UNILAP: A UNIFIED LATENT ACTION POLICY FOR CROSS-EMBODIMENT MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Latent Action Models (LAMs) can capture manipulation priors from action-free videos, but discrete latents usually impose a granularity bottleneck, limiting their ability to represent fine-grained action. Recent continuous alternatives mitigate this, but they are typically used as proxies for world modeling and require a further planning step to produce executable actions. We thus introduce UniLAP, a unified continuous latent-action policy that models the latent space via flow matching and bypasses reliance on future frames during inference. By distilling a LAM teacher with a flow-adaptive objective, UniLAP inherits the teacher's action representation while retaining the multimodal distribution of feasible behaviors. To decouple latent representations from task-irrelevant appearance, we extract a label-free motion target to isolate relevant kinematic cues from spurious visual dynamics. With the established embodiment-agnostic latent manifold, UniLAP can be easily adapted to novel platforms. Pretrained on multi-source videos at a compact 1.1B parameter scale, UniLAP outperforms state-of-the-art latent-action and Vision-Language-Action (VLA) baselines across hand pose forecasting, single-arm and humanoid manipulation, and real-world deployment.

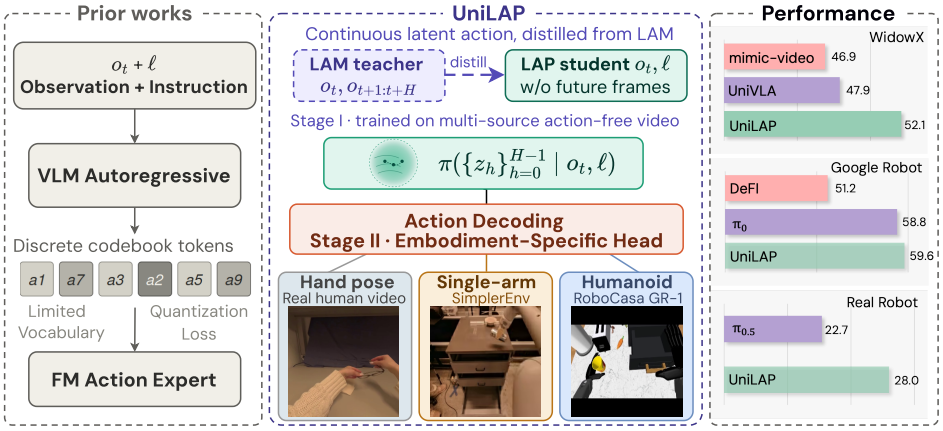

Figure 1: **UniLAP learns a continuous unified latent action policy (LAP) from action-free video.** *Left*, prior pipelines quantize the latent into discrete codebooks. *Middle*, UniLAP models it as a continuous flow distilled from a frozen LAM teacher. *Right*, performance comparison.

1 INTRODUCTION

Action-labeled data is expensive to collect at scale, and this scarcity limits the development of generalist manipulation policies (Zheng et al., 2025b). Action-free human and robot videos, by contrast, are abundant and dense with motion and dynamic cues. Latent Action Models (LAMs) (Schmidt & Jiang, 2023; Bruce et al., 2024; Ye et al., 2024) exploit such videos via extracting latent actions from consecutive frames by next-frame prediction without requiring explicit action annotations. Within Vision-Language-Action (VLA) systems, an action expert is then learned on limited labeled data to

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.

decode these latents into specific robot commands (Bu et al., 2025a; Chen et al., 2025c), separating the data-hungry acquisition of action priors from their data-scarce grounding on a target robot.

Despite their potential, existing LAM pipelines (Ye et al., 2024; Bu et al., 2025a; Chen et al., 2025c; Yang et al., 2026) suffer from two key limitations. First, most methods quantize them into discrete tokens and predict them via large VLMs autoregressively (Fig. 1 *left*). This discretization coarsens the latent onto the nearest code, hindering its ability to express the continuous, multimodal distribution that manipulation requires. Recent works (Gao et al., 2025; 2026) alternatively loosen such representational bottleneck and illustrate the effectiveness of continuous latent actions. However, these latents are usually inferred with access to future frames and are deployed by world-model prediction followed by planning (Goswami et al., 2025; Assran et al., 2025), leaving open how to turn continuous latent actions into a causal policy for robot control. Second, the standard pixel reconstruction latent learning entangles it with task-irrelevant appearance, a problem amplified by frequent camera motion. Prior work attempts to decouple this by factorizing the latent into separate tokens (Bu et al., 2025a; Li et al., 2025) or introducing optical flow (Bu et al., 2025b; Bi et al., 2025), but the former requires a dedicated design to enforce the separation, while the latter relies on an external optical flow estimator to generate dense motion targets for latent action learning.

We present UniLAP, a unified latent action policy that addresses these limitations. **(1) Continuous latent action policy.** Such a representation preserves the multimodal distribution that a discrete codebook cannot describe. We model the latent action as a continuous flow (Lipman et al., 2022) conditioned on current observation, and distill the LAM teacher's posterior via a *flow-adaptive objective*. This mechanism tightens the latent toward the teacher as the noise decreases. Acting without future frames, our method shortens the route from action-free pretraining to robot control. **(2) Label-free motion supervision.** Pixel reconstruction supervision binds the latent to appearance, making camera movement contaminate it. UniLAP instead derives a label-free motion target from cross-frame correspondences in a frozen ViT, isolating task-relevant information from unrelated visual changes. This supervision needs neither explicit factorization nor an external flow estimator. **(3) Cross-embodiment adaptation.** The continuous latent forms an embodiment-agnostic action space shared across human and multi-robot videos. A new robot reuses this pretrained latent and adapts with only a fresh action decoder and light policy finetuning, rather than per-robot retraining.

We evaluate UniLAP on hand pose forecasting (Hoque et al., 2025), single-arm manipulation (Li et al., 2024), humanoid manipulation (Bjorck et al., 2025), and tabletop manipulation on a real robot platform. At a smaller 1.1B scale, UniLAP compares favorably with latent-action and VLA baselines, surpassing recent action-free pretraining methods. Analyses and ablations further support the importance of continuous latent-action modeling, flow-adaptive posterior distillation, and correspondence-based motion supervision for learning transferable action representations.

2 Related Work

Policy Learning from Videos. Videos inherently capture rich dynamic information about the world transitions, making them valuable for robot learning. Early contributions have provided effective visual representations from videos via self-supervision, which can be transferred to policy learning. Another line of work incorporates world models (Ha & Schmidhuber, 2018), aiming to estimate future states and derive action via motion planning (Goswami et al., 2025; Assran et al., 2025) or inverse dynamic models (Kim et al., 2025; Chen et al., 2025a). Instead, recent world action models (Liang et al., 2025b; Hu et al., 2024; Pai et al., 2025; Ye et al., 2026) are proposed to co-train video generation with action prediction, treating video latents as structured variables for underlying dynamics. Meanwhile, other works like EgoMimic (Kareer et al., 2025) and EgoScale (Zheng et al., 2026) explore bridging the domain gap between egocentric videos and robot execution, treating humans as a special embodiment and hand motion as an action proxy. Our work differs from these in that we model a unified latent policy aligning various video sources without requiring explicit action supervision, and then transfer the latent action across embodiments.

Latent Action Learning. Latent Action Models (LAM) are normally learned from consecutive frames without explicit action annotations (Schmidt & Jiang, 2023; Bruce et al., 2024). The dominant paradigm (Ye et al., 2024; Bu et al., 2025a; Chen et al., 2025c; Yang et al., 2026) casts latent actions as *intermediate representation* within vision-language-action (VLA) frameworks, with flow-matching (Lipman et al., 2022; Black et al., 2024) or diffusion (Chi et al., 2025; Yin et al., 2025)

action experts. However, discrete latent actions via VQ-VAE (Van Den Oord et al., 2017) limit action granularity to the codebook size. IGOR (Chen et al., 2024) and CLAM (Liang et al., 2025a) learn continuous latents via variational objectives, while AdaWorld (Gao et al., 2025) adopts a β-VAE bottleneck that DreamDojo (Gao et al., 2026) scales to large-scale data. To address motion-scene entanglement, LatBot (Li et al., 2025) factorizes the latent into separate tokens, while UniVLA (Bu et al., 2025a) learns to filter out irrelevant information. Villa-X (Chen et al., 2025b) couples latent and real action diffusion with two experts, while others (Jiang et al., 2026; Zhang et al., 2025b) directly encode real actions across embodiments. We instead transplant the generative formulation onto *latent actions*, yielding a shared *latent flow-matching policy* across embodiments.

Visual Foresight for Manipulation. Instead of just relying on static image-action pairs, visual foresight has been shown to boost policy learning by implicitly encoding necessary dynamics. FLARE (Zheng et al., 2025a) aligns learnable future tokens with vision-language embeddings of a subsequent observation, while LDA-1B (Lyu et al., 2026) co-denoises action and upcoming visual latents. DreamVLA (Zhang et al., 2025a) broadens the signal across modalities, forecasting depth and semantics simultaneously. LAOF (Bu et al., 2025b) augments pixel reconstruction with optical flow to disentangle physical motion from photometric change, while Motus (Bi et al., 2025) entirely uses optical flow targets. These methods rely on external flow estimators and thus inherit their biases. In contrast, UniLAP derives a dense *correspondence residual* intrinsically from key vectors in a frozen ViT, requiring no auxiliary network to generate targets for latent policy pretraining.

3 METHODOLOGY

In this paper, we present UniLAP, which learns a shared latent policy from action-free videos across embodiments. It comprises three components trained in two stages. In Stage I, the *unified latent action policy* (Sec. 3.1) models latent actions as a continuous flow-matching process distilled from a frozen LAM teacher, while the *correspondence-driven motion supervision* (Sec. 3.2) regularizes the latent space with a label-free target. In Stage II, an *action decoder* (Sec. 3.3) maps the latents into executable commands through an embodiment-specific head. Fig. 2 shows the overall architecture.

3.1 UNIFIED LATENT ACTION POLICY

Prevailing methods (Bu et al., 2025a) predict discrete latent action tokens, fixing granularity at the codebook size, while the continuous latents of recent LAMs (Gao et al., 2026) are more expressive but inferred with access to future frames, hence not directly usable as a causal policy.

At time t, the policy predicts a horizon-H latent action chunk from observation o_t and instruction ℓ, while the LAM teacher reads the frame window $\{o_{t+j}\}_{j=0}^{H}$. Let $\tau \sim \mathcal{U}(0,1)$ be the flow timestep, $h \in \{0,\dots,H-1\}$ the chunk index, and $x_{0,h} \sim \mathcal{N}(0,I)$ the initial noise. The latent action $z_h \in \mathbb{R}^{d_z}$ encodes the transition (o_{t+h}, o_{t+h+1}), whose non-causal Gaussian posterior from LAM is:

$$q(z_h \mid o_{t+h}, o_{t+h+1}) = \mathcal{N}(\mu_h, \Sigma_h), \tag{1}$$

where μ_h and diagonal covariance Σ_h come from the LAM encoder on (o_{t+h}, o_{t+h+1}). The latent policy $\pi(\{z_h\}_{h=0}^{H-1} | o_t, \ell)$ outputs the entire H-step chunk in a single forward pass, though we keep per-step notation for brevity. We parameterize it as a flow-matching network (Lipman et al., 2022), following the paradigm in visuomotor policies (Chi et al., 2025; Black et al., 2024). Two frozen ViTs encode o_t, DINOv2 (Oquab et al., 2023) for fine-grained spatial features and SigLIP2 (Tschannen et al., 2025) for vision-language semantics, and a pre-trained language backbone embeds ℓ. A stack of transformer blocks conditions on the flow timestep τ through Adaptive Layer Normalization (AdaLN) (Perez et al., 2018; Peebles & Xie, 2023) and visual and language tokens via per-block cross-attention, predicting the velocity field v_θ along the linear interpolant $x_{\tau,h} = (1-\tau)x_{0,h} + \tau z_h$. With target $z_h \sim q(\cdot \mid o_{t+h}, o_{t+h+1})$, we optimize the policy under the flow-matching objective:

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{\tau, x_{0,h}, z_h}\Big[\big\| v_\theta(x_{\tau,h}, o_t, \ell) - (z_h - x_{0,h}) \big\|_2^2\Big]. \tag{2}$$

The policy is conditioned only on (o_t, ℓ), whereas the teacher infers latent actions using future frames. $\mathcal{L}_{\text{FM}}$ therefore targets the teacher posterior marginalized over these unobserved frames.

$$p^*\big(\{z_h\}_{h=0}^{H-1} \mid o_t, \ell\big) = \mathbb{E}_{\{o_{t+j}\}_{j=1}^{H} | o_t, \ell}\Big[\prod_h q(z_h \mid o_{t+h}, o_{t+h+1})\Big]. \tag{3}$$

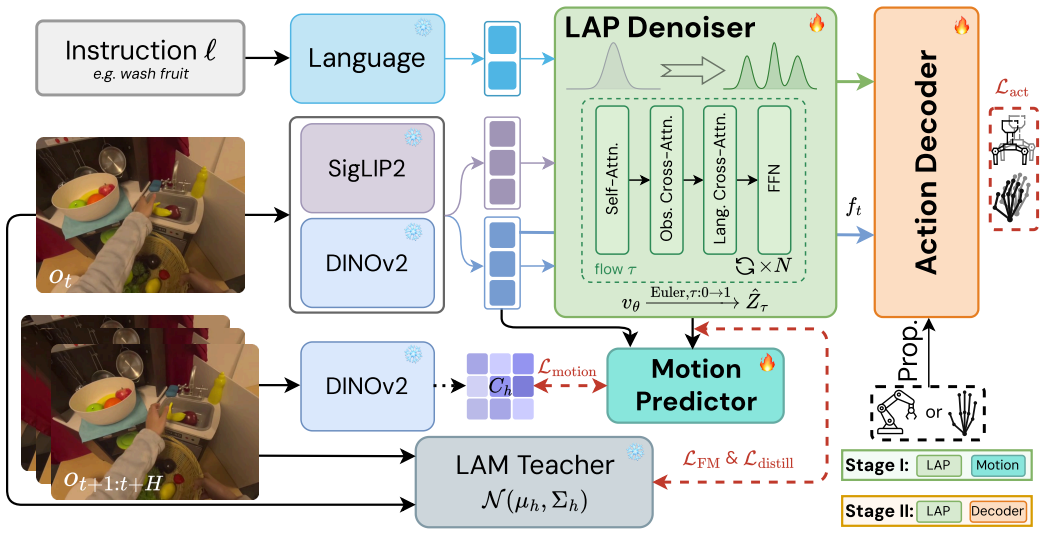

Figure 2: **Overview of UniLAP.** *Stage I* pretrains a continuous flow-matching latent action policy (LAP) on action-free videos under $\mathcal{L}_{\text{FM}}$. A frozen LAM teacher's posterior is distilled via the flow-adaptive $\mathcal{L}_{\text{distill}}$, and a cross-frame correspondence target supervises the motion predictor to regularize the latent. *Stage II* attaches a per-embodiment decoder ϕ_{dec}, trains it from scratch, and finetunes the latent policy on the target embodiment's labeled actions.

