EVO-WORLD: A LIGHTWEIGHT VISION-LANGUAGE-ACTION MODEL WITH FUTURE MOTION PREDICTION VIA OPTICAL-FLOW SUPERVISION

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language-action (VLA) models provide a unified framework for visual understanding, language grounding, and robot control, yet anticipating how the scene will evolve beyond the current observation can provide additional information for control. Action supervision alone does not explicitly encourage such anticipation, while future-image prediction entangles task-relevant motion with largely static appearance. To address these limitations, we introduce **Evo-World**, a lightweight VLA model that augments policy learning with future motion prediction through optical-flow supervision, thereby providing a task-relevant predictive signal focused on motion rather than appearance. We further design the training procedure to preserve these predictive representations during policy learning, regulating representation diversity while decoupling representation learning from action optimization. Across simulation benchmarks, Evo-World demonstrates strong performance, achieving a tier-averaged success rate of **91.1%** on Meta-World MT50 , with a 13.0% relative improvement over the non-predictive baseline. On real-world AgileX robots, Evo-World outperforms the evaluated baselines across single-arm and bimanual manipulation tasks while using only **0.87B** parameters and delivering lower GPU memory usage and higher inference efficiency. These results demonstrate that future motion prediction through optical-flow supervision provides an effective and lightweight approach to enhancing the predictive capabilities of VLA models.

1 INTRODUCTION

Vision-language-action (VLA) models provide a unified framework for visual understanding, language grounding, and robot control. By adapting pretrained visual and linguistic knowledge to action generation, they support a broad range of manipulation behaviors (Brohan et al., 2023; Kim et al., 2024; Black et al., 2026). Successful manipulation also depends on anticipating how a scene should change as a task unfolds: grasping, placing, and opening require coordinated changes in the robot and its surroundings. Yet action supervision alone does not explicitly require a policy's representations to anticipate these changes. Providing this predictive structure is therefore an opportunity to improve VLA policies beyond their understanding of the current observation.

Future-observation prediction offers temporal supervision for learning such anticipation (Wu et al., 2024; Cheang et al., 2024), but the prediction target determines which aspects of scene evolution the model must represent. Future RGB images entangle motion with appearance, including background textures and objects that remain largely unchanged. Preserving these details can support visual recognition, but it does not directly emphasize the changes needed for control. Optical flow provides a complementary target by describing image-space displacement while suppressing static texture. This motivates learning future motion representations that complement the policy's visual and linguistic context, as illustrated in Figure 1.

We introduce **Evo-World**, a lightweight VLA model that augments policy learning with future motion prediction through optical-flow supervision. Evo-World learns to anticipate motion from the current observation and task instruction, and uses this prediction to inform action generation. The key idea is to make scene change an effective learning target while retaining the appearance and semantics

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

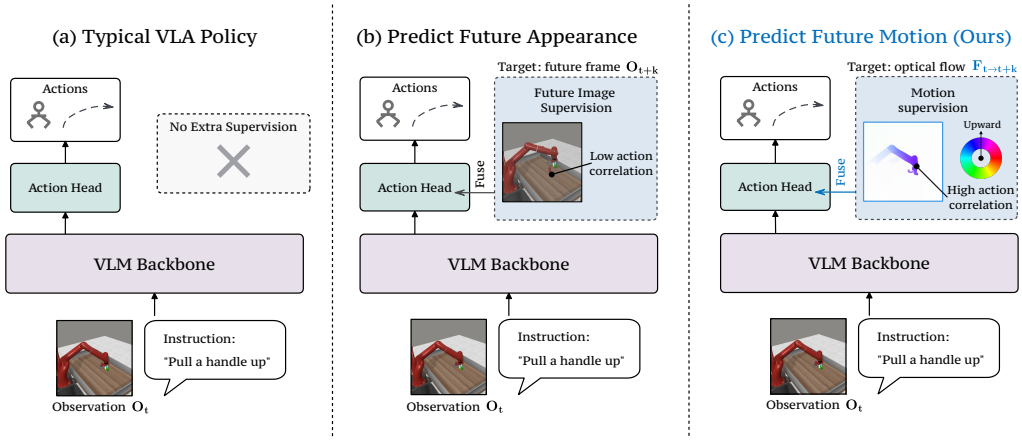

Figure 1: **Prediction targets for a VLA policy.** (a) An action-only policy; (b) prediction of future appearance; and (c) prediction of future motion with optical-flow supervision. The two predictive branches condition action generation on latent representations. The flow illustration depicts the training target. The action-correlation labels express the motivation for motion supervision qualitatively.

available to the vision-language backbone. Optical flow computed from demonstration videos supplies supervision during training; a compact predictive branch learns its latent representation.

For the predictive signal to remain useful, policy learning must preserve informative motion representations. We address two complementary requirements: maintaining representation diversity and preserving the motion-learning objective during action optimization. We adopt Sketched Isotropic Gaussian Regularization (SIGReg) from LeJEPA to discourage collapse of learned representations (Balestriero & LeCun, 2025). We also separate the gradient paths for representation learning and action optimization, allowing the policy to learn how to use the motion features while the predictive branch is trained by its own objective.

Evo-World contains only **0.87B** parameters, including its predictive branch. At deployment, it uses current camera observations, the instruction, and robot state; future observations and optical-flow estimation are unnecessary. Motion prediction operates in a latent space, avoiding the need to decode future images or flow fields during action generation. This design incorporates anticipation within a compact policy suitable for deployment on a physical robot.

We evaluate Evo-World on Meta-World MT50, RoboTwin, and three real-world manipulation tasks (Yu et al., 2020)(Chen et al., 2025a). It achieves **91.1%** tier-averaged success on Meta-World, a 13.0% relative improvement over the non-predictive baseline, and **41.8%** average success across Clean2Clean and Clean2Random on RoboTwin (Table 1). On an AgileX dual-arm robot, Evo-World outperforms OpenVLA-OFT and π_0 on all three tasks, with 63.3% average success (Table 2). These tasks cover single-arm floral arrangement and bimanual collaboration in mug hanging and microwave placement. Component ablations support the training design: regularization counters representation contraction, and both regularization and gradient separation improve aggregate success in the tested setting.

Our contributions are threefold:

1. We introduce optical-flow-supervised future motion prediction into a lightweight VLA policy, providing a strong predictive signal that complements visual understanding and language grounding.
2. We combine distribution regularization with separate gradient paths to preserve predictive representations during policy learning, and integrate these representations into action generation through a compact conditioning module.
3. We conduct comprehensive experiments across simulation and real-world robotic manipulation tasks. With a compact 0.87B-parameter model, Evo-World achieves strong task performance while retaining low computational overhead.

2 Related Work

2.1 Vision-language-action policies and efficient deployment

VLA models connect visual understanding and language grounding to robot control. RT-2 and OpenVLA adapt pretrained vision-language models by predicting action tokens (Brohan et al., 2023; Kim et al., 2024). For continuous control, Diffusion Policy learns action sequences through denoising (Chi et al., 2025) and π_0 integrates a flow-matching action expert with a vision-language backbone (Black et al., 2026). SmolVLA and Evo-1 emphasize compact architectures and efficient adaptation of multimodal features (Shukor et al., 2025; Lin et al., 2026d). These advances improve policy capacity and deployment practicality, but action supervision does not explicitly require anticipation of future scene changes.

2.2 Predictive learning for robot control

Predictive learning supplements action labels with supervision about how observations evolve. GR-1 and GR-2 learn future RGB observations alongside actions (Wu et al., 2024; Cheang et al., 2024). V-JEPA 2 learns video embeddings through masked prediction and action-conditioned latent dynamics for planning (Assran et al., 2025). Moto learns discrete motion tokens through conditional frame reconstruction and transfers motion priors to control (Chen et al., 2025b). These methods connect visual prediction to control through both generative and latent representations. The prediction target determines which scene properties the model must preserve: RGB reconstruction couples motion with appearance, while learned representations inherit the structure of their supervision. Motion-specific targets offer a complementary way to emphasize displacement.

