DEMO-JEPA: JOINT-EMBEDDING PREDICTIVE ARCHITECTURE FOR ONE-SHOT CROSS-EMBODIMENT IMITATION

Anonymous authors
Paper under double-blind review

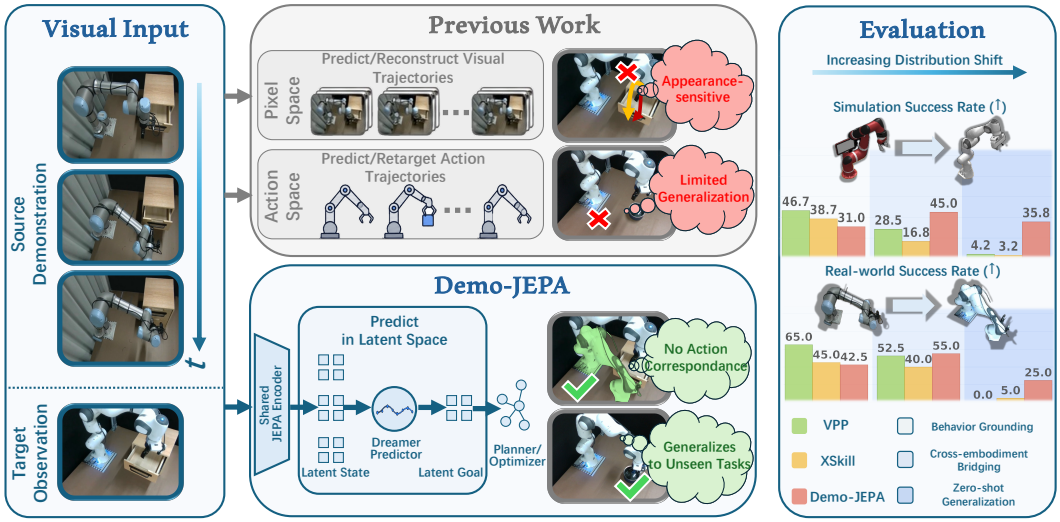

Figure 1: **Overview of Demo-JEPA.** Demo-JEPA performs cross-embodiment imitation in a JEPA latent space, where the Dreamer Predictor infers target-compatible goals from source demonstrations for planning. We evaluate it across three increasingly shifted suites: *Behavior Grounding* for seen tasks, *Cross-Embodiment Bridging* for unseen actions, and *Zero-Shot Generalization* for unseen configurations. Demo-JEPA achieves larger gains as distribution shift increases, showing the robustness of latent goal inference for cross-embodiment transfer.

ABSTRACT

Cross-embodiment imitation learning is often treated as reproducing demonstrated actions, but actions are inherently embodiment-specific. Differences in morphology, kinematics, and joint constraints can cause the same trajectory to produce different outcomes across robots. More critically, differences in reachable workspace or scene configuration may require the target robot to adopt a different execution strategy. Methods that reproduce or retarget the source robot's execution are therefore constrained by how the source robot performs the task. We instead view demonstrations as specifications of what should be achieved: the target robot infers the intended future state and generates actions end-to-end under its own dynamics.

We propose **Demo-JEPA**, a cross-embodiment imitation framework that decouples demonstration intent from embodiment-specific execution. Built on a JEPA-based world model, Demo-JEPA translates source visual demonstrations into target-compatible future latent trajectories in a shared predictive representation space. The target robot then uses these latent trajectories as subgoals and optimizes actions toward them in its own action space. Because Demo-JEPA avoids action-level correspondence and requires only visual demonstrations together with the target robot's own interaction experience, it supports flexible imitation across heterogeneous embodiments. Experiments on RLBench and real-world manipulation tasks show that Demo-JEPA matches specialized in-domain methods and generalizes to unseen tasks and embodiment configurations where prior methods fail.

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.

