CLOSING THE ACT-PREDICT-EVALUATE LOOP WITH A SHARED VIDEO-PRETRAINED REPRESENTATION FOR ROBOTIC MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Video-pretrained representations provide rich visual and temporal structure for embodied learning, but acting, prediction, and evaluation are still typically developed in separate representation spaces. This separation limits model-based policy improvement because predicted future states cannot be directly consumed by the policy and value model. We introduce Pivot, a framework that instead uses a frozen video-pretrained representation as the shared state space of the entire *Act-Predict-Evaluate* loop. The policy reads this state, action-conditioned dynamics predicts its successor, and a value model evaluates the predicted state under the instruction, enabling recursive latent imagination without observation reconstruction or representation translation. To make this generic video representation effective for manipulation, we introduce task-aware supervision at two levels. First, we train task-relevant latent dynamics using a semantic affordance prior derived from image-instruction attributions, emphasizing the interaction-critical changes. Second, we learn a task-regularized value function with temporal-difference learning and dense image-instruction supervision, providing progress estimates for imagined states. These estimates allow latent rollouts to directly guide policy improvement. On CALVIN, Pivot improves the imitation policy's average successful sequence length from 3.946 to 4.294. On a five-stage real-robot task, it increases the average number of completed stages from 1.6 to 4.3, with controlled ablations showing that both forms of task-aware supervision contribute to the improvement.

1 INTRODUCTION

Building embodied agents that follow language instructions in the physical world is a central goal of robot learning (Brohan et al., 2022; Zitkovich et al., 2023; Kim et al., 2024; Black et al., 2024). Progress remains constrained by the limited scale and diversity of robot interaction data, which is costly to collect and covers only a small fraction of the scenes, objects, and embodiments encountered in practice (O'Neill et al., 2024). Large-scale video provides a complementary source of supervision. Without requiring robot action labels, video-pretrained representations capture visual and temporal structure that has proven useful for manipulation policies (Hu et al., 2025; Yang et al., 2026a), action-conditioned world models that predict how a scene evolves (Assran et al., 2025; Zhu et al., 2025), and estimators of task progress (Ma et al., 2023b;a).

Most existing work, however, uses video-pretrained representations within individual modules rather than using them to connect the full decision-making loop. This becomes important in model-based policy improvement, where a policy acts from the current state, a dynamics model predicts the consequence of an action, and a value model evaluates whether the predicted state advances the task. When they operate in different representation spaces, an imagined successor must be reconstructed as an observation or mapped through a learned adapter before it can be acted on or evaluated (Quevedo et al., 2025; Sharma et al., 2026; Yang et al., 2026b). The issue is not only the additional computation. The dynamics model is optimized in a space that the policy and value model do not directly use, creating a mismatch between what is predicted and what downstream decisions depend on.

We study *whether a video-pretrained representation can instead serve as the state space shared by acting, prediction, and evaluation*. Our framework, **Pivot**, builds this state space from a frozen

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.

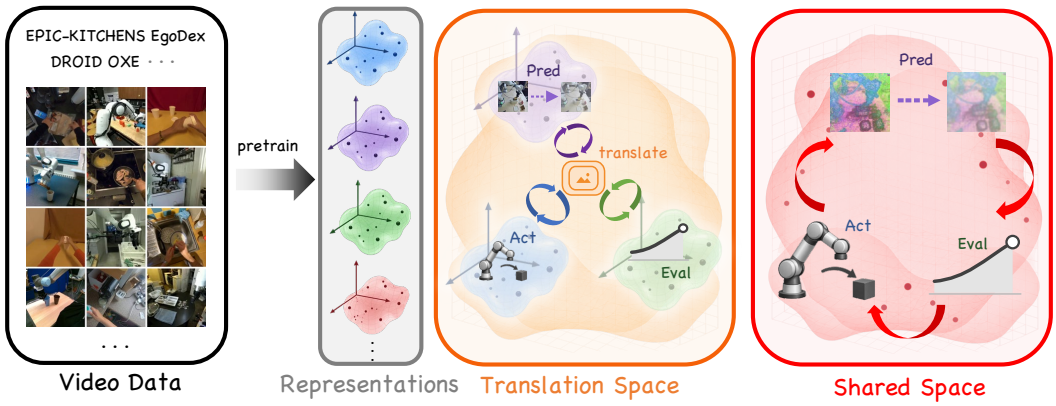


Figure 1: **One representation, three roles?** Systems that act, predict, and evaluate typically give each operation its own representation, so a predicted successor must be translated, by rendering to pixels or through a learned adapter, before another module can read it (*left*). We ask whether a frozen video-pretrained representation can instead serve as a shared state space for all three operations, allowing the policy to act on it, the dynamics model to predict future states within it, and the value model to evaluate those states directly, without reconstruction or representation translation (*right*).

V-JEPA 2 encoder (Assran et al., 2025). As shown in Figure 1, the policy acts on this state, the dynamics model predicts its successor, and the value model evaluates the predicted state under the instruction. Since predicted states directly usable by all three modules, Pivot supports recursive latent imagination whose value estimates provide supervision for policy improvement.

To make this generic video representation align with the fine-grained requirements of manipulation, we introduce task-aware supervision at two levels. For dynamics learning, a small change around a target object can determine success while contributing little to a loss averaged over the scene. We use image-instruction attributions from a frozen LIV model (Ma et al., 2023a) as a *semantic affordance prior*, giving greater weight to instruction-relevant regions in the dynamics loss. For evaluation, success and failure labels indicate the final outcome but offer limited supervision for judging progress at intermediate states. We therefore train a value model in the shared state space using temporal-difference (TD) learning, together with dense semantic targets derived from image-instruction similarity. During policy improvement, this value function scores latent rollouts, turning predicted value gains into supervision for policy refinement.

We evaluate Pivot on the CALVIN manipulation benchmark and a five-stage long-horizon real-world task. On CALVIN, Pivot improves the imitation policy's average successful sequence length from 3.946 to 4.294. Significantly, on the real-robot task, it increases completed stages from 1.6 to 4.3, demonstrating substantial gains in long-horizon execution. Our contributions are as follows:

- We introduce Pivot, a model-based policy improvement framework in which policy, dynamics, and value operate on a shared video-pretrained predictive state, enabling closed-loop latent imagination and policy refinement without observation reconstruction or representation translation.
- We develop *task-relevant latent dynamics* with a semantic affordance prior that emphasizes instruction-relevant regions for modeling interaction-critical state changes.
- We learn a *task-regularized value function* by combining temporal-difference learning with dense image-instruction supervision, providing reliable progress estimates for imagined states.

