LEARNING VISUOMOTOR POLICIES FROM JEPA LATENT TRANSITION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Video pretraining offers more than visual representations: predictive models also learn how visual states evolve. The success of recent generative world-action models (WAMs) suggests that pretrained transition models can themselves provide a transferable foundation for policy learning. We ask whether this principle extends to latent transition models learned through joint-embedding predictive architectures (JEPAs): beyond transferring the visual encoder, can a predictor that *consumes* robot actions be repurposed into a visuomotor policy that *produces* them? We introduce LTM-Policy, which adapts the latent transition model through two complementary routes. A policy route employs learnable action queries to generate robot actions, while a world route retains action-conditioned prediction of future representations from demonstrated transitions. At deployment, the resulting policy generates actions without pixel reconstruction, future-video rollout, or model-predictive search. Controlled experiments show that pretrained transition priors remain beneficial after repurposing the model for action generation. Retaining future-latent prediction during adaptation provides an additional 5.4-point benefit, and the transfer advantage persists in low-data regimes with limited robot demonstrations. Across the four LIBERO suites, LTM-Policy achieves 97.4% average success using only a single forward pass at deployment, without pixel reconstruction or future-video rollout. We further evaluate the method on LIBERO-Plus and real-world manipulation.

1 INTRODUCTION

Large-scale video pretraining provides a promising source of visual knowledge for robot learning. By learning from temporal changes in diverse videos, these models capture regularities of objects, motion, and state transitions at a scale far beyond typical robot demonstrations. A common transfer paradigm is to reuse the visual encoder while learning a new action model from robot data. Predictive video models, however, contain another potential transferable component: a transition model that captures how visual states evolve. Whether this transition model can itself be repurposed as the backbone of a direct visuomotor policy has received less attention. Unlike a visual encoder, which represents individual observations, a transition model captures relations between current and future states under action. Transferring it is therefore a distinct problem from transferring the encoder. The temporal structure it learned for prediction must remain useful once the model is optimized for direct action generation.

Recent world-action models (WAMs) provide encouraging evidence for this broader transfer perspective. By adapting pretrained video generators to jointly model future observations and robot actions, these methods reuse predictive video backbones for robot control (Kim et al., 2026; Ye et al., 2026; Liang et al., 2025; Gu et al., 2026; Yuan et al., 2026). Their success suggests that knowledge captured by a pretrained predictive model can remain useful when the model is adapted for action generation. Existing approaches, however, have largely explored this idea through generative video models that represent transitions by predicting future observations in pixel space, with the predictive backbone remaining tied to a generative model. This raises a complementary question: can the same transfer principle extend to a predictor that models state transitions directly in representation space?

Joint-embedding predictive architectures such as V-JEPA provide a direct setting in which to study this question (Assran et al., 2025). A JEPA consists of an encoder that maps observations into a latent

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.

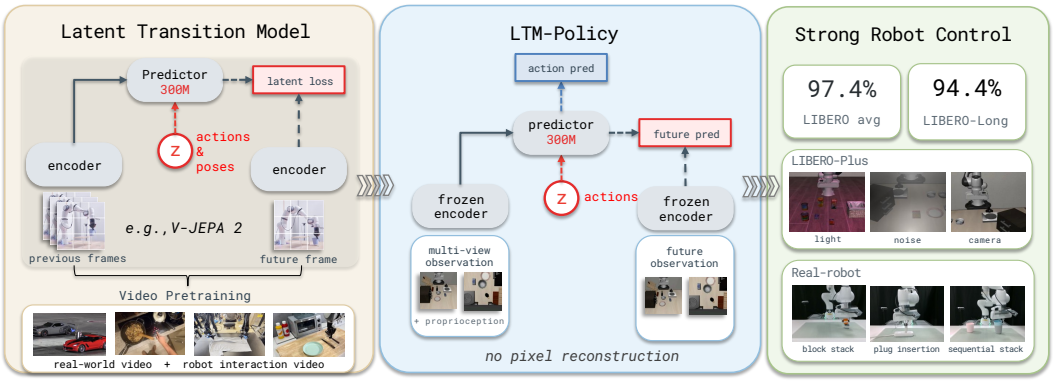


Figure 1: **Learning actions from a latent transition model.** A predictive video model provides both a visual encoder and a latent transition model. We repurpose a pretrained action-conditioned transition model as the backbone of a direct multi-view visuomotor policy and retain latent future prediction during robot adaptation. The resulting policy generates actions without pixel reconstruction, future-video rollout, or test-time planning.

space and a predictor that models how representations evolve. When conditioned on robot actions, this predictor functions as a latent transition model that predicts the resulting future representations. The pretrained transition model was originally used to evaluate actions through model-predictive control rather than to generate them directly. Direct control changes this inference problem: the model must infer unknown actions from the current observation, motivating an adaptation scheme that couples action learning with transition prediction.

We introduce LTM-Policy, which transfers a pretrained action-conditioned latent transition model into the backbone of a direct visuomotor policy. It is adapted through two complementary routes. In the policy route, learned action queries produce action features that condition a lightweight flow-matching expert for continuous action generation. In the world route, demonstrated actions replace these queries and the same model learns their consequences through dense future-latent prediction. The visual encoder remains frozen, while the transition model and action expert are optimized jointly to robot demonstrations. At deployment, only the policy route is required: each replanning step uses a single forward pass, without an image decoder, future-video rollout, or model-predictive search.

Controlled comparisons distinguish the value of the transferred transition prior from that of future-representation supervision during adaptation. With the visual encoder and policy architecture held fixed, pretrained transition priors improve performance even under action-only training; retaining future prediction further improves the transferred policy. On LIBERO-Long, these comparisons yield gains of 8.6 and 5.4 percentage points, respectively, supporting the complementary roles of transition-model transfer and predictive adaptation. The transfer advantage persists across the evaluated late-training checkpoints and most tasks, and remains evident in low-data regime with limited robot demonstrations.

Across the four LIBERO suites, LTM-Policy reaches 97.4% average success. We further evaluate the same direct policy under controlled visual shifts in LIBERO-Plus and on physical manipulation with a Franka Research 3. These results demonstrate that a pretrained action-conditioned latent transition model can be transferred into the backbone of a strong direct policy. Our contributions are threefold:

- We study **transition-model transfer** as a complement to conventional encoder transfer, and show that a pretrained action-conditioned JEPA predictor can be repurposed as the latent transition backbone of a direct visuomotor policy.
- We introduce a dual-route adaptation scheme that converts an action-conditioned transition model into a direct action-generating policy while preserving latent world modeling.
- Controlled experiments isolate the benefits of predictor transfer and future-state supervision, including their behavior under different training stages and limited robot data.

