MoAct: Learning Visual Transition Dynamics for Robot Action Generation

Anonymous authors
Paper under double-blind review

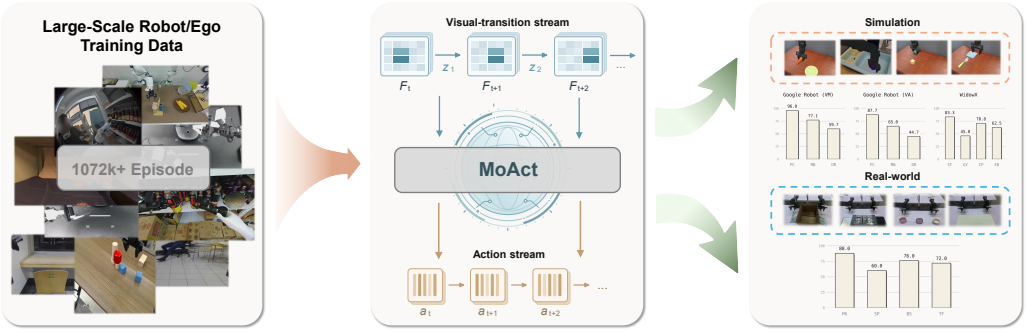

Figure 1: **Overview of MoAct.** MoAct is a visual-transition-aware vision-language-action model that learns where and how scenes evolve before generating actions. Rather than mapping observations directly to actions, it extracts compact transition tokens from large-scale robot and egocentric human videos and couples them with action generation through a dual-stream Mixture-of-Transformers.

Abstract

Modeling future observations allows vision-language-action (VLA) models to anticipate scene changes and improve action generation. However, extracting latent representations that retain the information needed for accurate prediction and control remains a key challenge. To address this challenge, we introduce MoAct, a framework that uses compact visual-transition tokens to model changes in future visual context and guide action generation. MoAct learns visual-transition dynamics from large-scale video data and connects a visual-transition dynamics expert with an action expert through a dual-stream Mixture-of-Transformers (MoT) architecture. This design allows predicted scene changes to guide policy learning. Extensive experiments in simulation and the real world show that MoAct outperforms existing future-guided methods. Egocentric human videos also help MoAct retain much of its performance with fewer robot demonstrations. For scene adaptation, MoAct requires only additional egocentric human videos from the target scenes, with no new robot demonstrations.

1 Introduction

Vision-language-action (VLA) models integrate visual and linguistic representations with continuous control, enabling robots to generate actions from visual and linguistic inputs (Kim et al., 2024; Black et al., 2024; Intelligence et al., 2025; Bjorck et al., 2025; Team et al., 2025b). Action supervision trains these policies to imitate demonstrations (Kim et al., 2024; Black et al., 2024), but does not explicitly require them to model the visual changes that accompany task progress. Predictive learning offers a complementary signal, as successive observations reveal how a scene evolves, even in videos without robot action annotations (Assran et al., 2023; Bardes et al., 2024; Assran et al., 2025). A central question is how to represent this evolution and use it to guide action generation.

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.

Existing approaches incorporate future information into robot control through several complementary routes. World-action models jointly model future visual observations and robot actions (Zhu et al., 2025; Bi et al., 2026; Li et al., 2026; Ye et al., 2026). Latent action models encode motion in compact intermediate representations learned from videos (Chen et al., 2025; Bu et al., 2025b; Zhang et al., 2026a). Feature-alignment approaches align policy representations with those of future observations (Zheng et al., 2025). Condition-space approaches learn compact future conditions through action supervision, then train a vision-language model (VLM) to predict them alongside actions (Su et al., 2026). Related visual-foresight and latent-world-modeling methods further explore how future visual information can guide planning and control (Zhao et al., 2025; Zhang et al., 2026c; Yang et al., 2026c; Zhang et al., 2026e). Despite their different routes, these approaches share a common challenge of extracting compact representations from information-rich future observations while preserving information useful for precise action generation. Focusing on visual transitions offers a natural route because it directly describes where and how a scene changes, including object motion and evolving spatial relationships during manipulation (Zhou et al., 2024; Yang et al., 2026a). However, visual changes may reflect both action-related motion and external variation (Zhang et al., 2026b). A useful representation must therefore preserve fine-grained transition information while allowing the policy to assess its relevance to the current observation and task. This observation motivates our insight:

Can the transition dynamics embedded in adjacent visual contexts provide the future information required for action generation in a sufficiently compact and effective form?

To answer this question, we introduce **MoAct**, a VLA model that learns visual-transition dynamics and uses them to guide action generation. MoAct learns compact representations of visual transitions by predicting subsequent visual features conditioned on those of the preceding observation, following the feature-space prediction perspective explored in prior visual world models (Zhou et al., 2024; Bardes et al., 2024). This design links future modeling to where and how a scene changes, providing predictive information about scene evolution for policy learning. A dual-stream Mixture-of-Transformers (MoT) architecture connects the visual-transition dynamics expert and action expert, allowing the latter to attend to these representations. The action expert combines these representations with the current observation, instruction, and robot state, conditioning action generation on both the present scene and anticipated changes.

To connect general visual-transition learning with task-specific action generation, we train MoAct in two stages. First, we pretrain general visual-transition dynamics on a large-scale mixture of robot and egocentric human manipulation videos. Second, we jointly adapt these dynamics to target scenes while training a newly initialized action expert on robot demonstrations. This training recipe enables MoAct to anticipate visual transitions in a compact latent space before generating actions. Extensive evaluations on simulated and real-world manipulation tasks show that MoAct outperforms existing future-guided action policies. Furthermore, egocentric human manipulation videos help MoAct retain much of its performance with fewer robot demonstrations. MoAct also supports scene adaptation by adding only egocentric human videos from target scenes, without collecting additional robot demonstrations.

In summary, our main **contributions** are threefold:

- We propose MoAct, a VLA framework that models compact visual-transition dynamics with transition tokens and uses them to guide action generation.
- We develop a dual-stream MoT architecture and a two-stage training recipe that connect visual-transition dynamics learning with task-specific action generation using heterogeneous robot and egocentric human videos.
- We validate MoAct through simulation and real-world experiments, showing improved performance over existing future-guided action policies. Egocentric human videos further reduce the need for robot demonstrations. Scene adaptation requires only additional egocentric human videos from the target scenes, with no new robot demonstrations.

2 Related Work

Vision-Language-Action Models. VLA policies use pretrained VLM backbones to guide action generation with visual observations and language instructions (Kim et al., 2024; Black et al., 2024; Bjorck et al., 2025; Team et al., 2025b;a; Jiang et al., 2025). Semantic knowledge from large-scale pretraining, together with diverse robot experience, supports generalization across objects, scenes, and instructions (Intelligence et al., 2025; Zheng et al., 2026). Recent work also explores action tokenization and policy backbones to improve the flexibility of VLA inference (Pertsch et al., 2025; Hou et al., 2025; Wang et al., 2025). However, image-text pretraining and action supervision alone do not explicitly require a policy to anticipate how a scene changes during manipulation. To supply this predictive signal, MoAct introduces explicit visual-transition supervision from videos and uses the learned dynamics as an additional condition to guide improved action generation.

Future Representations for Action Generation. Future prediction extends robot policies with information about how a scene may evolve, providing guidance beyond the current observation (Zhao et al., 2025; Zhang et al., 2026c; Liu et al., 2026a; Zhang et al., 2026e). Existing approaches jointly model future observations and actions or encode their relationship in shared latent representations (Zhu et al., 2025; Bi et al., 2026; Routray et al., 2026; Ma et al., 2026). Other methods use visual foresight or future-video reasoning to anticipate task-relevant states before action execution (Yang et al., 2026c; Zhang et al., 2026e). Latent-action models further compress motion into intermediate representations learned from robot or human videos (Chen et al., 2025; Bu et al., 2025b; Zhang et al., 2026a). Others align policy features with future observations or learn compact future conditions through action supervision (Zheng et al., 2025; Su et al., 2026). Unlike these approaches, MoAct uses feature-transition supervision to encode where and how scenes change. The dual-stream MoT supplies these compact dynamics representations directly to the action expert, making visual transitions an explicit condition for action generation.