1 INTRODUCTION

In robotic imitation learning, a central bottleneck is the embodiment gap: the physical and kinematic mismatch between the demonstrator and the learner Torabi et al. (2018); Hussein et al. (2017). Humans can often infer the intent of an observed behavior and adapt it to their own body, but robots typically struggle to translate demonstrations from another embodiment, such as a human or a robot with different degrees of freedom, into their own control space.

Many existing approaches bridge this gap through action-level alignment. Vision-Language-Action models Brohan et al. (2022; 2024); Black et al. (2026); Intelligence et al. (2025) and recent cross-embodiment imitation methods Zakka et al. (2021); Doshi et al. (2024); Fan et al. (2025); Bauer et al. (2025); Li et al. (2025) often rely on large-scale co-training across embodiments, shared or aligned action spaces, explicit action correspondences, or heuristic retargeting. While effective in certain settings, this action-centric paradigm becomes fundamentally limited for heterogeneous robots: the same semantic behavior, such as grasping an object, may correspond to very different motor commands, torque profiles, and joint configurations across embodiments. As a result, learning cross-embodiment skills through low-level action alignment can require substantial multi-embodiment data while still struggling to generalize across mismatched morphologies.

In this work, we decouple the intent of a demonstration from its execution. Rather than asking how an action should be reproduced, we ask what future state the demonstration is trying to realize. Under this objective-centric view, a visual demonstration is not treated as a sequence of motor primitives, but as an implicit specification of a desired temporal outcome. We therefore formulate cross-embodiment imitation as latent goal-conditioned planning in a shared predictive representation space Zakka et al. (2021); Finn & Levine (2017); Liu et al. (2022). Given a source visual demonstration, our framework infers embodiment-compatible future latent trajectories that serve as subgoals. The target robot then realizes these subgoals by planning under its own forward dynamics.

To instantiate this formulation, we introduce **Demo-JEPA**, a novel framework grounded in Joint Embedding Predictive Architecture (JEPA) LeCun et al. (2022); Assran et al. (2023); Bardes et al. (2024); Assran et al. (2025); Mur-Labadia et al. (2026), as shown in Figure 1. The key design choice in Demo-JEPA is to perform cross-embodiment imitation in a predictive latent space rather than in pixel or action space. Unlike pixel-level generative models Goodfellow et al. (2014); Esser et al. (2021); Ho et al. (2020); Song et al. (2022); Lipman et al. (2023) or standard reconstructive autoencoders Kingma & Welling (2022); Vincent et al. (2008); He et al. (2021), which may allocate substantial representational capacity to task-irrelevant details such as background textures, lighting, or embodiment-specific hardware appearance, JEPA latent spaces are optimized to capture abstract, predictive world structure. This property makes JEPA well-suited for cross-embodiment transfer, where the goal is to preserve high-level behavioral intent while abstracting away low-level perceptual and morphological differences.

At its core, Demo-JEPA features an embodiment-aware *Dreamer Predictor* that translates visual observations from a source demonstration into target-compatible future latent states within this shared predictive space. These inferred latent goals then guide action generation through iterative planning in the learned latent dynamics, using the Cross-Entropy Method Rubinstein (1997); Chua et al. (2018); Assran et al. (2025) (CEM). Crucially, Demo-JEPA does not require action-level correspondences, shared action spaces, or manually designed retargeting rules. The framework relies only on offline visual demonstrations from the source embodiment and the target robot's own interaction experience, reducing the need for costly cross-embodiment action annotation or data engineering.

We comprehensively evaluate Demo-JEPA in both simulated environments (RLBench James et al. (2020)) and on real-world robotic manipulation tasks. Our empirical results demonstrate that Demo-JEPA not only matches the asymptotic performance of specialized, in-domain planners but also exhibits robust generalization capabilities to zero-shot tasks and radically different embodiment configurations where prior baselines persistently fail. By capturing the underlying predictive dynamics of a demonstration rather than memorizing surface-level statistics, Demo-JEPA successfully translates abstract task progression into an embodiment-compatible goal space, facilitating highly flexible cross-embodiment imitation.