2 Related Work

2.1 Visuomotor Policies and Vision-Language-Action Models

Visuomotor policies directly map visual observations and robot states to actions. Representative methods include ACT, which predicts temporally coherent action chunks (Zhao et al., 2023), and Diffusion Policy, which models multimodal action distributions through iterative denoising (Chi et al., 2023). More recent generalist policies and vision-language-action (VLA) models scale robot learning with diverse data and large pretrained backbones, including RT-2, Octo, and OpenVLA (Zitkovich et al., 2023; Ghosh et al., 2024; Kim et al., 2024). π_0 and $\pi_{0.5}$ further combine large vision-language backbones with continuous action generation to support broad task and environment generalization (Black et al., 2025; Physical Intelligence et al., 2025). These methods primarily focus on action prediction rather than explicitly modeling future visual dynamics. Recent lightweight policies such as Patch Policy show that dense pretrained patch features can support effective control with compact architectures (Zhou et al., 2026). Our work follows this setting, but transfers a learned model of latent state transitions in addition to pretrained visual representations.

2.2 World-Action Models for Visuomotor Control

World-action models (WAMs) jointly model actions and how the environment evolves, allowing predictive structure learned from videos to inform robot control. A prominent line builds on generative video models: Video Policy combines video generation with action prediction (Liang et al., 2025), while Cosmos Policy and DreamZero repurpose pretrained video models for robot control (Kim et al., 2026; Ye et al., 2026). Other approaches condition actions on imagined visual futures, while Fast-WAM retains future modeling during training without explicit future generation at deployment (Gu et al., 2026; Yuan et al., 2026). These works demonstrate the value of predictive world models, but retain much of the machinery of generative video modeling.

Concurrent work explores WAMs directly in latent spaces. LaWAM and LiLa-WAM couple latent future prediction with action generation (Chen et al., 2026b; Yang et al., 2026), while StageWAM predicts semantically distant task stages in pretrained representations (Liu et al., 2026). JEPA-WAM similarly couples dense transition prediction in a pretrained V-JEPA space with continuous action generation through a shared Qwen-based predictor (Lin et al., 2026). These methods establish latent future modeling as an effective training signal for visuomotor policies, but learn the transition model anew during policy training. Our focus is complementary: we transfer a transition model already learned before downstream policy training.

2.3 JEPA for Robot Learning

Joint-Embedding Predictive Architectures (JEPAs) predict targets in representation space rather than reconstructing observations. I-JEPA introduces this paradigm for images (Assran et al., 2023), V-JEPA extends it to video (Bardes et al., 2024), and V-JEPA 2 scales latent video prediction to temporal understanding and planning (Assran et al., 2025). Related extensions apply JEPA-style prediction to vision-language and structured world modeling (Chen et al., 2026a; Nam et al., 2026; Maes et al., 2026).

For robot control, V-JEPA 2-AC learns an action-conditioned predictor over frozen V-JEPA 2 representations and uses it for model-predictive control (Assran et al., 2025). Closest to our setting, JEPA Policy jointly predicts demonstrated action chunks and their future representations in a shared transformer, studying how paired future supervision improves direct imitation learning (Xu et al., 2026). JEPA-WAM instead learns latent transition modeling during downstream robot-policy training in a frozen V-JEPA representation space, with a shared predictor whose action representations condition a flow-matching expert (Lin et al., 2026). In contrast, we initialize the policy backbone from the action-conditioned V-JEPA 2-AC predictor, which has already been trained to model latent state transitions under robot actions, and ask whether this pretrained transition model itself transfers to direct control. Our dual-route adaptation retains action-conditioned transition prediction from demonstrated actions while introducing learned action queries for direct policy learning; controlled initialization and future-supervision ablations then separate transition-model transfer from predictive supervision during downstream adaptation.

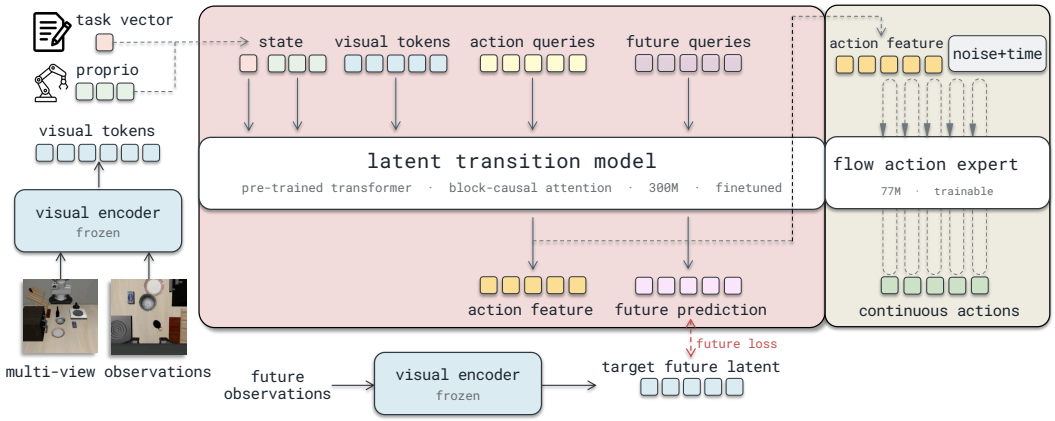


Figure 2: **Overview of LTM-Policy.** A frozen visual encoder maps synchronized multi-view observations to dense latent tokens. Together with robot state and task conditioning, these tokens enter an action-conditioned JEPA latent transition model that is finetuned for control. Block-causal attention organizes the sequence into state, action, and future blocks. In the policy route, learnable action queries produce action features that condition a lightweight flow-matching expert, while future queries predict the successor latent state. During training, a complementary world-modeling route replaces the learned action queries with demonstrated actions and trains the same transition model to predict their resulting future state. At inference, only the direct policy interface is used.