World Modeling in Latent Space. Predicting visual features offers a route to world modeling without reconstructing future images (Assran et al., 2023; Bardes et al., 2024; Assran et al., 2025; Maes et al., 2026). DINO-WM uses pretrained visual representations to learn action-conditioned dynamics for planning (Zhou et al., 2024). More broadly, masked and latent world models study how predictive representations can support planning and control without exhaustive pixel-level reconstruction (Seo et al., 2023; Hafner et al., 2023; Hansen et al., 2024). Related video-learning methods transfer visual experience to robot control through latent motion or action representations (Bu et al., 2025b; Zhang et al., 2026a; Chen et al., 2025; Routray et al., 2026). Recent work also investigates continuous latent motion and compact state representations for robot learning (Yang et al., 2026b; Liu et al., 2026b; Zhang et al., 2026b). MoAct places visual transitions at the center of latent world modeling, focusing on where changes occur and how they unfold. Using preceding observation features as a reference for successor-feature prediction, it learns compact transition representations that capture the location and evolution of visual changes.

3 MoAct

We propose MoAct (Figure 2), a VLA model that learns compact visual-transition dynamics from robot manipulation data and egocentric human videos and uses them to guide action generation. Section 3.1 describes the MoAct architecture, and Section 3.2 presents our two-stage training strategy.

3.1 Architecture

Learning visual-transition representations. MoAct represents scene evolution as transitions between adjacent observations, rather than as independently predicted future states. Our Qwen3.5-0.8B VLM backbone (Qwen Team, 2026) encodes the current observation o_t and language instruction ℓ into contextual tokens $\mathbf{H}$. A visual-transition dynamics expert uses N learnable queries $\mathbf{Q} \in \mathbb{R}^{N \times D}$, where D is its hidden dimension. Alternating cross-attention to $\mathbf{H}$ with query self-attention produces transition tokens $\mathbf{Z} = \{\mathbf{z}_i\}_{i=1}^{N}$. Each token is trained to encode the change from a preceding feature grid to its successor: a frozen DINOv3 encoder (Siméoni et al., 2025) extracts feature grids $\mathbf{F}_i$ from sampled observations, and a predictor estimates $\hat{\mathbf{F}}_i = g_\phi(\mathbf{F}_{i-1}, \mathbf{z}_i)$. Specifically, $\mathbf{z}_i$ first generates spatially varying gates over $\mathbf{F}_{i-1}$ to localize where changes occur, and then mod-

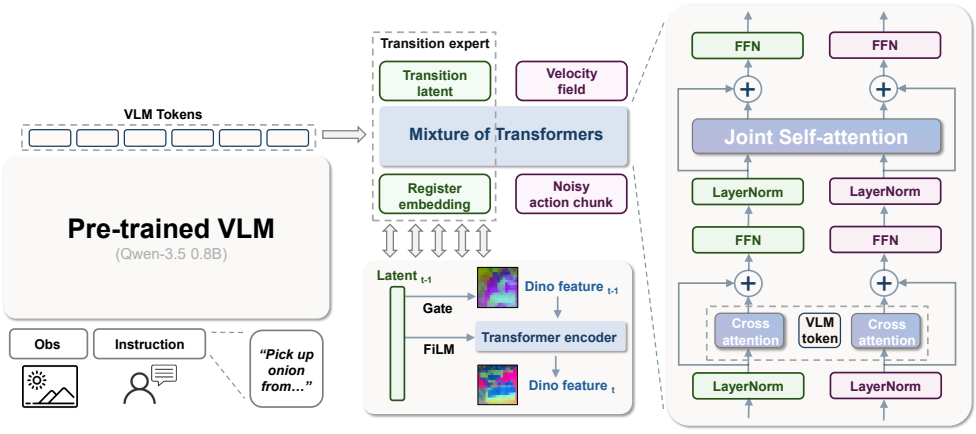

Figure 2: **MoAct Model Architecture.** MoAct models where and how scenes change to guide action generation. The visual-transition dynamics expert predicts compact transition representations from VLM backbone features. A dual-stream MoT passes these representations to the action expert, which generates actions through flow matching.

ulates the predictor layers to control how those changes unfold. This makes each token an explicit interface for the local visual dynamics between adjacent observations, supervised by the transition loss in Section 3.2. Further architectural details are provided in Appendix A.1.

Conditioning actions through a dual-stream MoT. We connect the visual-transition dynamics expert and action expert through a dual-stream MoT with separate parameters. The interface follows the principle of insulating predictive representations from action optimization (Driess et al., 2026). At paired layer l, action queries $\mathbf{Q}_A^{(l)}$ attend jointly to their own keys and values and to the dynamics pair $(\mathrm{sg}(\mathbf{K}_D^{(l)}), \mathrm{sg}(\mathbf{V}_D^{(l)}))$, where sg denotes stop-gradient. The dynamics keys and values provide layer-wise transition information for action generation. Cross-attention to $\mathbf{H}$ alternates with this joint attention, allowing the action expert to combine current scene context with predicted visual transitions. Further architectural details are provided in Appendix A.2.

Flow-matching action expert. The action expert follows Transformer-based continuous action generators (Bjorck et al., 2025; Hou et al., 2025). Its conditioning $\mathbf{c}_t$ comprises the vision-language context $\mathbf{H}$, proprioceptive state $\mathbf{s}_t$, and dynamics keys and values supplied by the MoT. The noisy action chunk $\mathbf{a}_\tau$ receives flow-time and temporal-position embeddings and is concatenated with an encoded proprioceptive token. Non-causal attention within this sequence lets each action token use the full chunk and the current robot state. An output head predicts the conditional velocity field $v_\theta(\mathbf{a}_\tau, \tau; \mathbf{c}_t)$, trained with the flow-matching objective in Section 3.2 (Lipman et al., 2022; Black et al., 2024). At inference, this field transforms a Gaussian noise sample into an action chunk under the same conditioning. Further architectural details are provided in Appendix A.3.

3.2 Training Recipe

Predictive pretraining provides visual priors for robot learning (Assran et al., 2025; Zhu et al., 2025). MoAct uses two training stages. The first learns general visual-transition dynamics from broad video data, and the second jointly adapts these dynamics to the target task while training the action expert under a shared objective. Detailed training settings are provided in Appendix B.2.

Stage 1. Visual-transition pretraining. We pretrain general visual-transition dynamics on a mixture of robot manipulation videos and egocentric human videos, with the complete data composition provided in Appendix B.1. With feature prediction as supervision (Bardes et al., 2024), we optimize the VLM backbone, visual-transition dynamics expert, and feature predictor while keeping DINOv3

frozen. For a video sequence $V \sim \mathcal{D}_{\mathrm{V}}$, the objective is

$$\mathcal{L}_{\mathrm{trans}} = \mathbb{E}_{V \sim \mathcal{D}_{\mathrm{V}}} \left[\frac{1}{N} \sum_{i=1}^{N} \mathrm{MSE}\left(\hat{\mathbf{F}}_i, \mathbf{F}_i\right) \right], \tag{1}$$

where MSE denotes mean squared error.