2.3 Motion supervision for manipulation

Motion guides manipulation through point trajectories (Wen et al., 2023; Bharadhwaj et al., 2024), predicted flow (Xu et al., 2024; Lin et al., 2026a), and motion-guided visual prediction (Gao et al., 2025; Guo et al., 2026; Zhong et al., 2026; Wei et al., 2026). Recent flow-supervised policies use this signal in different roles. Fang et al. (2025) augment VLA training with motion-image diffusion in a frozen VAE latent space. This auxiliary head supervises the shared VLM and is omitted from policy inference. FlowWAM jointly models RGB and flow latents, using generated flow for action prediction and supplied flow to condition world modeling (Chen et al., 2026). LaMP supplies future 3D motion priors through a frozen expert's one-step denoising features, avoiding full motion reconstruction (Wang et al., 2026). TemporalFlow-VLA instead supervises execution-history queries with robot-surface flow derived from kinematics and camera calibration (Yang et al., 2026a).

Evo-World predicts future motion by directly regressing learned embeddings of 2D optical-flow renderings from the current observation and instruction. These embeddings remain available to action generation without motion-denoising or flow-reconstruction objectives. The motion branch continues optimizing its prediction objective and distribution regularization (Balestriero & LeCun, 2025), with action gradients blocked; bounded FiLM combines current and predicted embeddings to condition the policy (Perez et al., 2018). Its RGB-derived targets require neither depth nor robot-geometry labels.

3 Method

Given camera observations O_t, a language instruction ℓ, and robot state s_t, Evo-World predicts an action chunk $a_{t:t+H-1}$. Figure 2 gives an overview of the framework, which we describe in the following four parts.

3.1 Motion Representation Module

This world-model branch predicts latent motion from one designated view O_t^w and the instruction. Encoders E_I and E_T, initialized from DINOv2 and CLIP, respectively, followed by projection heads P_I and P_T, produce D-dimensional features $z_t = P_I(E_I(O_t^w))$ and $e_\ell = P_T(E_T(\ell))$ (Oquab et al., 2023; Radford et al., 2021). A multilayer perceptron predicts instruction-conditioned demonstration

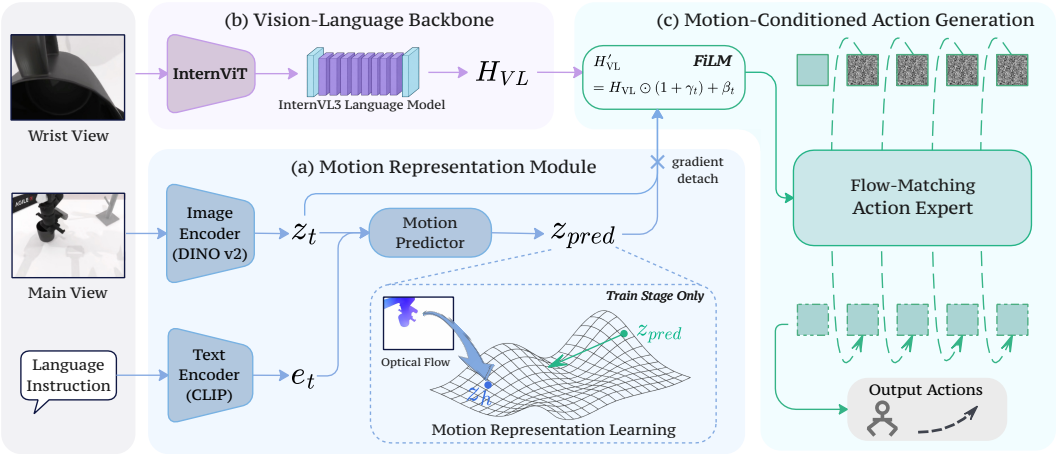

Figure 2: **Evo-World architecture.** (a) The Motion Representation Module predicts a latent motion representation from DINOv2 image features and CLIP instruction features. In the dashed box, DINOv2 encodes an optical-flow RGB rendering as a target used only during training. (b) The InternVL3-1B vision-language backbone provides visual and linguistic context. (c) The current and predicted representations are concatenated and detached from action gradients, then projected into FiLM scale and shift parameters. These parameters modulate the backbone features to condition the flow-matching action expert for action generation.

motion over an interval Δ as $z_{\text{pred}} = P_M([z_t; e_\ell])$. The pair (z_t, z_{pred}) represents the current scene and anticipated motion, without candidate-action conditioning or flow decoding.

During training, the prediction target is encoded from a three-channel optical-flow rendering. We precompute $F_{t\to t+\Delta} = \text{RAFT}(O^w_t, O^w_{t+\Delta})$ with $\Delta = 15$ (Teed & Deng, 2020). A renderer $\mathcal{R}$ with a fixed magnitude scale produces $Y_t = \mathcal{R}(F_{t\to t+\Delta})$, which is encoded as $z_h = P_I(E_I(Y_t))$. The current and target images share E_I, P_I. The optical-flow processing results for RoboTwin and Meta-World are presented in Figure 3. The target pathway is used only during training.

3.2 Vision-Language Backbone

The available camera views are first encoded into visual tokens by the InternViT vision encoder. These visual tokens are then combined with the instruction ℓ and processed by the InternVL3 LLM to obtain the joint vision-language representation $H_{\text{VL}} = E_{\text{LLM}}(E_{\text{ViT}}(O_t), \ell)$, where E_{ViT} denotes the InternViT encoder and E_{LLM} denotes the InternVL3 LLM. The resulting tokens provide visual and linguistic context for action generation (Lin et al., 2026d; Zhu et al., 2025).

3.3 Motion-Conditioned Action Generation

FiLM computes scale and shift from the current and predicted representations as $(\gamma_t, \beta_t) = \alpha \tanh(G_\phi([\text{sg}(z_t); \text{sg}(z_{\text{pred}})]))$ and modulates the vision-language tokens (Perez et al., 2018):

$$H'_{\text{VL}} = H_{\text{VL}} \odot (1 + \gamma_t) + \beta_t. \tag{1}$$

Modulation is bounded by $\alpha = 0.1$ and broadcast across tokens. The generator's final layer is zero-initialized, giving identity modulation. Stop-gradient sg blocks action gradients into the motion representation module while allowing the action loss to train FiLM and the policy.

The action expert A_θ is a flow-matching transformer (Lin et al., 2026d; Lipman et al., 2022). It embeds noisy actions as tokens with temporal positions, cross-attends to H'_{VL} and an MLP-encoded state token, and injects a time embedding into each feed-forward block. An output MLP maps the aggregated action tokens to chunk velocities.

For a normalized action chunk a, draw $u \sim \mathcal{U}(-1, 1)$ and set $\widetilde{u} = M \odot u$, where the binary mask M excludes padded action coordinates. During training, we sample $\tau \sim \mathrm{Beta}(2, 2)$, clip it to $[0.02, 0.98]$, and interpolate $x_\tau = (1 - \tau)\widetilde{u} + \tau a$. The expert predicts the velocity $\hat{v}_\theta = A_\theta(x_\tau, \tau, s_t, H'_{\mathrm{VL}})$.

At inference, we start from $x^0 = \widetilde{u}$ and apply N forward-Euler steps:

$$x^{k+1} = M \odot \left[x^k + \frac{1}{N} A_\theta(x^k, k/N, s_t, H'_{\mathrm{VL}}) \right]. \tag{2}$$

Here, $k = 0, \ldots, N - 1$, and the resulting chunk is $\hat{a} = x^N$. The motion representation module remains active; no future image or optical-flow estimator is required.