3 Method

Our goal is to repurpose an action-conditioned JEPA predictor as the transition backbone of a direct visuomotor policy. This requires resolving a structural mismatch: the pretrained transition model assumes that robot actions are given, whereas a policy must infer actions from observations. As shown in Fig. 2, LTM-Policy addresses this mismatch with a frozen visual encoder E, a latent transition model P_θ initialized from an action-conditioned JEPA predictor, and a lightweight flow-matching action expert F_ϕ. The encoder maps multi-view observations into dense latent states. We fine-tune the transition model to learn action-relevant features and predict how these states evolve under actions. The action expert uses the resulting features to generate continuous robot actions.

We reuse the pretrained transition model during robot adaptation, rather than relying on the visual representation alone. During training, direct policy learning and action-conditioned future prediction share the same transition model. Future-state prediction is supervised entirely in the JEPA latent space, without an image decoder or a pixel-reconstruction objective. At deployment, a single forward pass through the latent transition model P_θ provides the features used by action expert F_ϕ to generate actions; no video rollout or model-predictive search is required.

3.1 Setup and Notation

At control step t, the policy receives synchronized RGB observations $o_t^{1:K}$ from K cameras, proprioceptive state q_t, and task identifier c. It predicts an action chunk

$$A_t = a_{t:t+H-1} \in \mathbb{R}^{H \times d_a}, \tag{1}$$

where H is the prediction horizon and d_a is the action dimension. The controller executes the first $h \leq H$ actions before predicting the next chunk from an updated observation.

The frozen visual encoder E processes each view independently:

$$z_t^k = E(o_t^k) \in \mathbb{R}^{N \times D}, \qquad k = 1, \ldots, K, \tag{2}$$

where N is the number of patch tokens per view and D is their feature dimension. We denote the collection of encoded views by $z_t^{1:K}$. During training, each example also includes the future observation $o_{t+H}^{1:K}$, recorded after the H-step action chunk. Its encoding, $z_{t+H}^{1:K}$, serves as the target for future-state prediction.

3.2 JEPA LATENT TRANSITION MODEL

Predictor initialization. We initialize P_θ from the action-conditioned predictor of V-JEPA 2-AC (Assran et al., 2025). In V-JEPA 2-AC, the vision encoder is trained on a variety of large-scale video datasets and the predictor is further trained on DROID dataset (Khazatsky et al., 2024). The pretrained predictor predicts future visual representations from the current representation, robot state, and actions. For policy adaptation, we retain the pretrained transformer backbone and transition-model weights, adapt its input interface to multi-view observations, and introduce learnable action queries for direct action inference. The visual encoder remains frozen, while the transition model and newly introduced policy modules are jointly optimized on downstream robot demonstrations.

Multi-view state. We retain the dense patch tokens from each view without spatial pooling, preserving view-specific object and geometric information that may be lost in a single global representation. The state block combines these tokens with robot-state and task conditioning:

$$B_t^{\text{state}} = \left[e_s(q_t, c); e_v(z_t^1); \ldots; e_v(z_t^K)\right], \tag{3}$$

where e_s embeds the proprioceptive state and task identifier, e_v projects visual features into the predictor's token space, and semicolons denote concatenation along the token dimension. All views share the same encoder. We add learnable type embeddings to identify proprioceptive, primary-view, wrist-view, action, and future-query tokens within the shared predictor.

Block-causal attention. The predictor processes a sequence of three blocks:

$$B_t^{\text{state}} \;\rightarrow\; B_t^{\text{action}} \;\rightarrow\; B_t^{\text{future}}.$$

We use block-causal attention: tokens interact bidirectionally within each semantic block, while information flows causally across blocks. State tokens therefore attend only within the state block; action tokens attend to the complete state and action blocks; and future tokens attend to all preceding content as well as other future queries. This attention pattern allows the predictor to produce action features and future-state predictions in one forward pass, without autoregressive decoding. We use the same pattern in two training routes, which differ in the inputs to the action block.

Policy route. For direct policy learning, the action block contains learnable queries Q_a:

$$S_{\text{pol}} = [B_t^{\text{state}};\, Q_a;\, R_t], \tag{4}$$

where R_t denotes learnable future queries. The outputs at the action-query positions provide the features

$$H_t^a = P_\theta(S_{\text{pol}})\big|_{Q_a} \tag{5}$$

used by the action expert. The future-query outputs predict the future latent state $\hat{z}_{t+H,\text{pol}}^{1:K}$. Both outputs are inferred from the current observations, robot state, and task identifier, without access to the demonstrated action chunk. The demonstrated actions and the encoded future state are used only as supervision.

World route. For action-conditioned future prediction, we replace Q_a with embeddings of the demonstrated action chunk, U_t^{gt}:

$$S_{\text{world}} = [B_t^{\text{state}};\, U_t^{\text{gt}};\, R_t]. \tag{6}$$

The future-query outputs now predict $\hat{z}_{t+H,\text{world}}^{1:K}$ conditioned on the specified actions. This route retains the action-conditioned prediction task of the pretrained JEPA predictor during robot adaptation.

The two routes share the same transition model weights and follow the same causal dependency structure: current state → action representation → future state. The policy route must infer action-relevant structure from the current state because no action is provided as input. The world route instead observes the demonstrated action and preserves the original transition-modeling problem of predicting its latent consequence. Both routes are supervised using future latent states encoded from the demonstrations.

3.3 Training and Inference

Flow-matching action expert. The action expert F_ϕ generates a continuous action chunk conditioned on H_t^a and the proprioceptive state q_t. Let $\bar{A}_t$ denote the normalized demonstration chunk. We sample Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$ and flow time $\tau \sim \mathcal{U}(0, 1)$, then construct

$$X_\tau = (1 - \tau)\epsilon + \tau \bar{A}_t, \qquad v^\star = \bar{A}_t - \epsilon. \tag{7}$$

The expert predicts the velocity field

$$v_\phi = F_\phi(X_\tau, \tau; H_t^a, q_t) \tag{8}$$

and is trained with the flow-matching loss

$$\mathcal{L}_{\text{act}} = \mathbb{E}\left[\|v_\phi - v^\star\|_2^2\right]. \tag{9}$$

Gradients from this loss also update P_θ through the conditioning features H_t^a. This separation keeps iterative generative computation confined to the lightweight action decoder, while the substantially larger transition model is evaluated only once per control step.