Stage 2. Visual-transition fine-tuning and action expert training. Stage 2 jointly adapts the pretrained dynamics to downstream scenes and trains the action expert. We initialize the VLM backbone, visual-transition dynamics expert, and feature predictor from Stage 1 and reinitialize the action expert; task-specific adjacent observations and instructions provide transition supervision, while robot demonstrations provide action supervision through flow matching (Lipman et al., 2022; Black et al., 2024). Given a demonstrated action chunk $\mathbf{a}$, we sample Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and flow time $\tau \in [0, 1]$, form $\mathbf{a}_\tau = (1-\tau)\boldsymbol{\epsilon} + \tau\mathbf{a}$, and use $\mathbf{a} - \boldsymbol{\epsilon}$ as the target velocity. The action expert minimizes

$$\mathcal{L}_{\mathrm{act}} = \mathbb{E}_{\mathbf{a},\boldsymbol{\epsilon},\tau} \left[\mathrm{MSE}(v_\theta(\mathbf{a}_\tau, \tau; \mathbf{c}_t), \mathbf{a} - \boldsymbol{\epsilon}) \right]. \tag{2}$$

The condition $\mathbf{c}_t$ combines $\mathbf{H}$, the proprioceptive state $\mathbf{s}_t$, and the visual-transition representations supplied as keys and values by MoT.

The transition and action objectives are optimized jointly with

$$\mathcal{L}_{\mathrm{joint}} = \mathcal{L}_{\mathrm{trans}} + \mathcal{L}_{\mathrm{act}}. \tag{3}$$

DINOv3 remains frozen in both stages; the VLM backbone, visual-transition dynamics expert, feature predictor, and action expert are updated jointly in Stage 2, while the MoT interface and its stop-gradient pathway are retained.

4 Experiments

Benchmarks. We conduct a multi-suite simulation evaluation on SimplerEnv (Li et al., 2024), a challenging real-to-sim benchmark for closed-loop manipulation under visual and environmental variation. The protocol covers two robot embodiments and seven tasks: three Google Robot tasks evaluated under Visual Matching (VM) and Variant Aggregation (VA), and four contact-rich WidowX tasks. Together, these suites test grasping, collision-aware object transport, constrained-axis manipulation, and precise placement across varied visual conditions. We use the Fractal and Bridge subsets of Open X-Embodiment (O'Neill et al., 2024) to jointly fine-tune MoAct's visual-transition dynamics and action generation for the Google Robot and WidowX suites, respectively, and evaluate the resulting policies under the corresponding protocols. Detailed implementation settings are provided in Appendix B.2.

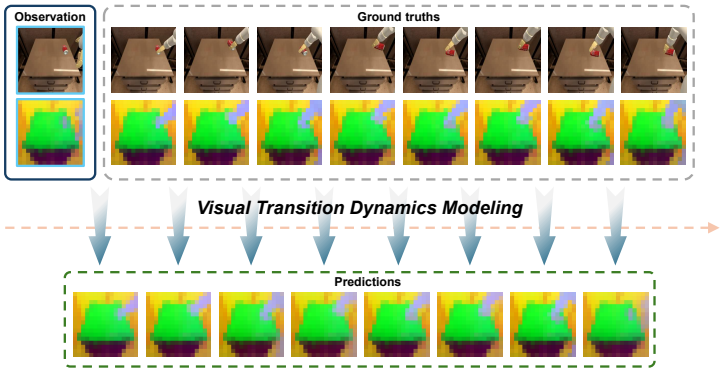

Figure 3: **Visual-transition dynamics modeling in SimplerEnv.** Given adjacent visual contexts, MoAct compactly models their visual transition with the visual-transition expert. The current visual feature is transformed into a successor feature through the corresponding visual-transition token. We visualize this process on a representative manipulation task.

Baselines. To evaluate MoAct's effectiveness, we compare it with two categories of methods. The first comprises conventional VLA policies, including π_0 (Black et al., 2024), π_0-FAST (Pertsch et al., 2025), OpenVLA (Kim et al., 2024), and GR00T N1 (Bjorck et al., 2025); the second comprises future-guided methods, including Moto (Chen et al., 2025), UniVLA (Bu et al., 2025b), DeFI (Zhang et al., 2026d), VITA (Ma et al., 2026), ViPRA (Routray et al., 2026), and WoG (Su et al., 2026). Detailed simulation and real-world evaluation settings for MoAct are provided in Appendices B.4 and B.5.

Table 1: **Google Robot evaluation in SimplerEnv.** Success rates (%) under VM and VA across the three tasks. The highest score in each column is shown in bold.

		VM				VA				Overall
Model	Size	PC	MN	DR	Avg.	PC	MN	DR	Avg.	Avg.
π_0 (Black et al., 2024)	3B	72.7	65.3	38.3	58.8	75.2	63.7	25.6	54.8	56.8
π_0-FAST (Pertsch et al., 2025)	3B	75.3	67.5	42.9	61.9	77.6	68.2	31.3	59.0	60.5
OpenVLA (Kim et al., 2024)	7B	16.3	46.2	35.6	32.7	54.5	47.7	17.7	39.8	33.8
GR00T N1 (Bjorck et al., 2025)	2B	47.0	70.0	18.1	45.0	78.8	62.5	13.2	51.5	48.4
Moto (Chen et al., 2025)	0.5B	74.0	60.4	43.1	59.2	–	–	–	–	–
VITA (Ma et al., 2026)	2B	57.5	55.8	58.9	57.4	–	–	–	–	–
DeFI (Zhang et al., 2026d)	2B	54.2	60.7	38.6	51.2	53.9	58.2	24.0	45.4	48.3
WoG (Su et al., 2026)	7B	89.0	**82.5**	**62.5**	**78.0**	**87.9**	**75.0**	19.3	60.7	69.4
MoAct	1B	**96.0**	77.1	59.7	77.6	87.7	65.0	**44.7**	**65.8**	**71.7**

Table 2: **WidowX evaluation in SimplerEnv.** Grasp rate (GR) and task success rate (SR) (%) across the four tasks. Overall averages the four tasks. The highest score in each column is shown in bold.

		ST		GY		CP		EB		Overall	
Model	Size	GR	SR	GR	SR	GR	SR	GR	SR	GR	SR
π_0 (Black et al., 2024)	3B	45.8	29.1	25.0	0.0	50.0	16.7	91.6	62.5	53.1	27.1
π_0-FAST (Pertsch et al., 2025)	3B	62.5	29.1	58.5	21.9	54.0	10.8	83.3	66.6	64.6	32.1
OpenVLA (Kim et al., 2024)	7B	4.1	0.0	33.0	0.0	12.5	0.0	8.3	4.1	14.5	1.0
GR00T N1 (Bjorck et al., 2025)	2B	83.3	62.5	54.2	45.8	70.8	16.7	41.7	20.8	62.5	36.5
UniVLA (Bu et al., 2025b)	7B	76.4	52.8	66.7	2.8	**79.2**	55.6	87.5	66.7	77.5	44.5
ViPRA (Routray et al., 2026)	7B	79.2	66.7	62.5	**54.2**	54.2	50.0	91.7	79.2	71.9	62.5
WoG (Su et al., 2026)	7B	**95.8**	79.2	75.0	33.0	70.8	50.0	**100.0**	**91.7**	**85.4**	63.5
MoAct	1B	87.5	**83.3**	**79.2**	45.8	75.0	**70.8**	95.8	62.5	84.4	**65.6**

4.1 Main Results

With approximately 1B parameters, MoAct achieves the highest overall task success rates among the compared methods on both Google Robot and WidowX (Tables 1 and 2). By learning compact visual-transition dynamics embedded in future visual context, as illustrated in Figure 3, MoAct models the evolution of future scenes and generates actions consistent with these changes, leading to strong performance.

Evaluation Results on Google Robot. As shown in Table 1, MoAct achieves an overall success rate of 71.7%. It also achieves the highest average success rate under VA at 65.8% and a highly competitive average success rate of 77.6% under VM. Under VA, MoAct's largest task-level advantage appears on Open/Close Drawer, where it reaches a success rate of 44.7%. This task requires coordinated contact and sustained motion along a constrained axis. These demands align with the complementary functions of spatial gating, which localizes relevant changes, and transition-conditioned modulation, which models their evolution. Conditioning successor-feature prediction on the pre-

ceding state further anchors these dynamics to the observed scene. This phenomenon indicates that modeling local visual change can provide effective guidance for constrained manipulation.