2 Related Work

Cross-Embodiment Imitation via Action Alignment. The embodiment gap, driven by profound mismatches in morphology, kinematics, and low-level action spaces, persists as a primary bottleneck in transferring learned skills across diverse robots Argall et al. (2009); Taylor & Stone (2009); Torabi et al. (2018); Hussein et al. (2017). Conventional methodologies typically attempt to bridge this divide through direct action-level alignment, including explicit action retargeting and correspondence learning across embodiments Gupta et al. (2017); Devin et al. (2017). More recent approaches construct shared action spaces or learn embodiment-agnostic control representations Zheng et al. (2025); Chen et al. (2026). Recent large-scale Vision-Language-Action (VLA) architectures address this challenge by leveraging massive multi-embodiment co-training datasets to implicitly internalize cross-robot control regularities Brohan et al. (2022; 2024); Collaboration et al. (2025); Kim et al. (2024). Other frameworks focus on discovering reusable and compositional skill primitives that generalize across embodiments Xu et al. (2023). Nevertheless, these action-centric paradigms face a fundamental limitation: identical semantic intents or tasks frequently demand radically different motor commands depending on the specific embodiment. Demo-JEPA fundamentally diverges from these approaches by entirely avoiding action-level correspondences, instead formulating cross-embodiment imitation strictly as semantic intent alignment within a shared, abstract latent space.

Imitation from Observation. Learning from visual-only demonstrations aims to recover executable control policies without access to expert action labels Torabi et al. (2018); Burnwal et al. (2025); Chen et al. (2024); Xie et al. (2025); Kedia et al. (2025). A dominant paradigm is predictive modeling, where future observations serve as a surrogate supervisory signal. Methods such as Visual Foresight and Video Prediction Policy (VPP) Finn & Levine (2017); Hu et al. (2025), as well as latent world models Hafner et al. (2020), learn action-conditioned dynamics to generate future trajectories and derive control by optimizing toward desired visual outcomes. However, these approaches typically operate in pixel space, making them computationally expensive and prone to compounding errors over long horizons. More importantly, pixel-level prediction often captures irrelevant visual details rather than task-relevant structure, limiting generalization under embodiment variations. Demo-JEPA instead performs goal inference in the latent space of a Joint Embedding Predictive Architecture (JEPA) LeCun et al. (2022), suppressing nuisance visual variation and abstracting away embodiment-specific noise.

World Models and Latent Goal Planning. World models enable agents to simulate environment dynamics and plan in learned latent spaces, significantly improving efficiency and generalization Ha & Schmidhuber (2018); Ye et al. (2026). Traditional approaches often rely on pixel reconstruction objectives, which can hinder abstraction and scalability. In contrast, JEPA-based methods focus on predicting latent representations instead of reconstructing pixels, leading to more semantically meaningful and robust representations for downstream reasoning Assran et al. (2023; 2025); Mur-Labadia et al. (2026). Our framework builds upon V-JEPA 2.1 Mur-Labadia et al. (2026), which learns structured, action-conditioned latent dynamics from videos. Within this paradigm, we integrate latent representation learning with goal-conditioned planning using sampling-based optimization such as the Cross-Entropy Method (CEM) Rubinstein (1997). By inferring target-compatible latent goals from visual demonstrations, Demo-JEPA enables the robot to execute behaviors using its own dynamics model, effectively bridging the gap between high-level intent extraction and embodiment-specific control.

3 Method

We propose Demo-JEPA, a framework that formulates cross-embodiment imitation as latent goal-conditioned planning in a learned world model. We first define the problem setting, then introduce the latent dynamics predictor for embodiment-aware goal inference, and finally present the implementation details, including training objectives and the inference procedure.