3.4 Training Objectives

World loss. For a batch of B samples, we jointly optimize a prediction loss and a regularization loss as world loss. The prediction loss measures the discrepancy between the predicted and target motion representations, with gradients retained on both sides. To discourage representation collapse, we further apply Sketched Isotropic Gaussian Regularization (SIGReg) to both the current and target batch representations z_t and z_h (Balestriero & LeCun, 2025):

$$\mathcal{L}_{\mathrm{pred}} = \frac{1}{BD} \sum_{b=1}^{B} \| z_{\mathrm{pred},b} - z_{h,b} \|_2^2, \qquad \mathcal{L}_{\mathrm{SIGReg}} = \frac{1}{2} \left[\mathcal{S}(z_t) + \mathcal{S}(z_h) \right]. \tag{3}$$

The statistic $\mathcal{S}$ matches characteristic functions of random one-dimensional projections to a standard Gaussian; its discrete form is given in the Appendix A.2. SIGReg directly regularizes z_t, z_h, influencing z_{pred} through the prediction loss.

Action loss. With target velocity $v^* = a - \widetilde{u}$, we average the squared velocity error over valid coordinates j, including the chunk dimension:

$$\mathcal{L}_{\mathrm{action}} = \frac{\sum_{b,j} M_{b,j} (\hat{v}_{\theta,b,j} - v^*_{b,j})^2}{\sum_{b,j} M_{b,j} + 10^{-8}}. \tag{4}$$

Joint objective. We optimize

$$\mathcal{L} = \mathcal{L}_{\mathrm{action}} + \lambda_{\mathrm{world}} \left(\mathcal{L}_{\mathrm{pred}} + \lambda_{\mathrm{sig}} \mathcal{L}_{\mathrm{SIGReg}} \right). \tag{5}$$

4 Experiments

We evaluate the performance of manipulation on Meta-World MT50, RoboTwin, and three real-world tasks on an AgileX dual-arm robot. Meta-World and RoboTwin are reported together in a single simulation table (Table 1): Meta-World provides the main quantitative setting across four difficulty tiers, while RoboTwin tests the model in a separate training and deployment setting. Real-world experiments assess performance on a physical robot.

4.1 Meta-World and RoboTwin Experimental setup

Evo-World is initialized from general-purpose pretrained encoders and trained on four NVIDIA RTX 4090 GPUs, without additional robot-policy pretraining. Detailed configurations are provided in Appendix Table A1.

Optical-flow supervision. Figure 3 illustrates the optical-flow renderings used to supervise the motion representation module on both simulation benchmarks. Compared with the corresponding RGB observations, these renderings emphasize image-space displacement and suppress static scene appearance. They serve as training targets for latent motion prediction; no optical-flow estimator is required at inference.

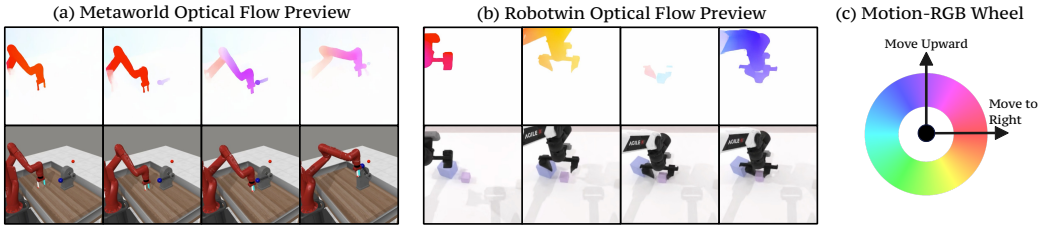

Figure 3: **Optical-flow supervision on Meta-World and RoboTwin.** Panels (a) and (b) pair optical-flow renderings (top) with corresponding RGB observations (bottom) from Meta-World and RoboTwin, respectively. Colored regions emphasize motion, while nearly static regions appear close to white. Panel (c) shows the motion-to-color mapping: hue indicates displacement direction, and weaker motion produces colors closer to white.

Baselines. For Meta-World we compare with π_0 (Black et al., 2026), SmolVLA (Shukor et al., 2025), RoboTron-Mani (Yan et al., 2024), Evo-1 (Lin et al., 2026d), Evo-Depth (Lin et al., 2026b), LA4VLA (Lin et al., 2026c), and FabriVLA (Yang et al., 2026b). For RoboTwin we compare against DP (Chi et al., 2025), ACT (Zhao et al., 2023), RDT(Liu et al., 2025), DP3(Ze et al., 2024), starVLA (Community, 2026), AHA-WAM(Cai et al., 2026a), GalaxeaVLA(Liu et al., 2026), and Xiaomi Robotics-0 (Cai et al., 2026b). Table 1 also lists parameter counts and whether large-scale robot-data pretraining was used. The comparison contextualizes performance.

4.2 Meta-World and RoboTwin results

Table 1 reports both simulation benchmarks. On Meta-World MT50, Evo-World reaches a tier-averaged success rate of 91.1%, which is 10.5 percentage points above the reported Evo-1 score of 80.6% and 1.1 points above FabriVLA's 90.0%. Evo-World obtains the best score on the Hard tier and on the tier average, and the second-best score on the other three tiers, so the four tier scores are consistently high rather than driven by a single difficulty level. On RoboTwin, Evo-World achieves a Clean success rate of 67.1% and a Random success rate of 16.5%. Averaged across the Clean and Random settings, Evo-World achieves the highest overall success rate of 41.8% among the compared methods. Notably, Evo-World uses about 0.87B parameters and no robot-data pretraining, yet exceeds larger and robot-pretrained baselines such as π_0 (3.5B) and LA4VLA (1.0B) on the Meta-World average. Example episode trajectories from the simulation benchmarks are provided in the Appendix E.

4.3 Real-world experiments

We evaluate Evo-World on an AgileX dual-arm robot in three tasks covering both single-arm manipulation and bimanual collaboration (Figure 4). Floral arrangement is a single-arm task in which one arm inserts a flower stem into a vase. Mug hanging and microwave placement are bimanual collaboration tasks: the two arms work together to hang mugs on a rack or to open a microwave door, put the plate inside, and close the door, respectively. Appendix F describes the hardware setup, demonstration collection, and supervised fine-tuning protocol.

We compare Evo-World with OpenVLA-OFT (Kim et al., 2025) and π_0 (Black et al., 2026). Each model is evaluated for 20 trials per task, and we report task success rates and their unweighted mean. All three models run on the same NVIDIA H20 GPU in BF16 precision. We also report parameter count, GPU memory usage, and measured inference frequency for each deployment configuration.

Evo-World achieves the highest success rate on all three tasks: 55.0% for floral arrangement, 70.0% for mug hanging, and 65.0% for microwave placement. Its average success rate is 63.3%, compared with 51.7% for π_0 and 23.3% for OpenVLA-OFT. The gains over π_0 are 15.0, 15.0, and 5.0 percentage points, respectively, covering both single-arm manipulation and bimanual collaboration.

With 0.87B parameters, Evo-World uses 2.54 GB of GPU memory and achieves a measured inference frequency of 6.16 Hz, giving the lowest memory usage and highest inference frequency among the evaluated models. Relative to π_0, it uses 62.0% less GPU memory and has a 13.7% higher inference

Table 1: **Meta-World MT50 and RoboTwin success rates (%).** Meta-World reports success within four difficulty tiers and their equal-weight tier average; RoboTwin reports the Clean and Randomized settings and their average. Params are in billions, and Pretrain marks large-scale robot-data pretraining. The best score in each column is **bold** and the second-best is underlined; "–" denotes a value not reported or not yet evaluated. Evo-World (Ours) is evaluated as described above.