Evaluation Results on WidowX. On WidowX, MoAct achieves the highest average task success rate of 65.6% (Table 2). Its average grasp rate of 84.4% is also competitive among the compared methods. Across the four tasks, it attains success rates of 83.3%, 45.8%, 70.8%, and 62.5% on Put Spoon on Towel, Stack Green on Yellow, Put Carrot on Plate, and Put Eggplant in Basket, respectively. These results show that successful grasping does not directly translate into successful placement, because the policy must continue to regulate the object's motion relative to the destination during transport and release. By anticipating these visual transitions, MoAct can guide actions beyond the initial grasp and improve completion of the full manipulation sequence.

4.2 Real-world Experiments

We evaluate MoAct on an AgileX PiPERX tabletop dual-arm platform (Figure 4). Three Intel RealSense D435 cameras provide one external view and two wrist-mounted views. The evaluation suite comprises four manipulation tasks: package storage (PK), sausage placement (SP), bowl stacking (BS), and towel folding (TF). We collect 100 expert robot demonstrations per task for standard fine-tuning and run 50 evaluation episodes per task to compute success rates. For experiments involving human videos, we additionally capture egocentric videos with a GoPro 13 camera. We study matched real-world performance, reduced reliance on robot demonstrations, and target-scene adaptation using egocentric human videos without collecting additional robot demonstrations. Detailed training, mixed-data fine-tuning, and deployment settings are provided in Appendices B.2–B.5.

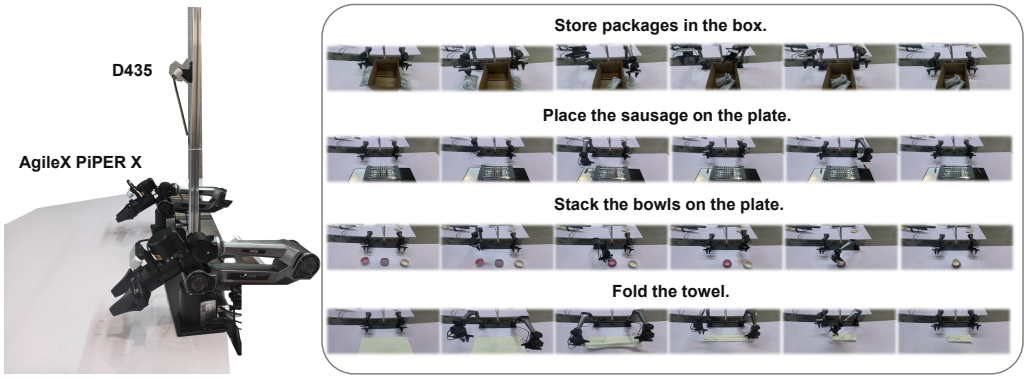

Figure 4: **Real-world setup and tasks.** The figure shows our robotic platform and sensors (left), and the execution of four real-world tasks (right), including package storage, sausage placement, bowl stacking, and towel folding. Frames in each row follow temporal order from left to right.

Real-world Task Performance. For a matched real-world comparison, we fine-tune the officially released $\pi_{0.5}$ base model and MoAct with 100 robot demonstrations for each task and compare the resulting policies. As shown in Fig. 5, MoAct reaches an average success rate of 74.0%, compared with 65.0% for $\pi_{0.5}$. The consistent advantage across the task suite indicates that explicitly modeling visual-transition dynamics can supply useful predictive structure for action generation in physical manipulation.

Reducing Reliance on Expert Demonstrations. We investigate whether egocentric human videos can reduce the robot demonstrations needed for task learning by comparing MoAct trained with 50 expert demonstrations, 50 expert demonstrations plus 50 human videos, and 100 expert demonstrations per task. Representative in-distribution egocentric human-video examples are shown in Appendix C.1. As shown in Fig. 5 (left), adding human videos raises the average success rate from 58.0% to 68.5% compared with using only half as many robot demonstrations. These results suggest that learning visual-transition dynamics from human videos can complement robot action supervision, reducing reliance on expert demonstrations while retaining much of the policy's performance.

Adapting to New Scenes with Human Videos. We further evaluate scene adaptation by adding only egocentric human videos from the target scenes, without collecting additional robot demonstrations. Starting from MoAct trained on 100 robot demonstrations, we retain the existing robot dataset and add 50 egocentric human videos collected in out-of-distribution package-storage scenes. Representative video examples from these target scenes are shown in Appendix C.2. As shown in Fig. 5 (right), the average success rate increases from 66.7% to 76.0% across the three modified scenes. These results support scene adaptation with target-scene egocentric human videos as the only newly collected data, without additional robot demonstrations.

4.3 Ablation Studies

We conduct ablation studies on SimplerEnv to address two questions: (1) Does MoAct's visual-transition learning pipeline improve downstream control over direct action-only fine-tuning? (2) Does including egocentric human videos in visual-transition pretraining further improve downstream control?

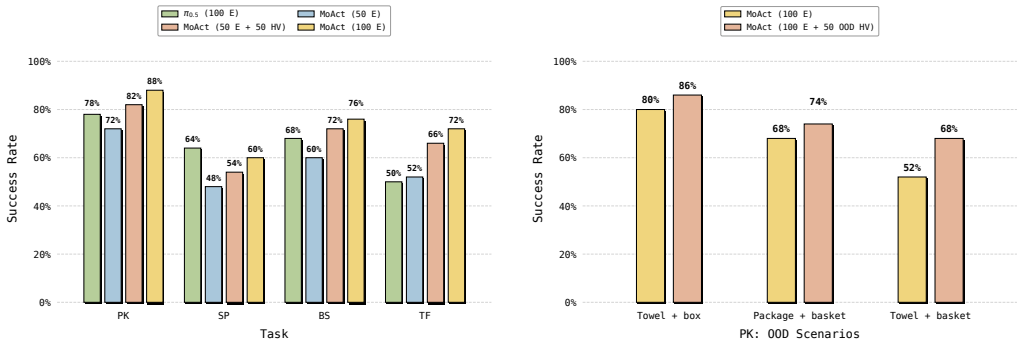

Figure 5: **Real-world task performance.** The left panel compares $\pi_{0.5}$ and MoAct on the four in-distribution tasks. The right panel compares MoAct before and after adaptation to out-of-distribution package-storage scenes using additional egocentric human videos from those scenes. Both settings use the same 100 robot demonstrations, with no additional robot demonstrations collected for adaptation. E denotes expert demonstrations and HV denotes egocentric human videos.

Table 3: **Ablation studies on Google Robot.** Success rates (%) under VM and VA. MoAct (action-only) optimizes only the action-generation objective, while MoAct w/o human videos excludes egocentric human videos from visual-transition pretraining. The highest score in each column is shown in bold.

	VM				VA				Overall
Model	PC	MN	DR	Avg.	PC	MN	DR	Avg.	Avg.
MoAct (action-only)	72.7	58.8	38.4	56.6	61.3	41.5	34.4	45.7	51.2
MoAct w/o human videos	91.0	76.3	52.8	73.3	80.0	62.9	**45.8**	62.9	68.1
MoAct	**96.0**	**77.1**	**59.7**	**77.6**	**87.7**	**65.0**	44.7	**65.8**	**71.7**

Effectiveness of Visual-Transition Learning. To assess the contribution of visual-transition learning in MoAct, we compare the full model with MoAct (action-only). This variant skips Stage 1 visual-transition pretraining and is fine-tuned on downstream robot demonstrations using only the action-generation loss $\mathcal{L}_{\text{act}}$. In both settings, the action expert is randomly initialized before downstream training, and the same downstream robot demonstrations and evaluation protocols are used. Detailed training procedures are provided in Appendix B.6.

As shown in Tables 3 and 4, full MoAct achieves overall success rates of 71.7% on Google Robot and 65.6% on WidowX, compared with 51.2% and 44.8% for MoAct (action-only), respectively.