3.1 Problem Formulation

We consider a cross-embodiment imitation setting with a source embodiment $\mathcal{E}^s$ and a target embodiment $\mathcal{E}^t$. The source provides a demonstration trajectory $\tau^s = \{o_k^s\}_{k=1}^T$ consisting only of observations. The target embodiment $\tau^t = \{o_k^t, s_k^t, a_k^t\}_{k=1}^T$ interacts with the environment through

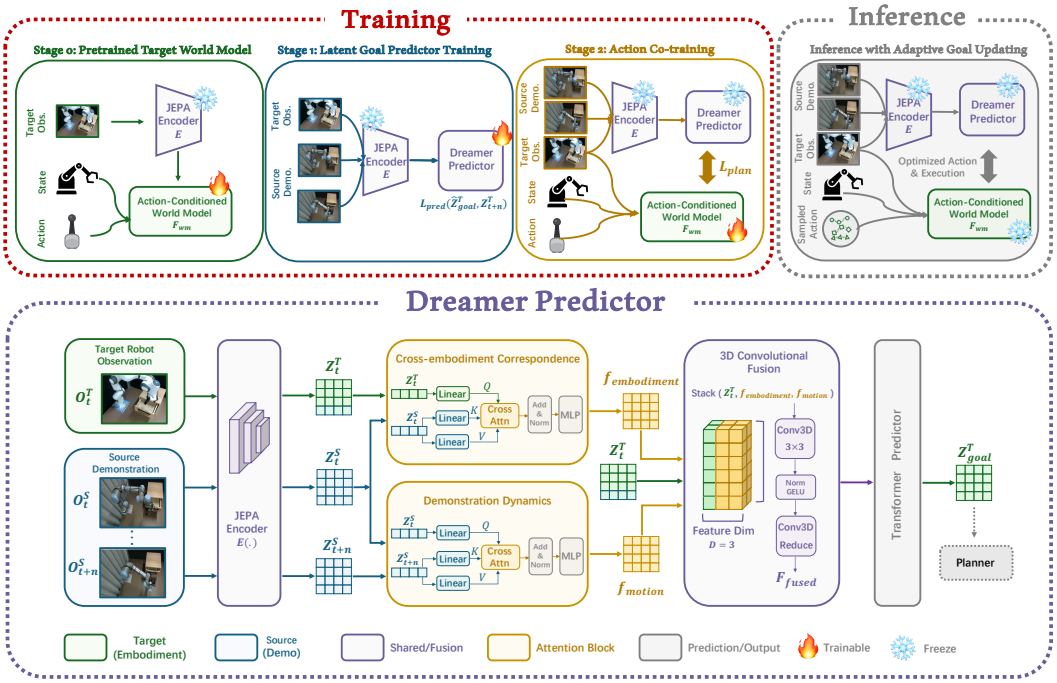

Figure 2: **Demo-JEPA training and inference pipeline.** The top panels show the overall training and inference stages, from target world-model initialization to closed-loop planning with adaptive goal updating. The bottom panel highlights the Dreamer Predictor, which uses JEPA latents, cross-attention, and 3D convolutional fusion to translate source demonstrations into target-compatible future latent goals.

observations o_k^t, states s_k^t, and actions $a_k^t \in \mathcal{A}^t$, where the two embodiments may differ in morphology, kinematics, action spaces, and visual appearance.

At each time step k, Demo-JEPA takes as input the current target observation o_k^t and a pair of source frames (o_k^s, o_{k+n}^s) sampled from τ^s, where n denotes a temporal offset aligned with the planning frequency. The model outputs a sequence of target actions $a_{k:k+H-1}^t$. Instead of directly predicting actions from demonstrations, Demo-JEPA first infers a target-compatible latent goal z_{goal}^t from the source frame pair and the current target observation, representing the intended future state implied by the demonstration. An action-conditioned world model then generates the action sequence by optimizing, via the Cross-Entropy Method, action rollouts whose predicted latent states match z_{goal}^t.

3.2 Demo-JEPA

Demo-JEPA formulates cross-embodiment imitation as latent goal-conditioned planning within a learned world model with a three-stage training procedure. Figure 2 presents the overall framework: given a source demonstration and the current target observation, the Dreamer Predictor infers an embodiment-compatible latent goal, which is then achieved through latent planning in action-conditioned world model. We now describe action-conditioned world model and dreamer predictor in detail.