2 Related Work

Visual representations for embodied learning. Pretrained visual encoders transfer representations learned from large-scale image and video datasets to robotic control (Nair et al., 2022; Majumdar et al., 2023; Karamcheti et al., 2023). More recent video-based approaches further exploit temporal structure through frame prediction (Wu et al., 2024; Cheang et al., 2024), feature forecasting (Zhang et al., 2025b; Sun et al., 2026; Zhang et al., 2025a), and latent-action pretraining (Ye et al., 2025; Chen et al., 2025). VPP extracts predictive features from a frozen video diffusion model in a

single forward pass and conditions a diffusion policy on these features (Hu et al., 2025). CoWVLA separates structure and motion in its video representation and combines motion-latent prediction with keyframe and action modeling (Yang et al., 2026a) These methods use predictive objectives to learn representations that support downstream action generation. Task semantics provide a complementary source of supervision. LIV aligns image-language similarity with task progress (Ma et al., 2023a), whereas Mask World Model learns policy representations through semantic mask prediction (Lou et al., 2026). In our work, we use image-language similarity by LIV to derive a semantic affordance prior for weighting prediction errors and to provide auxiliary value targets to optimize value model.

World models for robot policy improvement. Beyond shaping policy representations, world models can provide simulated experience for policy evaluation and improvement. WorldGym evaluates policies using video rollouts and vision-language rewards (Quevedo et al., 2025), while World-Gymnast and World-VLA-Loop optimize policies using imagined trajectories (Sharma et al., 2026; Liu et al., 2026). WMPO derives trajectory-level rewards from a success classifier and applies GRPO for policy optimization (Zhu et al., 2025), whereas RISE combines temporal-progress and temporal-difference supervision to learn value functions for advantage-conditioned flow matching (Yang et al., 2026b). These approaches connect the world model to the policy through generated visual observations. An alternative is to reason directly in a learned visual state space. DINO-WM and V-JEPA 2-AC operate on frozen visual representations and use predicted latent states for image-goal planning (Zhou et al., 2024; Assran et al., 2025). Our method follows this latent-space paradigm, but uses pretrained visual features for value-guided policy improvement through model-based latent interaction (Hafner et al., 2020; Hansen et al., 2022). Specifically, the policy, dynamics, and value model share the same state space, so predicted outcomes can be evaluated and reused for subsequent actions and predictions without image decoding or re-encoding.

3 METHOD

Our method performs model-based policy improvement for language-conditioned embodied manipulation in a shared predictive state space defined by a frozen video-pretrained encoder, as shown in Figure 2. The policy, dynamics model, and value model operate directly on this common representation, allowing predicted states to be recursively acted on and evaluated without reconstructing observations or translating between representation spaces, as required by prior approaches such as WMPO (Zhu et al., 2025). To better align this representation with tasks, we introduce a semantic affordance prior for task-relevant dynamics learning and dense image-instruction supervision for value learning. The resulting models are then used for closed-loop latent imagination and policy refinement.

3.1 A SHARED PREDICTIVE STATE SPACE

At control step t, the robot receives a multi-view RGB observation o_t together with a language instruction g. The policy produces an action chunk $\mathbf{a}_t = (a_t, \dots, a_{t+H-1})$ of length H. We use the frozen V-JEPA 2 encoder E_ω (Assran et al., 2025), pretrained through masked prediction in video representation space, to define the visual state

$$z_t = E_\omega(o_t) \in \mathcal{Z}, \quad \mathcal{Z} = \mathbb{R}^{N \times d}, \tag{1}$$

where N denotes the number of visual tokens across the input views and d denotes the token dimension. We define all three components of model-based policy improvement directly on $\mathcal{Z}$. The policy $\pi_\theta(\mathbf{a}_t \mid z_t, g)$ generates an action chunk from the current state, the dynamics model $F\phi$ predicts how the state changes under an action sequence, and the value model $V_\psi(z_t, g)$ estimates the task value of a state under instruction g. The key property is that the output space of the dynamics model is also the input space of the policy and value model. A predicted successor therefore remains a valid decision state rather than an intermediate representation that must first be reconstructed or adapted. This closure enables recursive imagination. Once the first observation has been encoded, a predicted state can be acted on, evaluated, and advanced again entirely within $\mathcal{Z}$. As a result, prediction is optimized in the same representation that downstream decision making directly consumes.

Policy initialization. We first learn a behavioral policy from language-annotated demonstrations $\mathcal{D}_{\text{demo}}$. A Perceiver module (Jaegle et al., 2021) compresses the visual tokens, which are combined

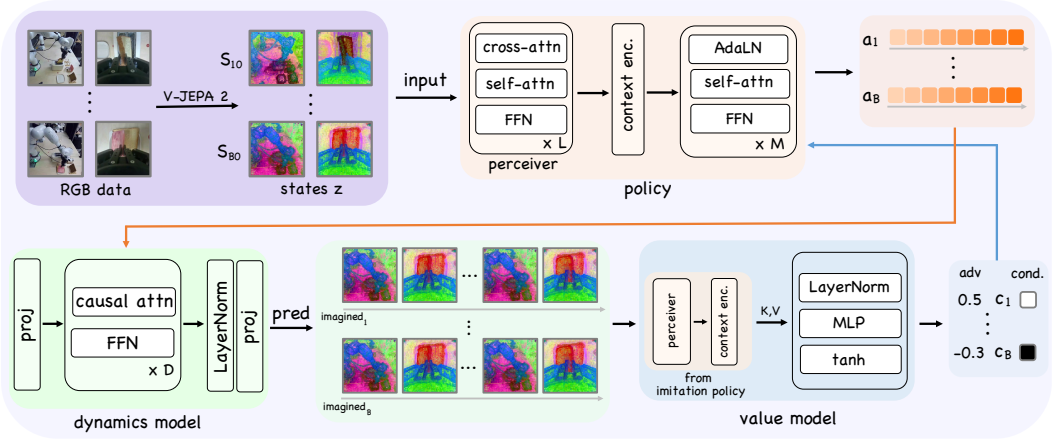


Figure 2: **Policy improvement in a shared predictive state space.** A frozen V-JEPA 2 encoder defines the latent state consumed by the policy, dynamics model, and value model. During imagination, the policy proposes an action chunk, the dynamics model predicts its latent consequences, and the value model evaluates the resulting progress toward the instruction.