			Meta-World MT50					RoboTwin		
Method	Params	Pretrain	Easy	Medium	Hard	Very Hard	Avg	Clean	Random	Avg
DP	0.1B	No	23.1	10.7	1.9	6.1	10.5	28.0	0.6	14.3
ACT	0.08B	No	–	–	–	–	–	29.7	1.7	15.7
RDT	1.2B	Yes	–	–	–	–	–	34.5	13.7	24.1
starVLA	4.68B	No	85.7	60.0	46.7	44.0	59.1	46.5	3.2	24.8
DP3	0.26B	No	–	–	–	–	–	55.2	5.0	30.1
π_0	3.5B	Yes	71.8	48.2	41.7	30.0	47.9	46.4	16.3	31.4
AHA-WAM	7.35B	No	–	–	–	–	–	64.3	3.2	33.8
GalaxeaVLA	2.86B	Yes	–	–	–	–	–	62.7	12.7	37.7
Xiaomi Robotics-0	6.7B	Yes	–	–	–	–	–	62.9	**18.2**	40.5
SmolVLA	2.25B	No	87.1	51.8	70.0	64.0	68.2	–	–	–
RoboTron-Mani	4B	Yes	85.5	67.7	76.7	81.0	77.7	–	–	–
Evo-1	0.77B	No	89.2	76.8	77.2	79.2	80.6	65.7	12.5	39.1
Evo-Depth	0.9B	No	83.1	84.7	87.3	82.4	84.4	–	–	–
LA4VLA	1.0B	Yes	88.9	**94.5**	66.7	**100.0**	87.5	–	–	–
FabriVLA	0.88B	No	**95.0**	88.2	86.7	90.0	90.0	–	–	–
Evo-World (Ours)	0.87B	No	93.2	89.1	**88.3**	94.0	**91.1**	**67.1**	16.5	**41.8**

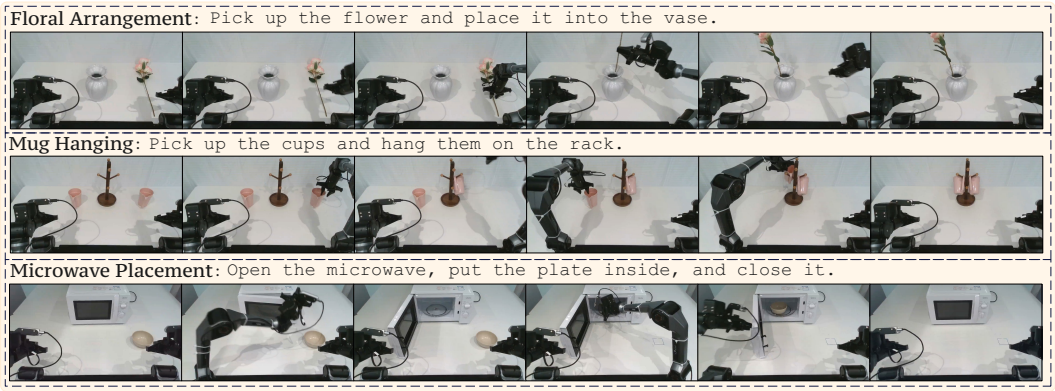

Figure 4: **Successful real-world executions of Evo-World on an AgileX robot.** The rows show floral arrangement (top), a single-arm task, and mug hanging (middle) and microwave placement (bottom), two bimanual collaboration tasks.

frequency under the tested deployment settings. These measurements support the practical efficiency of the compact policy alongside its higher task success rates.

5 Ablations and Representation Analysis

Table 3 organizes our Meta-World study along four axes: prediction target (A), FiLM conditioning content (B), fusion mechanism (C), and optimization components (D). Representation contraction and linear decodability of future motion are analyzed in Appendix C and Appendix D, respectively. All ablation variants are trained and evaluated under the same training budget, using checkpoints after 100,000 second-stage training steps.

Table 2: **Real-world performance and deployment efficiency on an AgileX dual-arm robot.** Each model is evaluated for 20 trials per task. All models use the same NVIDIA H20 GPU and BF16 precision. Task success rates and Avg SR are percentages; Avg SR weights the three tasks equally. Lower parameter count and GPU memory usage, and higher inference frequency and success rates, are better. The best value in each row is bold and the second-best is underlined.

Metric / Task	OpenVLA-OFT	π_0	**Evo-World (Ours)**
Params (B) ↓	7.73	<u>3.50</u>	**0.87**
GPU Mem. (GB) ↓	15.56	<u>6.69</u>	**2.54**
Infer. Freq. (Hz) ↑	2.41	<u>5.42</u>	**6.16**
Floral arrangement	25.0	<u>40.0</u>	**55.0**
Mug hanging	25.0	<u>55.0</u>	**70.0**
Microwave placement	20.0	<u>60.0</u>	**65.0**
Avg SR	23.3	<u>51.7</u>	**63.3**

Table 3: **Ablations on Meta-World MT50 (SR, %).** Tier Avg. weights the four difficulty buckets equally. Each entry gives absolute SR and, in parentheses, its difference from Evo-World (Ours) in percentage points. Positive, negative, and zero differences are green, red, and gray, respectively. The best and second-best scores in each column are bold and underlined; ties share the same formatting. The shared full model is bold, and differences are computed from the displayed SRs.

ID	Configuration	Easy	Medium	Hard	Very Hard	Tier Avg.
A. Prediction target						
A1	None	89.2 (−4.0)	76.8 (−12.3)	77.2 (−11.1)	79.2 (−14.8)	80.6 (−10.5)
A2	Future RGB	<u>90.7</u> (−2.5)	<u>80.0</u> (−9.1)	<u>78.3</u> (−10.0)	<u>88.0</u> (−6.0)	<u>84.3</u> (−6.8)
A3	**Optical flow (Ours)**	**93.2** (—)	**89.1** (—)	**88.3** (—)	**94.0** (—)	**91.1** (—)
B. FiLM conditioning content						
B1	z_t	89.2 (−4.0)	<u>80.9</u> (−8.2)	76.7 (−11.6)	<u>88.0</u> (−6.0)	83.7 (−7.4)
B2	z_{pred}	<u>90.4</u> (−2.8)	79.0 (−10.1)	<u>86.7</u> (−1.6)	86.0 (−8.0)	<u>85.5</u> (−5.6)
B3	Noise	86.5 (−6.7)	77.3 (−11.8)	76.0 (−12.3)	84.0 (−10.0)	81.0 (−10.1)
B4	$\boldsymbol{z_t + z_{\text{pred}}}$ **(Ours)**	**93.2** (—)	**89.1** (—)	**88.3** (—)	**94.0** (—)	**91.1** (—)
C. Fusion mechanism						
C1	Concatenation	81.1 (−12.1)	<u>37.3</u> (−51.8)	48.3 (−40.0)	46.0 (−48.0)	53.2 (−37.9)
C2	Cross-attention	<u>92.1</u> (−1.1)	**89.1** (0.0)	<u>78.3</u> (−10.0)	<u>86.0</u> (−8.0)	<u>86.4</u> (−4.7)
C3	**FiLM (Ours)**	**93.2** (—)	**89.1** (—)	**88.3** (—)	**94.0** (—)	**91.1** (—)
D. Component ablation						
D1	w/o SIGReg	91.0 (−2.2)	87.3 (−1.8)	86.7 (−1.6)	78.0 (−16.0)	85.8 (−5.3)
D2	w/o Grad. Detach	92.1 (−1.1)	84.6 (−4.5)	68.3 (−20.0)	84.0 (−10.0)	82.2 (−8.9)
D3	**Ours**	**93.2** (—)	**89.1** (—)	**88.3** (—)	**94.0** (—)	**91.1** (—)

Shared model: A3, B4, C3, and D3 repeat the same Evo-World (Ours) result, using optical-flow supervision, current and predicted representations, and FiLM; — marks a self-comparison.