The optimal flow under $\mathcal{L}_{\text{FM}}$ transports $\mathcal{N}(0, I)$ to p^* (Lipman et al., 2022), a Gaussian mixture over plausible futures that can capture action multimodality. At any timestep τ, an estimate of the latent action is recovered as $\hat{z}_{\tau,h} = x_{\tau,h} + (1-\tau)\, v_\theta(x_{\tau,h}, o_t, \ell)$.

Flow-Adaptive Posterior Distillation. The FM loss matches samples from the teacher posterior and preserves the stochasticity of possible latent actions. We further introduce a distillation term that provides a covariance-aware anchor to the high-probability region of the teacher posterior by aligning $\hat{z}_{\tau,h}$ with the posterior mean μ_h under the inverse-covariance metric Σ_h^{-1}. At early timesteps ($\tau \to 0$), where noisy velocity dominates $\hat{z}_{\tau,h}$, a constant penalty would bias the velocity field toward the posterior mean and hinder exploration of the transport path. We therefore weight the penalty by τ, yielding the flow-adaptive distillation loss:

$$\mathcal{L}_{\text{distill}} = \mathbb{E}_{\tau,\, x_{0,h},\, z_h}\left[\frac{\tau}{2}\left\|\hat{z}_{\tau,h} - \mu_h\right\|^2_{\Sigma_h^{-1}}\right]. \tag{4}$$

3.2 Correspondence as Motion Supervision

Latent action models typically rely on pixel reconstruction, which entangles the latent space with task-irrelevant content. Prior work factorizes the latent (Bu et al., 2025a; Li et al., 2025) or introduces optical flow (Bu et al., 2025b; Bi et al., 2025). Factorization requires a dedicated design to enforce the decomposition, while optical flow relies on an external estimator to generate motion targets. We instead leverage the cross-patch correspondence of the frozen DINOv2 encoder $\mathcal{E}$, whose stable cross-frame features expose motion as a feature-space residual.

Correspondence-Derived Motion Field. For each chunk step h, we extract the patch tokens $f_{t+h}, f_{t+h+1} \in \mathbb{R}^{N \times d_{\text{obs}}}$ and last-block key vectors $k_{t+h}, k_{t+h+1} \in \mathbb{R}^{N \times d_k}$ from the frozen encoder $\mathcal{E}$. These keys act as cross-frame descriptors (Oquab et al., 2023), from which we form a softmax correspondence matrix $C_h \in \mathbb{R}^{N \times N}$ whose rows softly match current frame to the next:

$$C_h = \text{Softmax}\left(\frac{k_{t+h}\, k_{t+h+1}^\top}{\sqrt{d_k}}\right). \tag{5}$$

Warping f_{t+h+1} with C_h yields a dense correspondence residual Δf_h:

$$\tilde{f}_h = C_h\, f_{t+h+1}, \qquad \Delta f_h = \tilde{f}_h - f_{t+h}. \tag{6}$$

Algorithm 1 Stage I training of UniLAP.

Require: dataset $\mathcal{D}$, horizon H, weights $\lambda_{\text{d}}, \lambda_{\text{m}}$, learning rate η, frozen $\mathcal{E}$ and LAM q

repeat
  $\big(\{o_{t+j}\}_{j=0}^{H},\, \ell\big) \sim \mathcal{D},\ \{f_{t+j},\, k_{t+j}\}_{j=0}^{H} \leftarrow \mathcal{E}\big(\{o_{t+j}\}_{j=0}^{H}\big)$
  $(\mu_h, \Sigma_h) \leftarrow q(o_{t+h}, o_{t+h+1}),\ z_h \sim \mathcal{N}(\mu_h, \Sigma_h)$ ▷ teacher target
  $C_h \leftarrow \mathrm{Softmax}\big(k_{t+h}\, k_{t+h+1}^{\top}/\sqrt{d_k}\big),\ \Delta f_h \leftarrow C_h\, f_{t+h+1} - f_{t+h}$ ▷ motion target
  $\tau \sim \mathcal{U}(0,1),\ x_{0,h} \sim \mathcal{N}(0, I),\ x_{\tau,h} \leftarrow (1-\tau)\, x_{0,h} + \tau\, z_h$
  $\hat{v}_h \leftarrow v_\theta(x_{\tau,h},\, o_t,\, \ell),\ \hat{z}_{\tau,h} \leftarrow x_{\tau,h} + (1-\tau)\, \hat{v}_h$ ▷ student forward
  $\{\Delta\hat{f}_h\}_{h=0}^{H-1} \leftarrow \phi_{\text{mot}}(\hat{Z}_\tau,\, f_t)$ ▷ motion prediction
  $\theta \leftarrow \theta - \eta\, \nabla_\theta\big(\mathcal{L}_{\text{FM}} + \lambda_{\text{d}}\mathcal{L}_{\text{distill}} + \lambda_{\text{m}}\mathcal{L}_{\text{motion}}\big)$
until converged

Since DINOv2 features stay stable across adjacent frames (Oquab et al., 2023), $\tilde{f}_h$ and f_{t+h} share identity content and differ mainly in motion. Their residual Δf_h is a feature-space proxy for that motion, measuring how matched content changes once re-indexed onto the current patches.

Latent-Conditioned Motion Predictor. We connect the latent action policy to the target Δf_h with a motion predictor ϕ_{mot} that estimates a per-step residual over the chunk-anchor patch tokens f_t, conditioned on $\hat{Z}_\tau = \{\hat{z}_{\tau,h}\}_{h=0}^{H-1}$. We pass the latents via a causal self-attention, so each $\bar{z}_{\tau,h} = \mathrm{CausalAttn}_h(\hat{Z}_\tau)$ aggregates only from its predecessors, and then modulates f_t via AdaLN:

$$[\gamma_h, \beta_h] = \psi_{\text{mod}}(\bar{z}_{\tau,h}), \quad u_{h,n} = \mathrm{LN}(f_{t,n}) \odot (1+\gamma_h) + \beta_h, \tag{7}$$

where ψ_{mod} is a shallow MLP. Since f_t comes from the frozen $\mathcal{E}$, no gradient flows through it. By construction, f_t carries the static scene structure shared across all H steps, while $\bar{z}_{\tau,h}$ provides the per-step action signal. A patch-wise decoder $\mathcal{P}$ maps the modulated tokens $u_{h,n}$ to per-patch residuals $\Delta\hat{f}_{h,n} = \mathcal{P}(u_{h,n})$, supervised by:

$$\mathcal{L}_{\text{motion}} = \frac{1}{N}\sum_{n=1}^{N}\left(1 - \frac{\Delta\hat{f}_{h,n}^{\top}\Delta f_{h,n}}{\|\Delta\hat{f}_{h,n}\|\,\|\Delta f_{h,n}\|}\right). \tag{8}$$

This supervision trains ϕ_{mot} as an action-conditioned latent dynamics model that predicts correspondence residuals from the current visual features and latent actions, encouraging the latent actions to encode motion dynamics. We jointly optimize the losses on action-free videos:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{FM}} + \lambda_{\text{d}}\mathcal{L}_{\text{distill}} + \lambda_{\text{m}}\mathcal{L}_{\text{motion}}. \tag{9}$$

At inference, we obtain the final latent chunk $\hat{Z} = \{\hat{z}_h\}_{h=0}^{H-1}$ by integrating v_θ from $\tau{=}0$ to 1 given o_t and ℓ, and discard the continuous LAM teacher and motion predictor. Algorithm 1 summarizes the training process.

3.3 Embodiment-Specific Action Decoding

To deploy the pretrained latent policy on a target embodiment, a decoder is required to map the latent chunk to its action space. We train this decoder from scratch and finetune the latent policy on the target's labeled actions. Let $p_t \in \mathbb{R}^{d_p}$ denote the proprioceptive state and $f_t \in \mathbb{R}^{N\times d_{\text{obs}}}$ the chunk-anchor patch tokens reused from Sec. 3.2. The decoder ϕ_{dec} produces the embodiment-specific action chunk $\hat{A} = \{\hat{a}_h\}_{h=0}^{H-1}$ with $\hat{a}_h \in \mathbb{R}^{d_a}$ from $\hat{Z} = \{\hat{z}_h\}_{h=0}^{H-1}$, conditioned on p_t and f_t along distinct pathways. As a global hardware descriptor, p_t passes through an MLP ψ_{prop} and modulates every step feature-wise via AdaLN, conditioning the chunk uniformly on the target hardware. As a spatially structured sequence, f_t is instead fed into cross-attention, allowing each step to attend to relevant regions. We augment $\hat{Z}$ with a learned positional embedding, compose the streams through L stacked layers, and read out $\hat{A}$ with an MLP head. We optimize the decoder by a chunk-level mean-squared error:

$$\mathcal{L}_{\text{act}} = \frac{1}{H\, d_a}\sum_{h=0}^{H-1}\|\hat{a}_h - a_h\|_2^2, \tag{10}$$

where d_a is the action dimension. Action labels enter only through $\mathcal{L}_{\text{act}}$, so Stage I stays action-free and the latent policy serves as a shared initialization that Stage II adapts to each embodiment.

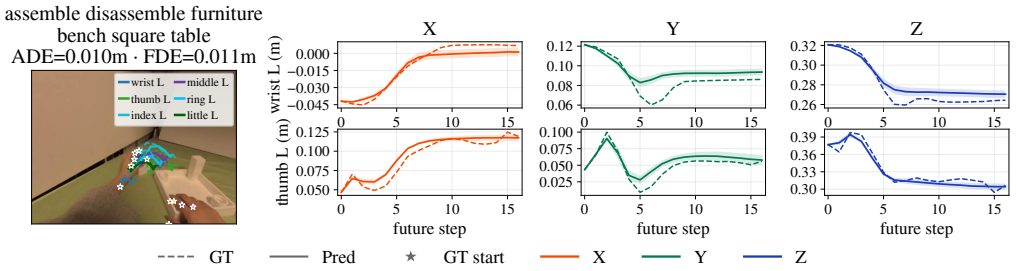

Figure 3: Qualitative hand pose forecasting on EgoDex. *Left* overlays the predicted 1.6 s keypoint trajectories on the egocentric frame. *Right* plots each keypoint's $X/Y/Z$ position over the horizon.

4 Experiment

Implementation Details. Our latent policy is a 16-block flow-matching Transformer predicting an $H{=}16$ latent chunk of dimension $d_z{=}32$, conditioned on frozen DINOv2 (Oquab et al., 2023) and SigLIP2 (Tschannen et al., 2025) visual features and Flan-T5 (Chung et al., 2024) text embeddings. The motion supervision features f_t and keys k_t come from DINOv2, and the LAM teacher is DreamDojo (Gao et al., 2026). Excluding the frozen LAM, the model totals 1.1B parameters, of which only 590M are trainable, since the backbones are frozen and only the latent policy, motion predictor, and the 8-block action decoder are learned. Stage I pretrains the latent policy and motion predictor under $\mathcal{L}_{\text{total}}$ on label-free training data of human (EgoDex (Hoque et al., 2025)) and multi-robot (RoboCasa-GR1 (Bjorck et al., 2025), Fractal (O'Neill et al., 2023), BridgeData V2 (Walke et al., 2023), RoboCasa365 (Nasiriany et al., 2026), DreamDojo-GR1 (Gao et al., 2026)) videos for 250k steps at batch size 128. Stage II trains an action decoder from scratch and finetunes the policy under $\mathcal{L}_{\text{act}}$ for each target embodiment. Refer to Sec. A for more architectural and training details.