with an embedding of the language instruction to form the policy context. A DiT-based action generator (Peebles & Xie, 2023) cross-attends to this context and is trained using conditional flow matching (Lipman et al., 2023). For a demonstration action chunk $\mathbf{a}$, we sample $\epsilon \sim \mathcal{N}(0, I)$ and $\tau \sim \mathcal{B}eta(1, 1.5)$, and construct the interpolation $\mathbf{a}^\tau = (1-\tau)\epsilon + \tau\mathbf{a}$. The policy is trained with

$$\mathcal{L}_{\text{act}}(\theta) = \mathbb{E}_{(z_t, g, \mathbf{a})\sim\mathcal{D}_{\text{demo}}}\Big[\big\|v_\theta(\mathbf{a}^\tau, \tau \mid z_t, g) - (\mathbf{a} - \epsilon)\big\|^2\Big], \tag{2}$$

where v_θ denotes the learned velocity field. Integrating this field from Gaussian noise generates an action chunk. This imitation policy the behavioral initialization for subsequent improvement.

3.2 Task-Relevant Latent Dynamics

The dynamics model predicts the consequences of candidate actions directly in the shared state space. Given the current rollout history, it jointly processes visual states, actions, and instruction tokens with causal attention over control steps. Starting from $\hat{z}_0 = z_0$, a rollout follows

$$\hat{z}_{t+1} = F_\phi(\hat{z}_{0:t}, a_{0:t}, g), \quad t = 0, \ldots, T-1. \tag{3}$$

Each predicted state is fed back into the dynamics model when predicting subsequent states. During dynamics learning, actions are taken from trajectories in $\mathcal{D}_{\text{demo}}$, while the prediction targets are the corresponding frozen encoder representations $z_t = E_\omega(o_t)$.

A standard latent prediction objective weights every visual token equally. This is poorly matched to manipulation, where the state change that determines task progress may occupy only a small fraction of the scene. Moving a target object or changing the configuration of an interaction region may be critical for the instruction while contributing little to a loss dominated by static background tokens. We therefore bias dynamics learning toward visual regions that are semantically relevant to the current task. We obtain this relevance signal from a frozen LIV model (Ma et al., 2023a). Given an observation o_t and instruction g, LIV produces an image-instruction similarity $s_{\text{LIV}}(o_t, g)$. We apply Grad CAM++ (Chattopadhay et al., 2018) to this similarity score, normalize the resulting spatial attribution to $[0, 1]$, and pool it onto the visual token grid used by the latent state. The resulting map $m(o, g) \in [0, 1]^N$ acts as a semantic affordance prior. It does not define what the future state should be. Instead, it indicates where prediction accuracy matters most for completing the instruction.

For a training sequence of length T, let $m_t = m(o_t, g)$. We optimize the dynamics model with

$$\mathcal{L}_{\text{dyn}}(\phi) = \mathbb{E}\Big[\frac{1}{TN}\sum_{t=1}^{T}\sum_{i=1}^{N}(1 + \lambda_{\text{sp}} m_{t,i})\|\hat{z}_{t,i} - z_{t,i}\|_1\Big], \tag{4}$$

where $\lambda \mathrm{sp} \geq 0$ controls the strength of semantic emphasis. The unit weight preserves supervision over the entire visual state, while the additional term allocates more learning capacity to regions that the instruction makes relevant. This design preserves the broad scene structure captured by video pretraining while improving the fidelity of interaction critical state changes. The semantic affordance prior is used only during dynamics training. Latent rollout itself depends solely on the learned dynamics model and therefore requires no attribution computation at imagination time.

3.3 Task-Regularized Value Learning

Latent imagination becomes useful for policy improvement only if predicted states can be evaluated according to task progress. We therefore learn a value function $V_\psi(z_t, g)$ that combines sparse task outcomes with dense semantic supervision. Since the value model consumes the same state representation produced by both the encoder and the dynamics model, the same evaluator can score real encoded states and imagined latent states without changing its input interface.

Outcome supervision. We first ground the value function in observed task outcomes. Let $\mathcal{D}_{\mathrm{all}}$ contain both successful and failed trajectories with outcome labels. States within a terminal window receive reward $+1$ for a successful trajectory and -0.6 for a failed trajectory. Other states receive zero immediate reward and obtain supervision through temporal difference bootstrapping. With terminal indicator $d_t \in \{0, 1\}$, the TD target and loss are

$$y_t = r_t + \gamma(1 - d_t)V_{\bar{\psi}}(z_{t+1}, g), \quad \mathcal{L}_{\mathrm{TD}}(\psi) = \mathbb{E}_{\mathcal{D}_{\mathrm{all}}}\big[(V_\psi(z_t, g) - \mathrm{sg}(y_t))^2\big], \tag{5}$$

where γ denotes the discount factor and sg stops gradients through the target. The target parameters are updated by an exponential moving average (EMA). This objective propagates observed success and failure information from the end of a trajectory to earlier states and gives the value function an explicit grounding in task outcome.

Semantic supervision. Outcome supervision alone provides only indirect information about how progress unfolds before termination. We therefore complement it with a dense semantic signal derived from the alignment between the observation and the language instruction. Specifically, we use the image-instruction similarity produced by the frozen LIV model as an auxiliary regression target

$$\mathcal{L}_{\mathrm{sem}}(\psi) = \mathbb{E}_{\mathcal{D}_{\mathrm{demo}}}\big[(V_\psi(E_\omega(o_t), g) - \kappa(s_{\mathrm{LIV}}(o_t, g)))^2\big], \tag{6}$$

where the affine calibration κ places the semantic targets on a scale compatible with the TD targets. Therefore, the complete objective of value model is

$$\mathcal{L}_{\mathrm{value}}(\psi) = \mathcal{L}_{\mathrm{TD}}(\psi) + \lambda_{\mathrm{sem}}\mathcal{L}_{\mathrm{sem}}(\psi). \tag{7}$$

The two objectives play complementary roles. Temporal difference learning grounds value in eventual task success or failure, while semantic supervision measures its alignment with the instruction. Their combination yields a task dependent evaluator that remains compatible with predicted latent states and can therefore provide supervision inside imagined rollouts.

