# LATENT ACTION REPRESENTATION LEARNING FROM DELTA OBSERVATIONS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Latent action models aim to learn action representations from video, enabling visuomotor learning beyond action-labeled demonstrations. Since actions cause state changes, their representations should reflect the visual changes they produce. We present **DOLA** (Delta-Observation Latent Actions), which builds this prior into a jointly learned visual encoder and inverse dynamics model: latent actions are constructed from differences between consecutive observation embeddings, with autoregressive forward prediction and action grounding shaping the representation. We then evaluate latent action models as action representation learners using two standard measures: linear probing of frozen representations, and downstream task performance of policies trained in latent action space. Across four simulated benchmarks, **DOLA** achieves the lowest linear probe error and strong downstream policy performance, using less than 7% of the parameters of the strongest baseline. Trained on unpaired demonstrations from two robot embodiments, **DOLA** learns shared action structure that supports cross-embodiment open-loop trajectory transfer. On a real Franka arm, a single model trained on random task-agnostic calibration data enables zero-shot execution of generated video plans across diverse manipulation tasks, including language-conditioned object pickup in clutter. Videos are available at https://dola-anon.github.io.

## 1 INTRODUCTION

A scale explosion is underway in robotics. In the last two years, the amount of collected robot data grew by five orders of magnitude, from several hours to hundreds of thousands of hours (O'Neill et al., 2024; Bjorck et al., 2025; Black et al., 2024; Team, 2026; Team et al., 2025), and remains five orders of magnitude smaller than Internet-scale videos available in the petabytes. These videos contain only visual observations and no action labels telling robots exactly how to move. Latent action models (LAMs) (Bruce et al., 2024; Ye et al., 2024; Gao et al., 2025; 2026; Zhang et al., 2025; Routray et al., 2025; Chen et al., 2025; Schmidt & Jiang, 2023; Klepach et al., 2026) aim to solve this problem by extracting an action representation from actionless videos.

What should a latent action represent? What exactly is an action? One view, rooted in neuroscience, is that an action is whatever causes the next observation given the current one. In other words, actions can be inferred from the sensory change they produce (Held & Hein, 1963; Prinz, 1997; O'regan & Noë, 2001). This view directly motivates the prevalent recipe for latent action models. Concretely, latent action models (LAMs) train an inverse dynamics model (IDM) that maps a pair of observations $(o_t, o_{t+1})$ to a bottlenecked latent $z_t$, jointly with a forward dynamics model (FDM) that predicts the next observation $o_{t+1}$ given $(o_t, z_t)$, reconstructing in raw pixel space (Schmidt & Jiang, 2023; Ye et al., 2024; Gao et al., 2026) or JEPA-style (Assran et al., 2023; Cui et al., 2024; Assran et al., 2025) in the observation embedding space.

The premise of representation learning for visuomotor control is that good visual embeddings reflect environment states (Majumdar et al., 2023; Xiao et al., 2022; Grill et al., 2020; He et al., 2020; Cui et al., 2024; Sermanet et al., 2018; Karamcheti et al., 2023; Shang et al., 2024; Wang et al., 2025). By the same logic, good latent action embeddings should reflect agent actions. Since actions cause state changes, ground-truth actions should be correlated with state deltas. Therefore, latent action embeddings should correlate with visual embedding deltas. Existing methods hope this correspondence emerges from bottlenecked reconstruction.



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.

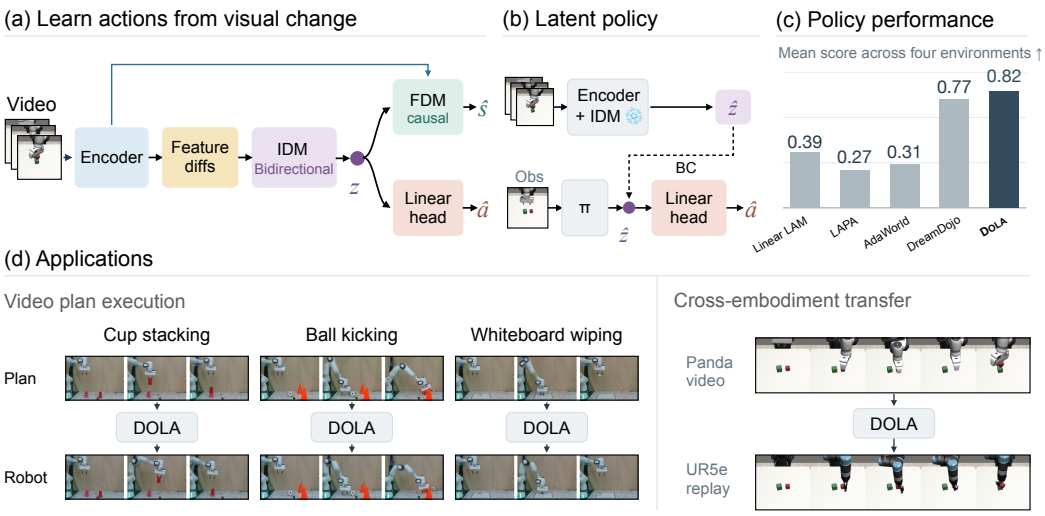



Figure 1: (a) **DOLA** learns a latent action model from actionless video by encoding consecutive observations, forming delta observation embeddings, and mapping them to latent actions with an inverse dynamics model. (b) The visual encoder and inverse dynamics model are frozen for downstream latent policy learning. (c) **DOLA** achieves strong downstream latent policy performance across environments. (d) **DOLA** enables zero-shot execution of generated video plans and cross-embodiment transfer by decoding shared latent actions across different robot embodiments.

In this work, we present **DOLA** (Delta-Observation Latent Actions), a JEPA-style (LeCun et al., 2022) method that builds the delta-embedding prior directly into the model. The IDM constructs latent actions from observation feature differences, jointly learning the visual encoder, while a causal FDM predicts future embeddings autoregressively. Since visual changes also reflect other agents, object dynamics, and camera motion, observation-only training does not guarantee alignment with the agent's actions (Nikulin et al., 2025; Liang et al., 2025; Zhang et al., 2025). Following this established practice, **DOLA** jointly grounds its latents with limited ground-truth action supervision.

To demonstrate the effectiveness of **DOLA**, we evaluate our action representation on four simulated environments, and a real Franka arm. Our main findings are summarized below:

1. We introduce **DOLA**, a JEPA-style LAM that explicitly encodes visual change into the inverse dynamics model, and a unified protocol that evaluates latent actions by linear probing and downstream policy learning with a fixed ground-truth action budget.
2. **DOLA** achieves the lowest linear probe error on all four simulated benchmarks (Table 2) and the strongest latent policy performance (Table 3), at a fraction of the size of prior LAMs.
3. **DOLA** enables the transfer of cross-embodiment actions through a shared latent action space learned from unpaired data across two different robots (Section 3.4).
4. **DOLA** enables executing zero-shot video plans from an off-the-shelf video generator on a real Franka robot, using task-agnostic random calibration data (Section 3.5).
5. We ablate each component of **DOLA** and report its performance impact.

All of our datasets and training and evaluation code will be made publicly available. Videos are available at https://dola-anon.github.io.

## 2 METHOD

Building on the principle that an action is characterized by its effect on the environment, **DOLA** constrains the latent action to be a function of the visual change by construction. Given a sequence of raw visual observations $(o_1, \ldots, o_T)$, we jointly train a visual encoder $f : o_t \mapsto s_t$, a latent inverse dynamics model $q : \Delta s_{t:t+h-1} \mapsto z_{t:t+h-1}$, and a forward dynamics model $p : (s_{t:t+h-1}, z_{t:t+h-1}) \mapsto s_{t+1:t+h}$ on the visual observations. An action head $g : z_t \mapsto a_t$

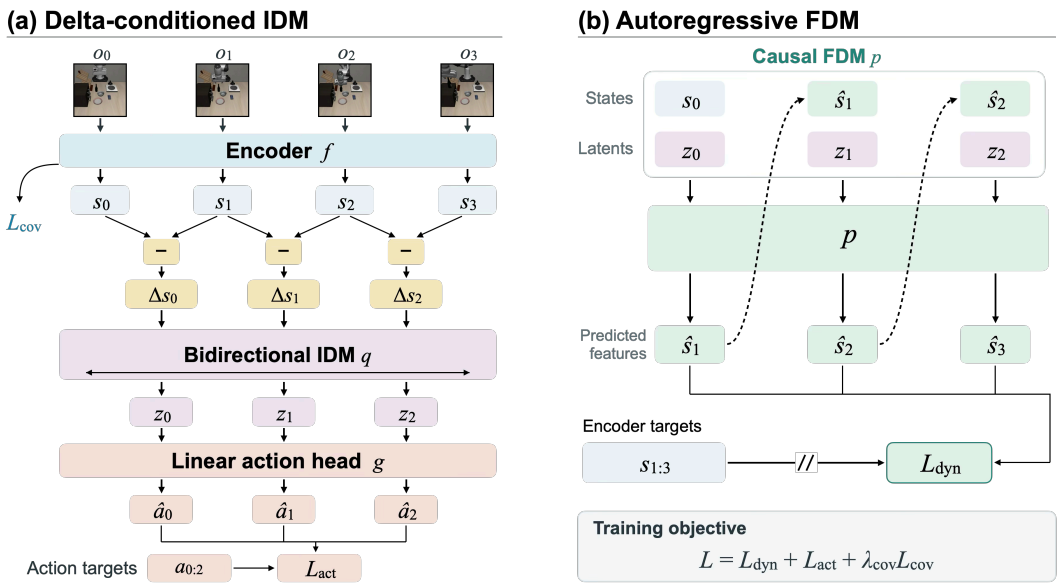



Figure 2: **DOLA** jointly learns a visual encoder, a delta-conditioned inverse dynamics model, an action head, and a forward dynamics model. The latent action is constructed from observation embedding differences and trained with forward prediction, action grounding, and covariance regularization.

learns to map from latent actions to ground-truth actions when such labels are available. The architecture is illustrated in Figure 2.