Evaluation Benchmarks. We evaluate UniLAP on three benchmarks, *EgoDex* (Hoque et al., 2025) for bimanual hand pose forecasting, *SimplerEnv* (Li et al., 2024) (*WidowX* and *Google Robot*) for real-to-sim single-arm manipulation, and *RoboCasa-GR1* (Bjorck et al., 2025) for humanoid tabletop manipulation. The three benchmarks span different action spaces, a 48-D bimanual hand pose for EgoDex, a 7-DoF single arm for SimplerEnv, and a 29-D humanoid joint space for RoboCasa-GR1. We report Average and Final Displacement Error (ADE / FDE) for EgoDex and success rate for the SimplerEnv and RoboCasa-GR1 benchmarks. See Sec. B for data statistics and evaluation protocols. We also deploy UniLAP on an xArm7, an embodiment unseen in Stage I pretraining, for three tabletop tasks, reporting success rate over 50 rollouts per task.

4.1 Main Results

Hand Pose Forecasting. Tab. 1 reports our results on EgoDex, comparing against BC, DDPM, and FM variants of the decoder-only (Dec) and encoder-decoder (EncDec) architectures from X-IL (Jia et al., 2025). Best-of-S reports the lowest error across S samples, so the deterministic BC baselines stay flat in S while the stochastic diffusion and flow-matching baselines improve with it. Our method posts 0.029 / 0.038 m ADE / FDE at $S{=}10$, and its relative ADE margin over the strongest baseline widens from 13% at $S{=}1$ to 24% at $S{=}10$. Since this gap also holds against the flow-matching baselines, it stems from modeling actions in a latent space, not from flow matching itself. Fig. 3 plots a predicted trajectory whose keypoints track the ground truth and diverge only late in the horizon, indicating that the policy captures fine finger-level motion.

Real-to-Sim Evaluation on SimplerEnv. In Tab. 3, we group the baselines into policies optimized end-to-end on labeled robot data (e.g., π_0 (Black et al., 2024), GR00T-N1 (Bjorck et al., 2025)) and methods that prepend an action-free pretraining stage (UniVLA (Bu et al., 2025a), mimic-video (Pai et al., 2025)). Since AdaWorld and DreamDojo are world models rather than standalone control policies, we use such an encoder as our frozen LAM teacher rather than as a policy baseline. CLAM learns a separate policy for each task, whereas UniLAP learns one language-conditioned policy for all tasks. On *WidowX*, UniLAP scores 52.1% while optimizing only 590M of its 1.1B parameters, exceeding UniVLA and mimic-video by 4.2 and 5.2, respectively. We attribute this advantage to modeling latent actions as a continuous flow rather than a discrete codebook, which retains the mul-

Methods	ADE (m)↓			FDE (m) ↓		
	S=1	S=5	S=10	S=1	S=5	S=10
Dec + BC	0.039	0.039	0.039	0.053	0.053	0.053
Dec + DDPM	0.054	0.041	0.039	0.071	0.050	0.044
Dec + FM	0.052	0.041	0.038	0.071	0.051	0.044
EncDec + BC	0.040	0.040	0.040	0.053	0.053	0.053
EncDec + DDPM	0.058	0.044	0.041	0.074	0.052	0.046
EncDec + FM	0.059	0.045	0.041	0.077	0.054	0.047
UniLAP	**0.034**	**0.031**	**0.029**	**0.047**	**0.040**	**0.038**

Table 1: EgoDex hand pose forecasting results. ADE and FDE are best-of-S errors in meters.

Tasks	GR00T-N1.6	StarVLA-π	Qwen3-DiT	UniLAP
BottleToCabinet	51.5	26.0	50.0	42.0
CanToDrawer	13.0	62.0	48.0	78.0
CupToDrawer	8.5	42.0	42.0	24.0
MilkToMicrowave	14.0	50.0	38.0	44.0
PotatoToMicrowave	41.5	42.0	18.0	34.0
WineToCabinet	16.5	32.0	28.0	34.0
Avg	24.2	42.3	37.3	**42.7**

Table 2: RoboCasa-GR1 success rate (%). Avg is over the six tasks.

Models	Params	Put Spoon on Towel	Put Carrot on Plate	Stack Green Block on Yellow Block	Put Eggplant in Yellow Basket	Avg
RT-1-X	35M	0.0	4.2	0.0	0.0	1.1
DINOv2 + DP	86M	0.0	8.0	0.0	26.0	8.5
Octo-Base	93M	15.8	12.5	0.0	41.7	17.5
OpenVLA	7B	8.3	4.2	0.0	45.8	14.6
π_0	3B	29.1	0.0	16.6	62.5	27.1
GR00T-N1	2B	62.5	45.8	16.7	20.8	36.5
RoboVLMs	2B	45.8	20.8	4.2	79.2	37.5
villa-X	3B	48.3	24.2	19.2	71.7	40.8
UniVLA	7B	52.8	55.6	2.8	80.6	47.9
mimic-video	2B	37.5	37.5	12.5	100.0	46.9
UniLAP(Ours)	1.1B	58.3	54.2	12.5	83.3	**52.1**

Models	Params	*Visual Matching (VM)*				*Variant Aggregation (VA)*			
		Pick Can	Move Near	Open/Close Drawer	Avg	Pick Can	Move Near	Open/Close Drawer	Avg
RT-1-X	35M	56.7	31.7	59.7	49.4	49.0	32.3	29.4	36.9
DINOv2 + DP	86M	31.0	5.0	60.0	32.0	-	-	-	-
Octo-Base	93M	17.0	4.2	22.7	14.6	0.6	3.1	1.1	1.6
OpenVLA	7B	18.0	56.3	63.0	45.8	60.8	67.7	28.8	52.4
π_0	3B	72.7	65.3	38.3	58.8	75.2	63.7	25.6	54.8
GR00T-N1	2B	47.0	70.0	18.1	45.0	78.8	62.5	13.2	51.5
GR00T-N1.5	2B	51.7	54.0	27.8	44.5	69.3	68.7	35.8	**57.9**
villa-X	3B	81.7	55.4	38.4	58.5	-	-	-	-
DeFI	2.5B	54.2	60.7	38.6	51.2	-	-	-	-
UniLAP(Ours)	1.1B	68.3	67.1	43.5	**59.6**	84.3	51.5	35.5	57.1

Table 3: Per-task success rate (%) on SimplerEnv (Li et al., 2024). *Top*: WidowX. *Bottom*: Google Robot. Baselines are RT-1-X (O'Neill et al., 2023), DINOv2 + DP (Dong et al., 2026), Octo-Base (Team et al., 2024), OpenVLA (Kim et al., 2024), π_0 (Black et al., 2024), GR00T-N1 and N1.5 (Bjorck et al., 2025), RoboVLMs (Liu et al., 2025), villa-X (Chen et al., 2025b), UniVLA (Bu et al., 2025a), mimic-video (Pai et al., 2025), and DeFI (Zhang et al., 2026).

timodal structure of plausible futures. *Google Robot* instead probes robustness to visual variation. UniLAP achieves 59.6% under Visual Matching and 57.1% under the perturbed Variant Aggregation protocol, a degradation of only 2.5. We trace this stability to our motion predictor, since a latent grounded in DINOv2 correspondences encodes motion structure largely invariant to appearance.

Manipulation on RoboCasa-GR1. Tab. 2 compares UniLAP on 6 articulated multi-step RoboCasa-GR1 tasks against GR00T-N1.6 (Bjorck et al., 2025), StarVLA-π (Community, 2026), and Qwen3DiT (Ma et al., 2026). Unlike these, UniLAP adapts its action-free pretrained latent to the humanoid rather than learning the policy end-to-end on labeled actions. Aggregated over the six tasks, it manages a comparable 42.7% average, a low-cost transfer to a high-DoF embodiment.

Real-World Deployment. We deploy UniLAP on a physical xArm7 with a ZED 2i (*left view*) camera, comparing with $\pi_{0.5}$ (Intelligence et al., 2025) on three tabletop tasks (Fig. 4). Following our train recipe, we collect 50 teleoperated demonstrations per task, fit an 8-DoF joint-space decoder from scratch, and finetune the latent policy under $\mathcal{L}_{\text{act}}$. The three tasks share a pick-and-place

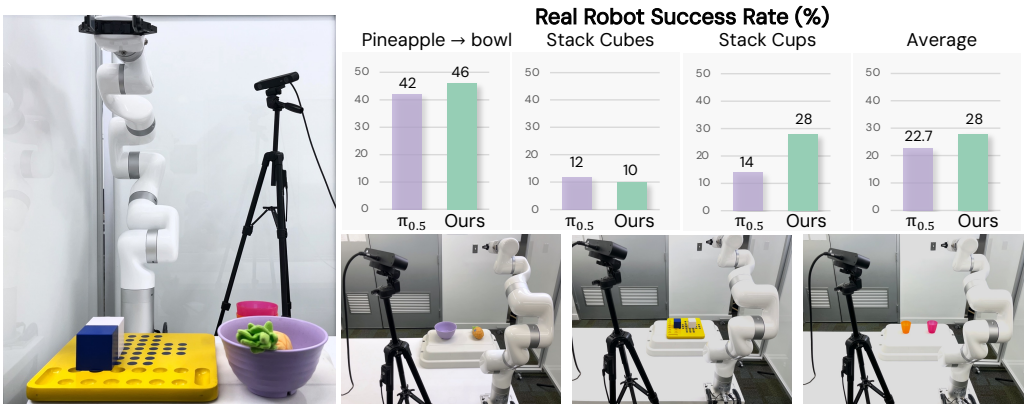

Figure 4: Real-world setup and results. *Left*, the single-camera xArm7 tabletop workspace. *Right*, per-task and average success rate over 50 rollouts. See the supplementary video for details.

structure but escalate in placement precision, from a tolerant drop (`Pineapple→Bowl`) through aligned nesting (`Stack Cups`) to a tight stack against soft-contact rebound (`Stack Cubes`). Across 50 rollouts evaluation per task, UniLAP averages 28.0% against $\pi_{0.5}$'s 22.7%. Results track this precision ordering and localize where the latent helps. Because of the task-intrinsic difficulty, both methods struggle with cube stacking. In contrast, UniLAP doubles $\pi_{0.5}$'s success rate on cup stacking, benefiting from a continuous latent that preserves the fine motor distinctions. On an unseen compositional instruction ("pick up the red cup and put it into the bowl"), UniLAP succeeds on 23 of 50 rollouts, recombining learned grounding and skills rather than replaying memorized trajectories. For more real-world analysis, inference latency, and per-task diagnostics see Sec. C.

4.2 Analysis and Ablation

Latent Action Probing. We probe learned latent $\hat{z}_h$ along three axes, each isolating a property the policy needs. *Teacher fidelity.* The LAM teacher forms posterior from (o_{t+h}, o_{t+h+1}) while our policy predicts $\hat{z}_h$ from o_t alone. Decoded from o_{t+h} by the *same* frozen LAM decoder (Gao et al., 2026), $\hat{z}_h$ reconstructs o_{t+h+1} close to the teacher's (Fig. 6, where o_t is shown for reference). The policy thus recovers the teacher's predictive content without observing the future. *Language sensitivity.* The LAM teacher takes no instruction, so understanding language is a capability the policy must learn. The same frame recurs under different instructions with different teacher targets, and ℓ is the only input that separates them. Holding o_t fixed while varying the instruction, $\hat{z}_h$ stays near the floor under paraphrase or object swaps but shifts 2.7× and 4.3× under verb or location changes and 17.5× when language is removed (Fig. 7), tracking what motion and where. *Multimodality.* To evaluate latent diversity, we decode 64 latents per observation using the frozen action decoder. For 46.5% of 512 EgoDex observations, the predicted endpoints split into two distinct clusters (median separation 4.5 cm), compared to only 2 of 100 unimodal controls. This confirms that our model captures genuinely multimodal action distributions rather than unimodal averages (Fig. 5, Sec. D.2).

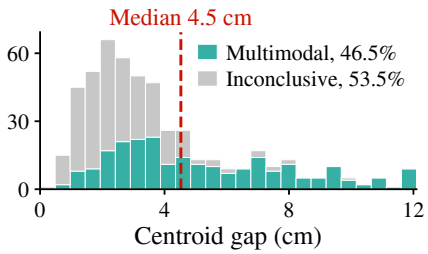

Figure 5: Latent multimodality.

Ablation. We vary Stage I while keeping the Stage II training recipe and evaluation protocol unchanged (Tab. 4 & Fig. 7). *Stage I pretraining.* A randomly initialized latent policy, adapted by the same Stage II, peaks at 33.3% WidowX SR against the full model's 44.4% (Fig. 7), confirming that Stage I pretraining drives the downstream gain. *Distillation objective.* The $\mathcal{L}_{\text{FM}}$-only variant reaches 40.3% WidowX SR, our τ-weighting raises it to 44.4%, and a constant weight drops it to 38.9%. This constant weight enforces the alignment to the posterior mean even when the latent is still noise, which is why it ends up worse than no distillation at all. Both ablated latents also reconstruct the future at lower PSNR (Sec. D.5). *Training data.* Pretraining on EgoDex or robot videos alone trails

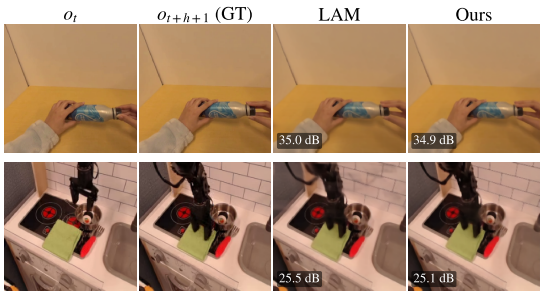

Figure 6: Latent action quality probing. Each row shows o_t for reference, the GT o_{t+h+1}, and reconstructions from the teacher latent and from our $\hat{z}_h$.