3.4 Closed-Loop Policy Improvement in Latent Space

We finally use the learned dynamics and value models to improve the policy without collecting additional environment interaction. We freeze E_ω, F_ϕ, and V_ψ, and refine the policy using the advantage-conditioning formulation. A binary condition c is introduced through an additional conditioning pathway in the policy. Here, both the states on which the policy acts and the consequences used to label its actions can be generated recursively inside the shared predictive state space.

Closed-loop imagination. We initialize an imagined trajectory from an observation sampled from the demonstration data, $\hat{z}_0 = E_\omega(o_0)$. The policy and dynamics model then alternate for J action chunks. At chunk boundary $t = jH$, the policy samples

$$\mathbf{a}_j \sim \pi_\theta(\cdot \mid \hat{z}_{jH}, g, c = 1), \qquad j = 0, \ldots, J - 1. \tag{8}$$

The dynamics model applies Eq. 3 for the H actions in this chunk while conditioning on the accumulated rollout history. The resulting state $\hat{z}_{(j+1)H}$ is fed directly back to the policy to produce the next action chunk. Consequently, after the initial observation, the policy is conditioned entirely on model-predicted states, forming a closed-loop interaction with the world model. These imagined state-action pairs expose the policy to states beyond the demonstration distribution.

Algorithm 1 Closed-loop policy improvement in shared latent space

Require: Frozen E_ω, F_ϕ, V_ψ; initialized π_θ; chunk length H; rollout length J; mixture ratio β

for each policy update **do**
  Sample demonstration chunks, encode their observations, and assign $c = 1$ to form $\mathcal{B}_{\text{demo}}$.
  Initialize $\mathcal{B}_{\text{img}} \leftarrow \varnothing$.
  for each sampled $(o_0, g) \sim \mathcal{D}_{\text{demo}}$ **do**
    Set $\hat{z}_0 \leftarrow E_\omega(o_0)$ and initialize rollout history.
    for $j = 0, \dots, J-1$ **do**
      Set $t \leftarrow jH$.
      Sample $\mathbf{a}_t \sim \pi_\theta(\cdot \mid \hat{z}_t, g, c = 1)$.
      for $k = t, \dots, t + H - 1$ **do**
        $\hat{z}_{k+1} \leftarrow F_\phi(\hat{z}_{0:k}, a_{0:k}, g)$.
      end for
      Compute A_j and c_j using Eq. 9 and Eq. 10.
      Add $(\hat{z}_t, g, \mathbf{a}_t, c_j)$ to $\mathcal{B}_{\text{img}}$.
    end for
  end for
  Sample a training batch from $\mathcal{B}_{\text{mix}} = (1-\beta)\mathcal{B}_{\text{demo}} + \beta\mathcal{B}_{\text{img}}$ and update θ with Eq. 12.
end for
return $\pi_\theta(\cdot \mid E_\omega(o_t), g, c = 1)$

Policy improvement. For each imagined action chunk, we measure its predicted task progress using the change in value between the beginning and end of the chunk:

$$A_j = V_\psi(\hat{z}_{(j+1)H}, g) - V_\psi(\hat{z}_{jH}, g). \tag{9}$$

The use of a value difference makes the supervision depend on whether the proposed action improves the predicted state relative to where the rollout started. It also allows each action chunk to be evaluated locally without requiring the imagined trajectory to reach task completion. We convert this progress estimate into a binary improvement condition

$$c_j = \mathbb{I}\{A_j > \delta\}, \qquad \delta = 0. \tag{10}$$

A positive label indicates predicted improvement under the learned evaluator. The resulting imagined samples are collected as $\mathcal{B}_{\text{img}} = \{(\hat{z}_{jH}, g, \mathbf{a}_{jH}, c_j)\}$. We retain demonstrations throughout refinement by assigning their action chunks $c = 1$ and collecting them in $\mathcal{B}_{\text{demo}}$, which serves as a reference during improvement. We train on the mixture

$$\mathcal{B}_{\text{mix}} = (1-\beta)\mathcal{B}_{\text{demo}} + \beta\mathcal{B}_{\text{img}}. \tag{11}$$

where $\beta \in [0, 1]$ controls the contribution of imagined experience. Using the same flow matching parameterization as policy initialization, the refinement objective becomes:

$$\mathcal{L}_{\text{imp}}(\theta) = \mathbb{E}_{(z_t, g, \mathbf{a}, c) \sim \mathcal{B}_{\text{mix}}} \Big[\big\| v_\theta(\mathbf{a}^\tau, \tau \mid z_t, g, c) - (\mathbf{a} - \epsilon) \big\|^2 \Big]. \tag{12}$$

The policy is trained on both positive and negative imagined chunks under their corresponding conditions, and is queried with $c = 1$ during subsequent imagination and robot execution. In this way, value guided latent imagination is distilled into the action policy for execution. The complete procedure is summarized in Algorithm 1.

4 Experiments

4.1 Experimental Setup

CALVIN benchmark. We evaluate on CALVIN (Mees et al., 2022) using the standard ABC→D protocol. All models are trained only on demonstrations from environments A, B, and C and evaluated in the visually unseen environment D. To train the value model, we additionally collect a small set of failed trajectories exclusively in environments A, B, and C, ensuring that no data from the test environment D is used during training. In our experiments, the policy takes observations from both a

Table 1: **Results on CALVIN ABC→D.** Cumulative success rates measure completion of the first k instructions. The lower block compares each policy-improvement (PI) method with its corresponding imitation initialization. † indicates results obtained by re-evaluating the official checkpoint.