Action-conditioned World Model. We adopt VJEPA2.1 Mur-Labadia et al. (2026) as the action-conditioned world model to capture the dynamics of the target embodiment in a latent space. An JEPA encoder $E(\cdot)$ maps observations to latent states $z_k = E(o_k^t)$, and a dynamics predictor $F_{wm}(\cdot)$ models transitions conditioned on robot states and actions:

$$\hat{z}_{k+1} = F_{wm}(z_k, s_k^t, a_k^t).$$

Given a latent goal z_{goal}^t, actions are obtained by optimizing a sequence $\mathbf{a}_{k:k+H-1}^t$ such that the predicted latent rollout matches the goal:

$$\mathbf{a}_{k:k+H-1}^{t*} = \arg\min_{\mathbf{a}} d\big(F_{wm}(z_k, s_k^t, \mathbf{a}),\ z_{\text{goal}}^t\big),$$

where $d(\cdot,\cdot)$ denotes a latent distance metric. We solve this optimization using the Cross-Entropy Method (CEM). See Appendix B.2 for details and Algorithm 1 for pseudocode.

Dreamer Predictor. Directly imitating source trajectories is fundamentally challenging due to differences in morphology, kinematics, and interaction dynamics across embodiments. Instead, we aim to infer a future latent state that is compatible with the target embodiment while preserving the semantic intention of the source demonstration. We introduce the **Dreamer Predictor**, an embodiment-aware latent predictor that performs cross-embodiment future-state translation in a shared JEPA representation space. At each time step k, the predictor takes as input the current target observation o_k^t and a source frame pair (o_k^s, o_{k+n}^s), where $n > 0$ denotes a temporal offset capturing future motion in the demonstration. A shared encoder maps observations into latent representations $(z_k^t, z_k^s, z_{k+n}^s)$.

The predictor jointly models two factors: (1) cross-embodiment correspondence between source and target, and (2) temporal evolution of the demonstrated behavior. We capture these via two cross-attention modules:

$$f_{\text{emb}} = \text{Attn}(Q = z_k^t,\ K = z_k^s,\ V = z_k^s), \quad f_{\text{mot}} = \text{Attn}(Q = z_{k+n}^s,\ K = z_k^s,\ V = z_k^s).$$

The resulting features preserve the spatial structure of the JEPA latent space. We obtain the fused feature representation f_{fused} by aggregating z_k^t,f_{emb} and f_{mot} through a 3D convolution ϕ:

$$f_{\text{fused}} = \phi\left(\left[\mathbf{z}_k^t \oplus f_{\text{emb}} \oplus f_{\text{mot}}\right]\right)$$

A Transforme predictor $\mathcal{T}$ then decodes the fused feature into latent goal $\hat{z}_{\text{goal}}^t = \mathcal{T}(f_{\text{fused}})$. This latent goal serves as the objective for downstream planning in the action-conditioned world model.

3.3 Implemention Details

We now describe the implementation details of Demo-JEPA, including training procedure and inference pipeline. **Stage 0** first pretrains an action-conditioned world model (not the focus of this work; details in Appendix B.1). We focus on **Stage 1** and **Stage 2**, as well as the inference pipeline.

Stage 1: Latent Goal Predictor Training. In the first stage, we train the Dreamer Predictor using action-free cross-embodiment demonstrations. The objective is to map source-embodiment transitions into target-compatible latent goals within the shared JEPA representation space. Given a target future representation z_{k+n}^t, the predictor is optimized via a latent reconstruction loss:

$$\mathcal{L}_{\text{pred}} = \left\|\hat{z}_{\text{goal}}^t - z_{k+n}^t\right\|_2^2.$$

To enhance generalization against execution errors and imperfect planning trajectories during inference, we introduce **temporal perturbation**. Instead of sampling a fixed source frame pair (o_k^s, o_{k+n}^s), we perturb the current timestamp to obtain a randomly shifted pair $(o_{k+\delta}^s, o_{k+n}^s)$, where $\delta \sim \mathcal{U}(-r, r)$ with r being the perturbation radius. This temporal regularization exposes the predictor to diverse, off-distribution latent transitions, forcing it to maintain robust goal alignments. During this stage, only the Dreamer Predictor is optimized.