### 2.1 LATENT ACTION FROM DELTA OBSERVATION EMBEDDINGS

**DOLA** is agnostic to encoder architecture. For simplicity, we use a randomly initialized ResNet-34.

Given an observation window $o_{t:t+h}$, the encoder produces per-frame embeddings $s_i = f(o_i)$. The IDM consumes the temporal difference of consecutive embeddings and projects it to a latent action:

$$\Delta s_t = s_{t+1} - s_t; \quad z_{t:t+h-1} = q(\Delta s_{t:t+h-1}) \tag{1}$$

Because $q$ is a function of the embedding delta alone, $z_i$ is structurally constrained to encode visual change rather than absolute scene content. We instantiate $q$ as a bidirectional transformer. We find that allowing $q$ to attend to the whole context improves performance, as the explicit delta observation prior and a causally masked FDM prevent trivially copying future observations. At test time, we infer overlapping windows and take a mean over all predictions for each $t$ for temporal aggregation.

### 2.2 FORWARD DYNAMICS IN EMBEDDING SPACE

The FDM is a causal transformer predicting future embeddings from past embeddings and latents:

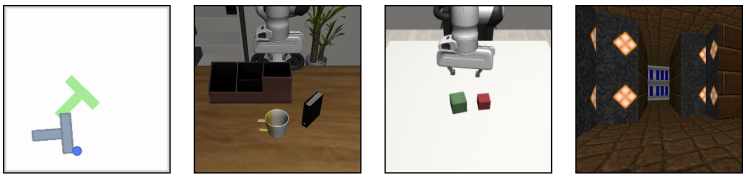
$$\hat{s}_{t+1:t+h} = p(s_{t:t+h-1}, z_{t:t+h-1}) \tag{2}$$

During training, the FDM is given $s_t$ and $z_{t:t+h-1}$ and reconstructs $s_{t+1:t+h}$ autoregressively. To prevent representation collapse to a trivial constant, we follow a SimSiam-style construction (Chen & He, 2021) and have the dynamics loss

$$\mathcal{L}_{\text{dyn}} = \text{MSE}(\hat{s}, \text{sg}[s]). \tag{3}$$

where the prediction target is $\text{sg}[s_{t+1:t+h}]$; $\text{sg}[\cdot]$ is the stop-gradient operator.

We find that the full $h$-step autoregressive rollout improves the action representation performance over one-step teacher forcing (see Table 4). We apply covariance regularization on the minibatch of encoder features following (Bardes et al., 2021; Cui et al., 2024). When ground-truth actions $a_t$ are available, an action head $g$ maps latents to ground-truth actions with an MSE loss $\mathcal{L}_{\text{act}}$; when no action labels are available, $\mathcal{L}_{\text{act}} = 0$.



(a) Push-T (b) LIBERO 90/10 (c) MimicGen Stack (d) VizDoom

Figure 3: We evaluate **DOLA** on four simulated benchmarks.

All four components are trained jointly end-to-end with a single objective:

$$\mathcal{L} = \mathcal{L}_{\text{dyn}} + \mathcal{L}_{\text{act}} + \lambda_{\text{cov}}\mathcal{L}_{\text{cov}} \tag{4}$$

with $\lambda_{\text{cov}} = 0.01$ across all settings. We set $h = 8$ for all runs by default. Optimizer, learning rate schedule, and per-environment dimensions are reported in Appendix E.

## 3 EXPERIMENTS

A latent action model is, in its essence, an action representation learner. The IDM is a representation function, and $z_t$ is the representation. We therefore evaluate it with the standard toolkit of representation learning, and report two numbers:

**Linear probe MSE**. We probe inferred $z_t$ against ground-truth $a_t$ on held-out validation trajectories. A high-quality latent action representation must preserve sufficient information to accurately recover the underlying ground-truth actions.

**Downstream task performance**. We train a downstream latent policy on latent actions inferred from video observations, and compare with an oracle policy trained on ground-truth actions.

**Environments** We evaluate LAMs across four simulated environments spanning manipulation and navigation: Push-T, LIBERO, MimicGen, and VizDoom. See Appendix D for full details.

**Push-T** (Chi et al., 2023): Pushing a T-block into a fixed goal area (206 expert demonstrations).

**LIBERO 90/10** (Liu et al., 2023): We evaluate held-out task transfer on the LIBERO 100 benchmark, training on 90 tasks and evaluating on 10 unseen tasks (50 demonstrations per task).

**MimicGen Stack** (Mandlekar et al., 2023): We use the Stack task in MimicGen with the UR5e embodiment, in which the robot stacks two blocks (1 000 demonstrations).

**VizDoom** (Wydmuch et al., 2018): We use the "My Way Home" task, in which an agent navigates to a fixed goal from random starts (1 000 demonstrations).

For LIBERO 90/10, all LAMs are trained on the full LIBERO 90 dataset observations and ground-truth actions, and evaluated on the full LIBERO 10 dataset as the held-out validation split. For Push-T, MimicGen Stack, and VizDoom, 5% of all episodes are held out as the validation split; the rest are for training: 90% are observations only, and 5% have ground-truth actions.

**Baselines** We compare **DOLA** with four representative LAM baselines. We fix the latent action dimension to 128 across all methods to avoid confounding evaluation results with representation capacity. For action grounding, we attach a linear action head to each LAM to predict ground-truth actions. Additional architecture and implementation details are provided in Appendix E.

**LAPA**: VQVAE on pairs of consecutive video frames from (Ye et al., 2024). We use the post-quantization codebook entry as the latent action representation. We attach a linear action head to the concatenated pre-quantization latent tokens for grounding.

**AdaWorld**: spatiotemporal VAE reconstructing future frame from (Gao et al., 2025). We use its continuous bottleneck as the latent action representation. We attach a linear action head to the posterior mean of this bottleneck for grounding.

**DreamDojo** (Gao et al., 2026): We evaluate its LAM and use its continuous bottleneck as the latent action representation. We attach a linear action head to the posterior mean for grounding.

**Linear LAM** (Zhang et al., 2025): a minimal LAM baseline with linear model components. We attach a linear action head to the latent following the paper.

**Mean baseline**: MSE of predicting the mean of all validation actions (or effectively, validation set variance). On our normalized datasets, this metric will be close to 1.0.

Our experiments are designed to answer the following key questions:

- Do latent action models learn decodable actions without grounding?
- How well does **DOLA** learn an action representation?
- Do **DOLA** latents enable downstream latent policy learning?
- What information is captured in **DOLA**'s latent space, and do latent actions learned across different embodiments map to a shared representation?
- Do **DOLA** latents generalize to novel tasks in both simulated and real-world environments?
- How does each component of **DOLA** affect the quality of the learned representations?

### 3.1 DO LAMS LEARN DECODABLE ACTIONS WITHOUT GROUNDING?

We first ask whether LAMs learn latent actions that are decodable without action grounding during training. We train all methods on Push-T with no ground-truth action supervision and evaluate the resulting latent representations using linear probing. As shown in Table 1, every method has linear probe error close to the mean-action baseline: a linear probe on the learned latents does little better than always predicting the average action. This is expected: the observation-only objectives constrain the latent actions only up to reparameterization. Any invertible transformation of $z$ explains the observed transitions as well as the true actions do, so the objective contains nothing that favors a representation aligned with the environment's action space. A small amount of action supervision is therefore needed to ground the learned latent actions. For all remaining experiments, we follow prior work (Nikulin et al., 2025; Zhang et al., 2025) and train with a limited amount of action grounding.

| Push-T | Mean baseline | Linear LAM | LAPA | AdaWorld | DreamDojo | **DOLA** (Ours) |
|---|---|---|---|---|---|---|
| 0% | 1.10 | $0.86 \pm 0.01$ | $1.03 \pm 0.01$ | $0.74 \pm 0.02$ | $0.86 \pm 0.01$ | $0.84 \pm 0.05$ |
| 5% | 1.10 | $0.70 \pm 0.10$ | $1.11 \pm 0.01$ | $0.35 \pm 0.17$ | $0.05 \pm 0.01$ | $0.01 \pm 0.01$ |

Table 1: Linear probe MSE (↓) on Push-T with 5% versus 0% ground-truth action supervision.

### 3.2 HOW WELL DOES **DOLA** LEARN AN ACTION REPRESENTATION?

As shown in Table 2, **DOLA** achieves the lowest linear probe error on all four simulated environments. On Push-T, LIBERO 90/10, and MimicGen Stack, **DOLA** outperforms the best baseline DreamDojo with 7% of the parameter count (47M vs 710M), demonstrating the effectiveness of the delta-observation inductive bias.

### 3.3 LATENT POLICY EVALUATIONS

We evaluate whether the learned latent actions support downstream behavioral cloning. For each method, we freeze the trained LAM, infer latent actions from demonstration videos, and train a policy to predict them. On Push-T, MimicGen Stack, and VizDoom, policy training uses the corresponding training demonstrations. On LIBERO 90/10, the LAM is trained on LIBERO 90, while the policy is trained on latent actions inferred from LIBERO 10 videos. For all methods, we freeze the LAM after pretraining and refit the linear action head by ordinary least squares using the same ground-truth action labels available during LAM training, then use it to decode latent action predictions.