Method	Consecutive tasks completed (%) ↑					Avg. Len ↑
	1	2	3	4	5	
GR-1 (Wu et al., 2024)	85.4	71.2	59.6	49.7	40.1	3.060
GR-MG (Li et al., 2025)	96.8	89.3	81.5	72.7	64.4	4.047
VPP† (Hu et al., 2025)	96.1	90.7	84.5	**80.4**	**74.4**	4.261
CoWVLA (Yang et al., 2026a)	96.8	91.2	84.4	77.9	70.8	4.211
WMPO (w/o PI) (Zhu et al., 2025)	93.9	85.9	78.0	71.9	65.3	3.950
WMPO	95.0	87.7	81.8	74.6	66.5	4.056
Pivot (w/o PI)	97.7	88.3	78.4	70.6	59.6	3.946
Pivot	**98.9**	**95.2**	**87.0**	78.9	69.4	**4.294**

Table 2: **Five-stage sandwich preparation on the real robot.** Each policy is evaluated over 10 trials. Stages follow the order specified in Section 4.1, and success is cumulative, using the same definition as CALVIN. The refined policy substantially improves long-horizon robustness.

Policy	Consecutive stages completed (%) ↑					Avg. Len ↑
	1	2	3	4	5	
Pivot w/o policy improvement	80	40	20	10	10	1.6
Pivot w/ policy improvement	**90**	**90**	**90**	**80**	**80**	**4.3**

global camera and a wrist camera. Evaluation consists of $1,000$ language-conditioned chains, each containing five consecutively sampled subtasks. A policy receives at most 360 control steps for each subtask. Following the benchmark protocol, we report the cumulative success rate after completing the first k subtasks, $k \in \{1, \ldots, 5\}$, together with the average successful sequence length (Avg. Len).

Real-robot benchmark. We additionally evaluate long-horizon execution on a physical robot using a five-stage sandwich assembly task observed from a global camera and a wrist camera. Each trial requires the robot to consecutively place the yellow bread into the toaster, move the yellow bread onto the plate, place the ham onto the yellow bread, place the black bread into the toaster, and finally place the black bread on top of the ham, as shown in Figure C.1 (Appendix C). We use the same cumulative metric as in CALVIN and report the success rate after each stage together with the average completed sequence length. Unless otherwise stated, each policy is evaluated for 10 complete trials.

Baselines. Our comparisons cover video-pretrained action prediction with GR-1 (Wu et al., 2024), generated visual goals with GR-MG (Li et al., 2025), prospective video features with VPP (Hu et al., 2025), and latent motion supervision with CoWVLA (Yang et al., 2026a). We also adapt WMPO (Zhu et al., 2025) to CALVIN by fine-tuning its world model and OpenVLA policy (Kim et al., 2024) on the same A-C demonstrations. For WMPO and Pivot, we report both the imitation initialization and the refined policy. Refinement in Pivot combines demonstrations with value-labeled latent experience while keeping the encoder, dynamics, and value models frozen, without additional environment interaction during this phase. Further implementation details are provided in Appendix A.

4.2 Main Results

Results on CALVIN. Table 1 shows that refinement improves Pivot at every sequence length. Avg. Len increases from 3.946 to 4.294, a relative improvement of 8.8%. The absolute gain in cumulative success grows from 1.2 percentage points for the first subtask to 9.8 points for the last subtask. This pattern shows that the benefit extends to sustained execution across successive instructions. The refined policy attains the highest average successful sequence length among the methods in the table and the highest success rates for the first three sequence lengths.

Results on the real robot. The real-robot results in Table 2 show a larger separation between imitation and refinement. Avg. Len increases from 1.6 to 4.3, and full-task completion rises from 1

Table 3: **Task-aware supervision for latent policy improvement.** Ablations use CALVIN ABC→D. Checkmarks indicate the supervision retained in each variant. The final column gives the change in Avg. Len relative to the full model.

Variant	Dynamics	Value		Avg. Len. ↑	Δ Full
	Affordance prior	TD	Semantic		
Uniform dynamics weights	–	✓	✓	4.147	−0.147
W/o semantic targets	✓	✓	–	4.239	−0.055
W/o TD targets	✓	–	✓	4.094	−0.200
Full model	✓	✓	✓	**4.294**	0.000
Imitation only	No imagined policy refinement			3.946	−0.348

of 10 trials to 8 of 10. The improvement is concentrated in the continuation of the task. First-stage completion changes from 8 to 9 trials, while completion through the fourth stage increases from 1 to 8 trials. When executed individually, the imitation policy achieves high success rates on each stage (Appendix C), but its performance drops substantially when the stages are executed consecutively. This suggests that transitions between tasks introduce additional difficulty, as each stage must operate on the state produced by the preceding one. In contrast, our method maintains a high success rate over the full sequence by focusing on instruction-relevant regions, which helps the policy handle these cross-stage transitions more reliably.

4.3 Ablation Studies

Table 3 studies the contributions of task-aware supervision in both dynamics and value learning. We ablate the semantic affordance prior used to weight dynamics prediction, as well as the TD and semantic targets used to train the value function, while keeping the remaining components unchanged.

Task-aware supervision for dynamics. Replacing the affordance-based weighting with uniform token weights reduces the average successful sequence length from 4.294 to 4.147. Both variants predict the same frozen encoder features and differ only in how prediction errors are weighted across spatial tokens. This result suggests that emphasizing instruction-relevant regions provides a more useful learning signal for latent dynamics, leading to more effective downstream policy refinement.

Outcome and semantic supervision for value. The two value-learning objectives provide complementary supervision. TD learning grounds the value function in eventual task outcomes, while semantic supervision supplies a dense, instruction-conditioned signal over intermediate states. Using either signal alone is insufficient to match the full model, with the average successful sequence length dropping to 4.094 without TD and to 4.239 without semantic supervision. Combining them reaches 4.294, indicating that semantic alignment provides additional structure beyond sparse outcome supervision. As we show next, this benefit becomes even more pronounced when the value model is asked to rank imagined successors.

4.4 Analysis of Task-Aware Learning

Instruction-dependent affordance priors. Figure 3 isolates the effect of task semantics by holding the RGB observation fixed while varying the language instruction. Despite identical visual input, the spatial prior shifts toward the objects and interaction regions relevant to the requested manipulation. This behavior shows that the affordance prior changes according to what the current instruction makes important. Together with the uniform-weight ablation in Table 3, these visualizations connect the spatial selectivity of the prior to its downstream effect on policy improvement.