5.1 Comparing prediction targets

Group A compares None prediction baseline, the future-frame-supervised variant, and the optical-flow-supervised model. The future-frame variant predicts an encoding of a future RGB image; the optical-flow variant predicts an encoding of rendered displacement. Neither variant reconstructs image pixels.

The future-frame variant exceeds the baseline result by 3.7 percentage points, and Evo-World further improves upon the future-frame variant by 6.8 points. This ordering supports optical-flow supervision among the evaluated configurations. Since all variants use the same training and evaluation protocol, these differences isolate the effect of the prediction target.

5.2 FiLM CONDITIONING CONTENT UNDER OPTICAL-FLOW SUPERVISION

Group B evaluates which representations should condition FiLM under optical-flow supervision. We compare the current representation z_t alone, the predicted representation z_{pred} alone, a noise condition, and the joint condition $z_t + z_{\text{pred}}$. Here, the joint condition denotes concatenation, and B4 repeats the full-model result.

Joint conditioning achieves 91.1% tier-averaged success, exceeding the current-only (83.7%), prediction-only (85.5%), and noise (81.0%) conditions by 7.4, 5.6, and 10.1 percentage points, respectively. It also achieves the highest success rate in all four difficulty buckets. Prediction-only conditioning exceeds current-only conditioning by 1.8 points on average, with higher scores on Easy and Hard; current-only conditioning performs better on Medium and Very Hard. The joint condition's consistent gains over both individual representations support combining current scene context with anticipated motion to condition action generation. The corresponding future-frame-supervised comparisons are reported separately in Appendix B.

5.3 COMPARING FUSION MECHANISMS

Group C compares concatenation, cross-attention, and FiLM for fusing motion representations with the policy's vision-language features. The FiLM row reproduces the highest recorded full-model result reported in Table 1.

FiLM achieves 91.1% tier-averaged success, exceeding concatenation and cross-attention by 37.9 and 4.7 percentage points, respectively. It obtains the highest reported score in the Easy, Hard, and Very Hard buckets and ties cross-attention in Medium. These results support the choice of FiLM among the reported fusion configurations. Earlier fusion comparisons under future-frame supervision are retained in Appendix B.

5.4 EFFECTS OF REGULARIZATION AND GRADIENT DETACHMENT

Group D compares the full model with variants that remove SIGReg or gradient detachment. The component study uses the same two-stage training schedule, optical-flow cache, and evaluation settings. Each variant changes the corresponding component throughout training. Absolute SRs are obtained by adding the reported component-ablation differences to the corresponding full-model scores.

Removing SIGReg reduces the overall success rate by 5.3 percentage points, with performance declining across all difficulty tiers. Thus, regularization improves aggregate performance consistently across all four difficulty tiers. Further analysis of the effect of the SIGReg loss on the learned representations is provided in the Appendix C.

Removing gradient detachment reduces success by 8.9 points, including a 20.0-point decline in the Hard tier. This result supports retaining separate gradient paths for motion representation learning and action optimization in the tested configuration.

6 CONCLUSION

We presented **Evo-World**, a compact VLA policy that learns instruction-conditioned predictions of future motion through optical-flow supervision. By directly predicting learned embeddings of flow renderings and using them to condition action generation, the model introduces motion anticipation without reconstructing future images or optical flow.

Experiments on Meta-World MT50, RoboTwin, and an AgileX robot support the effectiveness of this formulation. With **0.87B** parameters, it outperforms the evaluated real-world baselines across single-arm and bimanual tasks while using less GPU memory and achieving a higher measured inference frequency. Ablations further support joint conditioning on current and predicted representations, distribution regularization, and gradient separation. Because the current motion targets describe 2D image-plane displacement without explicit 3D geometry, future work will explore geometry-aware supervision while preserving efficient deployment.

AI USE STATEMENT

In this work, we used generative AI tools to assist with literature works. Additionally, we used these tools to improve manuscript organization, clarity, and language. We reviewed the AI-assisted work, checked references against original publications, and cross-checked quantitative statements against the reported experimental results and figures. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Section 3 describes the model architecture and training objectives. Appendix A documents the training hyperparameters, optical-flow preprocessing, and SIGReg implementation. Section 4 presents the experimental settings and evaluation metrics, while Section 5 and Appendix B report the ablation configurations and results. Appendices C and D describe the representation analyses. Appendix F provides the real-world hardware setup, demonstration collection procedure, and supervised fine-tuning setup.

REFERENCES

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.

Randall Balestriero and Yann LeCun. Lejepa: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025.

Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2act: Predicting point tracks from internet videos enables generalizable robot manipulation. In *European Conference on Computer Vision*, pp. 306–324. Springer, 2024.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael 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, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control, 2026. URL https://arxiv.org/abs/2410.24164.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Jisong Cai, Long Ling, Shiwei Chu, Zhongshan Liu, Jiayue Kang, Zhixuan Liang, Wenjie Xu, Yinan Mao, Weinan Zhang, Xiaokang Yang, et al. Aha-wam: Asynchronous horizon-adaptive world-action modeling with observation-guided context routing. *arXiv preprint arXiv:2606.09811*, 2026a.

Rui Cai, Jun Guo, Xinze He, Piaopiao Jin, Jie Li, Bingxuan Lin, Futeng Liu, Wei Liu, Fei Ma, Kun Ma, et al. Xiaomi-robotics-0: An open-sourced vision-language-action model with real-time execution. *arXiv preprint arXiv:2602.12684*, 2026b.

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

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025a.

Yi Chen, Yuying Ge, Weiliang Tang, Yizhuo Li, Yixiao Ge, Mingyu Ding, Ying Shan, and Xihui Liu. Moto: Latent motion token as the bridging language for learning robot manipulation from videos. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19752–19763. IEEE, 2025b.

Yixiang Chen, Peiyan Li, Yuan Xu, Qisen Ma, Jiabing Yang, Kai Wang, Jianhua Yang, Dong An, He Guan, Gaoteng Liu, Jianlou Si, Jun Huang, Jing Liu, Nianfeng Liu, Yan Huang, and Liang Wang. Flowwam: Optical flow as a unified action representation for world action models, 2026. URL https://arxiv.org/abs/2607.13017.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

StarVLA Community. Starvla: A lego-like codebase for vision-language-action model developing, 2026. URL https://arxiv.org/abs/2604.05014.

Yu Fang, Kanchana Ranasinghe, Le Xue, Honglu Zhou, Juntao Tan, Ran Xu, Shelby Heinecke, Caiming Xiong, Silvio Savarese, Daniel Szafir, Mingyu Ding, Michael S. Ryoo, and Juan Carlos Niebles. Robotic vla benefits from joint learning with motion image diffusion, 2025. URL https://arxiv.org/abs/2512.18007.

Chongkai Gao, Haozhuo Zhang, Zhixuan Xu, Zhehao Cai, and Lin Shao. Flip: Flow-centric generative planning as general-purpose manipulation world model, 2025. URL https://arxiv.org/abs/2412.08261.

Jun Guo, Xiaojian Ma, Yikai Wang, Min Yang, Huaping Liu, and Qing Li. Flowdreamer: A rgb-d world model with flow-based motion representations for robot manipulation. *IEEE Robotics and Automation Letters*, 11(3):2466–2473, 2026.