Table 4: **Ablation studies on WidowX.** Grasp rate (GR) and task success rate (SR) (%). MoAct (action-only) optimizes only the action-generation objective, while MoAct w/o human videos excludes egocentric human videos from visual-transition pretraining. The highest score in each column is shown in bold.

	ST		GY		CP		EB		Overall	
Model	GR	SR	GR	SR	GR	SR	GR	SR	GR	SR
MoAct (action-only)	70.8	54.2	75.0	16.7	66.7	54.2	79.2	54.2	72.9	44.8
MoAct w/o human videos	**95.8**	**91.7**	**79.2**	37.5	**75.0**	62.5	91.7	58.3	**85.4**	62.5
MoAct	87.5	83.3	**79.2**	**45.8**	**75.0**	**70.8**	**95.8**	**62.5**	84.4	**65.6**

These correspond to improvements of 20.5 and 20.8 percentage points. The results support the effectiveness of the proposed visual-transition learning pipeline as a whole for improving downstream control over direct action-only fine-tuning.

Effectiveness of Egocentric Human Videos. To evaluate the contribution of egocentric human videos, we compare the full MoAct model with the MoAct w/o human videos variant. This variant uses the same architecture, robot videos, and downstream robot demonstrations as MoAct, but does not use egocentric human videos during visual-transition pretraining. As shown in Tables 3 and 4, MoAct w/o human videos achieves average success rates of 68.1% on Google Robot and 62.5% on WidowX. Incorporating human videos raises the averages to 71.7% and 65.6%, respectively, suggesting that human manipulation videos improve performance on downstream action-generation tasks by further enriching visual-transition dynamics modeling.

5 Conclusion

In this paper, we introduce MoAct, a VLA framework that learns compact visual-transition dynamics from robot videos and egocentric human videos and uses them to guide continuous action generation. MoAct connects a visual-transition expert and an action expert through a dual-stream MoT and adopts a two-stage training strategy that first learns general visual-transition dynamics and then adapts them using task-relevant robot demonstrations. Simulation and real-world experiments show that visual-transition modeling improves the performance of robot action policies. Furthermore, MoAct demonstrates the potential of using egocentric human videos to reduce robot demonstration requirements and adapt to new target scenes without collecting additional robot demonstrations, paving the way toward next-generation robot policies with greater generality and stronger generalization.

6 Limitations and Future Work

For platforms with more complex interactions, such as bimanual humanoid systems and dexterous hands, current visual-transition representations may still be insufficient to capture their rich manipulation dynamics and may require finer-grained transition modeling. In addition, although target-scene egocentric human videos support scene adaptation without new robot demonstrations, MoAct cannot yet acquire human skills in a zero-shot manner from these videos alone. Future work will further explore in-context learning to strengthen the correspondence between human and robot manipulation.

AI Usage Statement

During the preparation of this manuscript, AI tools were used to improve the writing, including grammar, word choice, and sentence clarity.

REFERENCES

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629. IEEE, 2023.

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

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *arXiv preprint arXiv:2404.08471*, 2024.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, et al. Motus: A unified latent action world model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35101–35113, 2026.

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

Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025a.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2505.06111*, 2025b.

Yi Chen, Yuying Ge, Yizhuo Li, Yixiao Ge, Mingyu Ding, Ying Shan, and Xihui Liu. Moto: Latent motion token as the bridging language for robot manipulation. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2025.

Danny Driess, Jost Springenberg, Brian Ichter, Lili Yu, Adrian Li-Bell, Karl Pertsch, Allen Ren, Homer Walke, Quan Vuong, Lucy Xiaoyang Shi, et al. Knowledge insulating vision-language-action models: Train fast, run fast, generalize better. *Advances in Neural Information Processing Systems*, 38:102867–102888, 2026.

Zipeng Fu, Tony Z Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. *arXiv preprint arXiv:2401.02117*, 2024.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

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

Ryan Hoque, Peide Huang, David Yoon, Jian Zhang, et al. Egodex: Learning dexterous manipulation from large-scale egocentric video. In *International Conference on Learning Representations*, volume 2026, pp. 4218–4237, 2026.

Zhi Hou, Tianyi Zhang, Yuwen Xiong, Haonan Duan, Hengjun Pu, Ronglei Tong, Chengyang Zhao, Xizhou Zhu, Yu Qiao, Jifeng Dai, et al. Dita: Scaling diffusion transformer for generalist vision-language-action policy. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7686–7697. IEEE, 2025.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Tao Jiang, Tianyuan Yuan, Yicheng Liu, Chenhao Lu, Jianning Cui, Xiao Liu, Shuiqi Cheng, Jiyang Gao, Huazhe Xu, and Hang Zhao. Galaxea open-world dataset and g0 dual-system vla model. *arXiv preprint arXiv:2509.00576*, 2025.

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. *arXiv preprint arXiv:2403.12945*, 2024.

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

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, et al. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026.

Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, et al. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024.

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

Chenghao Liu, Jiachen Zhang, Chengxuan Li, Zhimu Zhou, Shixin Wu, Songfang Huang, and Huiling Duan. Ttf-vla: Temporal token fusion via pixel-attention integration for vision-language-action models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 18452–18459, 2026a.

Mingyu Liu, Jiuhe Shu, Hui Chen, Zeju Li, Canyu Zhao, Jiange Yang, Shenyuan Gao, Hao Chen, and Chunhua Shen. Stamo: Unsupervised learning of generalizable robot motion from compact state representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35014–35024, 2026b.

Xiangkai Ma, Lekai Xing, Han Zhang, Wenzhong Li, and Sanglu Lu. Unifying perception and action: A hybrid-modality pipeline with implicit visual chain-of-thought for robotic action generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22380–22390, 2026.

Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. Leworld-model: Stable end-to-end joint-embedding predictive architecture from pixels. *arXiv preprint arXiv:2603.19312*, 2026.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903. IEEE, 2024.

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.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL `https://qwen.ai/blog?id=qwen3.5`.

Sandeep Kumar Routray, Hengkai Pan, Unnat Jain, Shikhar Bahl, and Deepak Pathak. Vipra: Video prediction for robot actions. In *International Conference on Learning Representations*, volume 2026, pp. 68823–68857, 2026.

Younggyo Seo, Danijar Hafner, Hao Liu, Fangchen Liu, Stephen James, Kimin Lee, and Pieter Abbeel. Masked world models for visual control. In *Conference on Robot Learning*, pp. 1332–1344. PMLR, 2023.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Yue Su, Sijin Chen, Haixin Shi, Mingyu Liu, Zhengshen Zhang, Ningyuan Huang, Weiheng Zhong, Zhengbang Zhu, Yuxiao Liu, and Xihui Liu. World guidance: World modeling in condition space for action generation. *arXiv preprint arXiv:2602.22010*, 2026.

Gemini Robotics Team, Abbas Abdolmaleki, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Ashwin Balakrishna, Nathan Batchelor, Alex Bewley, Jeff

Bingham, et al. Gemini robotics 1.5: Pushing the frontier of generalist robots with advanced embodied reasoning, thinking, and motion transfer. *arXiv preprint arXiv:2510.03342*, 2025a.

Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, et al. Gemini robotics: Bringing ai into the physical world. *arXiv preprint arXiv:2503.20020*, 2025b.

Yating Wang, Haoyi Zhu, Mingyu Liu, Jiange Yang, Hao-Shu Fang, and Tong He. Vq-vla: Improving vision-language-action models via scaling vector-quantized action tokenizers. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11089–11099. IEEE, 2025.

Kun Wu, Chengkai Hou, Jiaming Liu, Zhengping Che, Xiaozhu Ju, Zhuqin Yang, Meng Li, Yinuo Zhao, Zhiyuan Xu, Guang Yang, et al. Robomind: Benchmark on multi-embodiment intelligence normative data for robot manipulation. *arXiv preprint arXiv:2412.13877*, 2024.