Does the value model rank action consequences? We evaluate whether the value model can reliably distinguish the consequences of candidate actions on held-out start states. For each start state, repeated simulator continuations determine which of two candidate action chunks has the higher success rate. Ranking is correct when the value assigns a higher score to that candidate's successor. We evaluate both encoded observed successors and successors predicted by F_ϕ. Since each pair shares the same initial state, ranking successor values is equivalent to ranking the value gains used

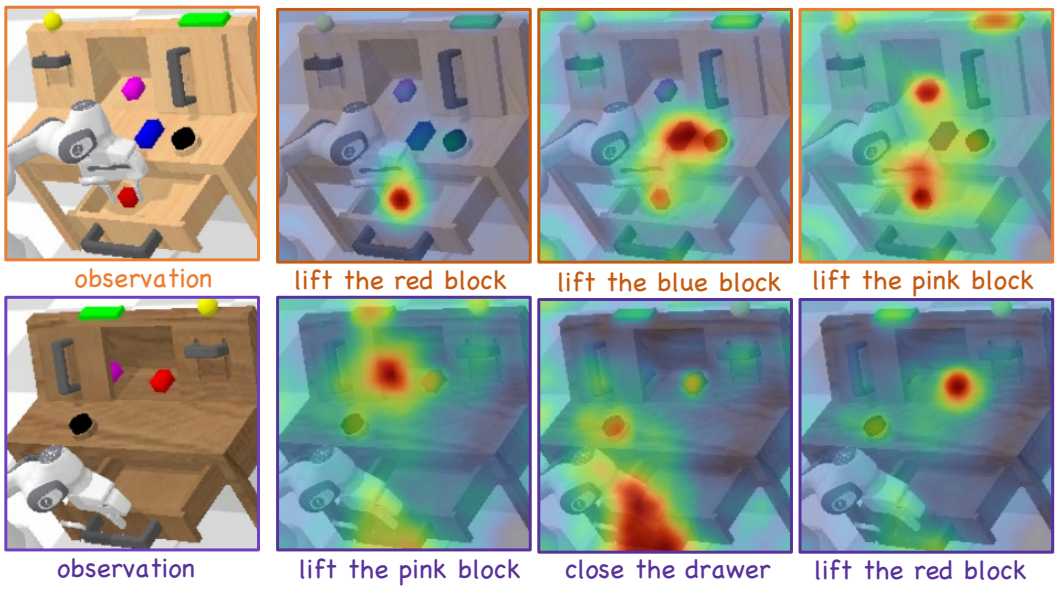


Figure 3: **Task semantics determine the affordance prior.** Within each group, the RGB observation is unchanged and the instruction varies. The maps assign greater prediction weight to objects and interaction regions relevant to each instruction.

Table 4: **Action ranking on observed and imagined successors.** Accuracy is measured on held-out action pairs ordered by success rates from repeated simulator continuations.

Value supervision	Pairwise ranking accuracy (%) ↑	
	Observed successor	Imagined successor
Semantic targets	66.2	52.1
TD targets	71.8	66.9
TD + Semantic targets	**81.7**	**80.3**

during policy refinement. Table 4 shows that the two supervision signals are highly complementary. Semantic targets alone achieve 66.2% accuracy on observed successors but only 52.1% on imagined successors, indicating limited discrimination on model-predicted states. Using TD loss improves these results to 71.8% and 66.9%, respectively. Combining both objectives further raises accuracy to 81.7% and 80.3%, with a particularly pronounced gain on imagined successors. Moreover, the observed-imagined accuracy gap shrinks to just 1.4 percentage points, compared with 4.9 points for TD alone. These results highlight the importance of combining outcome grounding with dense semantic supervision to obtain reliable value estimates on imagined states, where action consequences are evaluated to guide policy refinement.

5 Conclusions and Limitations

We introduced Pivot, a model-based policy improvement framework that closes the Act-Predict-Evaluate loop in a shared video-pretrained state space. A frozen V-JEPA 2 encoder provides the common representation for policy, dynamics, and value learning, enabling recursive latent imagination without observation reconstruction or representation translation. Task-aware dynamics supervision focuses prediction on instruction-relevant regions, while semantic regularization improves value estimation on imagined states. Experiments on CALVIN and a five-stage real-robot task show consistent improvements in long-horizon execution over the imitation policy. Our current study uses a single video-pretrained encoder, and latent prediction errors still accumulate over longer imagination horizons. In addition, the comparison with observation-space imagination is not fully controlled because the systems differ in backbone and visual inputs. Evaluating the framework with other pretrained representations is a direction for future work.

AI USE STATEMENT

Generative AI tools were used only for language editing, including improving grammar, clarity, conciseness, and overall presentation of the manuscript. They were not used to originate the research ideas, design the proposed method, conduct experiments, analyze experimental results, or formulate scientific conclusions. All technical contributions, experimental procedures, reported results, and scientific interpretations were developed and verified by the authors. The authors take full responsibility for the correctness, originality, and final content of this work.

REFERENCES

Mahmoud 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.

Kevin Black, Noah Brown, Danny Driess, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N. Balasubramanian. Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks. In *IEEE Winter Conference on Applications of Computer Vision*, pp. 839–847, 2018.

Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, et al. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024.

Yi Chen, Yuying Ge, Yizhuo Li, Yixiao Ge, Mingyu Ding, Ying Shan, and Xihui Liu. Moto: Latent motion token as the bridging language for robot manipulation. In *ICCV*, 2025.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *International Conference on Learning Representations*, 2020.

Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. In *International Conference on Machine Learning*, 2022.

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. In *International Conference on Machine Learning*, 2025.

Andrew Jaegle, Felix Gimeno, Andy Brock, Andrew Zisserman, Oriol Vinyals, and João Carreira. Perceiver: General perception with iterative attention. In *International Conference on Machine Learning*, 2021.

Siddharth Karamcheti, Suraj Nair, Annie S. Chen, Thomas Kollar, Chelsea Finn, Dorsa Sadigh, and Percy Liang. Language-driven representation learning for robotics. *arXiv preprint arXiv:2302.12766*, 2023.

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.

Peiyan Li, Hongtao Wu, Yan Huang, Chilam Cheang, Liang Wang, and Tao Kong. Gr-mg: Leveraging partially-annotated data via multi-modal goal-conditioned policy. *IEEE Robotics and Automation Letters*, 10(2):1912–1919, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Xiaokang Liu, Zechen Bai, Hai Ci, Kevin Yuchen Ma, and Mike Zheng Shou. World-vla-loop: Closed-loop learning of video world model and vla policy. *arXiv preprint arXiv:2602.06508*, 2026.