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.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success, 2025. URL https://arxiv.org/abs/2502.19645.

Sixu Lin, Junliang Chen, Huaiyuan Xu, Zhuohao Li, Guangming Wang, Yixiong Jing, Sheng Xu, Runyi Zhao, Brian Sheil, Lap-Pui Chau, et al. Roboflow4d: A lightweight flow world model toward real-time flow-guided robotic manipulation. *arXiv preprint arXiv:2605.17522*, 2026a.

Tao Lin, Yuxin Du, Jiting Liu, Nuobei Zhu, Yunhe Li, Yuqian Fu, Yinxinyu Chen, Hongyi Cai, Zewei Ye, Bing Cheng, et al. Evo-depth: A lightweight depth-enhanced vision-language-action model. *arXiv preprint arXiv:2605.14950*, 2026b.

Tao Lin, Yuxin Du, Yiran Mao, Zewei Ye, Yilei Zhong, Bing Cheng, Yiming Wang, Jiting Liu, Yang Tian, Junchi Yan, et al. La4vla: Learning to act without seeing via language-action pretraining. *arXiv preprint arXiv:2606.27295*, 2026c.

Tao Lin, Yilei Zhong, Yuxin Du, Jingjing Zhang, Jiting Liu, Yinxinyu Chen, Encheng Gu, Ziyan Liu, Hongyi Cai, Yanwen Zou, et al. Evo-1: Lightweight vision-language-action model with preserved semantic alignment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13397–13406, 2026d.

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

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. In *International Conference on Learning Representations*, volume 2025, pp. 29982–30009, 2025.

Yicheng Liu, Zibin Dong, Baijun Ye, Tianyuan Yuan, Tao Jiang, Anqi Yang, Shicheng Cao, Haonan Liu, Yue Sun, Zihan Guo, et al. G0. 5: One autoregressive stream for robot reasoning and action. *arXiv preprint arXiv:2608.11739*, 2026.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

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

Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pp. 402–419. Springer, 2020.

Xinkai Wang, Chenyi Wang, Yifu Xu, Mingzhe Ye, Fucheng Zhang, Jialin Tian, Xinyu Zhan, Lifeng Zhu, Cewu Lu, and Lixin Yang. Lamp: Learning vision-language-action policy with 3d scene flow as latent motion prior, 2026. URL https://arxiv.org/abs/2603.25399.

Haoyu Wei, Xiuwei Xu, Ziyang Cheng, Hang Yin, Angyuan Ma, Bingyao Yu, Jie Zhou, and Jiwen Lu. F2f-ap: Flow-to-future asynchronous policy for real-time dynamic manipulation. *arXiv preprint arXiv:2604.02408*, 2026.

Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point Trajectory Modeling for Policy Learning. *arXiv preprint arXiv:2401.00025*, 2023. URL https://arxiv.org/abs/2401.00025v3.

Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. In *International Conference on Learning Representations*, volume 2024, pp. 10641–10662, 2024.

Mengda Xu, Zhenjia Xu, Yinghao Xu, Cheng Chi, Gordon Wetzstein, Manuela Veloso, and Shuran Song. Flow as the cross-domain manipulation interface. *arXiv preprint arXiv:2407.15208*, 2024.

Feng Yan, Fanfan Liu, Liming Zheng, Yufeng Zhong, Yiyang Huang, Zechao Guan, Chengjian Feng, and Lin Ma. Robotron-mani: All-in-one multimodal large model for robotic manipulation. *arXiv preprint arXiv:2412.07215*, 2024.

Jiarui Yang, Yehao Lu, Yuning Su, Yu Zhong, Yufeng Xie, Yazhou Zhang, Haiyu Lan, Kaixiang Lu, Peiwen Lin, Chuang Wang, Junwei Liang, and Enyu Li. Temporalflow-vla: Learning physically grounded execution history for long-horizon robot manipulation, 2026a. URL https://arxiv.org/abs/2608.26821.

Shiyuan Yang, Borong Zhang, Jizheng Zhang, Zhijia Tao, Junfei Guo, Donglai Ran, Xu Bian, and Qingbiao Li. Fabrivla: A lightweight vision-language-action model for precise multi-task manipulation. *arXiv preprint arXiv:2607.08575*, 2026b.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pp. 1094–1100. PMLR, 2020.

Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. *arXiv preprint arXiv:2403.03954*, 2024.

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.

Zhide Zhong, Haodong Yan, Junfeng Li, Xiangchen Liu, Xin Gong, Tianran Zhang, Wenxuan Song, Jiayi Chen, Xinhu Zheng, Hesheng Wang, et al. Flowvla: Visual chain of thought-based motion reasoning for vision-language-action models. *IEEE Transactions on Robotics*, 2026.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

A IMPLEMENTATION DETAILS

A.1 REPRESENTATIVE TRAINING HYPERPARAMETERS ON META-WORLD

Table A1 summarizes the two-stage Meta-World training configuration. Stage 1 trains the motion representation module, FiLM, and action expert with the vision-language backbone frozen. Stage 2 initializes from the Stage 1 model, unfreezes the backbone, and starts a new optimizer and learning-rate schedule for a further 100,000 steps. Both motion encoders, their projectors, and the motion predictor remain trainable in both stages. The global batch contains 64 samples: 16 per GPU on four NVIDIA RTX 4090 GPUs, with one optimizer update per batch.

Table A1: **Representative Training Hyperparameters on Meta-World.** Evo-World (Ours) uses two training stages. Entries spanning both stage columns apply to both stages; the loss coefficients follow Eq. (5).

Hyperparameter	**Stage 1**	**Stage 2**
Training stages		
Training steps	10,000	100,000
Vision-language backbone	Frozen	Trainable
Motion representation module	Trainable	Trainable
FiLM and action expert	Trainable	Trainable
Optimization		
Optimizer	AdamW	
Peak learning rate	10^{-5}	
Learning-rate schedule	Linear warmup + cosine decay	
Warmup steps per stage	1,000	
Weight decay	10^{-3}	
Adam coefficients (β_1, β_2)	$(0.9, 0.999)$	
Adam ϵ	10^{-8}	
Gradient clipping (maximum norm)	1.0	
Batch size per GPU	16	
Effective global batch size	64	
Gradient accumulation steps	1	
Motion conditioning and regularization		
World-loss weight λ_{world}	0.1	
SIGReg coefficient λ_{sig}	0.09	
FiLM scale α	0.1	
Detachment at the policy interface	Enabled	
Action expert dropout	0.2	
Runtime and input settings		
Training GPUs	$4\times$ NVIDIA RTX 4090	
Mixed precision	BF16	
Distributed training	DeepSpeed ZeRO stage 2	
Activation checkpointing	Enabled	
Policy image resolution	448×448	
Motion-branch image resolution	224×224	
Action chunk length	50	
Policy augmentation probability	0.5 per view	

Each stage uses 1,000 steps of linear learning-rate warmup followed by cosine decay. Biases, normalization parameters, and other one-dimensional parameters are excluded from weight decay. Policy-image augmentation is applied independently with probability 0.5 per view and combines a random resized crop with area scale $[0.95, 1.0]$, rotation within $[-5^\circ, 5^\circ]$, and color jitter with brightness, contrast, saturation, and hue strengths of 0.3, 0.4, 0.5, and 0.08, respectively. Current

images and optical-flow targets in the motion branch use the deterministic preprocessing described below.