We follow the VQ-BeT training setup of Zhou et al. (2026), using identical policy hyperparameters across all methods within each environment. All policies use global features from the same frozen DynaMo-pretrained ResNet-34 encoder. We also evaluate with WebSSL CLS features, a strong representation identified in the same study, and observe broadly consistent trends (Section F). The oracle uses the same setup and demonstrations but predicts ground-truth actions directly. We report mean final target coverage for Push-T and success rate for the other environments, selecting the best checkpoint in each run and reporting the mean and standard deviation across three seeds.

| Method | Push-T | LIBERO 90/10 | MimicGen Stack | VizDoom |
|---|---|---|---|---|
| Mean baseline | 1.10 | 0.91 | 1.21 | 0.99 |
| Linear LAM | $0.70 \pm 0.10$ | $0.56 \pm 0.04$ | $0.53 \pm 0.01$ | $0.57 \pm 0.01$ |
| LAPA | $1.11 \pm 0.01$ | $0.74 \pm 0.12$ | $0.74 \pm 0.05$ | $0.63 \pm 0.03$ |
| AdaWorld | $0.35 \pm 0.17$ | $0.14 \pm 0.01$ | $0.67 \pm 0.04$ | $0.62 \pm 0.03$ |
| DreamDojo | $0.05 \pm 0.01$ | $0.13 \pm 0.01$ | $0.24 \pm 0.01$ | $0.42 \pm 0.01$ |
| **DOLA** (Ours) | $\mathbf{0.01 \pm 0.01}$ | $\mathbf{0.10 \pm 0.01}$ | $\mathbf{0.17 \pm 0.01}$ | $\mathbf{0.39 \pm 0.01}$ |

Table 2: Linear probe MSE (↓) on held-out trajectories.

Table 3 shows that **DOLA** recovers much of the oracle policy's performance and is competitive with DreamDojo at less than 10% of its model size.

| Method | Push-T | LIBERO 90/10 | MimicGen Stack | VizDoom |
|---|---|---|---|---|
| Oracle | $0.64 \pm 0.01$ | $0.90 \pm 0.01$ | $0.92 \pm 0.02$ | $1.00 \pm 0.01$ |
| Linear LAM | $0.09 \pm 0.01$ | $0.26 \pm 0.06$ | $0.22 \pm 0.02$ | $\mathbf{0.99 \pm 0.01}$ |
| LAPA | $0.07 \pm 0.01$ | $0.04 \pm 0.04$ | $0.09 \pm 0.01$ | $0.88 \pm 0.09$ |
| AdaWorld | $0.17 \pm 0.06$ | $0.62 \pm 0.05$ | $0.02 \pm 0.01$ | $0.41 \pm 0.10$ |
| DreamDojo | $\mathbf{0.57 \pm 0.02}$ | $0.71 \pm 0.01$ | $0.82 \pm 0.04$ | $\mathbf{0.96 \pm 0.01}$ |
| **DOLA** (Ours) | $\mathbf{0.56 \pm 0.02}$ | $\mathbf{0.84 \pm 0.01}$ | $\mathbf{0.90 \pm 0.01}$ | $\mathbf{0.98 \pm 0.02}$ |

Table 3: Latent policy performance (↑) across environments with DynaMo features.

### 3.4 CROSS-EMBODIMENT LATENT ACTION LEARNING

A useful latent action representation should capture transition structure that is shared across robot embodiments. We evaluate whether **DOLA** learns a latent action space jointly across embodiments while using only lightweight embodiment-specific action decoders.

We use the MimicGen Stack task with two embodiments: a 7-DoF Franka Panda arm, and a 6-DoF UR5e arm. Demonstrations are collected separately for each embodiment (no paired data). Actions are embodiment-normalized joint velocity commands. We train a shared **DOLA** model on both demonstration sets, grounded by separate action heads for each robot embodiment, $g^{\text{ur5}}$ and $g^{\text{panda}}$.

**Cross-embodiment open-loop transfer.** We infer latent actions from a Panda video, and replay on UR5e with its action head (Figure 4). **DOLA** produces UR5e joint trajectories that track the original Panda demonstration, indicating that **DOLA** captures action content decodable across embodiments.

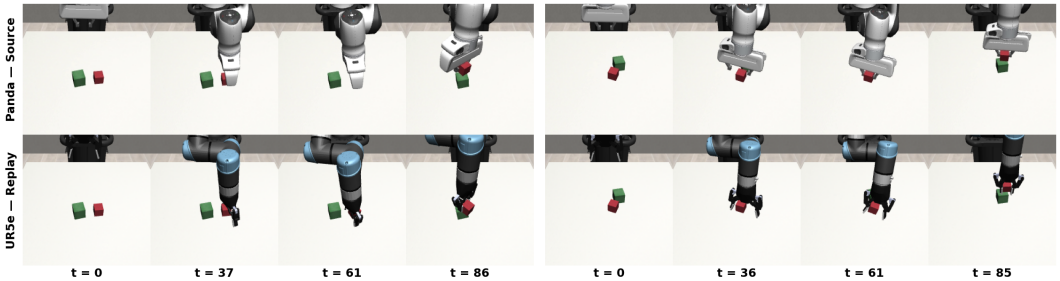



Figure 4: Cross-embodiment open-loop transfer on MimicGen Stack. Despite swapping a 7-DoF Panda for a 6-DoF UR5e, the UR5e tracks the source trajectory and stacks cube A onto cube B.

**Latent semantic analysis.** We test whether physically interpretable end-effector directions emerge despite supervising auxiliary action heads only on per-robot delta joint commands $\Delta\mathbf{q}$ and a binary gripper command. From held-out demonstrations of both embodiments, we extract $\Delta x$, $\Delta y$, $\Delta z$, $\Delta r_z$, and the gripper command. We omit $\Delta r_x$ and $\Delta r_y$ because they remain near zero in the essentially planar demonstrations. Pooling latents across embodiments, we fit one-dimensional canonical correlation with each target, recovering a shared latent direction $w$ per target.

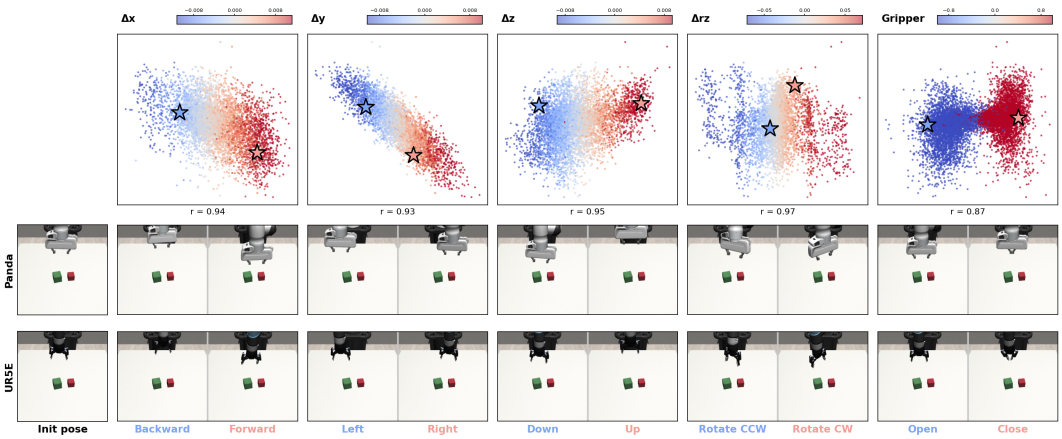



Figure 5: Latent semantic analysis on `stack`. Top: pooled Panda and UR5e latents projected onto the CCA direction $w$ recovered for each end-effector target; color and inset bar indicate the target value, and $r$ denotes the canonical correlation. The blue/red $\star$ mark the low/high target latents used as $z_{\text{neg}}$ and $z_{\text{pos}}$. Bottom: Panda and UR5e rollouts after decoding the same $z_{\text{neg}}/z_{\text{pos}}$ through their respective action heads from a fixed initial pose.

Latent projections along these directions correlate with their targets at $r = 0.94, 0.93, 0.95, 0.97, 0.87$, suggesting that each aligns with a single linear direction. Decoding latents from the bottom/top 5% of each target value through the action heads produces Panda and UR5e rollouts along the expected axes, suggesting embodiment-agnostic action structure (Fig. 5).

### 3.5 Executing zero-shot video plans

We test whether **DOLA** can convert generated video plans into robot actions for tasks unseen during training. We train **DOLA** with background augmentation on a calibration dataset of task-agnostic Franka arm motion, without target objects or task instructions (Fig. 8). At deployment, an off-the-shelf video generator produces a plan from an initial scene image and a task instruction. **DOLA** extracts latent actions from the plan, and its action head maps them to robot commands for open-loop execution. We use the same model across tasks without task-specific fine-tuning.

**Diverse zero-shot video plan execution.** We test the same model on four tasks unseen during calibration: *cup stacking*, *button pressing*, *ball kicking*, and *whiteboard wiping*. For each task, we generate a video plan from the initial real image and execute it with **DOLA**, without task-specific robot data or additional fine-tuning (Fig. 6(a)). We also evaluate language-conditioned pickup in a cluttered scene containing five objects. Given an instruction specifying the target object, the video generator produces a pickup plan that **DOLA** converts into robot actions. Fig. 6(b) shows temporal overlays of the generated plans and robot executions for all five targets. Finally, we test stage-wise replanning, executing each stage open-loop and using the resulting real image to generate the next plan. The robot successfully opens the oven and places the bread inside, but fails to close it (Fig. 6(c)).

**Failure modes and limitations.** We observe failures in both video generation and inverse dynamics (Figure 6d). Generated videos may depict visually plausible motion that is physically unrealistic or inconsistent with the scene geometry. In particular, the limited monocular view makes object depth ambiguous, which can lead to generated motion at the wrong depth relative to the target object. In the ball-kicking example, the generated video shows the ball rolling after the gripper touches it and stops, despite no kicking motion. The IDM can also predict inaccurate end-effector motion from the generated frames, leading to missed contacts or failed grasps. In the second example, the gripper closes about 1 cm above the bread and fails to grasp it. Since most failures stem from the video generation, we manually screen the generated plans and report qualitative execution results.

### 3.6 Ablations

We ablate the components of **DOLA** and measure their impact following the same protocol as Section 3.2, reporting the linear probe MSE across three random seeds.