Fuxiang Yang, Donglin Di, Lulu Tang, Xuancheng Zhang, Lei Fan, Hao Li, Wei Chen, Tonghua Su, and Baorui Ma. Chain of world: World model thinking in latent motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6675–6684, 2026a.

Jiange Yang, Yansong Shi, Haoyi Zhu, Mingyu Liu, Kaijing Ma, Yating Wang, Gangshan Wu, Tong He, and Limin Wang. Como: Learning continuous latent motion from internet videos for scalable robot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 42352–42363, 2026b.

Yi Yang, Xueqi Li, Yiyang Chen, Jin Song, Yihan Wang, Zipeng Xiao, Jiadi Su, You Qiaoben, Pengfei Liu, and Zhijie Deng. Mantis: A versatile vision-language-action model with disentangled visual foresight. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 42505–42515, 2026c.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026.

Chubin Zhang, Jianan Wang, Zifeng Gao, Yue Su, Tianru Dai, Cai Zhou, Jiwen Lu, and Yansong Tang. Clap: Contrastive latent action pretraining for learning vision-language-action models from human videos. *arXiv preprint arXiv:2601.04061*, 2026a.

Chuheng Zhang, Tim Pearce, Pushi Zhang, Kaixin Wang, Xiaoyu Chen, Wei Shen, Li Zhao, and Jiang Bian. What do latent action models actually learn? *Advances in Neural Information Processing Systems*, 38:146676–146697, 2026b.

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

Wenyao Zhang, Bozhou Zhang, Zekun Qi, Wenjun Zeng, Xin Jin, and Li Zhang. Disentangled robot learning via separate forward and inverse dynamics pretraining. *arXiv preprint arXiv:2604.16391*, 2026d.

Zhuoyang Zhang, Shang Yang, Qinghao Hu, Luke J Huang, James Hou, Yufei Sun, Yao Lu, and Song Han. Foreact: Steering your vla with efficient visual foresight planning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 37195–37205, 2026e.

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

Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, et al. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. In *International Conference on Learning Representations*, volume 2026, pp. 60580–60606, 2026.

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

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. Dino-wm: World models on pre-trained visual features enable zero-shot planning. *arXiv preprint arXiv:2411.04983*, 2024.

Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. *arXiv preprint arXiv:2504.02792*, 2025.

A Detailed Model Architecture

A.1 Learning Visual-Transition Representations

The visual-transition dynamics expert represents future scene changes with N transition tokens. Learnable queries $\mathbf{Q} \in \mathbb{R}^{N \times D}$ alternate between cross-attention to the VLM context $\mathbf{H}$ and self-attention, producing $\mathbf{Z} = \{\mathbf{z}_i\}_{i=1}^{N}$. Each token $\mathbf{z}_i$ conditions the prediction of DINOv3 features $\mathbf{F}_i$ from the preceding observed features $\mathbf{F}_{i-1}$. The predictor uses spatial gating to model *where* changes occur and layer-wise modulation to model *how* features change.

Spatial gating assigns a weight to each patch and feature channel. For patch p, the transition token and positional embedding $\mathbf{p}_p$ determine the gate and predictor input:

$$\begin{aligned} \mathbf{g}_{i,p} &= 2\sigma(f_G(\mathrm{SiLU}(f_P(\mathbf{p}_p) + f_Z(\bar{\mathbf{z}}_i)))), \\ \mathbf{u}_{i,p}^{(0)} &= (\bar{\mathbf{f}}_{i-1,p} + \mathbf{p}_p) \odot \mathbf{g}_{i,p}. \end{aligned} \tag{4}$$

Here, bars denote internal feature normalization, f_P, f_Z, and f_G are learned affine maps, and $\odot$ denotes element-wise multiplication. The sigmoid function σ bounds gate values between zero and two, allowing suppression or amplification at each spatial location.

Stacking the gated patch features gives $\mathbf{U}_i^{(0)}$, which passes through four Transformer encoder blocks. After each block, the transition token supplies channel-wise scale and shift parameters:

$$\begin{aligned} \left[\boldsymbol{\gamma}_i^{(m)}; \boldsymbol{\beta}_i^{(m)}\right] &= f_m(\bar{\mathbf{z}}_i), \\ \mathbf{U}_i^{(m)} &= \mathrm{Enc}_m(\mathbf{U}_i^{(m-1)}) \odot (1 + \boldsymbol{\gamma}_i^{(m)}) + \boldsymbol{\beta}_i^{(m)}, \\ \hat{\mathbf{F}}_i &= f_{\mathrm{out}}(\mathbf{U}_i^{(4)}). \end{aligned} \tag{5}$$

Here, $m = 1, \ldots, 4$, each affine map f_m produces parameters shared across patches, and f_{out} applies final normalization and projection. The modulation maps are initialized to zero, making the initial scale-and-shift operation an identity. Together, gating and modulation condition successor-feature prediction on $\mathbf{z}_i$, supervised by the transition objective in Section 3.2. All transition tokens are inferred from the current observation and instruction; future observations provide training targets.

A.2 Conditioning Actions through a Dual-Stream MoT

The dual-stream MoT supplies dynamics information to the action expert through joint attention. At paired layer l, the dynamics expert extracts keys $\mathbf{K}_D^{(l)}$ and values $\mathbf{V}_D^{(l)}$ from the normalized inputs to its self-attention block. Both experts use matching attention-head dimensions, allowing action queries to attend to the combined action and dynamics tokens:

$$\begin{aligned} \bar{\mathbf{K}}^{(l)} &= [\mathbf{K}_A^{(l)}; \mathrm{sg}(\mathbf{K}_D^{(l)})], \\ \bar{\mathbf{V}}^{(l)} &= [\mathbf{V}_A^{(l)}; \mathrm{sg}(\mathbf{V}_D^{(l)})], \\ \mathbf{Y}_A^{(l)} &= \mathrm{softmax}\left(\frac{\mathbf{Q}_A^{(l)} \bar{\mathbf{K}}^{(l)\top}}{\sqrt{d_h}}\right) \bar{\mathbf{V}}^{(l)}. \end{aligned} \tag{6}$$

Here, semicolons concatenate tokens, d_h is the head dimension, and sg stops gradients through the dynamics keys and values. The output undergoes projection, residual addition, and a feed-forward update. Joint attention alternates with cross-attention to $\mathbf{H}$, combining visual-transition information with the current scene and instruction. The dynamics stream receives no action inputs, preserving one-way conditioning from dynamics to action generation.

A.3 Continuous Action Generation

The action expert generates a chunk $\mathbf{a} \in \mathbb{R}^{T_a \times d_a}$, where T_a is the horizon and d_a is the action dimension. Each noisy action is embedded with flow time τ and temporal position j, and an encoded proprioceptive token is prepended:

$$\begin{aligned} \mathbf{e}_{\tau,j} &= E_a(\mathbf{a}_{\tau,j}, \tau) + \mathbf{p}_j^A, \\ \mathbf{X}_A^{(0)} &= [E_s(\mathbf{s}_t); \mathbf{e}_{\tau,1}; \ldots; \mathbf{e}_{\tau,T_a}]. \end{aligned} \tag{7}$$

Here, E_a combines a projected action with a sinusoidal time embedding through a nonlinear projection, E_s is a two-layer state encoder, and $\mathbf{p}_j^A$ is learned. The action Transformer processes these tokens using the MoT interface, and a two-layer decoder maps the action-token outputs to velocities.

Flow matching trains the velocity field along $\mathbf{a}_\tau = (1-\tau)\boldsymbol{\epsilon} + \tau\mathbf{a}$ with target $\mathbf{a} - \boldsymbol{\epsilon}$, as specified in Section 3.2. At inference, K forward Euler steps transform Gaussian noise into an action chunk:

$$\begin{aligned} \mathbf{a}^{(0)} &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \\ \mathbf{a}^{(k+1)} &= \mathbf{a}^{(k)} + \frac{1}{K} v_\theta(\mathbf{a}^{(k)}, k/K; \mathbf{c}_t), \qquad k = 0, \ldots, K-1. \end{aligned} \tag{8}$$

The conditioning $\mathbf{c}_t$ comprises the VLM context, proprioceptive state, and dynamics keys and values. The VLM context and dynamics keys and values are reused throughout integration, without decoding future feature grids.