Variant	**EgoDex** (m)	**WidowX** (%)
	FDE ↓	SR (3 tasks) ↑
UniLAP (full)	**0.0384**	**44.4**
Stage I objective		
$\mathcal{L}_{\text{FM}}$ only	0.0391	40.3
Constant weight in $\mathcal{L}_{\text{distill}}$	0.0388	38.9
Stage I training data		
EgoDex only	0.0392	34.7
Robot only	0.0469	41.7
Stage I teacher (robot-only data, compare with Robot only)		
UniVLA VQ teacher	0.0482	29.2

Table 4: Component ablations. Every variant changes Stage I only and shares the same Stage II recipe and protocol.

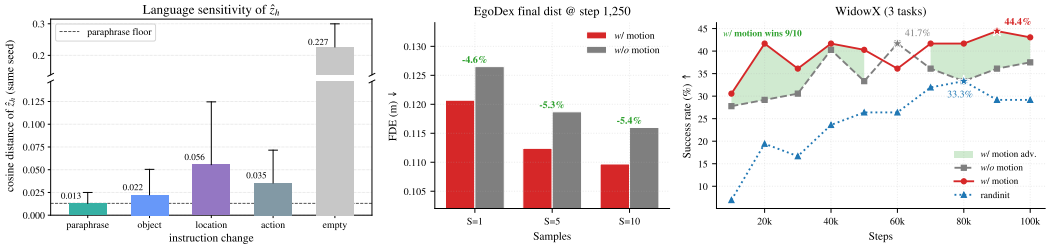

Figure 7: Language sensitivity (*left*), motion supervision, and pretraining (*middle & right*) ablations.

the mixture by 9.7 and 2.7 WidowX and is also worse on EgoDex itself, so human and robot data contribute complementary priors. *Teacher representation.* Replacing the continuous teacher (Gao et al., 2026) with the VQ LAM released by UniVLA (Bu et al., 2025a), which encodes each frame transition into four tokens from a 16×128 codebook, lowers WidowX SR from 41.7 to 29.2 and raises EgoDex FDE from 0.0469 to 0.0482 under the robot-only data setting that both runs share. *Motion supervision.* Adding $\mathcal{L}_{\text{motion}}$ reduces early-training EgoDex FDE by 4.6 to 5.4%, and the motion-supervised variant wins 9 of 10 WidowX checkpoints (Fig. 7), so the correspondence target grounds the latent in dynamics throughout training rather than merely warming it up.

5 Conclusion and Limitations

We presented UniLAP, a unified latent action policy for cross-embodiment manipulation, learned from action-free videos. It models the latent action as a continuous flow-matching policy distilled from a frozen LAM teacher under a flow-adaptive posterior objective, anchors the latent in motion via correspondence-based supervision in a frozen ViT, and grounds it on each robot with a separate action decoder and a light policy finetune. Across hand pose forecasting, single-arm and humanoid manipulation, UniLAP compares favorably with prior latent-action and VLA baselines while training far fewer parameters than the larger ones. We hope this continuous, motion-grounded latent offers a reusable, embodiment-agnostic interface for learning manipulation from large-scale action-free videos across human and robots.

UniLAP has several limitations that scope our claims and point to future work. At 1.1B parameters with 590M trainable, it keeps visual and language backbones frozen for efficiency, rather than co-training them end-to-end (Community, 2026), so scaling or unfreezing them to gain perceptual capacity is a natural step. Reliance on a single camera may limit performance under occlusion, so how to integrate complementary observations, such as a wrist view, remains open. Scaling the human corpus beyond EgoDex offers a further route to a stronger shared latent, since our ablation shows human and robot videos contribute complementary priors.

REPRODUCIBILITY STATEMENT

Sec. A specifies our overall architecture, the frozen LAM teacher, the two-stage training recipe and inference procedure, and Tab. 6 collects the detailed model and optimization settings. The Stage I pretraining corpus is listed per source with episode counts and frame rates in Tab. 7, and all six corpora are public datasets. Algorithm 1 gives the Stage I training step from a sampled video window to the parameter update, with the loss terms defined in Sec. 3.1 and Sec. 3.2. For Stage II, Tab. 8 gives each embodiment's action and proprioception dimensions, and its adaptation budget, and Sec. A states the normalization applied to proprioception and actions. Evaluation protocols for EgoDex, SimplerEnv and RoboCasa-GR1, including trial counts and episode lengths, are given in Sec. B. The real-robot platform, data collection and preprocessing, deployment rates, task suite and per-task rollout counts are described in Sec. C. We will release the code upon publication.

ETHICS STATEMENT

This work pretrains on six public datasets, EgoDex (Hoque et al., 2025), BridgeData V2 (Walke et al., 2023), Fractal (O'Neill et al., 2023), RoboCasa-GR1 (Bjorck et al., 2025), RoboCasa365 (Nasiriany et al., 2026) and DreamDojo-GR1 (Gao et al., 2026), and distills a public LAM teacher (Gao et al., 2026). EgoDex is egocentric human video, so the pretraining data contains people's hands and first-person views of their surroundings, and we neither add annotations to it nor redistribute it. The only data we collected are 150 real-robot demonstrations, recorded by the authors via VR teleoperation of a 7-DoF arm in a lab workspace, with no recruited participants and no third-party human subjects. The real-robot evaluation runs in a bounded workspace with an emergency stop available throughout, and the robot policy has not been validated for operation near people or in safety-critical settings. We do not release any new dataset with this paper.

AI USE STATEMENT

We used a large language model to aid and polish writing. Its role was limited to rewording and tightening the text the authors had drafted, and to checking terminology and cross-references for consistency throughout the paper. We did not use generative AI tools to generate research ideas or design experiments. We take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

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.

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. *arXiv preprint arXiv:2512.13030*, 2025.

Hongzhe Bi, Lingxuan Wu, Tianwei Lin, Hengkai Tan, Zhizhong Su, Hang Su, and Jun Zhu. H-rdt: Human manipulation enhanced bimanual robotic manipulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 18135–18143, 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. pi_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

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*, 2025a.

Xizhou Bu, Jiexi Lyu, Fulei Sun, Ruichen Yang, Zhiqiang Ma, and Wei Li. Laof: Robust latent action learning with optical flow constraints. *arXiv preprint arXiv:2511.16407*, 2025b.

Boyuan Chen, Tianyuan Zhang, Haoran Geng, Kiwhan Song, Caiyi Zhang, Peihao Li, William T Freeman, Jitendra Malik, Pieter Abbeel, Russ Tedrake, et al. Large video planner enables generalizable robot control. *arXiv preprint arXiv:2512.15840*, 2025a.

Xiaoyu Chen, Junliang Guo, Tianyu He, Chuheng Zhang, Pushi Zhang, Derek Cathera Yang, Li Zhao, and Jiang Bian. Igor: Image-goal representations are the atomic control units for foundation models in embodied ai. *arXiv preprint arXiv:2411.00785*, 2024.

Xiaoyu Chen, Hangxing Wei, Pushi Zhang, Chuheng Zhang, Kaixin Wang, Yanjiang Guo, Rushuai Yang, Yucen Wang, Xinquan Xiao, Li Zhao, et al. Villa-x: enhancing latent action modeling in vision-language-action models. *arXiv preprint arXiv:2507.23682*, 2025b.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19752–19763, 2025c.

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.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.

StarVLA Community. Starvla: A lego-like codebase for vision-language-action model developing. *arXiv preprint arXiv:2604.05014*, 2026.

Jiahua Dong, Yunze Man, Pavel Tokmakov, and Yu-Xiong Wang. Capturing visual environment structure correlates with control performance. *arXiv preprint arXiv:2602.04880*, 2026.

Shenyuan Gao, Siyuan Zhou, Yilun Du, Jun Zhang, and Chuang Gan. Adaworld: Learning adaptable world models with latent actions. *arXiv preprint arXiv:2503.18938*, 2025.

Shenyuan Gao, William Liang, Kaiyuan Zheng, Ayaan Malik, Seonghyeon Ye, Sihyun Yu, Wei-Cheng Tseng, Yuzhu Dong, Kaichun Mo, Chen-Hsuan Lin, et al. Dreamdojo: A generalist robot world model from large-scale human videos. *arXiv preprint arXiv:2602.06949*, 2026.

Raktim Gautam Goswami, Amir Bar, David Fan, Tsung-Yen Yang, Gaoyue Zhou, Prashanth Krishnamurthy, Michael Rabbat, Farshad Khorrami, and Yann LeCun. World models can leverage human videos for dexterous manipulation. *arXiv preprint arXiv:2512.13644*, 2025.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2(3):440, 2018.

Ryan Hoque, Peide Huang, David J Yoon, Mouli Sivapurapu, and Jian Zhang. Egodex: Learning dexterous manipulation from large-scale egocentric video. *arXiv preprint arXiv:2505.11709*, 2025.

Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. *arXiv preprint arXiv:2412.14803*, 2024.

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.

Xiaogang Jia, Atalay Donat, Xi Huang, Xuan Zhao, Denis Blessing, Hongyi Zhou, Han A Wang, Hanyi Zhang, Qian Wang, Rudolf Lioutikov, et al. X-il: Exploring the design space of imitation learning policies. *arXiv preprint arXiv:2502.12330*, 2025.

Guangqi Jiang, Yutong Liang, Jianglong Ye, Jia-Yang Huang, Changwei Jing, Rocky Duan, Pieter Abbeel, Xiaolong Wang, and Xueyan Zou. Cross-hand latent representation for vision-language-action models. *arXiv preprint arXiv:2603.10158*, 2026.

Simar Kareer, Dhruv Patel, Ryan Punamiya, Pranay Mathur, Shuo Cheng, Chen Wang, Judy Hoffman, and Danfei Xu. Egomimic: Scaling imitation learning via egocentric video. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 13226–13233. IEEE, 2025.

Hanjung Kim, Jaehyun Kang, Hyolim Kang, Meedeum Cho, Seon Joo Kim, and Youngwoon Lee. Uniskill: Imitating human videos via cross-embodiment skill representations. *arXiv preprint arXiv:2505.08787*, 2025.

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.

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.

Zuolei Li, Xingyu Gao, Xiaofan Wang, and Jianlong Fu. Latbot: Distilling universal latent actions for vision-language-action models. *arXiv preprint arXiv:2511.23034*, 2025.

Anthony Liang, Pavel Czempin, Matthew Hong, Yutai Zhou, Erdem Biyik, and Stephen Tu. Clam: Continuous latent action models for robot learning from unlabeled demonstrations. *arXiv preprint arXiv:2505.04999*, 2025a.

Junbang Liang, Pavel Tokmakov, Ruoshi Liu, Sruthi Sudhakar, Paarth Shah, Rares Ambrus, and Carl Vondrick. Video generators are robot policies. *arXiv preprint arXiv:2508.00795*, 2025b.

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

Huaping Liu, Xinghang Li, Peiyan Li, Minghuan Liu, Dong Wang, Jirong Liu, Bingyi Kang, Xiao Ma, Tao Kong, and Hanbo Zhang. Towards generalist robot policies: What matters in building vision-language-action models. *arXiv preprint arXiv:2412.14058*, 2025.

Jiangran Lyu, Kai Liu, Xuheng Zhang, Haoran Liao, Yusen Feng, Wenxuan Zhu, Tingrui Shen, Jiayi Chen, Jiazhao Zhang, Yifei Dong, et al. Lda-1b: Scaling latent dynamics action model via universal embodied data ingestion. *arXiv preprint arXiv:2602.12215*, 2026.

Teli Ma, Jia Zheng, Zifan Wang, Chunli Jiang, Andy Cui, Junwei Liang, and Shuo Yang. Dit4dit: Jointly modeling video dynamics and actions for generalizable robot control. *arXiv preprint arXiv:2603.10448*, 2026.

Soroush Nasiriany, Sepehr Nasiriany, Abhiram Maddukuri, and Yuke Zhu. Robocasa365: A large-scale simulation framework for training and benchmarking generalist robots. *arXiv preprint arXiv:2603.04356*, 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.

Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 1(2), 2023.

Jonas Pai, Liam Achenbach, Victoriano Montesinos, Benedek Forrai, Oier Mees, and Elvis Nava. mimic-video: Video-action models for generalizable robot control beyond vlas. *arXiv preprint arXiv:2512.15692*, 2025.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 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*, 2018.

Dominik Schmidt and Minqi Jiang. Learning to act without actions. *arXiv preprint arXiv:2312.10812*, 2023.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning*, pp. 1723–1736. PMLR, 2023.

Songlin Wei, Hongyi Jing, Boqian Li, Zhenyu Zhao, Jiageng Mao, Zhenhao Ni, Sicheng He, Jie Liu, Xiawei Liu, Kaidi Kang, et al. ψ_0: An open foundation model towards universal humanoid loco-manipulation. *arXiv preprint arXiv:2603.12263*, 2026.

Fuxiang Yang, Donglin Di, Lulu Tang, Xuancheng Zhang, Lei Fan, Hao Li, Chen Wei, Tonghua Su, and Baorui Ma. Chain of world: World model thinking in latent motion. *arXiv preprint arXiv:2603.03195*, 2026.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. *arXiv preprint arXiv:2410.11758*, 2024.

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.

Zeya Yin, Tin Lai, Lucas Barcelos, Jayadeep Jacob, Yonghui Li, and Fabio Ramos. Diverse motion planning with stein diffusion trajectory inference. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 15610–15616. IEEE, 2025.

Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, Xinqiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, Fan Lu, He Wang, et al. Dreamvla: a vision-language-action model dreamed with comprehensive world knowledge. *arXiv preprint arXiv:2507.04447*, 2025a.