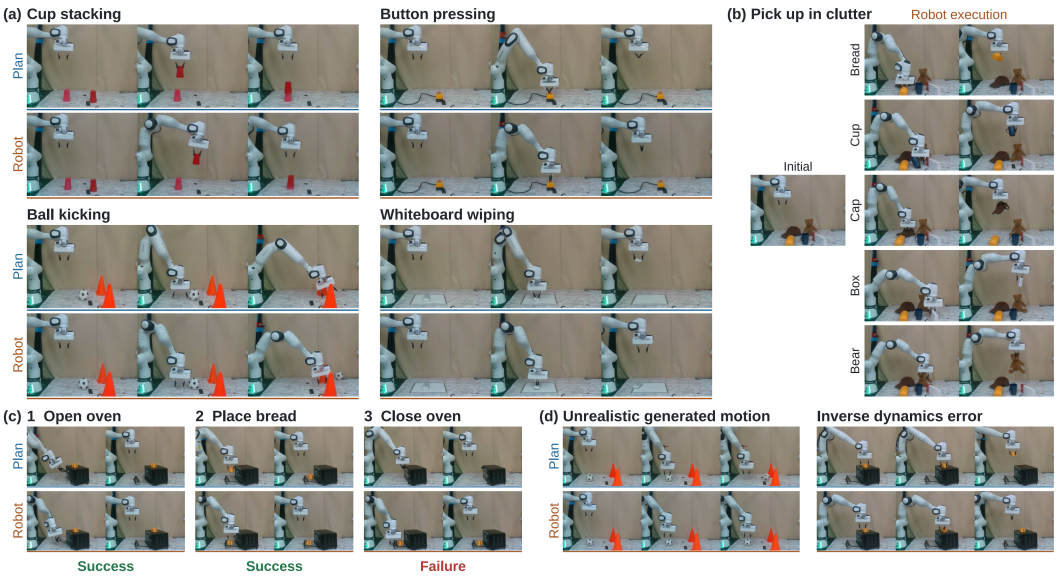



Figure 6: Zero-shot video plan execution. (a) Four unseen manipulation tasks. (b) Language-conditioned pickup in clutter. (c) Stage-wise oven task. (d) Two main failure modes.

Table 4: Ablations across our four simulated environments, linear probe MSE

| Method | Push-T | LIBERO 90/10 | MimicGen Stack | VizDoom |
|---|---|---|---|---|
| **DOLA** full model | $0.01 \pm 0.01$ | $0.10 \pm 0.01$ | $0.17 \pm 0.01$ | $0.39 \pm 0.01$ |
| No delta-obs | $0.62 \pm 0.02$ | $0.15 \pm 0.01$ | $0.34 \pm 0.01$ | $0.57 \pm 0.05$ |
| No covariance regularization | $0.02 \pm 0.01$ | $0.10 \pm 0.01$ | $0.14 \pm 0.01$ | $0.40 \pm 0.01$ |
| Causal IDM | $0.01 \pm 0.01$ | $0.20 \pm 0.01$ | $0.16 \pm 0.01$ | $0.41 \pm 0.01$ |
| No temporal aggregation | $0.02 \pm 0.01$ | $0.27 \pm 0.01$ | $0.20 \pm 0.01$ | $0.46 \pm 0.01$ |
| Short context ($h = 2$) | $0.02 \pm 0.01$ | $0.24 \pm 0.01$ | $0.23 \pm 0.01$ | $0.50 \pm 0.03$ |
| No autoregressive rollout | $0.01 \pm 0.01$ | $0.10 \pm 0.01$ | $0.16 \pm 0.01$ | $0.56 \pm 0.04$ |

**Delta-observation input.** Replacing feature differences with raw features significantly degrades performance, confirming that this inductive prior into model training is key to **DOLA**'s action representation learning performance.

**Covariance regularization.** Covariance regularization can help prevent collapse and stabilize training; we observe that **DOLA** can nevertheless train stably without it.

**Bidirectional IDM.** Using a causally masked IDM instead of a bidirectional one degrades performance on LIBERO and VizDoom, suggesting that access to the full context helps integrate transitions.

**Temporal aggregation.** Removing temporal aggregation during inference degrades performance.

**Observation context.** Reducing the observation context window from the default 8 frames to 2 frames degrades performance across all environments.

**Autoregressive rollout.** Teacher forcing significantly degrades performance on VizDoom.

## 4 Related Works

### 4.1 Latent Actions

Latent actions enable scalable robot learning from action-free video, supporting generalist foundation models (Bruce et al., 2024; Bjorck et al., 2025), world models (Gao et al., 2026; 2025; Garrido et al., 2026; Xie et al., 2025), and large-scale Vision-Language-Action (VLA) pipelines (Ye et al., 2024). To separate action from visual nuisance, prior work uses information bottlenecks ranging from discrete vector quantization (Schmidt & Jiang, 2023; Chen et al., 2025; Ye et al., 2024; Routray et al.,

2025) to continuous regularization via sparsity or noise (Garrido et al., 2026) and transformer-based continuous bridges (Tian et al., 2024). Auxiliary objectives such as optical flow consistency (Routray et al., 2025) and video reconstruction (Chen et al., 2025) further ground the latents in physical dynamics and discourage shortcut learning.

Despite this progress, evaluating the quality of latent actions in isolation remains an open question. Most prior work evaluates representations through end-to-end fine-tuning (Chen et al., 2025; Ye et al., 2024; Routray et al., 2025; Tian et al., 2024) or indirect visual generative metrics (Gao et al., 2026; 2025; Garrido et al., 2026). While isolated metrics like classification (Nie et al., 2026) or linear probing (Nikulin et al., 2025) exist, their correlation to physical execution remains ambiguous. **DOLA** treats latent action learning as a standard representation learning problem. Using a unified semi-supervised framework with limited action grounding, **DOLA** evaluates its frozen representations through both linear probing and downstream policy success, allowing us to directly quantify standalone representation quality independently of the downstream architecture.

### 4.2 Learning Behaviors from Videos

To reduce reliance on action-labeled data, recent work learns behaviors from diverse video sources, including internet-scale human videos (Mendonca et al., 2023; Majumdar et al., 2023; Hoque et al., 2025; Singh et al., 2025; Goswami et al., 2025), demonstrations (Bahl et al., 2022; Guzey et al., 2025), and robot videos (Wen et al., 2023; Tian et al., 2024; O'Neill et al., 2024; Khazatsky et al., 2024). Early approaches bridge the embodiment gap through reinforcement learning, keypoint-based behavior cloning, or dataset alignment (Bahl et al., 2022; Guzey et al., 2025; Haldar & Pinto, 2025; Xie et al., 2026; Cui et al., 2026). In parallel, World Action Models use video prediction to ground policy learning in physical dynamics (Li et al., 2025; Ye et al., 2026). More recently, video diffusion models have enabled generative imitation (Du et al., 2023; Ko et al., 2023; Patel et al., 2025), where executable actions are inferred from generated visual plans by inverse dynamics or correspondence tracking (Tian et al., 2024; Du et al., 2023; Ye et al., 2026; Ko et al., 2023; Bharadhwaj et al., 2024). Building on this direction, our method learns executable latent actions from action-free video, enabling latent policy learning in simulation and zero-shot execution of generated video plans on real robots.

### 4.3 Self-supervised Learning from Pixels

Self-Supervised Learning (SSL) has evolved from representation learning of images (Grill et al., 2020; Oquab et al., 2023; He et al., 2022) to videos (Tong et al., 2022; Bardes et al., 2024; Assran et al., 2025). Because annotations are expensive at scale, SSL techniques demonstrate superior scalability (Fan et al., 2025). Methods broadly split by how they avoid representation collapse. Contrastive approaches repel negative pairs (Oord et al., 2018; Chen et al., 2020; He et al., 2020; Sermanet et al., 2018), while non-contrastive approaches rely on architectural asymmetry or explicit regularizers on the embedding statistics (Chen & He, 2021; Oquab et al., 2023; Zbontar et al., 2021; Bardes et al., 2021; Balestriero & LeCun, 2025). **DOLA** adapts this toolkit to latent action learning.

## 5 Conclusion and Limitations

We presented **DOLA**, a latent action model that builds a delta-observation prior into jointly learned visual and action representations. By constructing latent actions from observation embedding differences and training with autoregressive forward prediction and action grounding, **DOLA** achieves the lowest linear probe error across four simulated benchmarks and supports strong downstream policy learning in latent action space. Our unified evaluation assesses frozen action representations through both linear decodability and downstream control. Beyond that, **DOLA** supports cross-embodiment trajectory transfer and zero-shot execution of generated video plans on a real Franka arm.

Consistent with prior work, our experiments show that delta-observation conditioning alone does not ensure alignment with the environment's action space. Real-robot execution remains sensitive to errors, particularly around depth and contact. Generating physically consistent videos remains an open challenge, and advances in this area could improve downstream execution. Future work should investigate scaling to more diverse videos and embodiments, incorporating pretrained representations, and closed-loop control to correct execution errors.

### AI USE STATEMENT

We used large language models to help refine wording and improve grammar. All technical claims, experimental results, and conclusions were written, checked, and approved by the authors.

### ETHICS STATEMENT

Our work studies latent action learning for robot control. While it may improve data efficiency, deploying learned action representations on physical robots can introduce safety risks due to prediction or execution errors. Our real-robot experiments are conducted in controlled settings, and broader deployment would require appropriate safety safeguards.

### REPRODUCIBILITY STATEMENT

All of our datasets, and training and evaluation code will be made publicly available. Videos of our trained policies can be seen here: https://dola-anon.github.io.