Future-state supervision. Both routes predict the dense future latent state supplied by the frozen encoder. We supervise each predicted patch token against its corresponding target token using

$$\ell(\hat{z}, z) = \frac{1}{KN} \sum_{k=1}^{K} \sum_{n=1}^{N} \left\| \frac{\hat{z}_n^k}{\|\hat{z}_n^k\|_2} - \frac{z_n^k}{\|z_n^k\|_2} \right\|_1. \tag{10}$$

Since the visual encoder is frozen, the target representations are treated as fixed supervision and gradients update only the predictor and policy modules. The prediction losses for the two routes are

$$\begin{aligned} \mathcal{L}_{\text{pol}-\text{fut}} &= \ell\left(\hat{z}_{t+H,\text{pol}}^{1:K}, z_{t+H}^{1:K}\right), \\ \mathcal{L}_{\text{world}-\text{fut}} &= \ell\left(\hat{z}_{t+H,\text{world}}^{1:K}, z_{t+H}^{1:K}\right). \end{aligned} \tag{11}$$

The policy-future loss supervises future-state prediction without demonstrated actions as input, whereas the world-future loss supervises prediction conditioned on those actions. The former couples inferred action features to the observed state transition, whereas the latter directly maintains the action-conditioned dynamics learned by the pretrained predictor. Both losses update the shared transition model.

Joint optimization. We combine action learning and future-state prediction through

$$\mathcal{L} = \mathcal{L}_{\text{act}} + \lambda_{\text{pol}} \mathcal{L}_{\text{pol}-\text{fut}} + \lambda_{\text{world}} \mathcal{L}_{\text{world}-\text{fut}}, \tag{12}$$

where λ_{pol} and λ_{world} weight the two prediction losses. Unless otherwise stated, each minibatch is split evenly between the policy and world routes. The action loss updates both the flow expert and, through H_t^a, the shared transition model. Future losses update only the transition model, while the visual encoder remains frozen throughout training.

Deployment. At test time, LTM-Policy operates as a direct closed-loop visuomotor policy. Given the current multi-view observation and proprioceptive state, the frozen visual encoder first produces latent visual tokens. The policy route then evaluates the latent transition model P_θ once to obtain action features H_t^a from the learned action queries. Conditioned on these features, the lightweight flow expert generates a continuous action chunk from Gaussian noise using a fixed number of refinement steps. The robot executes the first h actions, receives a new observation, and replans. Only the policy route is used during deployment: demonstrated actions and future-state targets are not required, and the predicted future representations are not used for action selection. Thus, inference involves no pixel reconstruction, future-video rollout, or model-predictive search.

4 Experiments

Our experiments address four research questions: **(RQ1)** Can a pretrained JEPA transition model support direct visuomotor control in simulation and real-world manipulation? **(RQ2)** Does transferring the pretrained transition model itself improve policy learning beyond the frozen visual encoder and model architecture? **(RQ3)** What additional benefit does future-state supervision provide during policy adaptation? **(RQ4)** How does an additional camera view affect the full model?

Table 1: Success rate (%) comparison on LIBERO benchmark. Gray rows denote pixel-based WAMs.

Method	Spatial	Object	Goal	Long	Average
Diffusion Policy (Chi et al., 2023)	78.3	92.5	68.3	50.5	72.4
OpenVLA (Kim et al., 2024)	84.7	88.4	79.2	53.7	76.5
π_0 (Black et al., 2025)	96.8	98.8	95.8	85.2	94.1
$\pi_{0.5}$ (Physical Intelligence et al., 2025)	**98.6**	98.2	**98.4**	92.4	96.9
OpenVLA-OFT (Kim et al., 2025)	97.6	98.4	97.9	94.5	97.1
LiLa-WAM (Yang et al., 2026)	98.0	98.8	97.2	94.2	97.1
JEPA-WAM (Lin et al., 2026)	95.6	**99.4**	97.2	**94.6**	96.7
Fast-WAM (Yuan et al., 2026)	98.2	100.0	97.0	95.2	97.6
Cosmos Policy (Kim et al., 2026)	98.1	100.0	98.2	97.6	98.5
LTM-Policy	**98.6**	**99.4**	97.2	94.4	**97.4**

4.1 Experimental Setup

Benchmarks and evaluation. We evaluate LTM-Policy across simulation and real-world manipulation. Our primary benchmark is LIBERO (Liu et al., 2023), where we report results on the Spatial, Object, Goal, and Long suites, each containing ten manipulation tasks. We evaluate each task over 50 simulator episodes and report average task success. To assess visual robustness, we additionally evaluate the same trained policies on LIBERO-Plus (Fei et al., 2026) without further adaptation, covering controlled shifts in background, camera viewpoint, lighting, object layout, robot appearance/state, sensor noise, and language. We further evaluate on three real-world manipulation tasks using a Franka Research 3.

Implementation details. Unless otherwise stated, the policy receives synchronized scene and wrist RGB observations, robot proprioception, and a task identifier. The V-JEPA 2 visual encoder remains frozen throughout training, while the action-conditioned transition model and flow-matching action expert are adapted on robot demonstrations. Training jointly optimizes the policy and world routes described in Sec. 3.2, with each minibatch divided evenly between the two routes. All controlled comparisons use the same training demonstrations and observation modalities. We use the same action-chunk horizon, execution horizon, and evaluation protocol across matched comparisons. Optimization hyperparameters, training schedules, flow-sampling details, benchmark-specific preprocessing, and additional evaluation settings are provided in Appendix A.

4.2 Main Results (RQ1)

LIBERO. Table 1 compares LTM-Policy with representative models on the four LIBERO suites. LTM-Policy achieves an overall average of 97.4%. This places it among the strongest methods in the comparison, substantially outperforming earlier direct policies such as Diffusion Policy, OpenVLA, and π_0, while remaining competitive with more recent VLA and WAM approaches. The gains are particularly notable on Spatial and Object. More importantly, it maintains 94.4% success on LIBERO-Long, comparable to strong recent policies such as OpenVLA-OFT, LiLa-WAM, and JEPA-WAM. These results suggest that repurposing a pretrained latent transition model does not merely preserve simple manipulation capability, but can also support challenging multi-step control. At the same time, LTM-Policy achieves this performance without future generation or model-predictive search.