A.2 Exact SIGReg statistic

We instantiate the characteristic-function form of SIGReg (Balestriero & LeCun, 2025) as follows. For each local training batch, sample $P = 1024$ independent unit directions $u_p \in \mathbb{R}^D$ by normalizing Gaussian vectors. Use $K = 17$ knots $\xi_k = 3(k-1)/(K-1)$ and $\delta = 3/(K-1)$. Define $q_1 = q_K = \delta$ and $q_k = 2\delta$ for interior knots. The implementation computes

$$\mathcal{S}(Z) = \frac{B}{P} \sum_{p=1}^{P} \sum_{k=1}^{K} q_k e^{-\xi_k^2/2} \times \left[\left(\frac{1}{B} \sum_{b=1}^{B} \cos(\xi_k u_p^\top z_b) - e^{-\xi_k^2/2} \right)^2 + \left(\frac{1}{B} \sum_{b=1}^{B} \sin(\xi_k u_p^\top z_b) \right)^2 \right].$$

These q_k are twice the conventional trapezoidal weights on $[0, 3]$, equivalently accounting for the symmetric integral on $[-3, 3]$. The code averages $\mathcal{S}(z_t)$ and $\mathcal{S}(z_h)$ and uses the same sampled directions for the two representations. Characteristic-function averages are computed within the local batch; the world-loss implementation does not gather embeddings across workers.

A.3 Optical-flow targets

The Meta-World preprocessing script computes RAFT flow from each source frame to the frame 15 steps ahead. Near an episode boundary, the destination is clipped to the final frame. The default renderer uses a magnitude scale fixed within each dataset, clips larger magnitudes while retaining direction, and saves the renderings in a video with the original frame indexing. Scale calibration, when not supplied explicitly, uses the 99th percentile of sampled magnitudes from the first five episodes. The exact cached scale and RAFT checkpoint should be obtained from the preprocessing manifest.

At training time, the dataset loads the cached flow rendering at the *current* frame timestamp. A new prediction interval requires a corresponding cache.

A.4 RoboTwin Evaluation Protocol

The combined comparison in Table 1 includes Evo-1 (Lin et al., 2026d), starVLA (Community, 2026), π_0 (Black et al., 2026), ACT (Zhao et al., 2023), and Xiaomi Robotics-0 (Cai et al., 2026b), among other baselines. We report results under the aligned RoboTwin Clean2Clean and Clean2Random settings, following the RoboTwin 2.0 evaluation protocol .

B Earlier future-frame fusion experiments

The earlier future-frame-supervised study reports the results in Table A2. These use appearance-target representations; Table 3, group C, reports the separate fusion comparison under optical-flow supervision. Within this earlier study, FiLM and the combined current/predicted condition obtain the highest recorded values.

C Representation contraction without SIGReg

To examine how SIGReg affects the learned motion representations, Figure A1 compares PCA projections of the current representation z_t and predicted representation z_{pred} with and without regularization. With SIGReg, both representations exhibit variation across observations. Without SIGReg, they concentrate near their respective means, indicating collapse toward nearly constant representations.

Table A2: **Fusion and conditioning comparisons for future-frame supervision (SR, %).** Groups A and B change different factors. Each entry gives absolute SR and its difference from the corresponding reference score in percentage points. Positive, negative, and zero differences are dark green, dark red, and gray, respectively. Within each group and column, the best score is bold and the second-best is underlined; ties share the same formatting.

ID	Configuration	Easy	Medium	Hard	Very Hard	Tier Avg.
A. Fusion mechanism						
A1	Concatenation	88.6 (−2.1)	<u>79.1</u> (−0.9)	71.7 (−6.6)	<u>86.0</u> (−2.0)	81.4 (−2.9)
A2	Cross-attention	<u>89.6</u> (−1.1)	74.5 (−5.5)	**80.0** (+1.7)	<u>86.0</u> (−2.0)	<u>82.5</u> (−1.7)
A3	**FiLM**	**90.7** (—)	**80.0** (—)	<u>78.3</u> (—)	**88.0** (—)	**84.3** (—)
B. FiLM conditioning content						
B1	z_t only	<u>89.3</u> (−1.4)	<u>78.2</u> (−1.8)	<u>73.3</u> (−5.0)	<u>86.0</u> (−2.0)	<u>81.7</u> (−2.6)
B2	Noise	<u>89.3</u> (−1.4)	76.4 (−3.6)	71.7 (−6.6)	78.0 (−10.0)	78.9 (−5.4)
B3	$\boldsymbol{z_t + z_{\text{pred}}}$	**90.7** (—)	**80.0** (—)	**78.3** (—)	**88.0** (—)	**84.3** (—)

Reference: A3 and B3 repeat the same future-frame baseline, using FiLM with current and predicted representations; — denotes a self-comparison. The joint condition $z_t + z_{\text{pred}}$ denotes concatenation.

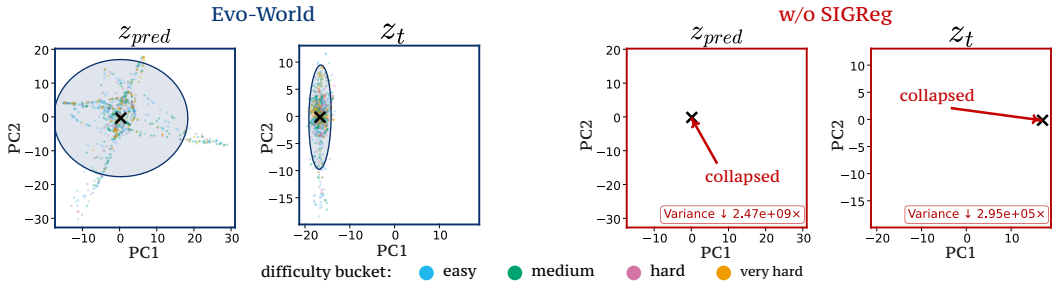

Figure A1: **Effect of SIGReg on latent representations.** PCA projections for Evo-World (left pair) and the variant without SIGReg (right pair). Each pair shows z_{pred} followed by z_t. Colors indicate task difficulty, and black crosses mark the plotted means. For each representation, the corresponding panels use the same displayed axis scales. The annotations report the variance reduction factors when SIGReg is removed.

PC1 and PC2 denote coordinates along the first and second principal components, respectively; their values are latent-space coordinates, not physical distances. For each representation, the panels with and without SIGReg have the same displayed axis ranges and tick spacing. The z_{pred} panels use 10-unit tick intervals on both axes, whereas the z_t panels use 10-unit intervals on PC1 and 5-unit intervals on PC2. Representation spread should therefore be compared between regularization conditions within the corresponding pair of panels.

As annotated in Figure A1, removing SIGReg reduces the variance of z_t by a factor of 2.95×10^5 and that of z_{pred} by a factor of 2.47×10^9. The contraction affects both representations and is substantially stronger in the predicted representation.

These observations complement the component ablation in Table 3, group D, where removing SIGReg lowers tier-averaged success from 91.1% to 85.8%, a decrease of 5.3 percentage points. Together, the representation analysis and control results support the role of SIGReg in maintaining input-dependent motion representations for action generation.

D Linear Decodability of Future Motion

Table A3 compares how well current and predicted representations encode future end-effector displacement. The reported evaluation keeps the representation models frozen and fits a linear ridge-regression decoder to displacement over 15 frames, using common samples and trajectory-level splits.

Displacement labels are derived from simulator state, independently of the optical-flow estimator used for training. Performance is measured by test R^2 and median angular error.

Table A3: **Linear decoding of future end-effector displacement.** Results compare current and predicted representations under future-frame and optical-flow supervision. Higher test R^2 and lower median angular error indicate better decoding. The best score in each metric column is bold and the second-best is underlined.