## REFERENCES

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15619–15629, 2023. 1

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025. 1, 9

Shikhar Bahl, Abhinav Gupta, and Deepak Pathak. Human-to-robot imitation in the wild. *arXiv preprint arXiv:2207.09450*, 2022. 9

Randall Balestriero and Yann LeCun. Lejepa: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025. 9

Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021. 3, 9, 16

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *arXiv preprint arXiv:2404.08471*, 2024. 9

Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2act: Predicting point tracks from internet videos enables generalizable robot manipulation. In *European Conference on Computer Vision*, pp. 306–324. Springer, 2024. 9

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025. 1, 8

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. $\pi_0$: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. 1

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024. 1, 8

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PmLR, 2020. 9

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15750–15758, 2021. 3, 9

Yi Chen, Yuying Ge, Weiliang Tang, Yizhuo Li, Yixiao Ge, Mingyu Ding, Ying Shan, and Xihui Liu. Moto: Latent motion token as the bridging language for learning robot manipulation from videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19752–19763, 2025. 1, 8, 9

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44:1684 – 1704, 2023. URL https://api.semanticscholar.org/CorpusID:257378658. 4, 15

Zichen J Cui, Hengkai Pan, Aadhithya Iyer, Siddhant Haldar, and Lerrel Pinto. Dynamo: In-domain dynamics pretraining for visuo-motor control. *Advances in Neural Information Processing Systems*, 37:33933–33961, 2024. 1, 3, 16

Zichen Jeff Cui, Omar Rayyan, Haritheja Etukuru, Bowen Tan, Zavier Andrianarivo, Zicheng Teng, Yihang Zhou, Krish Mehta, Nicholas Wojno, Kevin Yuanbo Wu, Manan H Anjaria, Ziyuan Wu, Manrong Mao, Guangxun Zhang, Binit Shah, Yejin Kim, Soumith Chintala, Lerrel Pinto, and Nur Muhammad Mahi Shafiullah. Contact-anchored policies: Contact conditioning creates strong robot utility models. *arXiv preprint arXiv:2602.09017*, 2026. 9

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *Advances in neural information processing systems*, 36:9156–9172, 2023. 9

David Fan, Shengbang Tong, Jiachen Zhu, Koustuv Sinha, Zhuang Liu, Xinlei Chen, Michael Rabbat, Nicolas Ballas, Yann LeCun, Amir Bar, et al. Scaling language-free visual representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 370–382, 2025. 9

Shenyuan Gao, Siyuan Zhou, Yilun Du, Jun Zhang, and Chuang Gan. Adaworld: Learning adaptable world models with latent actions. *arXiv preprint arXiv:2503.18938*, 2025. 1, 4, 8, 9, 15, 17

Shenyuan Gao, William Liang, Kaiyuan Zheng, Ayaan Malik, Seonghyeon Ye, Sihyun Yu, Wei-Cheng Tseng, Yuzhu Dong, Kaichun Mo, Chen-Hsuan Lin, et al. Dreamdojo: A generalist robot world model from large-scale human videos. *arXiv preprint arXiv:2602.06949*, 2026. 1, 4, 8, 9, 15, 18

Quentin Garrido, Tushar Nagarajan, Basile Terver, Nicolas Ballas, Yann LeCun, and Michael Rabbat. Learning latent action world models in the wild. *arXiv preprint arXiv:2601.05230*, 2026. 8, 9

Raktim Gautam Goswami, Amir Bar, David Fan, Tsung-Yen Yang, Gaoyue Zhou, Prashanth Krishnamurthy, Michael Rabbat, Farshad Khorrami, and Yann LeCun. World models can leverage human videos for dexterous manipulation. *arXiv preprint arXiv:2512.13644*, 2025. 9

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020. 1, 9

Irmak Guzey, Yinlong Dai, Georgy Savva, Raunaq Bhirangi, and Lerrel Pinto. Bridging the human to robot dexterity gap through object-oriented rewards. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3344–3351. IEEE, 2025. 9

Siddhant Haldar and Lerrel Pinto. Point policy: Unifying observations and actions with key points for robot manipulation. *arXiv preprint arXiv:2502.20391*, 2025. 9

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016. 16

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020. 1, 9

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 15979–15988. IEEE, 2022. 9

Richard Held and Alan Hein. Movement-produced stimulation in the development of visually guided behavior. *Journal of comparative and physiological psychology*, 56(5):872, 1963. 1

Ryan Hoque, Peide Huang, David J Yoon, Mouli Sivapurapu, and Jian Zhang. Egodex: Learning dexterous manipulation from large-scale egocentric video. *arXiv preprint arXiv:2505.11709*, 2025. 9

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. 1

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024. 9

Albina Klepach, Alexander Nikulin, Ilya Zisman, Denis Tarasov, Alexander Derevyagin, Andrei Polubarov, Nikita Lyubaykin, Igor Kiselev, and Vladislav Kurenkov. Object-centric latent action learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 22626–22634, 2026. 1

Po-Chen Ko, Jiayuan Mao, Yilun Du, Shao-Hua Sun, and Joshua B Tenenbaum. Learning to act from actionless videos through dense correspondences. *arXiv preprint arXiv:2310.08576*, 2023. 9

Yann LeCun et al. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open Review*, 62(1):1–62, 2022. 2

Seungjae Lee, Yibin Wang, Haritheja Etukuru, H Jin Kim, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. Behavior generation with latent actions. *arXiv preprint arXiv:2403.03181*, 2024. 19

Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. *arXiv preprint arXiv:2503.00200*, 2025. 9

Anthony Liang, Pavel Czempin, Matthew M Hong, Yutai Zhou, Jingzhen Wang, Erdem Biyik, and Stephen Tu. Clam: Continuous latent action models for robot learning from unlabeled demonstrations. *arXiv preprint arXiv:2505.04999*, 2025. 2

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023. 4, 15

Arjun Majumdar, Karmesh Yadav, Sergio Arnaud, Jason Ma, Claire Chen, Sneha Silwal, Aryan Jain, Vincent-Pierre Berges, Tingfan Wu, Jay Vakil, et al. Where are we in the search for an artificial visual cortex for embodied intelligence? *Advances in Neural Information Processing Systems*, 36: 655–677, 2023. 1, 9

Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. *arXiv preprint arXiv:2310.17596*, 2023. 4, 16

Russell Mendonca, Shikhar Bahl, and Deepak Pathak. Structured world models from human videos. *arXiv preprint arXiv:2308.10901*, 2023. 9

Dujun Nie, Fengjiao Chen, Qi Lv, Jun Kuang, Xiaoyu Li, Xuezhi Cao, and Xunliang Cai. Lary: A latent action representation yielding benchmark for generalizable vision-to-action alignment. *arXiv preprint arXiv:2604.11689*, 2026. 9

Alexander Nikulin, Ilya Zisman, Denis Tarasov, Nikita Lyubaykin, Andrei Polubarov, Igor Kiselev, and Vladislav Kurenkov. Latent action learning requires supervision in the presence of distractors. *arXiv preprint arXiv:2502.00379*, 2025. 2, 5, 9

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 9

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 9

J Kevin O'regan and Alva Noë. A sensorimotor account of vision and visual consciousness. *Behavioral and brain sciences*, 24(5):939–973, 2001. 1

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903. IEEE, 2024. 1, 9

Shivansh Patel, Shraddhaa Mohan, Hanlin Mai, Unnat Jain, Svetlana Lazebnik, and Yunzhu Li. Robotic manipulation by imitating generated videos without physical demonstrations. *arXiv preprint arXiv:2507.00990*, 2025. 9

Wolfgang Prinz. Perception and action planning. *European journal of cognitive psychology*, 9(2): 129–154, 1997. 1

Sandeep Routray, Hengkai Pan, Unnat Jain, Shikhar Bahl, and Deepak Pathak. Vipra: Video prediction for robot actions. *arXiv preprint arXiv:2511.07732*, 2025. 1, 8, 9

Dominik Schmidt and Minqi Jiang. Learning to act without actions. *arXiv preprint arXiv:2312.10812*, 2023. 1, 8

Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 1134–1141. IEEE, 2018. 1, 9

Jinghuan Shang, Karl Schmeckpeper, Brandon B May, Maria Vittoria Minniti, Tarik Kelestemur, David Watkins, and Laura Herlant. Theia: Distilling diverse vision foundation models for robot learning. *arXiv preprint arXiv:2407.20179*, 2024. 1

Himanshu Gaurav Singh, Antonio Loquercio, Carmelo Sferrazza, Jane Wu, Haozhi Qi, Pieter Abbeel, and Jitendra Malik. Hand-object interaction pretraining from videos. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3352–3360. IEEE, 2025. 9

Gemini Robotics Team, Abbas Abdolmaleki, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Ashwin Balakrishna, Nathan Batchelor, Alex Bewley, Jeff Bingham, et al. Gemini robotics 1.5: Pushing the frontier of generalist robots with advanced embodied reasoning, thinking, and motion transfer. *arXiv preprint arXiv:2510.03342*, 2025. 1

Generalist AI Team. Gen-1: Scaling embodied foundation models to mastery. *Generalist AI Blog*, 2026. https://generalistai.com/blog/apr-02-2026-GEN-1. 1

Yang Tian, Sizhe Yang, Jia Zeng, Ping Wang, Dahua Lin, Hao Dong, and Jiangmiao Pang. Predictive inverse dynamics models are scalable learners for robotic manipulation. *arXiv preprint arXiv:2412.15109*, 2024. 9

Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in neural information processing systems*, 35:10078–10093, 2022. 9

Mohammad Hassan Vali and Tom Bäckström. Nsvq: Noise substitution in vector quantization for machine learning. *IEEE Access*, 10:13598–13610, 2022. 17

Yixiao Wang, Mingxiao Huo, Zhixuan Liang, Yushi Du, Lingfeng Sun, Haotian Lin, Jinghuan Shang, Chensheng Peng, Mohit Bansal, Mingyu Ding, et al. Ver: Vision expert transformer for robot learning via foundation distillation and dynamic routing. *arXiv preprint arXiv:2510.05213*, 2025. 1

Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning. *arXiv preprint arXiv:2401.00025*, 2023. 9

Marek Wydmuch, Michał Kempka, and Wojciech Jaśkowski. Vizdoom competitions: Playing doom from pixels. *IEEE Transactions on Games*, 11(3):248–259, 2018. 4, 16

Tete Xiao, Ilija Radosavovic, Trevor Darrell, and Jitendra Malik. Masked visual pre-training for motor control. *arXiv preprint arXiv:2203.06173*, 2022. 1

Amber Xie, Oleh Rybkin, Dorsa Sadigh, and Chelsea Finn. Latent diffusion planning for imitation learning. *arXiv preprint arXiv:2504.16925*, 2025. 8

Sicheng Xie, Haidong Cao, Zejia Weng, Zhen Xing, Haoran Chen, Shiwei Shen, Jiaqi Leng, Zuxuan Wu, and Yu-Gang Jiang. Human2robot: Learning robot actions from paired human-robot videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 11078–11086, 2026. 9

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. *arXiv preprint arXiv:2410.11758*, 2024. 1, 4, 8, 9, 15, 17

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026. 9

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International conference on machine learning*, pp. 12310–12320. PMLR, 2021. 9

Chuheng Zhang, Tim Pearce, Pushi Zhang, Kaixin Wang, Xiaoyu Chen, Wei Shen, Li Zhao, and Jiang Bian. What do latent action models actually learn? *arXiv preprint arXiv:2506.15691*, 2025. 1, 2, 5, 15, 17

Gaoyue Zhou, Zichen Jeff Cui, Ada Langford, Bowen Tan, Yann LeCun, and Lerrel Pinto. Patch policy: Efficient embodied control via dense visual representations, 2026. URL https://arxiv.org/abs/2607.18236. 5, 19

## A Broader impact statement

This work studies how to learn latent action representations from video and limited action supervision. Such methods may help robots make better use of large video datasets while reducing the amount of action-labeled data needed for training. In the long term, better latent action representations could make robot learning more data-efficient, but deployment on physical systems should still involve careful testing, safety validation, and human oversight.