LIBERO-Plus. We evaluate the LIBERO-trained policy on perturbed LIBERO-Plus benchmark instances without further adaptation. As shown in Table 2, LTM-Policy obtains 67.4% under the benchmark's official aggregation. Its robustness profile varies substantially across perturbation types: performance remains strong under lighting and camera perturbations, while background shifts are considerably more challenging. LIBERO-Plus also includes a language-perturbation category. As our policy is conditioned on a task identifier rather than natural-language instructions, perturbing the instruction does not change its conditioning signal; we therefore report this category for benchmark completeness but do not interpret it as language robustness. Taken together, the applicable perturbations suggest that the current latent policy is relatively robust to some changes in geometry and illumination, but remains sensitive to large shifts in scene appearance. Per-suite results and the corresponding non-language aggregate are provided in Appendix B.

Table 2: Zero-shot LIBERO-Plus (%) by perturbation category. Baseline values follow the benchmark's original evaluation table (Fei et al., 2026); each category aggregates the four LIBERO suites.

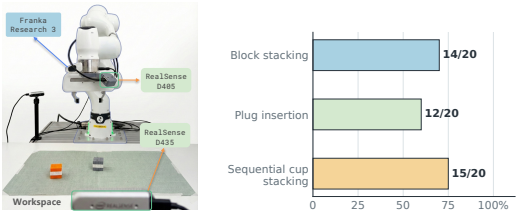
Method	Camera	Robot	Language	Light	Background	Noise	Layout	Overall
OpenVLA (Kim et al., 2024)	0.8	3.5	23.0	8.1	34.8	15.2	28.5	15.6
WorldVLA (Cen et al., 2025)	0.1	27.9	41.6	43.7	17.1	10.9	38.0	25.0
NORA (Hung et al., 2025)	2.2	37.0	65.1	45.7	58.6	12.8	62.1	39.0
UniVLA (Bu et al., 2025)	1.8	**46.2**	69.6	69.0	81.0	21.2	31.9	43.9
π_0 (Black et al., 2025)	13.8	6.0	58.8	85.0	81.4	**79.0**	68.9	53.6
π_0-FAST (Pertsch et al., 2025)	**65.1**	21.6	61.0	73.2	73.2	74.4	68.8	61.6
RIPT-VLA (Tan et al., 2025)	55.2	31.2	77.6	88.4	91.6	73.5	**74.2**	68.4
OpenVLA-OFT (Kim et al., 2025)	56.4	31.9	79.5	88.7	**93.3**	75.8	**74.2**	**69.6**
LTM-Policy	60.1	38.1	**93.8**†	**92.7**	54.8	67.6	71.0	67.4

4.2.1 Real-World Manipulation

Hardware and demonstrations. We use a 7-DoF Franka Research 3 arm with a front RealSense D435 camera and a wrist-mounted D405 (Fig. 3). We collect 40 successful demonstrations for each task through leader-follower teleoperation with two Franka arms, logging the actions, proprioception, and observations.

Evaluation protocol. We evaluate Block Stacking, Plug Insertion, and Sequential Cup Stacking with 20 physical rollouts per task. The policy receives the same front and wrist views during training and deployment: the front camera covers the workspace, while the wrist camera observes local contact geometry. At each replanning step, it generates an action chunk from the latest observations and robot state, without test-time video rollout or search.

Task success. LTM-Policy succeeds in 14/20 Block Stacking trials (70%), 12/20 Plug Insertion trials (60%), and 15/20 Sequential Cup Stacking trials (75%), as shown in Fig. 3. Together with the simulation results, these experiments show that the adapted JEPA predictor can support direct action generation and closed-loop execution without test-time video rollout or search. The physical trials establish deployment feasibility on the three tasks; the contribution of each training component is examined in the controlled ablations below. Representative rollout sequences are shown in Appendix C.



(a) Setup and cameras (b) Real-world success

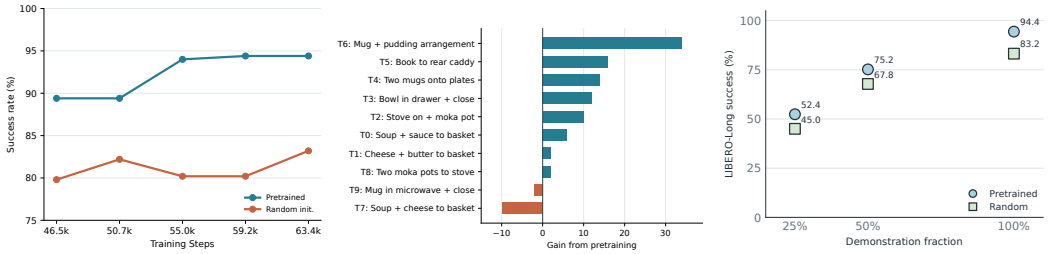
Figure 3: **Real-world manipulation.** (a) Franka Research 3 setup and camera placement. (b) Successful rollouts out of 20 trials per task.

4.3 Ablation and Analysis

Does the pretrained transition model itself matter? (RQ2) Table 3 first isolates the contribution of transition model initialization. With the same frozen visual encoder, architecture, training objective, and optimization budget, pretrained initialization improves LIBERO-Long from 83.2% to 94.4% under future-state supervision, a substantial 11.2-point gain. The improvement remains when future prediction is removed: under action-only training, pretraining raises Long from 80.4% to 89.0% and Goal from 94.2% to 96.4%. This shows that the benefit is not explained by the pretrained visual encoder or by downstream predictive supervision alone; the transferred transition-model weights themselves provide useful structure for policy learning.

Table 3: Ablations on LIBERO-Goal and LIBERO-Long (success rate %).

Transition init.	Training objective	Views	Goal	Long
Random	Action only	Dual	94.2	80.4
Random	Action + future	Dual	94.0	83.2
Pretrained	Action only	Dual	96.4	89.0
Pretrained	Action + future	Dual	**97.2**	**94.4**
Pretrained	Action + future	Single	87.2	71.2



(a) Training dynamics (b) Per-task differences (c) Data efficiency

Figure 4: **Effect of transition model pretraining on LIBERO-Long.** (a) Success rates of pretrained and randomly initialized models across checkpoints. (b) Per-task success differences at the final evaluated checkpoint. (c) Success rates under different demonstration budgets.