Yunfan Lou, Xiaowei Chi, Xiaojie Zhang, Zezhong Qian, Chengxuan Li, Rongyu Zhang, Yaoxu Lyu, Guoyu Song, Chuyao Fu, Haoxuan Xu, et al. Mask world model: Predicting what matters for robust robot policy learning. *arXiv preprint arXiv:2604.19683*, 2026.

Yecheng Jason Ma, William Liang, Vaidehi Som, Vikash Kumar, Amy Zhang, Osbert Bastani, and Dinesh Jayaraman. LIV: Language-image representations and rewards for robotic control. In *International Conference on Machine Learning*, 2023a.

Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. VIP: Towards universal visual reward and representation via value-implicit pre-training. In *International Conference on Learning Representations*, 2023b.

Arjun Majumdar, Karmesh Yadav, Sergio Arnaud, Yecheng Jason Ma, Claire Chen, Sneha Silwal, Aryan Jain, Vincent-Pierre Berges, Pieter Abbeel, Jitendra Malik, Dhruv Batra, Yixin Lin, Oleksandr Maksymets, Aravind Rajeswaran, and Franziska Meier. Where are we in the search for an artificial visual cortex for embodied intelligence? *arXiv preprint arXiv:2303.18240*, 2023.

Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. CALVIN: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7(3):7327–7334, 2022.

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3M: A universal visual representation for robot manipulation. In *Conference on Robot Learning*, 2022.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, et al. Open x-embodiment: Robotic learning datasets and rt-x models. In *ICRA*, pp. 6892–6903, 2024.

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.

Julian Quevedo, Ansh Kumar Sharma, Yixiang Sun, Varad Suryavanshi, Percy Liang, and Sherry Yang. WorldGym: World model as an environment for policy evaluation. *arXiv preprint arXiv:2506.00613*, 2025.

Ansh Kumar Sharma, Yixiang Sun, Ninghao Lu, Yunzhe Zhang, Jiarao Liu, and Sherry Yang. World-Gymnast: Training robots with reinforcement learning in a world model. *arXiv preprint arXiv:2602.02454*, 2026.

Jingwen Sun, Wenyao Zhang, Zekun Qi, Shaojie Ren, Zezhi Liu, Hanxin Zhu, Guangzhong Sun, Xin Jin, and Zhibo Chen. Vla-jepa: Enhancing vision-language-action model with latent world model. In *European Conference on Computer Vision*, pp. 478–497. Springer, 2026.

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

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. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026a.

Jiazhi Yang, Kunyang Lin, Jinwei Li, Wencong Zhang, Tianwei Lin, Longyan Wu, Zhizhong Su, Hao Zhao, Ya-Qin Zhang, Li Chen, Ping Luo, Xiangyu Yue, and Hongyang Li. RISE: Self-improving robot policy with compositional world model. *arXiv preprint arXiv:2602.11075*, 2026b.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, et al. Latent action pretraining from videos. In *International Conference on Learning Representations*, 2025.

Jianke Zhang, Yucheng Hu, Yanjiang Guo, Xiaoyu Chen, Yichen Liu, Wenna Chen, Chaochao Lu, and Jianyu Chen. Unijepa: Enhancing robot policy via unified continuous and discrete representation learning. *arXiv preprint arXiv:2510.10642*, 2025a.

Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, Xinqiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, Fan Lu, He Wang, Zhizheng Zhang, Li Yi, Wenjun Zeng, and Xin Jin. DreamVLA: A vision-language-action model dreamed with comprehensive world knowledge. *arXiv preprint arXiv:2507.04447*, 2025b.

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

Fangqi Zhu, Zhengyang Yan, Zicong Hong, Quanxin Shou, Xiao Ma, and Song Guo. WMPO: World model-based policy optimization for vision-language-action models. *arXiv preprint arXiv:2511.09515*, 2025.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pp. 2165–2183, 2023.

A Implementation Details

A.1 Components of the Shared-State Framework

Encoder and state layout. The encoder is the V-JEPA 2 ViT-G/16 with frozen weights. Each camera view is resized to 224×224; the current frame is duplicated to fill the temporal patch of two frames, which gives 196 tokens of dimension $d = 1408$ per view and $N = 392$ tokens per state, followed by a parameter-free layer normalization. The policy reads the two views as a concatenated sequence of 392 tokens, whereas the dynamics and value models read them interleaved along the width of a 14×28 grid. The two layouts differ by a fixed index permutation. Language goals are encoded by a frozen CLIP ViT-B/32 text encoder.

Policy. The resampler is a Perceiver (Jaegle et al., 2021) with 6 layers and 224 latents of width 384. The context encoder has 4 layers and reads one goal token, the 224 latents, and 16 learned future tokens. The decoder has 4 blocks with adaptive layer normalization (Peebles & Xie, 2023) and cross-attention, and maps $H = 8$ noisy action tokens to velocities. Actions are 7-dimensional relative end-effector motions with a gripper command at 30 Hz. The flow time follows a reflected $\mathrm{Beta}(1.5, 1)$ distribution and sampling uses four Euler steps; the whole chunk is executed before the next observation is encoded. The policy is trained on the language-annotated demonstrations of CALVIN scenes A-C and does not use proprioception.

Dynamics model. The predictor has 24 blocks of width 1024 with 16 heads, rotary position embeddings, and block-causal attention across frames; tokens are projected from 1408 to 1024 dimensions and back. Each frame carries one action token and one goal token, and one step is one control step at 30 Hz. Training uses windows of 25 frames from the demonstrations of scenes A-C and only the autoregressive rollout loss of Eq. 4, with the horizon curriculum $K = 8 \to 16 \to 24$. At rollout time a key-value cache carries the full imagined prefix.

Value model. The value reuses the resampler, goal projection, future tokens, and context encoder of the imitation policy, initialized from its weights, followed by a multi-head attention readout, a layer normalization, and a two-layer MLP with $\tanh$ output. The context path and the readout use learning rates of 3×10^{-5} and 3×10^{-4}.