A.4 Component Configurations and Inference Efficiency

Table 5 lists the architectural hyperparameters of MoAct's components. The action and state dimensions, d_a and d_s, depend on the robot platform.

Table 5: **Detailed component configurations of MoAct.** The feature predictor and frozen visual encoder provide visual-transition supervision during training.

Component	Configuration
VLM backbone	Qwen3.5-0.8B; input sequence length capped at 320 tokens.
Visual-transition dynamics expert	8 learnable transition tokens; 16 layers (8 cross-attention and 8 self-attention); hidden dimension 1152; 12 attention heads with head dimension 64; feed-forward dimension 4608; dropout 0.2.
Flow-matching action expert	16 layers (8 cross-attention and 8 joint-attention); hidden dimension 768; 12 attention heads with head dimension 64; feed-forward dimension 3072; dropout 0.2.
State encoder and action decoder	Two-layer MLPs with hidden dimension 1024: $d_s \to 1024 \to 768$ for state encoding and $768 \to 1024 \to d_a$ for action decoding.
Action and time encoder	Action projection $d_a \to 768$; concatenation with a 768-dimensional sinusoidal time embedding; projections $1536 \to 768 \to 768$; learned temporal position embeddings.
Visual feature predictor	4 Transformer encoder layers; hidden dimension 1152; 6 attention heads; feed-forward dimension 4608; dropout 0.0; spatial gating before the encoder and transition-conditioned modulation after each layer. Prediction targets use frame offsets $[4, 8, 12, 16, 20, 24, 28, 32]$.
Frozen visual encoder	DINOv3 ViT-S/16; 256×256 input; 16×16 patch grid with 384 channels per view. Up to 3 view slots are concatenated into 1152-dimensional predictor inputs.

During inference, MoAct computes the VLM context and dynamics keys and values once and reuses them throughout flow integration. The frozen DINOv3 encoder and visual feature predictor are not evaluated for action generation. We measure inference cost under BF16 inference on an NVIDIA RTX 4090 GPU with 48 GB of memory. Under this configuration, generating a 32-step action chunk takes 54.8 ms, including all 10 forward Euler integration steps, with a peak GPU memory usage of 2.9 GB.

B Detailed Dataset, Training, and Evaluation Settings

B.1 Pretraining Data Mixture

Table 6: **Pretraining data mixture.** Episode counts refer to the configured corpus after deduplication. Sampling proportions are rounded to one decimal place.

Dataset	Embodiment	Episodes	Data type	Sampling (%)
Open X-Embodiment (O'Neill et al., 2024)	Multiple robots	608,034	Robot demonstrations	17.0
RoboMIND (Wu et al., 2024)	Single- and dual-arm robots	40,747	Robot demonstrations	25.5
AgiBotWorld (Bu et al., 2025a)	Dual-arm robot	11,694	Robot demonstrations	17.0
DROID (Khazatsky et al., 2024)	Single-arm robot (Franka)	95,658	Robot demonstrations	12.8
ALOHA Mobile (Fu et al., 2024)	Mobile dual-arm robot	278	Robot demonstrations	3.8
ALOHA Static (Zhao et al., 2023)	Stationary dual-arm robot	790	Robot demonstrations	8.9
EgoDex (Hoque et al., 2026)	Human	314,839	Egocentric human videos	15.0

B.2 Training Settings

Table 7: **Training settings.** LR denotes the peak learning rate, and batch sizes are global. All experiments are trained on 32 NVIDIA H200 GPUs. Stage 2 jointly optimizes $\mathcal{L}_{\text{joint}} = \mathcal{L}_{\text{trans}} + \mathcal{L}_{\text{act}}$; the action expert is reinitialized for initial Stage 2 training and retained when continuing from a trained MoAct checkpoint for scene adaptation. DINOv3 remains frozen in both stages.

	Stage 1	Stage 2	
Setting	Pretraining	Simulation	Real world
Training objective	$\mathcal{L}_{\text{trans}}$	$\mathcal{L}_{\text{joint}}$	$\mathcal{L}_{\text{joint}}$
Training steps	100,000	50,000	20,000
Global batch size	1,024	1,024	1,024
VLM backbone LR	10^{-5}	10^{-5}	10^{-5}
Visual dynamics LR	10^{-4}	10^{-5}	10^{-5}
Action expert LR	–	5×10^{-5}	5×10^{-5}
Shared optimization settings			
Optimizer	AdamW		
AdamW (β_1, β_2)	$(0.9, 0.95)$		
AdamW ϵ	10^{-8}		
Weight decay	10^{-8}		
LR schedule	Linear warmup, then cosine decay		
Warmup duration	5% of training steps		
Minimum LR	10% of the peak LR		
Gradient clipping	Maximum gradient norm of 1.0		

B.3 Stage 2 Mixed-Data Fine-tuning

For the initial Stage 2 fine-tuning, we jointly train MoAct with robot expert demonstrations and egocentric human videos. Let $\mathcal{D}_{\mathrm{R}}$ denote the robot demonstration dataset and $\mathcal{D}_{\mathrm{H}}$ denote the egocentric human-video dataset. Both datasets provide supervision for visual-transition dynamics through adjacent observations, whereas only robot demonstrations provide action supervision. Thus, human videos contribute to $\mathcal{L}_{\mathrm{trans}}$ but are excluded from $\mathcal{L}_{\mathrm{act}}$. For subsequent scene adaptation, we retain all parameters from the existing MoAct checkpoint, reuse $\mathcal{D}_{\mathrm{R}}$ unchanged, and add only newly collected egocentric human videos from the target scenes to $\mathcal{D}_{\mathrm{H}}$. The two losses are combined using the joint objective defined in Section 3.2.

Algorithm 1 Stage 2 mixed-data fine-tuning with robot and egocentric human videos.

Require: Stage 1 checkpoint or existing MoAct checkpoint; robot demonstrations $\mathcal{D}_{\mathrm{R}}$; egocentric human videos $\mathcal{D}_{\mathrm{H}}$
Ensure: Fine-tuned MoAct model

if initial Stage 2 fine-tuning **then**
  Initialize the VLM, visual-transition dynamics expert, and feature predictor from Stage 1
  Initialize the action expert anew
else ▷ Scene adaptation
  Load all model parameters from the existing MoAct checkpoint
  Reuse $\mathcal{D}_{\mathrm{R}}$ unchanged; let $\mathcal{D}_{\mathrm{H}}$ contain newly collected target-scene videos
end if
Keep DINOv3 frozen
for each Stage 2 training iteration **do**
  Sample robot clips $\mathcal{B}_{\mathrm{R}}$ and human clips $\mathcal{B}_{\mathrm{H}}$ according to the configured mixture ratio
  Form the mixed video batch $\mathcal{B} = \mathcal{B}_{\mathrm{R}} \cup \mathcal{B}_{\mathrm{H}}$
  Compute $\mathcal{L}_{\mathrm{trans}}$ on $\mathcal{B}$ using adjacent-observation feature prediction
  Sample Gaussian noise and flow time for action chunks in $\mathcal{B}_{\mathrm{R}}$
  Compute $\mathcal{L}_{\mathrm{act}}$ on robot demonstrations only using flow matching
  Form $\mathcal{L}_{\mathrm{joint}} = \mathcal{L}_{\mathrm{trans}} + \mathcal{L}_{\mathrm{act}}$
  Update the VLM, visual-transition dynamics expert, feature predictor, and action expert
end for

B.4 Simulation Evaluation Settings

We evaluate MoAct on SimplerEnv (Li et al., 2024), a real-to-sim benchmark, using its Google Robot and WidowX task suites under BF16 inference. The Google Robot suite covers Pick Coke Can, Move Near, and Drawer, while the WidowX suite comprises four manipulation tasks. The policy receives 256×256 RGB observations, task instructions, and normalized proprioceptive states. Episode horizons, action post-processing, temporal ensembling, and score aggregation follow the protocol summarized in Algorithm 2.