What does future-state supervision contribute? (RQ3) Future-state prediction provides an additional benefit on top of predictor transfer, especially on LIBERO-Long. For the pretrained model, adding future supervision improves Long from 89.0% to 94.4%, while the randomly initialized model improves from 80.4% to 83.2%. The corresponding gains on Goal are smaller: 96.4% to 97.2% with pretrained initialization, while the randomly initialized model remains essentially unchanged (94.2% vs. 94.0%). These results suggest that future-state prediction is a complementary training signal, with its effect becoming more pronounced on the longer-horizon suite.

How important is multi-view observability? (RQ4) Multi-view observations provide another substantial source of performance. With pretrained initialization and future supervision fixed, adding the wrist camera raises success from 71.2% to 94.4% on LIBERO-Long and from 87.2% to 97.2% on LIBERO-Goal. The magnitude of this effect highlights that accurate transition and action prediction depend not only on the learned prior, but also on sufficient visual observability. We therefore use dual-view observations in all main experiments.

Is the transfer benefit consistent across training and tasks? (RQ2). We compare pretrained and randomly initialized transition models on LIBERO-Long under the same training and evaluation settings (Fig. 4(a,b)). Pretraining improves success at all five checkpoints from 46.5k to 63.4k training steps, with gains of 7.2–14.2%. At the last checkpoint, it improves performance on 8 of 10 tasks, yielding an average gain of 8.4%. These results show that the initialization benefit persists across the different training stages and is shared by most tasks.

Does the transferred transition prior improve data efficiency? (RQ2). We evaluate both initializations on LIBERO-Long using identical trajectory subsets sampled within each task and training settings (Fig. 4(c)). Pretraining provides a 7.4 % gain at both reduced demonstration budgets, increasing success from 45.0% to 52.4% with 25% of the data and from 67.8% to 75.2% with 50%. The advantage persists with the full dataset, where success increases from 83.2% to 94.4%. These gains show that the transferred predictor supports more effective policy learning from limited demonstrations while remaining beneficial as the demonstration budget increases.

5 Discussion and Future Work

Our results suggest that pretrained latent transition models can serve as reusable backbones for direct visuomotor policy learning, rather than only as auxiliary world models for prediction or planning. The transfer benefit persists under action-only adaptation and in low-data regimes, while future-representation supervision provides an additional source of improvement during downstream policy learning. An important direction for future work is to test whether this principle extends beyond the V-JEPA family to other predictive representation models and latent world-model backbones, and to study how transition-model transfer scales with larger and more diverse robot datasets. It would also be valuable to explore richer forms of policy adaptation, broader embodiment and task distributions, and stronger robustness to appearance shifts. More broadly, our results point toward transition-model transfer as a complementary paradigm to conventional encoder transfer, enabling predictive video models to contribute not only representations but also reusable dynamics for robot control.

AI USE STATEMENT

In this work, we used generative AI tools for implementation in experiment prototyping, as well as polishing the paper language. We have not used generative AI tools for original paper writting. We have reviewed all AI-assisted work. AI-generated code was verified and tested for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide the implementation and evaluation details. Optimization hyperparameters, training schedules, flow-sampling settings, benchmark-specific preprocessing, and evaluation protocols are described in Sec. 4 and Appendix A.

REFERENCES

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael G. Rabbat, Yann LeCun, and Nicolas Ballas. Self-Supervised Learning from Images with a Joint-Embedding Predictive Architecture. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pp. 15619–15629. IEEE, 2023. doi: 10.1109/CVPR52729.2023.01499. URL https://doi.org/10.1109/CVPR52729.2023.01499.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba, Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-JEPA 2: Self-Supervised Video Models Enable Understanding, Prediction and Planning. *arXiv preprint arXiv:2506.09985*, 2025. URL https://arxiv.org/abs/2506.09985.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. Revisiting Feature Prediction for Learning Visual Representations from Video. *Transactions on Machine Learning Research*, 2024, 2024. URL https://openreview.net/forum?id=QaCCuDfBk2.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A Vision-Language-Action Flow Model for General Robot Control. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.010. URL https://www.roboticsproceedings.org/rss21/p010.html.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Learning to Act Anywhere with Task-centric Latent Actions. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.014. URL https://www.roboticsproceedings.org/rss21/p014.html.

Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, Deli Zhao, and Hao Chen. WorldVLA: Towards Autoregressive Action World Model. *arXiv preprint arXiv:2506.21539*, 2025. URL https://arxiv.org/abs/2506.21539.

Delong Chen, Mustafa Shukor, Théo Moutakanni, Willy Chung, Lei Yu, Tejaswi Kasarla, Allen Bolourchi, Yann LeCun, and Pascale Fung. VL-JEPA: Joint Embedding Predictive Architecture for Vision-language. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 89791–89807, 2026a. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/9144aded4e536bc5fb7bfc660a2d7a3d-Abstract-Conference.html.

Jialei Chen, Kai Wang, Kang Chen, Shuaihang Chen, Feng Gao, Wenhao Tang, Zhiyuan Li, Weilin Liu, Zhuyu Yao, Boxun Li, Yuanbo Xu, and Chao Yu. LaWAM: Latent World Action Models for Efficient Dynamics-Aware Robot Policies. *arXiv preprint arXiv:2606.15768*, 2026b. URL https://arxiv.org/abs/2606.15768.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. In *Robotics: Science and Systems XIX, Daegu, Republic of Korea, July 10-14, 2023*, 2023. doi: 10.15607/RSS.2023.XIX.026. URL https://doi.org/10.15607/RSS.2023.XIX.026.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. LIBERO-Plus: A Progressive Robustness Benchmark for Visual-Language-Action Models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2026, Denver, CO, USA, June 3-7, 2026*, pp. 38574–38583. Computer Vision Foundation, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Fei_LIBERO-Plus_A_Progressive_Robustness_Benchmark_for_Visual-Language-Action_Models_CVPR_2026_paper.html.

Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Quan Vuong, Ted Xiao, Pannag R. Sanketi, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An Open-Source Generalist Robot Policy. In *Robotics: Science and Systems XX, Delft, The Netherlands, July 15-19, 2024*, 2024. doi: 10.15607/RSS.2024.XX.090. URL https://doi.org/10.15607/RSS.2024.XX.090.

Songen Gu, Yunuo Cai, Tianyu Wang, Simo Wu, and Yanwei Fu. Say, Dream, and Act: Learning Video World Models for Instruction-Driven Robot Manipulation. *arXiv preprint arXiv:2602.10717*, 2026. URL https://arxiv.org/abs/2602.10717.