## B Reproducibility statement

To support reproducibility, we will release datasets and codebase for **DOLA**. We provide evaluation details and model hyperparameters in the appendix. All methods are compared with the same labeled-action budget and downstream evaluation pipeline.

## C Baseline details

**LAPA (Ye et al., 2024).** LAPA learns discrete latent actions by training a VQ-VAE-style tokenizer on pairs of consecutive video frames. We use the post-quantization codebook entry, which is consumed by the downstream policy, as the latent action representation.

**AdaWorld (Gao et al., 2025).** AdaWorld learns continuous latent actions with a VAE-style video autoencoder. A spatiotemporal encoder produces per-transition Gaussian latents that are used to reconstruct future frames. We use the encoder bottleneck as the latent action representation.

**DreamDojo LAM (Gao et al., 2026).** DreamDojo learns latent actions as proxy actions for large-scale world-model pretraining from heterogeneous robot videos. We evaluate its LAM component, which follows a VAE-style spatiotemporal encoder-decoder design, and use its continuous bottleneck as the latent action representation.

**Linear LAM (Zhang et al., 2025).** Linear LAM restricts the latent action to be a linear function of consecutive observation embeddings, $z = C\phi(o_t) + D\phi(o_{t+1})$, with a linear forward model predicting the next embedding. We use the resulting 128-dimensional transition representation as the latent action.

## D Environment and dataset details

We evaluate latent action representations on four benchmarks: PushT, LIBERO 90/10, MimicGen, and VizDoom. These environments cover planar manipulation, multi-task robot manipulation, cross-embodiment robot manipulation, and egocentric navigation. For PushT, MimicGen, and VizDoom, each latent action model is trained primarily from observation sequences, with 5% of all trajectories paired with ground-truth actions to jointly supervise the latent action model and linear action head. LIBERO follows a separate task-transfer protocol, where we train on LIBERO 90 and evaluate transfer to LIBERO 10.

**PushT.** PushT is a planar contact-rich manipulation task in which a circular pusher moves a T-shaped rigid body into a fixed target region. We use the standard PushT dataset from Diffusion Policy (Chi et al., 2023), which contains 206 expert demonstrations. Observations are top-down RGB renderings of the workspace. Actions are two-dimensional relative pusher commands. During evaluation, we execute the learned policy in the environment and report final target coverage, where a score of 1.0 corresponds to perfect coverage of the target region. We use 100 evaluation rollouts.

**LIBERO 90/10.** LIBERO 90/10 is a suite of goal-conditioned simulated robot manipulation tasks from LIBERO (Liu et al., 2023). We use the full LIBERO 90 dataset for training and the LIBERO 10 tasks for evaluating task transfer. Each task is performed by a simulated Franka Panda robot. Observations contain two RGB camera views: a fixed third-person view and a wrist-mounted view. Actions are robot control commands normalized to $[-1, 1]$ following the dataset convention. We

jointly train the latent action model and linear action head across all LIBERO 90 tasks using ground-truth actions, and evaluate whether the learned representation transfers to the LIBERO 10 tasks. At evaluation time, we run 100 rollouts in total and report the average task success rate.

**MimicGen.** MimicGen is a robot manipulation benchmark generated from a small number of human demonstrations (Mandlekar et al., 2023). We use MimicGen to evaluate both downstream latent-policy performance and the cross-embodiment transfer properties of the learned latent action space. Observations contain two RGB camera views: an external agent-view camera and a wrist-mounted camera. Actions are represented in the robot joint velocity space and normalized, with a binary gripper command.

For the cross-embodiment experiments, we use the MimicGen Stack task with two robot embodiments: a Franka Panda arm and a UR5e arm. The two embodiments differ in kinematics, joint dimensionality, and gripper hardware. We therefore share the observation encoder and latent action projector across both embodiments, but use separate lightweight action heads for each robot. This setup tests whether the latent representation captures transition structure that is shared across embodiments, while allowing the final decoder to map the shared latent into embodiment-specific controls. For open-loop transfer, we infer a latent action sequence from a source demonstration in one embodiment, decode the same latents through the other embodiment's action head, and replay the resulting commands in the target robot environment.

**VizDoom.** VizDoom is an egocentric benchmark based on the Doom game engine (Wydmuch et al., 2018). We use a single continuous-control navigation task "My Way Home" with randomized agent initial states. The dataset contains 1000 trajectories. Observations are first-person RGB images at native resolution $160 \times 120$. The action space is three-dimensional and controls the agent camera yaw velocity, forward/backward velocity, and left/right velocity. We evaluate the learned latent policy over 100 rollouts and report average success rate.

**Real-world Franka calibration data.** The Franka calibration dataset contains task-agnostic data collected on a real Franka Panda arm. Each trajectory begins at a fixed front-facing home pose and randomly covers the workspace visible to the camera, with random rotations and gripper toggles at unscripted times. Observations contain a single RGB camera view at 30 Hz, paired with 60 Hz absolute end-effector states; 7-dim delta actions are computed offline from consecutive states. The position and rotation channels are normalized to zero mean and unit variance, and the gripper channel is encoded as a binary value ($-1$ open, $+1$ closed) carrying the absolute next-frame gripper state. The dataset contains no task-specific behavior and is used to train a single inverse-dynamics model for downstream video planning. We evaluate on manipulation tasks that are unseen during calibration.

# E Hyperparameters and implementation details

We provide implementation details for DOLA and every baseline used in our experiments. All methods follow the benchmark-specific training and evaluation protocols described in Section 3, and all methods produce a 128-dimensional latent action for the linear probe and downstream policy across all benchmarks. Method-specific quantities are reported per baseline below.

## E.1 Latent action model training

### E.1.1 **DOLA** (OURS)

**DOLA** consists of a ResNet-34 (He et al., 2016) visual encoder $f_\theta$, a bidirectional Transformer inverse-dynamics model that maps differences between visual embeddings to latent actions $z_t$, and a causal forward-dynamics Transformer. We jointly train a linear action head using the available ground-truth action supervision to align the latent space with the environment action space. The total loss is the sum of the forward-dynamics prediction loss, an off-diagonal covariance regularizer (Bardes et al., 2021; Cui et al., 2024) on the encoder features, and the action-head loss on the labeled subset.

Table 5: Environment-dependent hyperparameters for DOLA pretraining.

| | PUSH-T | LIBERO 90/10 | MIMICGEN | VIZDOOM |
|---|---|---|---|---|
| Encoder init | Random | Random | Random | Random |
| Latent action dim | 128 | 128 | 128 | 128 |
| SSL target | MSE | MSE | MSE | MSE |
| Batch size | 64 | 64 | 64 | 64 |
| Learning rate | 1e−4 | 1e−4 | 1e−4 | 2e−4 |
| FDM depth / heads / width | 8/8/512 | 8/8/512 | 8/8/512 | 6/6/120 |
| FDM dropout | 0 | 0 | 0 | 0.3 |

Table 6: Shared hyperparameters for DOLA pretraining.

| | |
|---|---|
| Optimizer / betas / clip | AdamW / $(0.9, 0.999)$ / 0.5 |
| IDM depth / heads / width | 8 / 8 / 512 |
| Covariance regularization coef. | 0.01 |
| Observation window $h$ | 8 |
| LR schedule | cosine + warmup |
| Number of epochs / warmup epochs | 40 / 5 |
| Weight decay | 0 |

### E.1.2 LINEAR LAM

Linear LAM (Zhang et al., 2025) does not have an official open-source implementation, so we re-implemented it directly from the paper. Following the paper's formulation, we use a bias-free linear IDM ($z = C\,\phi(o) + D\,\phi(o')$) and FDM ($\hat{\phi}(o') = A\,\phi(o) + B\,z$) over an ImageNet-initialized ResNet-34 feature $\phi(o)$ (with multi-view observations concatenated into a single feature vector), together with the linear auxiliary action head $\hat{a} = Ez$ with split-aware loss weight $\lambda = |\mathcal{D}_a|/|\mathcal{D}|$.

Table 7: Shared hyperparameters for Linear LAM.

| | |
|---|---|
| Encoder backbone | ResNet-34 |
| Latent action dim $d_z$ | 128 |
| Optimizer / LR / weight decay / betas | AdamW / 1e−3 / 0 / $(0.9, 0.999)$ |
| Gradient clip norm | 1.0 |
| Number of epochs | 40 |
| Frames per training sample / frame stride | 2 / 1 |

### E.1.3 LAPA

We use the official LAPA implementation (https://github.com/LatentActionPretraining/LAPA) from Ye et al. (2024) and adapt only the data loaders and probe head to match our protocol. The model is a ViViT-style spatiotemporal encoder–decoder with NSVQ (Vali & Bäckström, 2022) codebook quantization, producing a length-$L_z$ sequence of discrete codes per transition; the latent action passed to the probe and downstream policy is the concatenated post-quantization codebook embedding, of dimension $L_z \cdot d_q = 4 \cdot 32 = 128$ across all environments.