Wenyao Zhang, Bozhou Zhang, Zekun Qi, Wenjun Zeng, Xin Jin, and Li Zhang. Disentangled robot learning via separate forward and inverse dynamics pretraining. In *The Fourteenth International Conference on Learning Representations*, 2026.

Yang Zhang, Chenwei Wang, Ouyang Lu, Yuan Zhao, Yunfei Ge, Zhenglong Sun, Xiu Li, Chi Zhang, Chenjia Bai, and Xuelong Li. Align-then-steer: Adapting the vision-language action models through unified latent guidance. *arXiv preprint arXiv:2509.02055*, 2025b.

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*, 2025a.

Ruijie Zheng, Dantong Niu, Yuqi Xie, Jing Wang, Mengda Xu, Yunfan Jiang, Fernando Castañeda, Fengyuan Hu, You Liang Tan, Letian Fu, et al. Egoscale: Scaling dexterous manipulation with diverse egocentric human data. *arXiv preprint arXiv:2602.16710*, 2026.

Ying Zheng, Lei Yao, Yuejiao Su, Yi Zhang, Yi Wang, Sicheng Zhao, Yiyi Zhang, and Lap-Pui Chau. A survey of embodied learning for object-centric robotic manipulation. *Machine Intelligence Research*, 22(4):588–626, 2025b.

OVERVIEW

This supplementary material extends the main paper in four parts. Sec. A specifies the model architecture as well as the training and inference details. Sec. B describes the EgoDex, SimplerEnv, and RoboCasa-GR1 evaluation benchmarks and their protocols. Sec. C details the real-robot setup, the task suite, qualitative rollouts, and a per-task analysis. Sec. D reports additional analysis, covering latent probing, language sensitivity, hand-pose forecasts, and latent reconstructions. Tab. 5 collects the notation used in this paper.

Symbol	Meaning
t, H, h	time step, chunk horizon and chunk index
N, n	number of patch tokens per frame and patch index
o_t, ℓ	current observation frame and language instruction
τ	flow timestep
$z_h \in \mathbb{R}^{d_z}$	latent action of the transition (o_{t+h}, o_{t+h+1}), sampled from the LAM teacher
μ_h, Σ_h	teacher posterior mean and diagonal covariance
$\pi(\cdot \mid o_t, \ell)$, v_θ	latent policy and the velocity field
$x_{0,h}$, $x_{\tau,h}$	initial noise and its linear interpolant toward z_h at flow timestep τ
p^*	teacher posterior marginalized over the unobserved future, the target of $\mathcal{L}_{\text{FM}}$
$\hat{z}_{\tau,h}$	estimate of the latent action at flow timestep τ, with $\hat{z}_h$ its endpoint at $\tau{=}1$
$\mathcal{E}$	frozen DINOv2 encoder
$f_{t+h} \in \mathbb{R}^{N\times d_{\text{obs}}}$, $k_{t+h} \in \mathbb{R}^{N\times d_k}$	patch tokens and last-block keys of frame o_{t+h} from $\mathcal{E}$
f_t	patch tokens of the current frame (f_{t+h} at $h{=}0$), the chunk anchor shared by all H steps
$C_h \in \mathbb{R}^{N\times N}$, $\tilde{f}_h$	softmax correspondence matrix from the current frame to the next, and the next-frame tokens it warps onto the current patches
Δf_h	correspondence residual between the warped and current tokens, the motion target
ϕ_{mot}	motion predictor
$\bar{z}_{\tau,h}$	latent aggregated causally over steps up to h
ψ_{mod}	shallow MLP producing the AdaLN modulation
γ_h, β_h	AdaLN scale and shift
$u_{h,n}$	anchor token n modulated for step h
$\mathcal{P}$	patch-wise decoder
$\Delta\hat{f}_{h,n}$	predicted residual at patch n
$p_t \in \mathbb{R}^{d_p}$	proprioceptive state
ϕ_{dec}, ψ_{prop}	action decoder and the MLP embedding proprioception
$\hat{A}$, $\hat{a}_h$, $a_h \in \mathbb{R}^{d_a}$	predicted action chunk, its step-h action and the ground-truth action
$\mathcal{L}_{\text{FM}}$, $\mathcal{L}_{\text{distill}}$, $\mathcal{L}_{\text{motion}}$, $\mathcal{L}_{\text{total}}$	Stage I flow-matching, distillation and motion losses, and their weighted sum
λ_{d}, λ_{m}	weights of the distillation and motion losses in $\mathcal{L}_{\text{total}}$
$\mathcal{L}_{\text{act}}$	Stage II action loss

Table 5: Notation used in Sec. 3.

A MODEL ARCHITECTURE AND TRAINING DETAILS

A.1 MODEL ARCHITECTURE

Our latent action policy is a 16-block flow-matching (Lipman et al., 2022) Transformer that predicts an $H{=}16$ action chunk of latent dimension $d_z{=}32$ in a single forward pass. It conditions on the flow timestep with AdaLN (Perez et al., 2018; Peebles & Xie, 2023) and on the current observation frame o_t and instruction tokens via per-block cross-attention. Two complementary visual encoders supply the visual conditioning, DINOv2-base (B) (Oquab et al., 2023) for fine-grained spatial features and

Component / Hyperparameter	Value
Architecture	
Latent policy: blocks / width / heads	16 / 1024 / 16
Action decoder: blocks / width / heads	8 / 768 / 12
MLP ratio	4
Action decoder dropout	0.1
Latent dimension d_z	32
Action chunk horizon H	16
Observation history	1 frame
Spatial visual encoder	DINOv2-base
Semantic visual encoder	SigLIP2-base
Text encoder	Flan-T5-large
Total / trainable parameters	1.1B / 590M
Stage I pretraining (action-free)	
Optimizer	AdamW
Learning-rate schedule	cosine, 5% warmup
Peak learning rate	1×10^{-4}
Weight decay	0.01
Batch size	128 (64×2 GPUs)
Training steps	250k
Precision	bf16-mixed
Gradient clip	1.0
EMA decay	0.9999
Distillation weight λ_{d}	0.05
Motion weight λ_{m}	0.5
Stage II adaptation (per embodiment)	
Action decoder initialization	from scratch
Action decoder learning rate	1×10^{-4}
Latent policy finetune learning rate	1×10^{-5}
Batch size (effective)	256 (16 for the real robot)
Latent flow integration	4-step Euler
Per-embodiment d_a, d_p, steps	see Tab. 8

Table 6: Architecture and optimization settings of UniLAP.

SigLIP2-base (B) (Tschannen et al., 2025) for vision-language semantics, and the frozen Flan-T5-large (L) (Chung et al., 2024) backbone embeds the instruction. Both visual encoders consume a 224×224 single-view RGB image. DINOv2-B (patch size 14) yields $N=256$ patch tokens of width $d_{\text{obs}}=768$, and SigLIP2-B (patch size 16) yields 196 tokens, each linearly projected to the policy with width 1024. The motion supervision reuses the DINOv2 patch tokens and its last-block key vectors of dimension $d_k=768$. The motion predictor reads the latent chunk via a 4-head causal self-attention, modulates the frozen anchor features by AdaLN, and decodes patch-wise residuals with a two-layer MLP. This predictor is used only in Stage I and discarded for Stage II and inference. The embodiment-specific action decoder is an 8-block Transformer which conditions on proprioception via AdaLN and on the anchor patch tokens with cross-attention. Only the latent policy, motion predictor, and action decoder are trainable (590M of the 1.1B total parameters), while the visual and language backbones stay frozen. The 1.1B excludes the frozen DreamDojo LAM, whose 404.8M encoder provides the teacher posterior. Tab. 6 lists the architecture and optimization settings.

A.2 LAM Teacher and Latent Distillation

The teacher is a frozen DreamDojo (Gao et al., 2026) latent action encoder (24 blocks, width 1024, patch size 16) that maps each transition (o_{t+h}, o_{t+h+1}) to a diagonal-Gaussian posterior $\mathcal{N}(\mu_h, \Sigma_h)$ over the $d_z=32$ latent. Each transition is a consecutive frame pair at each corpus's native rate (stride 1), namely 5 Hz for BridgeData V2 (Walke et al., 2023), 3 Hz for Fractal (O'Neill et al., 2023), and 20 Hz for the RoboCasa-GR1 (Bjorck et al., 2025) and RoboCasa365 (Nasiriany et al., 2026) corpora. EgoDex (Hoque et al., 2025) is the exception, subsampled by $3\times$ to 10 Hz following H-RDT (Bi et al., 2026) and Ψ_0 (Wei et al., 2026). The flow-matching target is a latent z_h sampled from this posterior. We train with classifier-free guidance at an instruction-dropout rate of 0.1.

Pretraining corpus	Domain	Episodes	Frames	Rate (Hz)
EgoDex	Human, egocentric	338,234	90.0M	30→10
BridgeData V2	WidowX, real	53,192	1.9M	5
Fractal	Google robot, real	87,212	3.8M	3
RoboCasa-GR1	GR1, simulation	24,000	6.0M	20
DreamDojo-GR1	GR1, real	22,209	6.4M	20
RoboCasa365	RoboCasa, simulation	32,043	29.1M	20
Total		556,890	137.2M	

Table 7: Stage I pretraining corpus. Frames are native-rate totals, though we subsample only EgoDex to 10 Hz.

UniVLA VQ teacher. The LAM teacher ablation in Tab. 4 keeps the student, losses, and Stage II recipe fixed and replaces only the teacher with the discrete LAM released by UniVLA (Bu et al., 2025a), which encodes each frame transition into four tokens drawn from a 16×128 codebook. Evaluation follows the three-task WidowX protocol used in the ablations (Tab. 4).

A.3 Pretraining Corpus

Stage I pretrains on a label-free mixture of $556{,}890$ trajectories spanning egocentric human video and four robot embodiments (Tab. 7). It pairs a large human prior from EgoDex (Hoque et al., 2025) with single-arm manipulation from BridgeData V2 (Walke et al., 2023) and Fractal (O'Neill et al., 2023), simulated GR1 humanoid manipulation from RoboCasa-GR1 (Bjorck et al., 2025), real GR1 manipulation from DreamDojo-GR1 (Gao et al., 2026), and simulated kitchen manipulation from RoboCasa365 (Nasiriany et al., 2026). Spanning human, simulated, and real sources exposes the latent to diverse dynamics during pretraining.

A.4 Training and Inference Configuration

Stage I optimizes $\mathcal{L}_{\text{total}}$ on action-free videos for 250k steps at batch size 128, without any explicit action labels, weighting the distillation and motion terms by $\lambda_{\text{d}}{=}0.05$ and $\lambda_{\text{m}}{=}0.5$. We concatenate all corpora into one pool and sample uniformly from it, so each corpus contributes in proportion to its size. The flow-matching term regresses the velocity v_θ along the linear interpolant $x_{\tau,h}{=}(1-\tau)x_{0,h}+\tau z_h$ toward $v{=}z_h-x_{0,h}$, sampling $\tau\sim\mathcal{U}(0,1)$ under an MSE loss. The motion predictor supervises per-patch residuals at $M{=}4$ evenly strided steps ($h\in\{3,7,11,15\}$) of the $H{=}16$ chunk for training efficiency. Stage II trains a fresh action decoder on the target embodiment's labeled actions and adapts the latent policy (Tab. 8). Both stages use AdamW with a cosine learning-rate schedule, bf16-mixed precision, gradient clipping at 1.0, and weight EMA at decay 0.9999. The learning rate warms up linearly over the first 5% of steps to its peak (Tab. 6), then decays to $0.1\times$ the peak in Stage I and 0.01–$0.05\times$ in Stage II. In Stage II, the latent chunk is produced by 4-step Euler integration of the velocity field, and the action decoder maps it to executable commands.

A.5 Per-Embodiment Configuration

Stage II attaches an 8-block action decoder that translates the latent chunk to a d_a-dimensional action conditioned on a d_p-dimensional proprioceptive state, and adapts the shared latent policy at horizon $H{=}16$. For EgoDex, we freeze the latent policy because the DreamDojo teacher was pretrained on EgoDex and its latent already captures the domain, and train only the decoder. For the other embodiments, we finetune the latent policy at 1×10^{-5}, $0.1\times$ the decoder learning rate. We normalize every proprioception (and target action) dimension to $[-1,1]$ by $x\mapsto\operatorname{clip}\big(2(x-q_{01})/(q_{99}-q_{01})-1,-1,1\big)$, where q_{01} and q_{99} are the per-dimension percentiles of each corpus. GR1 first encodes its joint angles as $[\sin q,\cos q]$, which doubles its proprioception to 58 dimensions, whereas the other embodiments normalize the raw state. Tab. 8 reports the per-embodiment d_a, d_p, freeze setting, and Stage II step budget, while Tab. 6 collects the optimization settings shared across embodiments.

Benchmark	Embodiment	Action space (d_a)	d_p	Tasks	Metric	LAP	Steps
EgoDex	Egocentric human	48-D hand delta	48	–	ADE / FDE (m)	frozen	40k
SimplerEnv-WidowX	WidowX arm	7-D EE (delta)	8	4	Success rate (%)	finetuned	100k
SimplerEnv-Google	Google robot arm	7-D EE (delta)	8	3	Success rate (%)	finetuned	30k
RoboCasa-GR1	GR1 humanoid	29-D joints	58	6	Success rate (%)	finetuned	100k
Real world (ours)	xArm7 arm	8-D joints	8	3	Success rate (%)	finetuned	20k

Table 8: Per-embodiment overview. The action space gives the action dimension d_a and its form, d_p is the proprioception dimension, EE denotes end-effector, LAP marks whether the latent policy is frozen or finetuned, and Steps is the Stage II adaptation budget.

