations.
$O_k^\star, \hat{O}_k$	Target and predicted Object representations.
$R_k^\star, \hat{R}_k$	Target and predicted Region representations.
u_t	Temporal execution context from the latest N control steps.
c_t	Persistent interaction context retained from the latest refresh.
z_t^B	Boundary prediction logit.
b_t	Boundary supervision target for interaction-state refresh.
w_+	Positive-class weight for sparse Boundary supervision.
Interaction-conditioned action generation	
$K_t^{\text{act}}, V_t^{\text{act}}$	Step-wise action features from the current execution state.
$K_{\tau(t)}^{\text{int}}, V_{\tau(t)}^{\text{int}}$	Cached interaction features from the latest refresh.
$a_{t:t+H}$	Continuous future action chunk over horizon H.
$m_{t:t+H}$	Movement-direction and displacement target derived from the action chunk.
Training objectives	
$\mathcal{L}_{\text{stage}}$	Stage representation loss.
$\mathcal{L}_{\text{object}}$	Object representation loss.
$\mathcal{L}_{\text{region}}$	Region spatial-prediction loss.
$\mathcal{L}_{\text{boundary}}$	Boundary classification loss for interaction-state refresh.
$\mathcal{L}_{\text{motion}}$	Movement-direction and displacement supervision loss.
$\mathcal{L}_{\text{action}}$	Continuous action-generation objective.
$\mathcal{L}_{\text{S1}}$	Stage-1 interaction-representation objective.
$\mathcal{L}_{\text{S2}}$	Stage-2 interaction-conditioned action objective.

merged, while candidates producing excessively short intervals are suppressed. A short neighborhood around each retained boundary is assigned positive supervision to account for temporal uncertainty. Consecutive boundaries then partition the trajectory into temporally coherent interaction intervals. Importantly, these boundaries are determined from changes in physical execution rather than from semantic Stage labels; they indicate when the retained interaction context should be refreshed, not necessarily when the Stage should change.

Stage annotation. For each boundary-delimited interval, we construct a structured textual summary from the task instruction, interval position, boundary information, gripper states, dominant end-effector motion, and available object hypotheses. GPT-5.6 interprets this interval-level context and assigns a concise semantic Stage describing the currently executed interaction phase. We further enforce consistency with the observed trajectory context and adjacent intervals. The resulting Stage target is shared by all frames within the interval, providing persistent semantic context between refreshes. We do not require adjacent boundary-delimited intervals to have different Stage descriptions. When execution cues indicate a boundary without a change in high-level interaction intent, consecutive intervals may therefore share the same Stage description.

Object annotation. For each boundary-delimited interval, we sample observations across the interval to reduce sensitivity to occlusion or an unfavorable single frame. Conditioned on the task instruction and Stage description, Qwen3-VL identifies the entity involved in the current interaction and predicts its bounding box. Candidate predictions are filtered according to visual validity and temporal consistency. From the accepted localization, we construct a masked crop that preserves limited surrounding context while suppressing unrelated background. The crop is then encoded by the frozen FLUX autoencoder to obtain the continuous Object target, which is shared throughout the corresponding interval.

Region annotation. Region provides explicit spatial grounding for the current interaction. We use the accepted Qwen3-VL bounding box as a prompt to SAM and select a segmentation mask consistent with the grounded object and current Stage. The mask is converted into a normalized soft spatial response and projected to a 24×24 dense heatmap. Invalid or temporally inconsistent masks are discarded. The resulting Region target provides dense spatial supervision complementary to the semantic Stage and Object representations.

C Experimental Details

We provide additional benchmark-specific details omitted from the main text for conciseness.

LIBERO and LIBERO-Plus. For LIBERO (Liu et al., 2023), we follow the standard four-suite VLA evaluation protocol (Kim et al., 2025a) and report LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-10 (Long). LIBERO-Plus (Fei et al., 2026) evaluates robustness under seven controlled distribution shifts covering camera viewpoint, robot initial state, language instruction, lighting, background texture, sensor noise, and object layout.

RoboCasa. We evaluate diverse kitchen manipulation on the 24-task RoboCasa benchmark (Nasiriany et al., 2024), following the commonly used 24-task protocol (Feng et al., 2026; Mattes et al., 2026). The tasks are grouped into Pick & Place, Doors, Drawers, Knobs, Levers, Buttons, and Insertion, with the category scores aggregated over all 24 tasks.

RoboCasa365. We use the 50-task RoboCasa365 benchmark (Nasiriany et al., 2026), comprising 18 Atomic-Seen, 16 Composite-Seen, and 16 Composite-Unseen tasks. The Composite-Unseen split evaluates transfer to held-out composite tasks. We report the three split scores and the official Overall score.

RoboMME. RoboMME (Dai et al., 2026) evaluates history-dependent manipulation over 16 tasks grouped into Counting, Permanence, Reference, and Imitation, targeting complementary forms of temporal and task-context reasoning. We report the four suite scores and the benchmark average.

Evaluation protocol. Unless otherwise stated, we follow the official evaluation protocol of each benchmark and use task success rate as the primary metric. Each main-result table includes the baselines available under the corresponding benchmark protocol.

Table 7: Effect of human-video supervision on RoboCasa-24. We report success rate (%). Δ denotes the absolute improvement obtained by incorporating human videos.

Training Data	Pick & Place	Doors	Drawers	Knobs	Levers	Buttons	Insertion	Avg.
Robot Data Only	54.8	70.0	86.0	34.0	80.7	66.0	36.0	61.3
Robot Data + Human Video	**58.5**	**76.0**	**88.0**	**42.0**	**86.7**	**69.3**	**46.0**	**66.3**
Δ	+3.7	+6.0	+2.0	+8.0	+6.0	+3.3	+10.0	+5.0

Table 8: Effect of human-video supervision on RoboCasa365. We report success rate (%). Δ denotes the absolute improvement obtained by incorporating human videos.

Training Data	Atomic-Seen	Composite-Seen	Composite-Unseen	Overall
Robot Data Only	60.6	25.2	6.8	32.1
Robot Data + Human Video	**63.7**	**30.4**	**10.1**	**35.9**
Δ	+3.1	+5.2	+3.3	+3.8

D Additional Experimental Results

D.1 Human-Video Contribution

Under the same optimization budget, incorporating human videos improves the RoboCasa-24 average from 61.3% to 66.3% and the RoboCasa365 overall score from 32.1% to 35.9%. As detailed

in Tables 7 and 8, the gain is consistent across all seven RoboCasa-24 categories and is largest on Insertion ($+10.0$ points) and Knobs ($+8.0$ points). These skills involve tightly coordinated interaction phases and precise transitions between them, making them well matched to the structured Stage/Object/Region representation and boundary-driven asynchronous updates of AsyncMoT. On RoboCasa365, human-video supervision improves Atomic-Seen, Composite-Seen, and Composite-Unseen by 3.1, 5.2, and 3.3 points, respectively, with the largest gain on Composite-Seen. This trend is particularly relevant because composite tasks require multiple atomic skills to be executed sequentially: AsyncMoT organizes such execution through persistent interaction states and refreshes them only when physical execution makes the retained context outdated, allowing interaction-level knowledge learned from heterogeneous human videos to be reused across successive stages.