Supervision	Representation	Test R^2	Median error (deg.)
Future frame	z_t	0.497	29.6
Future frame	z_{pred}	0.421	35.2
Optical flow	z_t	**0.661**	**20.9**
Optical flow	z_{pred}	<u>0.508</u>	<u>29.4</u>

Optical-flow supervision yields higher test R^2 and lower median angular error for both current and predicted representations than future-frame supervision. These results indicate that future end-effector displacement is more accessible to linear decoding in the optical-flow-supervised representations. Under both supervision targets, z_t yields better decoding results than z_{pred}. This comparison assesses end-effector displacement and does not establish that the predictor adds linearly decodable motion information beyond the current representation.

E SIMULATION BENCHMARK EXAMPLES

Figures A2 and A3 provide example task sequences from Meta-World and RoboTwin, complementing the quantitative comparison in Table 1. The Meta-World examples illustrate object transport and interactions with articulated objects, including doors, drawers, and windows. The RoboTwin examples cover container manipulation, object handover between arms, coordinated placement, and stacking. Together, they illustrate the range of object motions and arm coordination involved in the evaluated benchmarks.

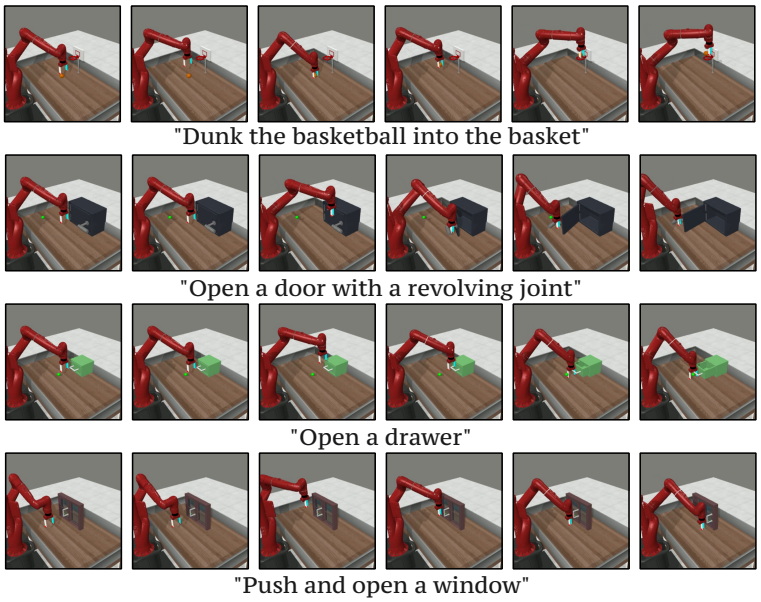

Figure A2: **Example task sequences from Meta-World.** From top to bottom: placing a basketball into a basket, opening a hinged door, opening a drawer, and opening a window. Each row shows six successive observations from left to right, with the corresponding language instruction below.

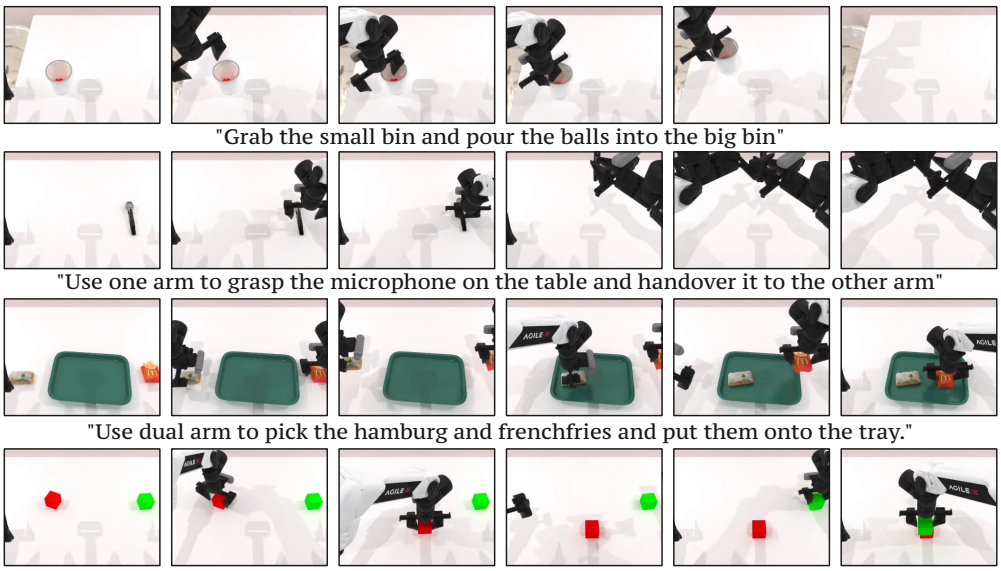

Figure A3: **Example task sequences from RoboTwin.** From top to bottom: pouring balls from a small container into a larger one, handing a microphone from one arm to the other, placing food items on a tray with both arms, and stacking a green block on a red block. Each row shows six successive observations from left to right, with the corresponding language instruction below.

F REAL-WORLD EXPERIMENTAL SETUP

Figure A4 shows the hardware used for real-world data collection and policy evaluation. The setup comprises two AgileX PiPER follower arms, a pair of leader arms for teleoperation, and Intel RealSense D435 cameras providing one elevated view and two wrist-mounted views of the workspace.

For each of the three tasks—floral arrangement, mug hanging, and microwave placement—we collect 100 teleoperated demonstration trajectories, yielding 300 demonstrations in total. Task-relevant object positions are varied across demonstrations to support generalization across spatial configurations. Floral arrangement is performed with a single arm, while mug hanging and microwave placement require bimanual collaboration. Using the same collected demonstrations, we separately perform supervised fine-tuning (SFT) of OpenVLA-OFT, π_0, and Evo-World before evaluating their task performance.

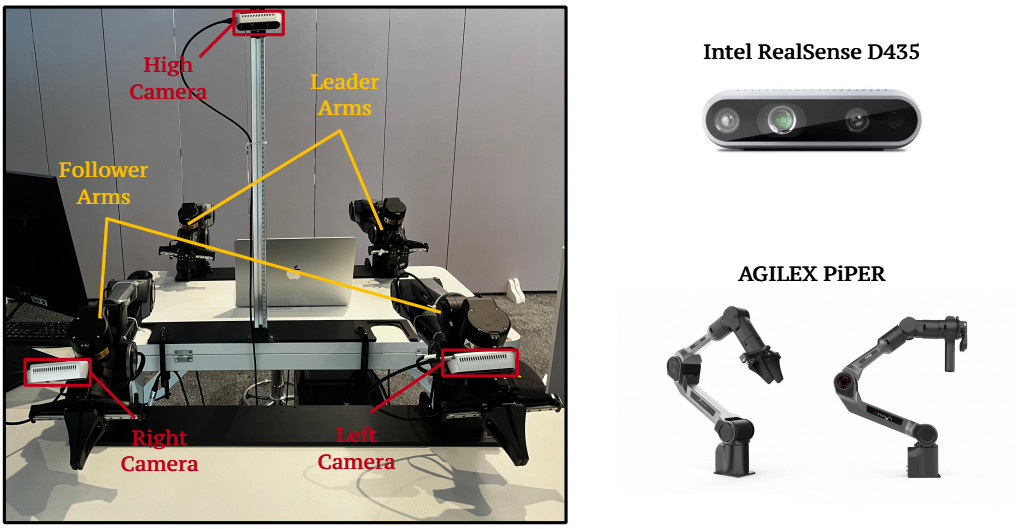

Figure A4: **Real-world experimental setup.** Left: the teleoperation apparatus, with leader arms controlling the two follower arms and cameras providing an elevated view and left and right wrist views. Right: the Intel RealSense D435 camera and AgileX PiPER robot used in the setup. We collect 100 teleoperated demonstrations per task with varied object positions and use the collected data to separately fine-tune OpenVLA-OFT, π_0, and Evo-World.