B Evaluation Benchmarks

We evaluate UniLAP across four evaluation settings: egocentric human, real-to-sim single-arm robots, a humanoid, and a physical robot. Tab. 8 summarizes each benchmark's embodiment, action and proprioception dimensions, task count, metric, and Stage II adaptation. We detail EgoDex, SimplerEnv, and RoboCasa-GR1 below and cover the physical robot in Sec. C.

EgoDex (Hoque et al., 2025) is an egocentric dataset of human dexterous manipulation captured with Apple Vision Pro, with 338k trajectories (90M frames) across 194 tasks. The 48-D action is the camera-frame delta of a bimanual hand pose, comprising a 9-D wrist pose (3-D position and 6-D rotation) and five fingertip positions (5×3) per hand. We adopt the official train/test split and report on its 3,243-trajectory test set, downsampling the 30 Hz capture to 10 Hz. From a single egocentric RGB frame and the current hand pose, UniLAP forecasts a 16-step (1.6 s) chunk. We score Average and Final Displacement Error (ADE / FDE, m) in the camera frame over the 12 keypoints (two wrists and ten fingertips), aggregated as best-of-S over $S \in \{1, 5, 10\}$ following the official evaluation protocol.

SimplerEnv (Li et al., 2024) provides reproducible real-to-sim proxies for single-arm manipulation, and we follow the evaluation setup of StarVLA (Community, 2026). We train the action decoder on BridgeData V2 (Walke et al., 2023) for the *WidowX* suite of four tabletop tasks (*Put Spoon on Towel*, *Put Carrot on Plate*, *Stack Green Cube on Yellow Cube*, and *Put Eggplant in Yellow Basket*), and on Fractal (O'Neill et al., 2023) for the *Google Robot* suite of three tasks (*Pick Coke Can*, *Move Near*, and *Open/Close Drawer*). We report per-task and average success rate, with 24 trials per WidowX task and 300, 240, and 216 trials for the Google Robot *Pick Coke Can*, *Move Near*, and *Open/Close Drawer* tasks respectively. WidowX runs at 5 Hz control with a 120-step episode, and Google Robot at 3 Hz with an 80-step episode (113 for the drawer). We evaluate WidowX under Visual Matching, and Google Robot under both Visual Matching and Variant Aggregation, which perturbs background, lighting, distractor objects, and surface texture.

RoboCasa-GR1 (Bjorck et al., 2025) offers humanoid tabletop manipulation in the RoboCasa platform (Nasiriany et al., 2026). The GR1 humanoid has two 7-DoF arms, two 6-DoF dexterous hands, and a 3-DoF waist, giving a 29-D joint-space action. The model observes a single egocentric RGB view. We evaluate 6 articulated multi-step tasks. Each pairs an object with a cabinet, drawer, or microwave and issues an instruction such as "pick up the bottle, place it into the cabinet, and close the cabinet". Success requires shutting the receptacle's door, and Tab. 2 abbreviates the tasks as, e.g., *BottleToCabinet*. We learn the action decoder from 1000 demonstrations per task and report success over 50 rollouts per task, evaluated in 10 parallel environments at a 720-step episode.

C Real-Robot Experiments

C.1 Setup

Platform. We use a 7-DoF UFACTORY xArm7 with a parallel-jaw gripper and a single uncalibrated Stereolabs ZED 2i stereo camera mounted on a tripod at the left rear of the arm (Fig. 8). We take its left-eye RGB at HD720 (1280×720) as the fixed third-person view, and our model operates on a 224×224 input. The view feeds UniLAP directly, and no extrinsic calibration is needed.



Figure 8: Real-robot platform: the 7-DoF xArm7 with a parallel-jaw gripper, the tripod-mounted ZED 2i that supplies the fixed third-person view, and the Meta Quest controllers used for VR teleoperation during data collection.

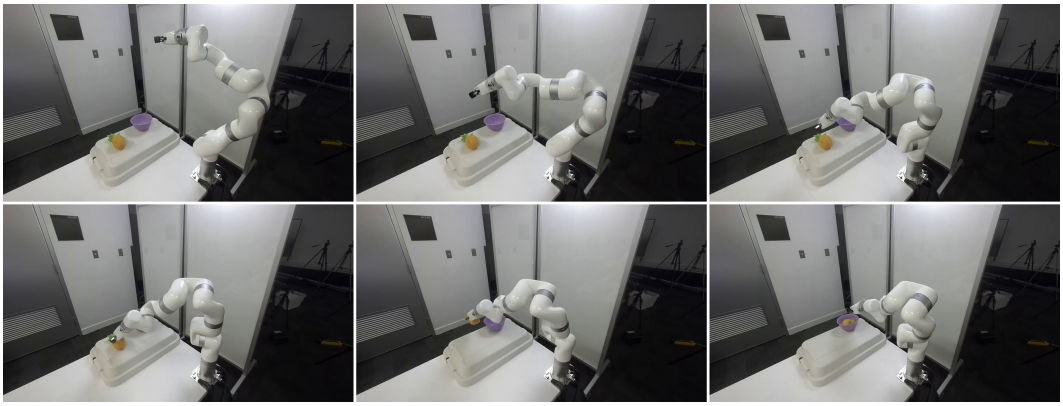

Figure 9: Teleoperated demonstration of *T1: Pineapple→Bowl* in the robot's observation view, read left to right, top to bottom. Zoom in for better viewing.

Observation and action. At each step, UniLAP receives the current left-view RGB frame and an 8-D proprioceptive vector of 7 arm joint angles and the gripper opening. It predicts a chunk of 16 actions, each 8-D, comprising relative joint-position targets $q_{t+1+h} - q_t$ and an absolute normalized gripper command. At execution, predicted joint offsets are denormalized and added to the measured configuration.

Data collection. We collect 50 demonstrations per task (150 total) via Meta Quest VR teleoperation that streams end-effector targets to the arm. Fig. 9-11 show one teleoperated demonstration snapshot for each task. Proprioception and the teleoperation command stream are logged at 100 Hz alongside the ZED stereo recording.

Data preprocessing. Each demonstration is uniformly subsampled in time to a fixed length of 49 frames. For each target timestamp we take the temporally nearest left-eye RGB frame, with alignment error within one camera frame (≤ 44 ms), together with the robot state and action sampled from the 100 Hz log, giving a one-to-one correspondence between frames and action targets. Clips are packaged in the LeRobot v2.0 format at a nominal 16 fps. Because every variable-duration episode is normalized to 49 frames, the effective temporal stride varies across demonstrations.



Figure 10: Teleoperated demonstration of *T2: Stack Cubes* in the robot's observation view, read left to right, top to bottom. Zoom in for better viewing.

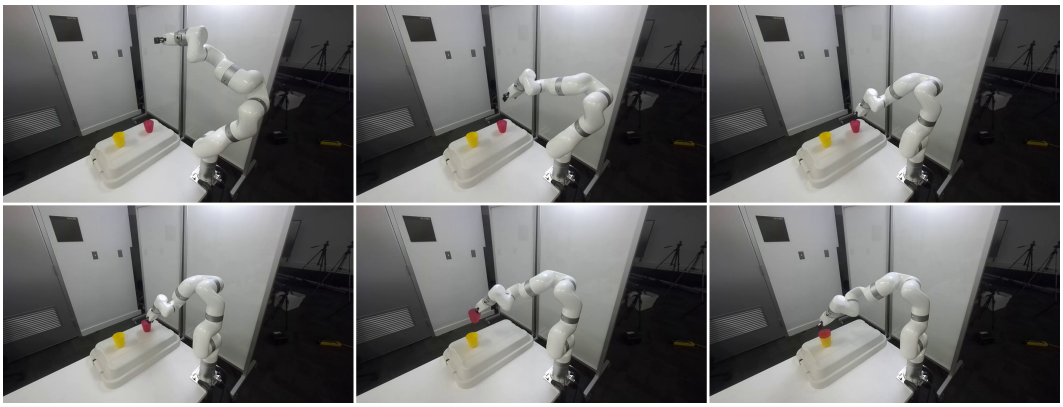

Figure 11: Teleoperated demonstration of *T3: Stack Cups* in the robot's observation view, read left to right, top to bottom. Zoom in for better viewing.

Deployment and control rates. The policy runs as an inference server that a robot-side client queries over ZMQ. Action chunking decouples the policy from the low-level controller across three rates, described next from fastest to slowest. The *servo rate* (200 Hz) is the rate at which the client sends a joint target to the arm's low-level controller, one every 5 ms, and is independent of model inference. The *action rate* (4.8 Hz) is the rate at which the chunk's 16 steps execute, the 16 fps demonstration frame rate scaled by a $0.3\times$ deployment speed, with each step interpolated into ≈ 42 servo commands. The *replanning rate* (~ 0.28 Hz) is the rate at which the client queries a fresh chunk after executing all 16 steps. Each cycle takes ≈ 3.5 s, including action execution and the next query, while the arm tracks each chunk open-loop. One query takes ~ 188 ms end to end (median) on an RTX PRO 5000 Laptop GPU, covering serialization, transport, and the model forward. Chunking lets the servo loop replay the precomputed chunk while no query is in flight, so this latency leaves the 200 Hz servo rate intact. It does age each chunk's observation by that amount and stall the loop once per chunk, lowering the achieved action rate from 4.8 to ~ 4.5 Hz.

C.2 Task Descriptions

The three evaluated tabletop tasks share a pick-and-place structure but escalate in the precision of the placement sub-skill. We additionally probe compositional generalization with an out-of-distribution task.

T1: Pineapple→Bowl. A tolerant drop-in placement with an irregularly shaped object, where a coarse release above the bowl suffices. The instruction is `pick up the pineapple and put it into the bowl`.

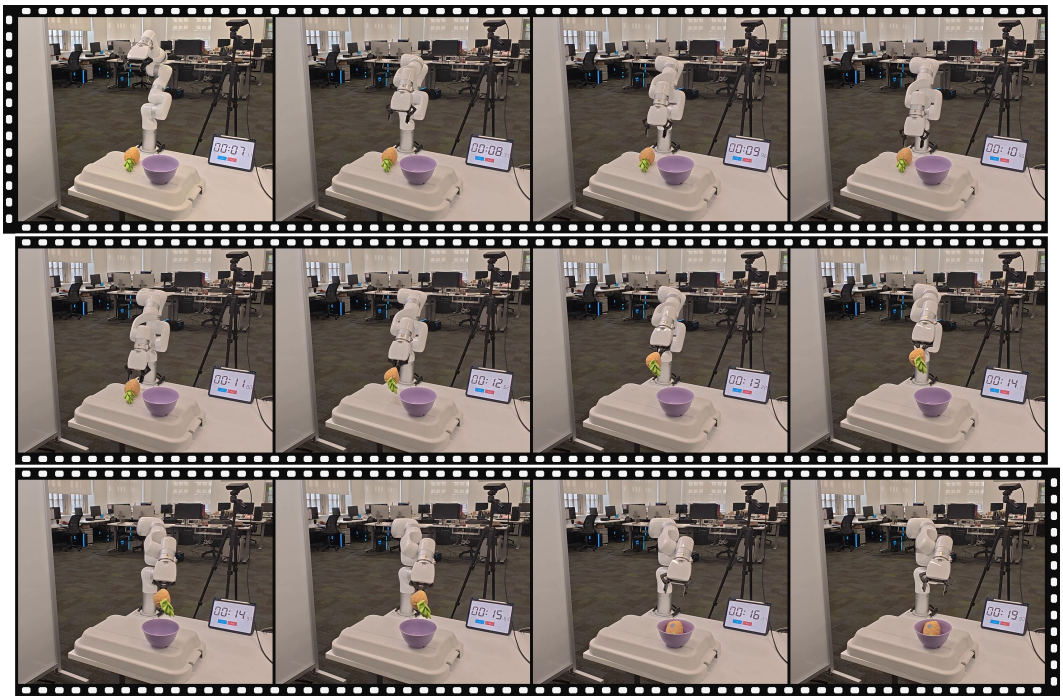

Figure 12: UniLAP rollout on *T1: Pineapple→Bowl*, key frames read left to right, top to bottom. See the supplementary video for the full rollout.

Method	Pineapple → Bowl	Stack Cubes	Stack Cups	**Average**
$\pi_{0.5}$	42	**12**	14	22.7
UniLAP(Ours)	**46**	10	**28**	**28.0**

Table 9: Per-task real-robot success rate (%) over 50 rollouts, comparing against $\pi_{0.5}$ (Intelligence et al., 2025).

T2: Stack Cubes. Stacking a white cube on a blue cube, which requires a precise grasp and aligned placement against soft-contact rebound from the compliant foam. We add a yellow background board, in both data collection and deployment, so the white cube is distinguishable from the white background. The instruction is `stack the white cube on the blue cube`.

T3: Stack Cups. Nesting a red cup on a yellow cup, which demands accurate alignment within a small placement tolerance on thin-walled objects. The instruction is `stack the red cup on the yellow cup`.

OOD: Cup→Bowl. An out-of-distribution compositional task that pairs the red cup (seen only in T3) with the bowl (seen only in T1) in a combination absent from training. The instruction is `pick up the red cup and put it into the bowl`.

C.3 Qualitative Rollouts

Fig. 12-15 show key-frame filmstrips of representative UniLAP rollouts on the three tabletop tasks and the unseen compositional instruction. We encourage viewing the supplementary video for the full sequences.