Chia-Yu Hung, Qi Sun, Pengfei Hong, Amir Zadeh, Chuan Li, U-Xuan Tan, Navonil Majumder, and Soujanya Poria. NORA: A Small Open-Sourced Generalist Vision Language Action Model for Embodied Tasks. *arXiv preprint arXiv:2504.19854*, 2025. URL https://arxiv.org/abs/2504.19854.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Yecheng Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Arhan Jain, Abraham Lee, Youngwoon Lee, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng Chi, Kyle Beltran Hatch, Shan Lin, Jingpei Lu, Jean Mercat, Abdul Rehman, Pannag R. Sanketi, Archit Sharma, Cody Simpson, Quan Vuong, Homer Rich Walke, Blake Wulfe, Ted Xiao, Jonathan Heewon Yang, Arefeh Yavary, Tony Z. Zhao, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, David Antonio Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Donovon Jackson, Charlotte Le, Yunshuang Li, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Nguyen, Abigail O'Neill, Rosario Scalise, Derick Seale, Victor Son, Stephen Tian, Emi Tran, Andrew E. Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, Osbert Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, Dinesh Jayaraman, Joseph J. Lim, Jitendra Malik, Roberto Martín-Martín, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Michael C. Yip, Yuke Zhu, Thomas Kollar, Sergey Levine, and Chelsea Finn. DROID: A Large-Scale In-The-Wild Robot Manipulation Dataset. In *Robotics: Science and Systems XX, Delft, The Netherlands, July 15-19, 2024*, 2024. doi: 10.15607/RSS.2024.XX.120. URL https://doi.org/10.15607/RSS.2024.XX.120.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Paul Foster, Pannag R. Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An Open-Source Vision-Language-Action Model. In *Conference on Robot Learning, 6-9 November 2024, Munich, Germany*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713. PMLR, 2024. URL https://proceedings.mlr.press/v270/kim25c.html.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-Tuning Vision-Language-Action Models: Optimizing Speed and Success. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.017. URL https://www.roboticsproceedings.org/rss21/p017.html.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, and Jinwei Gu. Cosmos Policy: Fine-Tuning Video Models for Visuomotor Control and Planning. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 71531–71552, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/748becc400a57c0e31cfe6a2e7951467-Abstract-Conference.html.

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*, 2025. URL https://arxiv.org/abs/2508.00795.

Yihan Lin, Jiawei He, Shifeng Bao, Chen Zhao, Yang Li, Xiaobo Wang, Yan Wang, Cheng Chi, and Jing Zhang. JEPA-WAM: Learning Vision-Language-Action Policies with Joint-Embedding World Modeling. *arXiv preprint arXiv:2608.09381*, 2026. URL https://arxiv.org/abs/2608.09381.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking Knowledge Transfer for Lifelong Robot Learning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, volume 36, pp. 44776–44791. Curran Associates, Inc., 2023. doi: 10.52202/075280-1939. URL http://papers.nips.cc/paper_files/paper/2023/hash/8c3c666820ea055a77726d66fc7d447f-Abstract-Datasets_and_Benchmarks.html.

Xiao Liu, Yuguang Yang, Xi Wang, Kai Jiang, Cheng Chi, Yong Xu, Wenchao Ding, Yilun Chen, and Yan Wang. StageWAM: Joint-Embedding Stage Prediction for World-Action Models in Robot Manipulation. *arXiv preprint arXiv:2608.10780*, 2026. URL https://arxiv.org/abs/2608.10780.

Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. LeWorldModel: Stable End-to-End Joint-Embedding Predictive Architecture from Pixels. *arXiv preprint arXiv:2603.19312*, 2026. URL https://arxiv.org/abs/2603.19312.

Heejeong Nam, Quentin Le Lidec, Lucas Maes, Yann LeCun, and Randall Balestriero. Causal-JEPA: Learning World Models through Object-Level Latent Masking. *arXiv preprint arXiv:2602.11389*, 2026. URL https://arxiv.org/abs/2602.11389.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. FAST: Efficient Action Tokenization for Vision-Language-Action Models. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.012. URL https://www.roboticsproceedings.org/rss21/p012.html.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a Vision-Language-Action Model with Open-World Generalization. *arXiv preprint arXiv:2504.16054*, 2025. URL https://arxiv.org/abs/2504.16054.

Shuhan Tan, Kairan Dou, Yue Zhao, and Philipp Krähenbühl. Interactive Post-Training for Vision-Language-Action Models. *arXiv preprint arXiv:2505.17016*, 2025. URL https://arxiv.org/abs/2505.17016.

Jie Xu, Kangjin Yu, Ziyi Jin, Junjie Gao, Liqing Chen, Yixian Li, Shuai Tian, and Zhongpu Xia. JEPA Policy: Diffusion-Free Imitation Learning via Paired Action and Future Representation Prediction. *arXiv preprint arXiv:2609.09630*, 2026. URL https://arxiv.org/abs/2609.09630.

Fan Yang, Yuting Su, Xiaobo Wang, Yuncheng You, Fugui Fan, Yuting Wu, Minghui Wu, Chenxu Zhao, JiaHong Ning, and Peiguang Jing. LiLa-WAM: Lightweight Latent Reasoning World-Action Model for Robotic Manipulation. *arXiv preprint arXiv:2608.03701*, 2026. URL https://arxiv.org/abs/2608.03701.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Malik, Kyungmin Lee, William Liang, Nadun Ranawaka, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, Johan Bjorck, Jing Wang, Gwanghyun Kim, Dantong Niu, Ruijie Zheng, Yuqi Xie, Jimmy Wu, Qi Wang, Ryan Julian, Danfei Xu, Yilun Du, Yevgen Chebotar, Scott Reed, Jan Kautz, Yuke Zhu, Linxi "Jim" Fan, and Joel Jang. World Action Models are Zero-shot Policies. *arXiv preprint arXiv:2602.15922*, 2026. URL https://arxiv.org/abs/2602.15922.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-WAM: Do World Action Models Need Test-time Future Imagination? *arXiv preprint arXiv:2603.16666*, 2026. URL https://arxiv.org/abs/2603.16666.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware. In *Robotics: Science and Systems XIX, Daegu, Republic of Korea, July 10-14, 2023*, 2023. doi: 10.15607/RSS.2023.XIX.016. URL https://doi.org/10.15607/RSS.2023.XIX.016.