A.2 Affordance Maps

AMaps are computed offline for every frame and both views. The attribution target is the cosine similarity between the LIV image embedding and the LIV embedding of the instruction. Grad-CAM++ maps are taken from the four residual stages of the LIV ResNet-50 trunk, rectified and min–max normalized per stage, fused with weights 0.05, 0.15, 0.30, and 0.50, and normalized again to $[0, 1]$. We use $\lambda = 1$ in Eq. 4. Maps are stored at 56×56 resolution, warped through the random crop applied to the encoder input, and average-pooled to the 14×14 token grid of each view.

A.3 Value Training

Objective. Each training window consists of nine consecutive states and one additional observation for bootstrapping. For a trajectory of length L, terminal rewards are assigned to the final $W(L) = \mathrm{clip}(\lceil 10(L/40)^2 \rceil, 1, 10)$ states, with $r^+ = 1$ for successful trajectories and $r^- = -0.6$ for failed trajectories. We use a discount factor of $\gamma = 0.995$, an exponential moving average (EMA) target network with an update rate of 0.01, and a semantic supervision weight of $\lambda_{\mathrm{sem}} = 0.01$. Both value predictions and TD targets are constrained to $[-1, 1]$, with the latter explicitly clipped to this range.

Failure endpoints. Failed rollouts are recorded for a fixed horizon of 360 steps. An agent-based pipeline identifies candidate failure endpoints using simulator-state checks and rendered storyboards, with each proposal subsequently verified by a human annotator. Only the trajectory prefix up to the first confirmed failure is retained for training, while the remaining steps are discarded.

A.4 Policy Improvement

We use a zero advantage threshold to construct binary improvement labels. Action chunks with positive advantage are assigned $c = 1$, those with negative advantage are assigned $c = 0$, and

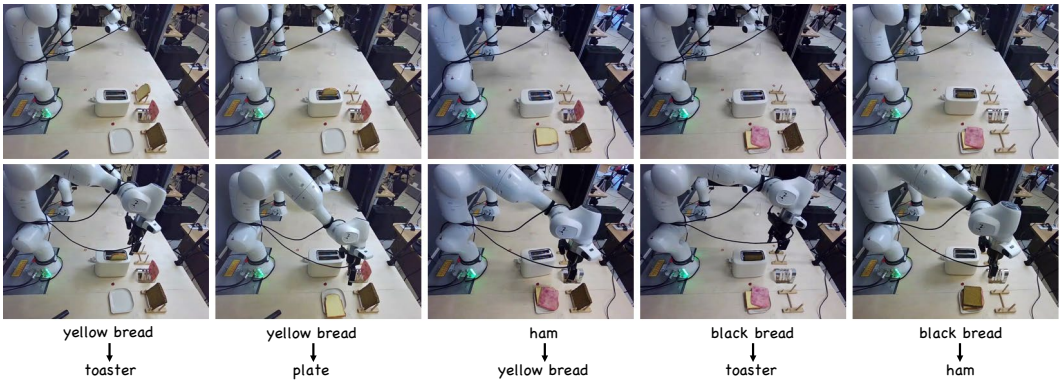


Figure C.1: **Five-stage real-robot sandwich assembly.** Each column corresponds to one stage in execution order, showing its initial (top) and final (bottom) observations from the global camera. The manipulated object and its target destination are indicated below each column.

zero-advantage chunks are excluded from both the flow-matching and velocity-matching objectives. The label is embedded, passed through a zero-initialized projection, and added to the flow-time embedding that modulates the decoder through adaptive layer normalization.

The policy is initialized from the imitation policy. The resulting training objective is

$$\mathcal{L} = 0.5\,\mathcal{L}_{\mathrm{CFM}}^{\mathrm{demo}} + 0.5\,\mathcal{L}_{\mathrm{CFM}}^{\mathrm{img}}.$$

The first $1{,}000$ updates use demonstration samples exclusively. Thereafter, each batch contains equal proportions of demonstration and imagined samples, with the latter generated from demonstration states. We train for $15{,}000$ updates using AdamW with a global batch size of 256 and gradient clipping at 1. The learning rate increases linearly from 2×10^{-6} to 3×10^{-4} over the first $2{,}000$ updates, remains constant until update $4{,}000$, and then follows a cosine decay to 10^{-6}. The conditioning projection uses ten times the base learning rate.

B Experimental Details

Evaluation protocol. All CALVIN evaluations use the same $1{,}000$ instruction chains with Mesa rendering . Each subtask is allocated at most 360 control steps. Our policy predicts action chunks of length 8, which are executed in full and generated using four Euler integration steps.

WMPO implementation. We fine-tune OpenVLA-7B using LoRA with rank 32 on demonstrations from environments A-C. The policy takes observations from a single static camera and predicts discrete tokens corresponding to action chunks of length 8. Following WMPO, we train a video-diffusion world model and a VideoMAE-based outcome classifier on the same A-C data, both of which remain frozen during policy refinement.

Ranking protocol. We sample initial states from imitation-policy rollouts in environments A-C. For each state, candidate action chunks are executed from the same restored simulator state, and their success rates are estimated through repeated continuations. We retain only pairs with different empirical success rates, using the higher-success candidate as the ranking target. Table 4 reports pairwise ranking accuracy on 71 held-out source episodes.

C Real-Robot Details

Figure C.1 illustrates the five consecutive stages of the real-robot sandwich assembly task, captured from the global camera. To distinguish individual skill execution from errors accumulated across stage transitions, we additionally evaluate the imitation policy from manually established initial conditions. For each setting, all preceding stages are completed by hand, and the policy executes the remaining stages consecutively. Table C.1 reports cumulative success rates over 10 trials per setting.

Table C.1: Imitation policy on the real robot, started with the first stages completed by hand.

Stages evaluated	Stage 1	Stage 2	Stage 3	Stage 4	Stage 5
Last 1	–	–	–	–	90%
Last 2	–	–	–	70%	70%
Last 3	–	–	70%	10%	10%
Last 4	–	70%	20%	10%	0%
All 5	80%	40%	20%	10%	10%

The imitation policy completes the final stage in 90% of trials when evaluated independently, but its success rate drops markedly as more preceding stages are included. This contrast highlights the difficulty of maintaining reliable execution across successive stage transitions, rather than executing the final manipulation skill in isolation.