C.4 Results and Per-Task Analysis

Across 50 rollouts per task, UniLAP averages 28.0% against $\pi_{0.5}$'s 22.7% (Tab. 9). The per-task rates track the precision ordering of the tasks and localize where the latent helps.

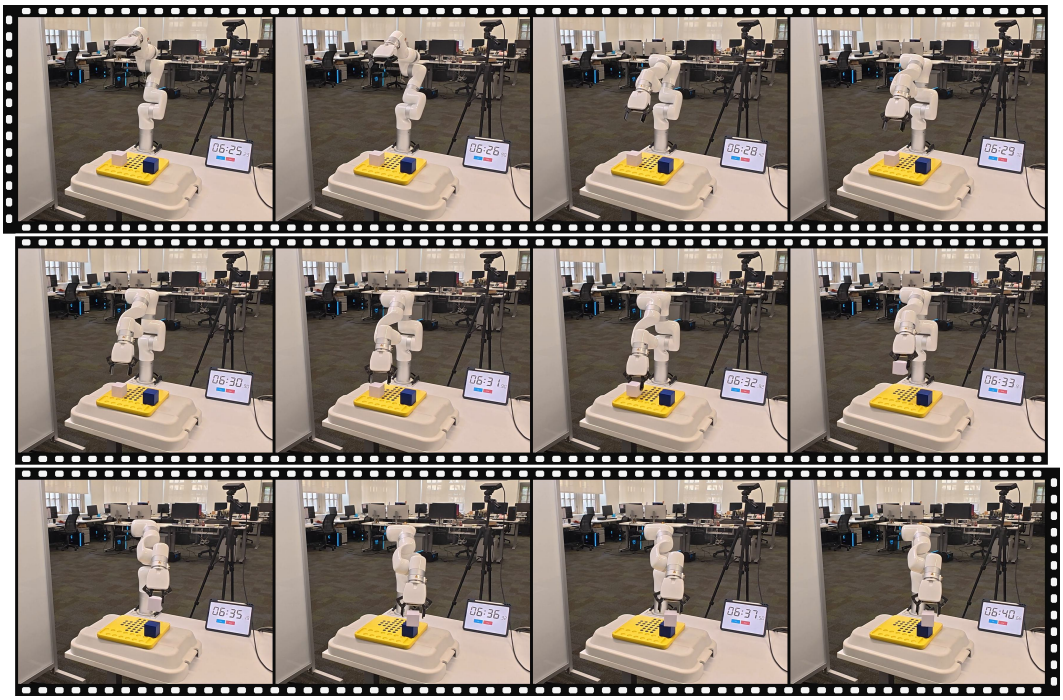

Figure 13: UniLAP rollout on *T2: Stack Cubes*, key frames read left to right, top to bottom. See the supplementary video for the full rollout.

Cube stacking. Both $\pi_{0.5}$ and UniLAP collapse on cube stacking (12% and 10%), which marks a task-intrinsic difficulty. Two factors compound. First, the cube demonstrations show the largest cross-demonstration variability of the release moment and the highest gripper-channel prediction error, which points to the release sub-skill as the likely bottleneck. Second, the foam cubes are compliant, so a slightly mistimed release lets the top cube deform on contact and spring off the stack.

Cup stacking. Nesting one thin-walled cup on another demands accurate alignment within a small placement tolerance. The cup demonstrations are the most consistent in proprioception, with the smallest spread in episode length, gripper cycles, and release height, so the supervision is reliable and the difficulty lies in placement precision itself. Here UniLAP reaches 28% and doubles $\pi_{0.5}$ at 14%, consistent with a continuous latent that preserves the fine motor distinctions.

Compositional generalization. An unseen instruction ("pick up the red cup and put it into the bowl") pairs an object and a target drawn from different training tasks. UniLAP succeeds on 23 of 50 rollouts of this unseen composition, recombining learned grounding and skills rather than replaying a memorized trajectory.

D Additional Analysis

D.1 Latent Probing

We probe the latent's action content by fitting a linear regressor from each latent source to the ground-truth action chunk and reporting the held-out R^2. Evaluation windows are drawn from EgoDex, BridgeData V2 (WidowX), and Fractal (Google Robot), about 3k windows per corpus. The regression target is the recorded action of that corpus, namely the 48-D camera-frame hand delta for EgoDex and the 7-D end-effector delta for Bridge and Fractal. We compare three sources. The policy latent $\hat{Z}$ is the mean of 4 flow-matching samples at guidance scale 1.0. The teacher mean μ is the LAM posterior mean, which observes o_{t+1} and acts as a non-causal reference. The appearance floor is a random projection of the frozen DINOv2 patch features to the latent dimension. Each



Figure 14: UniLAP rollout on *T3: Stack Cups*, key frames read left to right, top to bottom. See the supplementary video for the full rollout.

probe is a ridge regressor whose penalty is selected by RidgeCV, and we report R^2 as the mean and standard deviation over 5 held-out splits. For a latent source $x \in \mathbb{R}^{Hd_z}$ and the flattened ground-truth action chunk $a \in \mathbb{R}^C$ with $C{=}H\,d_a$ channels, the ridge probe predicts $\hat{a}{=}Wx{+}b$, and we average the per-channel coefficient of determination over the held-out windows n,

$$R^2 = \frac{1}{C}\sum_{c=1}^{C}\left(1 - \frac{\sum_n (a_{n,c} - \hat{a}_{n,c})^2}{\sum_n (a_{n,c} - \bar{a}_c)^2}\right), \tag{11}$$

with $\bar{a}_c$ the per-channel mean over those windows. The *Recover* column reports the policy latent's score relative to the oracle teacher mean,

$$\text{Recover} = \frac{R^2(\hat{Z})}{R^2(\mu)}, \tag{12}$$

averaged over the splits where $R^2(\mu){>}0$.

The policy latent recovers 62.4% of the oracle's R^2 on EgoDex, 56.5% on Bridge, and matches it on Fractal, far above the appearance floor (Tab. 10). The latent thus encodes action-predictive information from o_t alone, not generic feature variance.

D.2 Latent Multimodality

We evaluate whether the policy's latent carries more than one plausible action for the same observation. For each of 512 held-out EgoDex observations we draw 64 latents and decode each one with the same frozen action decoder, so the decoding is deterministic and all spread in the terminal 12-keypoint configuration comes from the latent. Standard unimodality tests are one dimensional, so we instead cluster the endpoints with 2-means in their top three principal components and compare the resulting separation against 199 draws from a Gaussian matched to the observed variance, rejecting at $\alpha = 0.05$. The null is rejected for 238 of 512 observations (46.5%), and for 2 of 100 matched controls built from unimodal samples (1 of 100 under a second seed). Among the rejected observations the centroid gap, measured between the two cluster centroids in the original coordinates, has

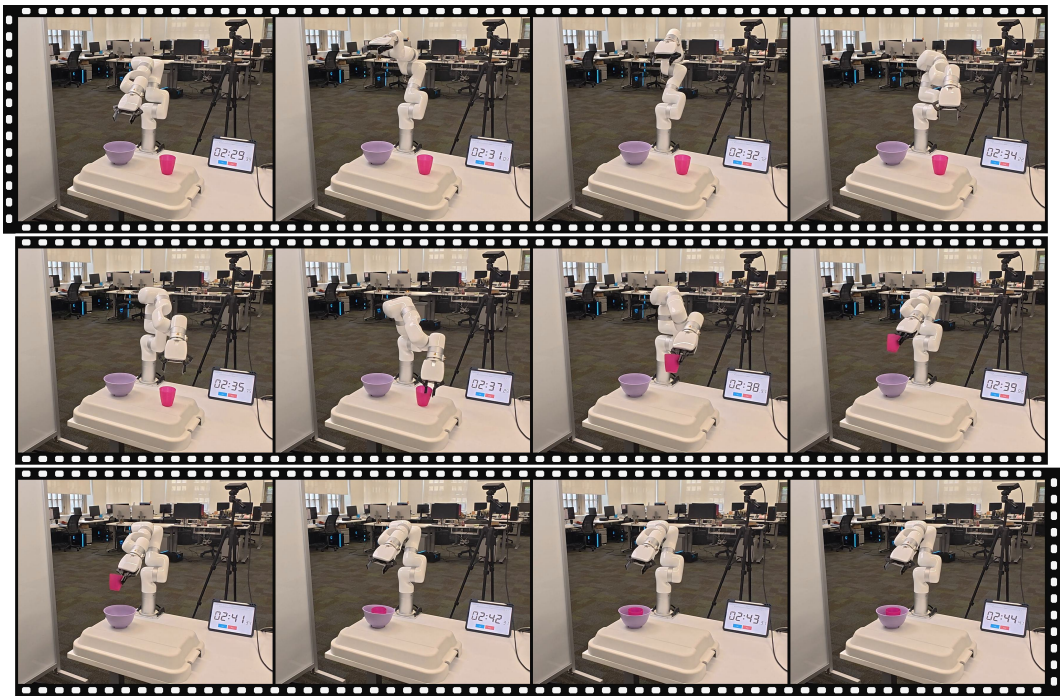

Figure 15: UniLAP rollout on the unseen compositional instruction `pick up the red cup and put it into the bowl`, key frames read left to right, top to bottom. See the supplementary video for the full rollout.

Latent source	**Action R^2 ↑**		
	EgoDex	Bridge	Fractal
Random Proj. f_t	$0.123_{\pm 0.002}$	$0.024_{\pm 0.008}$	$0.104_{\pm 0.007}$
Teacher μ (oracle)	$0.282_{\pm 0.002}$	$0.203_{\pm 0.002}$	$0.186_{\pm 0.007}$
UniLAP(Ours)	$0.176_{\pm 0.003}$	$0.115_{\pm 0.006}$	$0.194_{\pm 0.009}$
Recover	$62.4_{\pm 0.5}\%$	$56.5_{\pm 3.3}\%$	$104.1_{\pm 3.0}\%$

Table 10: Aggregate linear probing of $\hat{Z}$ for action prediction on EgoDex, BridgeData V2 (WidowX), and Fractal (Google Robot). Recover gives the policy latent's R^2 as a fraction of the oracle's.

median 4.53 cm with quartiles 2.98 and 7.29 cm, comparable to the $S{=}1$ FDE of 4.7 cm, and the minority cluster holds a quarter of the samples. Non-rejection is not acceptance, so the remaining observations are labeled inconclusive in Fig. 5 rather than unimodal.

D.3 Language Sensitivity

We quantify the instruction sensitivity of the predicted latent (Fig. 7, *left*). Holding the observation o_t fixed, we perturb the instruction across five categories: paraphrase, object swap, verb change, location change, and full removal. For each, we compute the cosine distance between the predicted latent chunk $\hat{Z}$ under the original and perturbed instruction at a shared flow-matching seed, which isolates the instruction-induced change from sampling noise. We average this distance over 32 observation frames and 4 seeds. The figure shows raw cosine distances, while relative changes are measured against the paraphrase-level distance. Paraphrase and object swap remain at this floor, whereas verb change, location change, and removal induce progressively larger shifts, indicating that the latent encodes the target action and location rather than surface phrasing.

D.4 Additional Hand Pose Forecasts

We extend the EgoDex qualitative study of Fig. 3, which plots the wrist and thumb of an assemble/disassemble furniture case. Fig. 16 adds the four finger keypoints omitted from that case, and Fig. 17 and Fig. 18 show the full six-keypoint set for a stack task and a vertical pick-and-place scene. The predictions broadly follow the ground-truth motion across keypoints, with local deviations over the forecasting horizon.

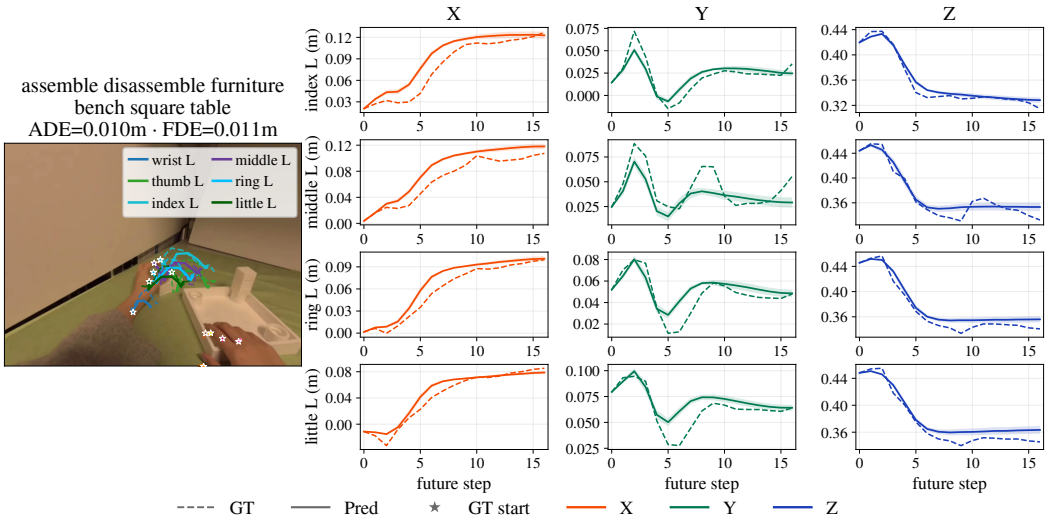

Figure 16: Four finger keypoints (index, middle, ring, little) of the acting hand for the *assemble disassemble furniture* case of Fig. 3, which plots only the wrist and thumb. Each panel shows one keypoint's $X/Y/Z$ over the 1.6 s horizon, with ground truth dashed and prediction solid.