Gaoyue Zhou, Zichen Jeff Cui, Ada Langford, Bowen Tan, Yann LeCun, and Lerrel Pinto. Patch Policy: Efficient Embodied Control via Dense Visual Representations. *arXiv preprint arXiv:2607.18236*, 2026. URL https://arxiv.org/abs/2607.18236.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong T. Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. In *Conference on Robot Learning, CoRL 2023, 6-9 November 2023, Atlanta, GA, USA*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183. PMLR, 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

A Implementation Details

Model and observations. Our LIBERO experiments use synchronized scene and wrist images at a resolution of 256×256. The frozen V-JEPA 2-AC ViT-G encoder processes the two views independently, producing 256 tokens of dimension 1,408 per view; we retain the complete token sequences rather than pooling them into global features. The policy is additionally conditioned on robot proprioception and a learned embedding of the known base-task index. Each demonstration is divided into eight-step chunks of seven-dimensional actions, and the observation following the chunk supplies the target for latent future prediction. We initialize the transition model from the released V-JEPA 2-AC action-conditioned predictor and keep only the visual encoder fixed during downstream training.

Training. We train the LIBERO policies on eight RTX A5000 GPUs with a global batch size of 48. Optimization uses AdamW with linear warmup followed by cosine decay. The learning rates are 4×10^{-5} for the transferred transition model, 2×10^{-4} for the flow-matching action expert, and 1.5×10^{-4} for the newly initialized query, positional, token-type, and task embeddings. Each minibatch is divided equally between the policy and world-modeling routes, so both objectives update the shared transition model throughout adaptation.

Policy execution and efficiency. At deployment, the policy executes the complete eight-step action chunk before observing the environment again and replanning. The flow expert uses four integration steps on LIBERO-Spatial, LIBERO-Object, and LIBERO-Goal, and eight on LIBERO-Long. Each replanning cycle requires one evaluation of the transition model; only the smaller action expert is evaluated iteratively. The predicted future representations provide training supervision but do not participate in action selection at test time. Table 4 reports parameter counts and measured computational cost for this dual-view policy. Memory and throughput are measured during cache-based training on eight RTX A5000 GPUs, whereas end-to-end latency includes online RGB encoding on a single RTX A5000.

Table 4: Inference uses one RTX A5000, 256×256 RGB from two views, action chunk size 8, and 8 flow integration steps. Training uses 8×RTX A5000 with global batch 48.

Measurement	Value
Frozen video encoder	1,012.2M params
Pretrained predictor	305.2M params
Flow action expert	77.0M params
Other policy adapters/heads	4.8M params
Trainable policy total	387.0M params
Checkpoint size	4.64 GB
Training peak memory	17.9 GiB/GPU
Training throughput	29.4 samples/s
Cached-latent policy latency	125.7 ms/chunk
Dual-RGB end-to-end latency	298.8 ms/chunk
Inference peak allocated memory	5.38 GiB

B LIBERO-Plus Breakdown

Table 5 reports LIBERO-Plus results for each LIBERO suite and perturbation category. The overall score aggregates all seven perturbation types, while the non-language score excludes instruction perturbations.

Task conditioning. LTM-Policy is conditioned on a base-task identifier rather than natural-language instructions. Accordingly, perturbing the instruction text does not change the conditioning signal provided to the policy. We therefore report the language category for completeness, but do not interpret it as a measure of language robustness.

Table 5: Per-suite LIBERO-Plus breakdown for LTM-Policy. The non-language score excludes instruction perturbations, while the overall score includes all seven categories.

Suite	Camera	Robot	Language	Light	Background	Noise	Layout	Non-language	Overall
Goal	62.7	33.5	96.6	97.1	53.4	76.3	60.9	62.4	67.9
Spatial	81.4	42.0	98.2	97.9	93.8	84.3	90.6	80.8	83.6
Object	69.9	30.4	99.7	99.0	60.1	67.1	73.0	65.5	70.3
Long	26.3	46.3	80.7	76.6	11.8	42.8	59.6	42.8	48.6
Overall	**60.1**	**38.1**	**93.8**	**92.7**	**54.8**	**67.6**	**71.0**	**62.6**	**67.4**

C Real-World Rollouts

Figure 5 shows representative rollouts for the three physical manipulation tasks. Each row presents a rollout from left to right.

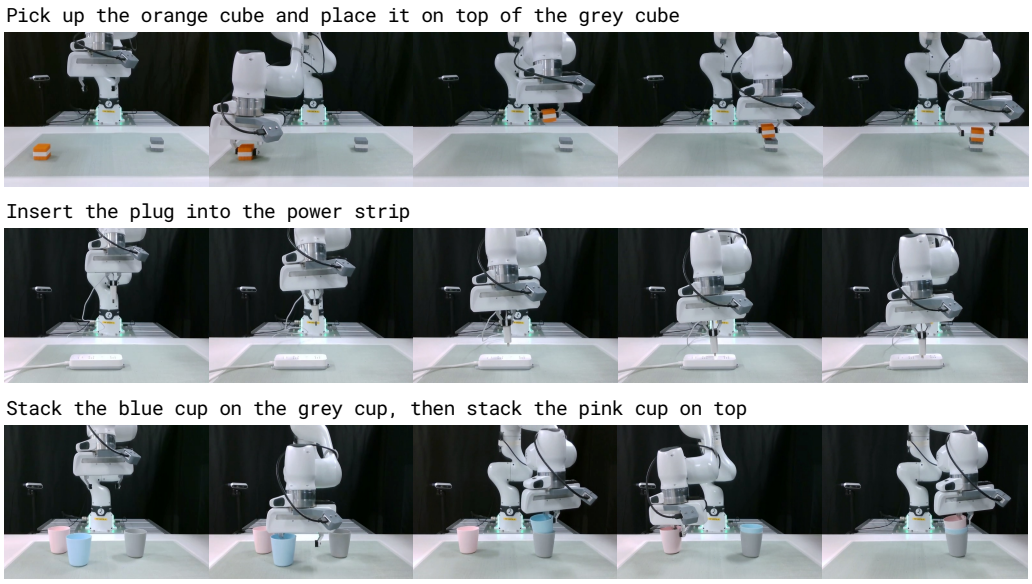


Figure 5: Representative real-world rollouts for block stacking (top), plug insertion (middle), and sequential cup stacking (bottom). Quantitative results over 20 trials per task are reported in Fig. 3.