Algorithm 2 Simulation evaluation and inference protocol.

Require: Policy π_θ, task instruction ℓ, environment $\mathcal{E}$, benchmark $b \in \{\text{Google Robot}, \text{WidowX}\}$
Ensure: Episode success flag y

1: Reset $\mathcal{E}$; set the horizon $H \leftarrow 80$ (pick/move), 113 (drawer), or 120 (WidowX)
2: Set the ensemble window $M \leftarrow 2$ for Google Robot and $M \leftarrow 7$ for WidowX; initialize the action queue
3: **for** $t = 0, \ldots, H-1$ until task termination **do**
4: Observe RGB image o_t and normalized proprioceptive state s_t
5: Encode (o_t, ℓ, s_t) with π_θ and sample $a^{(0)} \sim \mathcal{N}(0, I)$
6: **for** $k = 0, \ldots, 9$ **do** ▷ 10-step forward Euler solver
7: $a^{(k+1)} \leftarrow a^{(k)} + \frac{1}{10} v_\theta\big(a^{(k)}, \frac{k}{10}; c_t\big)$
8: **end for**
9: Store the resulting 16-step action chunk and align it with the latest M chunks
10: For each aligned action, compute cosine similarity s_j to the newest chunk and use $w_j \propto \exp(0.1 s_j)$
11: Average the aligned actions with weights w_j; convert Euler rotation increments to axis–angle commands
12: Apply benchmark-specific gripper processing and execute the first action; discard expired actions
13: **end for**
14: Set y according to the environment-defined success criterion

Google Robot episodes run for 80 control steps for picking and moving and 113 for drawer manipulation; WidowX episodes run for 120 steps. Google Robot uses relative gripper commands with a change threshold of 0.5 and a ten-step hold, whereas WidowX binarizes the ensembled gripper command at a threshold of 0.5. Google Robot scores are averaged hierarchically across episodes, variants, and task subgroups, with Drawer averaging opening and closing performance. We report unweighted means across the three Google Robot task categories within each evaluation setting and across the four WidowX tasks. For WidowX, the final-step success flag is used.

B.5 Real-world Evaluation Settings

For real-world evaluation, we retain the simulation settings for observation preprocessing, BF16 inference, flow integration, and action execution unless specified otherwise. The action expert predicts a 32-step chunk of absolute joint actions. To support asynchronous inference, we enable training-time real-time chunking (RTC) with an RTC window of 8. The controller executes 16 actions from the active chunk before requesting the next prediction, while the RTC context carries the most recent 8-step window across successive chunks. The complete closed-loop procedure is given in Algorithm 3.

Algorithm 3 Real-world evaluation and asynchronous inference protocol.

Require: Policy π_θ, task instruction ℓ, robot controller $\mathcal{R}$
Ensure: Episode success flag y

1: Enable training-time RTC with window size $W_{\mathrm{RTC}} \leftarrow 8$; reset $\mathcal{R}$ and initialize the active action chunk
2: **while** the task is not terminated **do**
3: Observe the camera images and robot state; apply the same preprocessing as in Algorithm 2
4: Start or continue asynchronous inference with the latest W_{RTC}-step RTC context
5: Sample $a^{(0)} \sim \mathcal{N}(0, I)$ and apply 10 forward Euler updates to obtain a 32-step absolute-joint chunk
6: Execute the next 16 actions from the active chunk while the next chunk is inferred asynchronously
7: When the new chunk is ready, replace the active chunk and update the RTC context with the latest 8 actions
8: **end while**
9: Set y according to the task success criterion

All remaining evaluation conventions, including the input resolution, solver steps, and reporting procedure, follow the simulation protocol unless stated above; the predicted absolute joint actions are sent directly to the robot controller.

B.6 Training Protocol for the MoAct (action-only) Variant

The MoAct (action-only) variant evaluates the overall contribution of the visual-transition learning pipeline. It initializes its VLM backbone from the same original pretrained checkpoint used to initialize full MoAct before Stage 1 and does not inherit the VLM parameters obtained after visual-transition pretraining. It skips Stage 1 visual-transition pretraining and does not use adjacent-observation feature-prediction targets.

The variant is fine-tuned directly on downstream robot demonstrations using only the action-generation loss $\mathcal{L}_{\mathrm{act}}$. Like full MoAct, its action expert is randomly initialized before downstream training. During this fine-tuning stage, the transition loss $\mathcal{L}_{\mathrm{trans}}$ and all future-frame targets are omitted. The downstream robot demonstrations and evaluation protocols are kept identical to those used for the full MoAct model.

C More Data Cases

C.1 In-distribution Demonstration Data

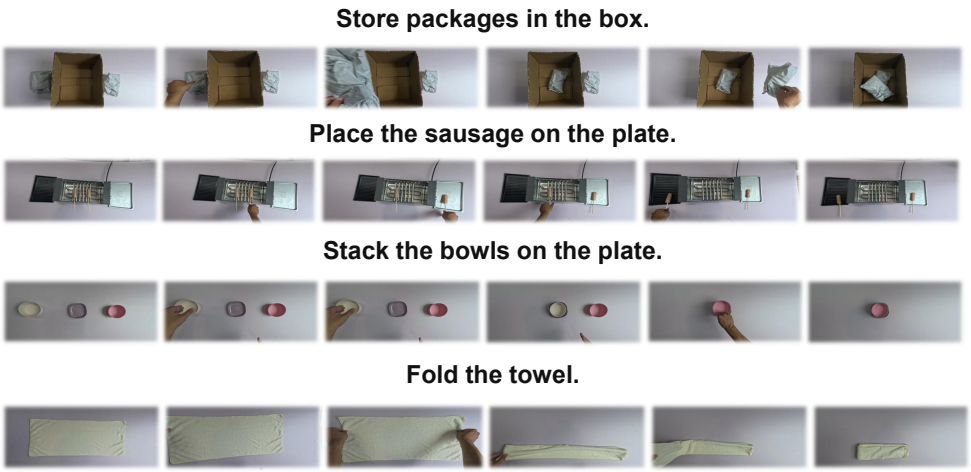

Figure 6: **In-distribution human demonstration data.** Representative egocentric human-video frames for package storage, sausage placement, bowl stacking, and towel folding.

C.2 Out-of-distribution Demonstration Data

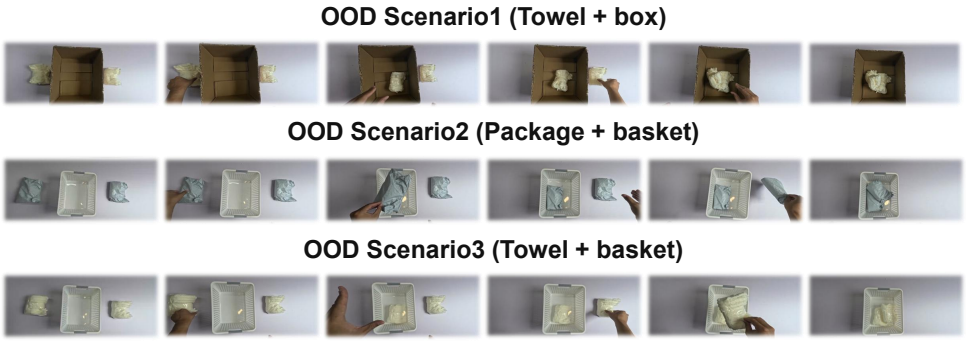

Figure 7: **Out-of-distribution human demonstration data.** Representative egocentric human-video frames from three out-of-distribution package-storage scenarios: towel + box, package + basket, and towel + basket.