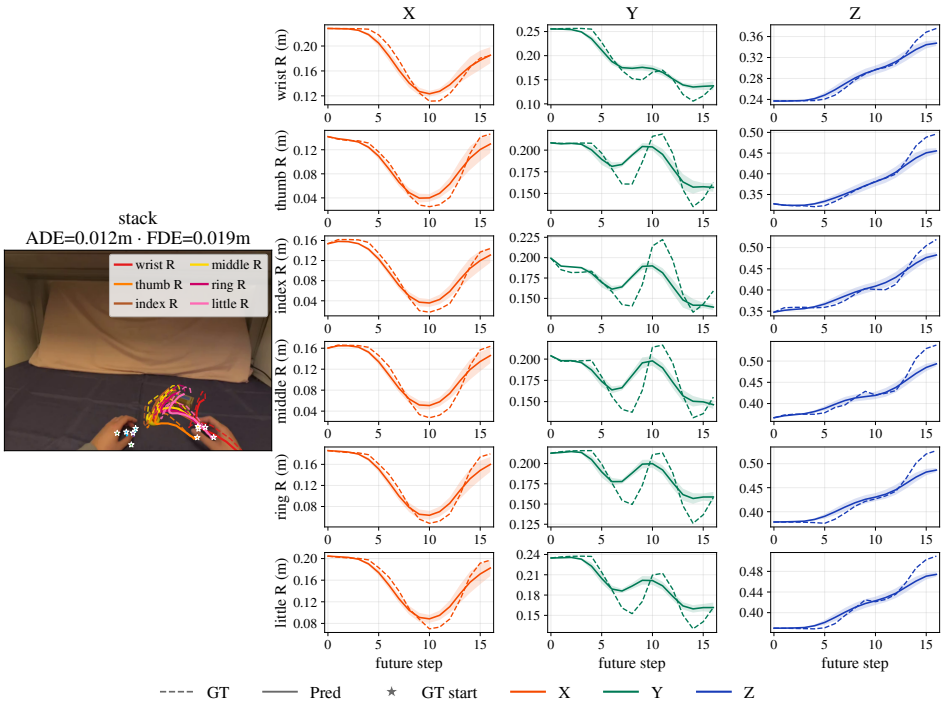

Figure 17: Full per-keypoint forecast on a *stack* task, across all six keypoints of the acting hand. Panels follow Fig. 16: each shows one keypoint's $X/Y/Z$ over the 1.6 s horizon, with ground truth dashed and prediction solid.

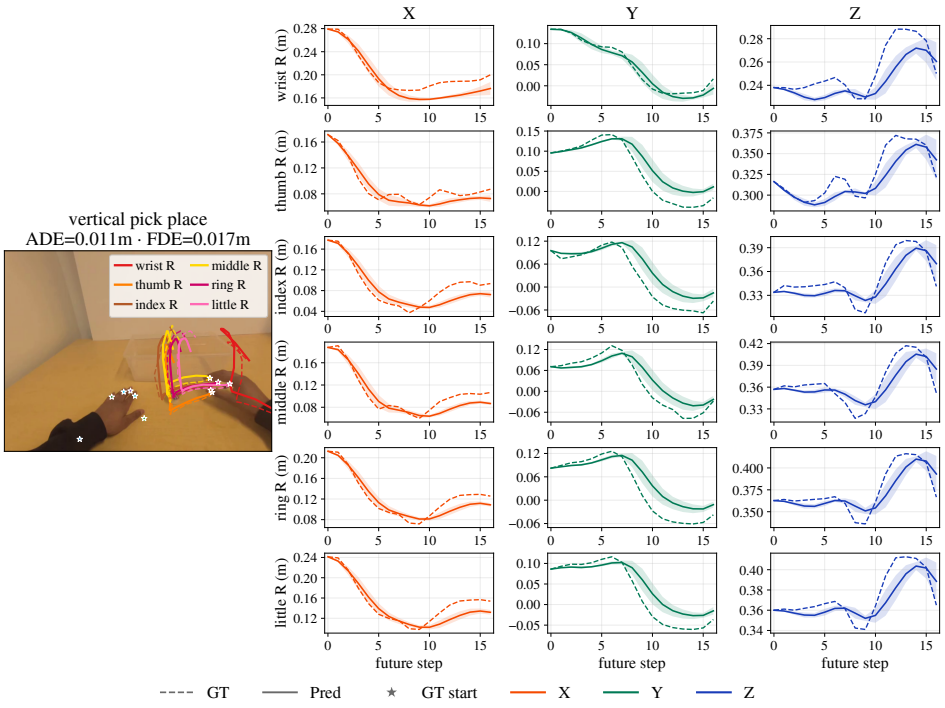

Figure 18: Full per-keypoint forecast on a *vertical pick-and-place* scene, across all six keypoints of the acting hand. Panels follow Fig. 16: each shows one keypoint's $X/Y/Z$ over the 1.6 s horizon, with ground truth dashed and prediction solid.

D.5 Additional Latent Reconstructions

Fig. 6 shows one LAM-decoder reconstruction on EgoDex and WidowX. Fig. 19-21 provide four additional examples each on EgoDex, WidowX (BridgeData V2), and Google Robot (Fractal). Across these examples, UniLAP reconstructs the future close to the LAM teacher in PSNR while the Uniform τ and FM-only variants degrade, so the flow-adaptive objective preserves the teacher's predictive content across embodiments.

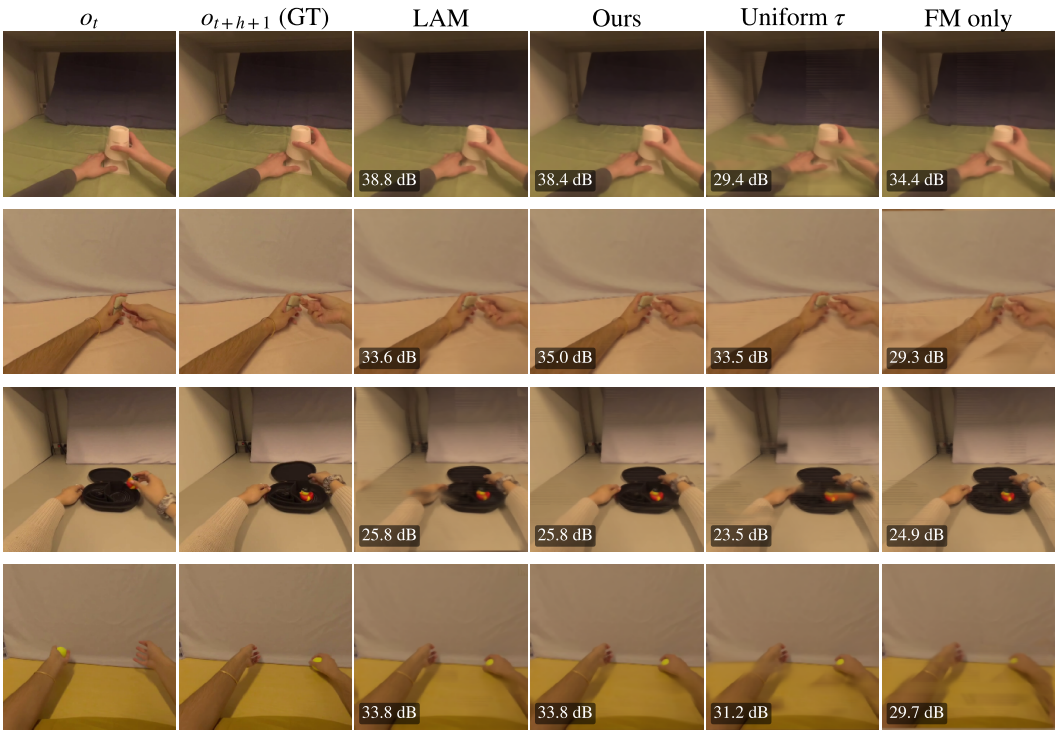

Figure 19: Additional latent reconstructions on EgoDex, four examples. Columns are o_t, the GT o_{t+h+1}, and reconstructions decoded from the teacher, UniLAP, Uniform τ, and FM-only latents, with PSNR against the GT. Each reconstruction is a single-step extrapolation from o_{t+h}, which is not shown; o_t is given for reference.

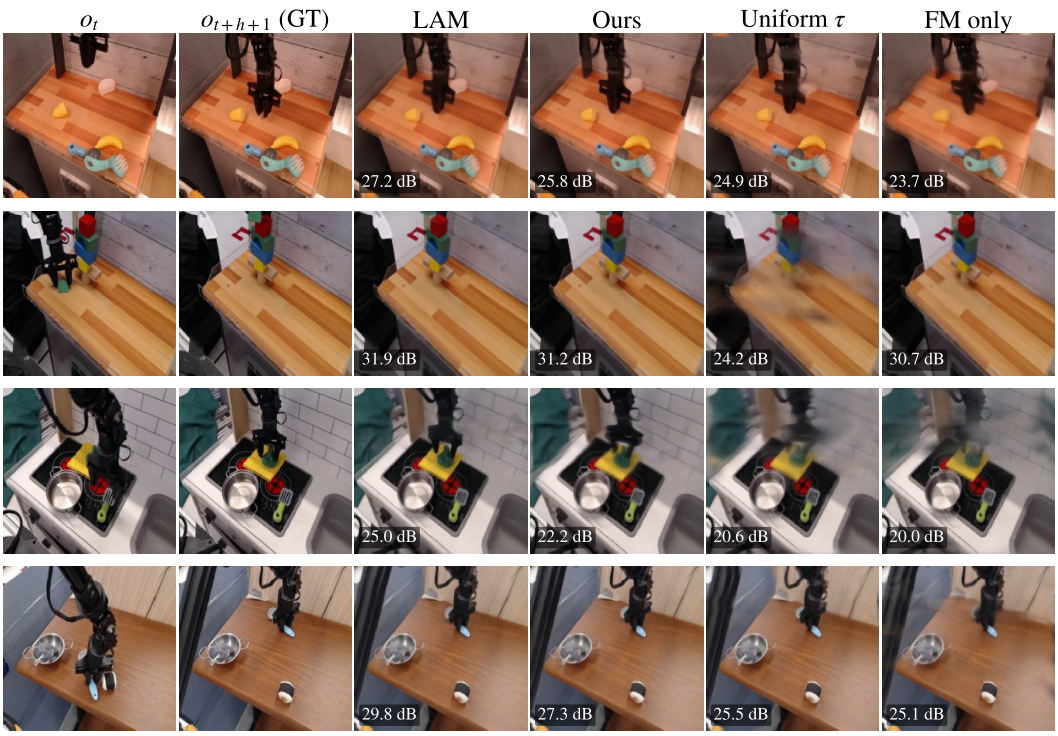

Figure 20: Additional latent reconstructions on WidowX (BridgeData V2). Columns follow Fig. 19.

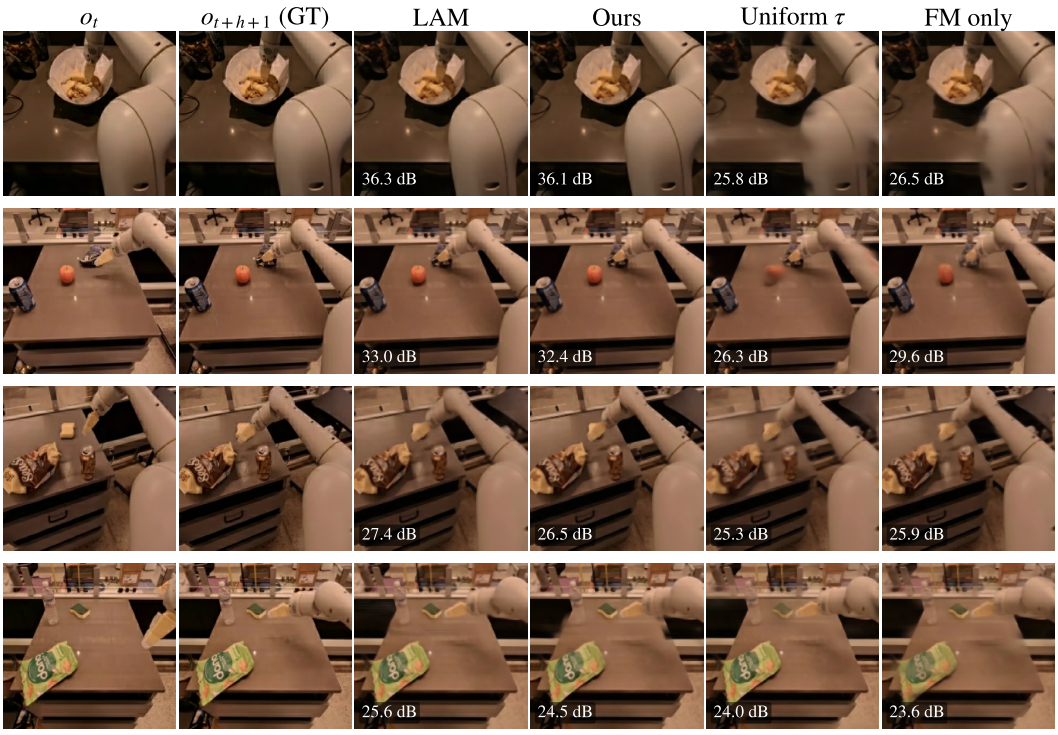

Figure 21: Additional latent reconstructions on Google Robot (Fractal). Columns follow Fig. 19.