### E.1.4 ADAWORLD

We use the official AdaWorld implementation (https://github.com/Little-Podi/AdaWorld) from Gao et al. (2025), keeping the spatiotemporal transformer encoder–decoder and the KL-regularized continuous latent unchanged. The encoder produces one continuous latent per transition that is consumed directly as the 128-dim latent action by the probe and the downstream policy.

Table 8: Hyperparameters for LAPA across all four benchmarks.

| | |
|---|---|
| Token dim $d$ / quant dim $d_q$ | 1024 / 32 |
| Codebook size | 8 |
| Code sequence length $L_z$ | 4 |
| Patch size / image size | 32 / 224 |
| Spatial / temporal transformer depth | 8 / 8 |
| Heads / per-head dim | 16 / 64 |
| Latent action dim ($L_z \cdot d_q$) | 128 |
| Optimizer / LR / weight decay / betas (LAM) | Adam / 1e−4 / 0 / $(0.9, 0.99)$ |
| Pretraining-time action head optim / LR / wd / betas | AdamW / 1e−4 / 1e−6 / $(0.9, 0.999)$ |
| Gradient clip norm | 0.5 |
| Batch size / frame offset | 64 / 1 |
| Number of epochs | 40 |

Table 9: Hyperparameters for AdaWorld across all four benchmarks.

| | |
|---|---|
| Model dim | 1024 |
| Latent action dim | 128 |
| Patch size | 16 |
| Encoder / decoder blocks | 16 / 16 |
| Number of heads | 16 |
| KL coefficient $\beta$ | 2e−4 |
| Optimizer / LR / weight decay / betas (LAM) | AdamW / 2.5e−5 / 1e−2 / $(0.9, 0.999)$ |
| Action head: hidden width / depth | 64 / 0 |
| Action head optimizer / LR / weight decay | AdamW / 1e−4 / 1e−6 |
| Gradient clip norm | 0.3 |
| Number of epochs | 40 |

#### E.1.5 DreamDojo LAM

We use the LAM component from the official DreamDojo release (https://github.com/NVIDIA/DreamDojo, Gao et al., 2026), retaining the deeper (24-block) encoder–decoder transformer; the surrounding world-model is not used in our experiments. The continuous latent action is consumed directly as the 128-dim interface to the probe and downstream policy.

Table 10: Hyperparameters for DreamDojo LAM across all four benchmarks.

| | |
|---|---|
| Model dim | 1024 |
| Latent action dim | 128 |
| Patch size | 16 |
| Encoder / decoder blocks | 24 / 24 |
| Number of heads | 16 |
| KL coefficient $\beta$ | 1e−6 |
| LAM dropout | 0.0 |
| Optimizer / LR / weight decay / betas (LAM) | AdamW / 2.5e−5 / 1e−2 / $(0.9, 0.999)$ |
| Action head: hidden width / depth | 64 / 0 |
| Action head optimizer / LR / weight decay | AdamW / 1e−4 / 1e−6 |
| Action loss weight | 1.0 |
| frame stride | 1 |
| Gradient clip norm | 0.3 |
| Number of epochs | 40 |

### E.2 Downstream latent policy

For all methods, we train the same downstream behavior cloning policy directly in the learned latent action space. We infer latent actions $z_t$ using the frozen encoder and latent IDM, and refit the linear action head on the same ground-truth actions with ordinary least squares for downstream evaluation.

We train a VQ-BeT policy (Lee et al., 2024) to predict the inferred latent actions for all baselines and **DOLA**, using identical hyperparameters across methods for each environment.

Table 11: Environment-dependent hyperparameters for the downstream VQ-BeT latent policy.

| Environment | Obs win. | Act. chunk | Batch | Epochs | VQ batch | Trans. (L/H/D) |
|---|---|---|---|---|---|---|
| PUSHT | 5 | 5 | 512 | 5000 | 1024 | 8/8/512 |
| LIBERO 90/10 | 10 | 1 | 64 | 5000 | 2048 | 6/6/120 |
| MIMICGEN | 5 | 1 | 256 | 200 | 1024 | 8/8/512 |
| VIZDOOM | 5 | 5 | 512 | 5000 | 1024 | 8/8/512 |

Table 12: Shared hyperparameters for the downstream VQ-BeT latent policy.

| | |
|---|---|
| Optimizer | AdamW |
| Learning rate | $5.5\text{e}{-5}$ |
| Weight decay | $2\text{e}{-4}$ |
| Betas | $(0.9, 0.999)$ |
| VQ-VAE latent dim | 512 |
| VQ-VAE codebook size $n_{\text{embed}}$ | 16 |
| VQ-VAE residual groups | 2 |
| VQ iterations | 1000 (PUSHT, MIMICGEN, VIZDOOM) / 300 (LIBERO 90/10) |
| Offset loss multiplier | 100 (LIBERO 90/10) / 10 (PUSHT, MIMICGEN, VIZDOOM) |
| Latent action input dim to BeT | 128 for all baselines and **DOLA** |

### E.3 COMPUTE RESOURCES

All LAM pretraining and runs were conducted on NVIDIA H200 GPUs.

## F WEBSSL LATENT POLICY RESULTS

To assess sensitivity to the visual representation, we repeat the latent-policy evaluation from Table 3 using WebSSL CLS features, following Patch Policy (Zhou et al., 2026), in place of the DynaMo features for all methods and environments. We observe similar rankings between baselines across encoder choices.

Table 13: Latent policy performance (↑) across environments with WebSSL CLS features.

| Method | Push-T | LIBERO 90/10 | MimicGen Stack | VizDoom |
|---|---|---|---|---|
| Oracle | $0.47 \pm 0.01$ | $0.97 \pm 0.01$ | $0.25 \pm 0.03$ | $1.00 \pm 0.01$ |
| Linear LAM | $0.09 \pm 0.01$ | $0.32 \pm 0.03$ | $0.09 \pm 0.01$ | $\mathbf{0.96 \pm 0.02}$ |
| LAPA | $0.07 \pm 0.01$ | $0.04 \pm 0.05$ | $0.04 \pm 0.02$ | $0.85 \pm 0.13$ |
| AdaWorld | $0.12 \pm 0.04$ | $0.72 \pm 0.02$ | $0.01 \pm 0.01$ | $0.24 \pm 0.05$ |
| DreamDojo | $\mathbf{0.40 \pm 0.01}$ | $0.82 \pm 0.03$ | $0.11 \pm 0.02$ | $\mathbf{0.95 \pm 0.03}$ |
| **DOLA** (Ours) | $\mathbf{0.42 \pm 0.02}$ | $\mathbf{0.90 \pm 0.01}$ | $\mathbf{0.27 \pm 0.02}$ | $\mathbf{0.96 \pm 0.02}$ |

## G ADDITIONAL CROSS-EMBODIMENT OPEN-LOOP EPISODES

As a supplement to the main-text result (Fig. 4), Fig. 7 shows four additional Panda→UR5e open-loop transfer episodes: two successes (from different source demos) and two failures covering the qualitatively distinct failure modes we observe. **No Grasp**: the UR5e tracks the reach but closes its gripper too early or misaligned and never lifts cube A. **No Stack**: cube A is lifted but accumulated open-loop drift causes the place to miss cube B. Even in these failures, the overall motion semantics decoded from the Panda latents are preserved on the UR5e: the robot reaches toward cube A and proceeds toward cube B in the right order, so the breakdown is localized to the contact-rich phases (grasping, placing) where compounding open-loop error is most visible.

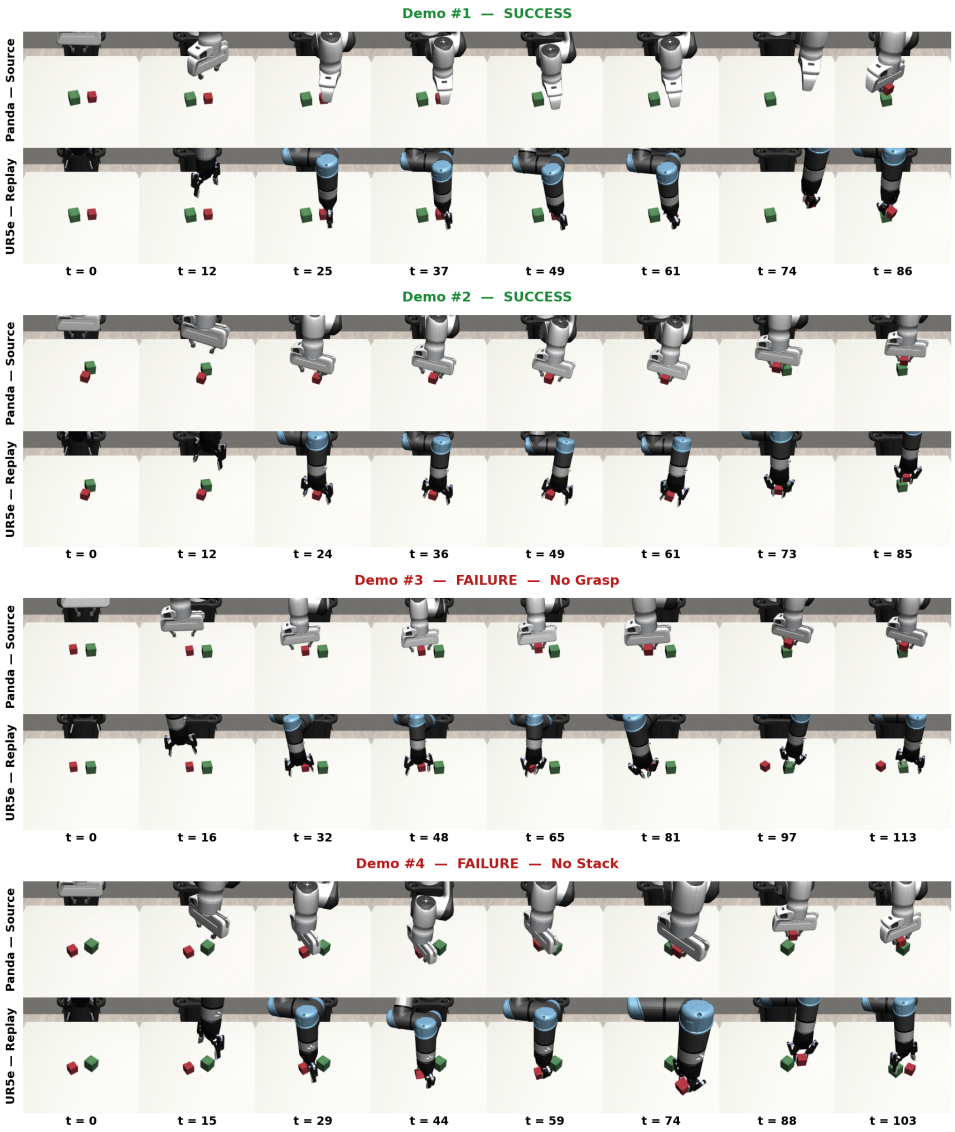



Figure 7: Additional Panda→UR5e open-loop transfer episodes. Each block: top row is the source Panda video, bottom row is the UR5e env executing actions decoded from the Panda latents through the UR5e head; columns are time-aligned. Top two: successes. Bottom two: *No Grasp* and *No Stack* failure modes.

# H Real-robot calibration and additional video-planning results

## H.1 Calibration dataset and background augmentation

**Calibration dataset.** We use 285 blue-screen demonstrations and 222 real-scene demonstrations for real-robot calibration. These clips are task-agnostic: they are collected without target objects or task instructions. The top two rows of Fig. 8 show examples of both types of calibration clips.

**Background augmentation.** We use background augmentation to account for appearance differences between calibration and deployment (Fig. 8). For the 285 blue-screen demonstrations, we replace the background using precomputed robot masks for each frame. We construct a bank of 196 backgrounds from seven arm-free images recovered from the 222 real-scene demonstrations. To recover these images, we take the mode of each pixel's color histogram over time, then fill the

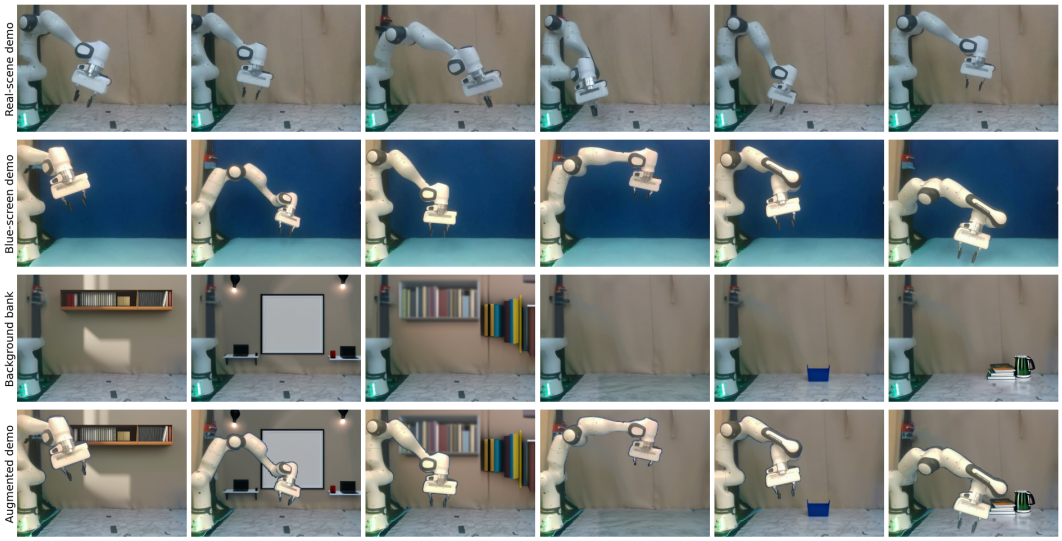



Figure 8: Calibration data and background augmentation on the real Franka. Top row: task-agnostic real-scene demonstrations, used unmodified. Second row: blue-screen demonstrations with per-frame robot masks. Third row: backgrounds generated by applying SDXL inpainting to arm-free images of the real scene, with backdrop and tabletop variations or added distractor objects. Bottom row: blue-screen frames composited over backgrounds from the bank.

remaining static arm region with the nearest clean pixels. Using SDXL inpainting, we create 112 variations of the backdrop and tabletop and another 84 images containing one to three distractor objects at random table locations. Pixels outside the inpainting masks remain unchanged. The third row of Fig. 8 shows examples from the bank.

For each training window, we uniformly sample a background $b$ from the training portion of the bank and composite each frame as

$$\tilde{x}_t = m_t \odot x_t + (1 - m_t) \odot b,$$

where $x_t$ is the original frame and $m_t$ is its robot mask. The bottom row of Fig. 8 shows the resulting composites. We keep the background fixed within each window and sample a new one each time the window is used during training. We train on both the augmented blue-screen demonstrations and the original real-scene demonstrations, using the same action-normalization statistics. We hold out 15% of the background images for validation and assign each validation window a fixed background from this set, allowing evaluation on backgrounds not seen during training.

### H.2 Additional video-planning episodes

Figure 9 supplements the main-text results in Fig. 6 with an additional successful execution, an inverse dynamics error, and two examples of unrealistic generated motion.

**Successful execution.** In Fig. 9(a), the real Franka successfully picks up the bread and places it on the plate in open loop, using actions decoded from the generated video's latent actions.

**Inverse dynamics error.** In Fig. 9(b), the generated video depicts a successful bear pickup, but the real robot misses the grasp and retracts without lifting the bear. This example illustrates a discrepancy between the generated plan and the motion produced by the decoded actions.

**Unrealistic generated motion.** Figure 9(c) shows physically implausible generated motion for cup stacking and oven closing. Because these plans were visibly implausible, we did not execute them on the real robot and show only the generated videos.

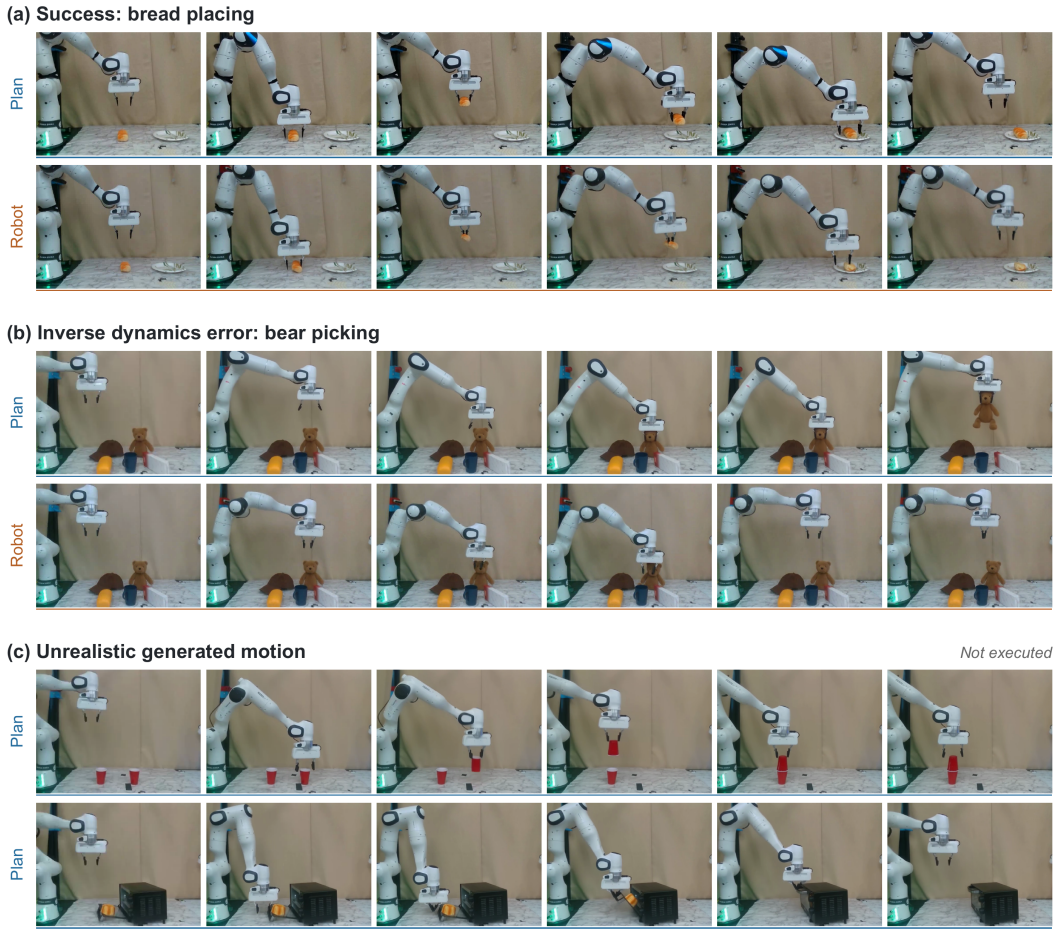



Figure 9: Additional real-Franka video-planning examples, with six frames per row. (a) Successful bread placement. (b) Inverse dynamics error during bear picking. In (a) and (b), *Plan* shows the generated source video and *Robot* shows the real Franka executing actions decoded from its latents. Paired rows are sampled at corresponding relative times within each clip. (c) Unrealistic generated motion for cup stacking (top) and oven closing (bottom); neither plan was executed on the real robot.