Stage 2: Action Co-Training. Once the Dreamer Predictor converges, it is frozen to provide stable latent goal guidance. We then unfreeze the dynamics predictor module F_{wm} of the action-conditioned world model. The objective is to adapt the world model's latent rollouts to the goal distribution of Dreamer Predictor outputs. Given inferred goal $\hat{z}_{\text{goal}}^t$, the world model predicts future latent goals conditioned on sampled action sequences $\mathbf{a}_{k:k+n-1}^t$. We optimize F_{wm} using a planning loss:

$$\mathcal{L}_{\text{plan}} = \left\|F_{wm}(z_k^t, s_k^t, \mathbf{a}_{k:k+n-1}^t) - \hat{z}_{\text{goal}}^t\right\|_2^2.$$

This co-training stage closely aligns the action-conditioned dynamics with the cross-embodiment intentions, significantly improving downstream generation stability. See Algorithm 2 for pseudocode of training pipeline.

Inference with Adaptive Goal Updating. During inference, the source demonstration τ^s serves as a sequence of reference segments. At each interaction step, the Dreamer Predictor infers a latent goal $\hat{z}_{\text{goal}}^t$ based on the current target observation o_k^t and a source reference pair $(o_i^s, o_{i+\Delta}^s)$. Target actions are then generated via CEM optimization in the world model. Directly advancing the source reference segment at a fixed frequency often leads to compounding errors due to cross-embodiment kinematic

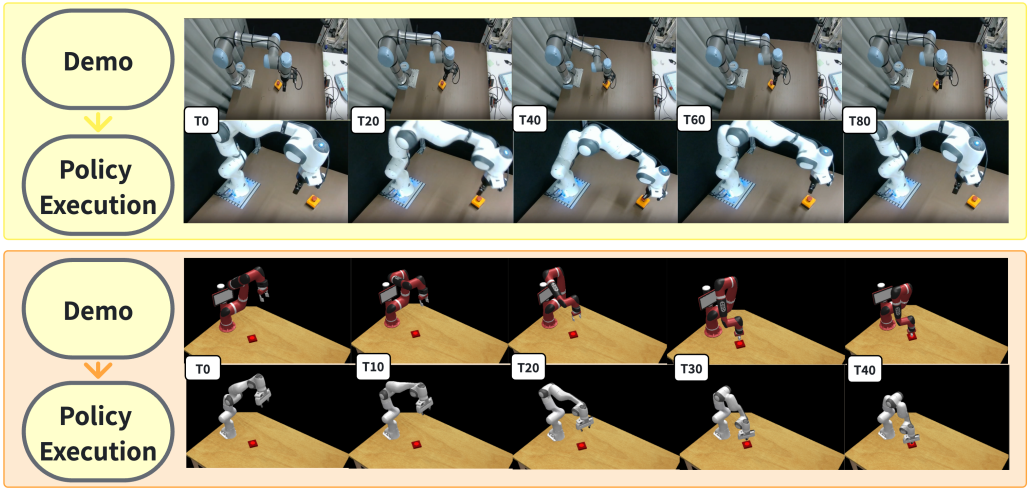

Figure 3: **From demo to policy execution.** Policy takes demo as input and executes action step by step. Top: real world deployment. Bottom: simulation deployment.

Table 1: **Pretraining data across transfer settings.** Table summarizes the training objective and data scale used in each stage. For each setting, all compared methods use the same number of tasks and trajectories, while differing in their Stage I and Stage II training objectives.

Setting	Method	Stage I			Stage II		
		Training stage	Tasks	Trajectories	Training stage	Tasks	Trajectories
Simulation	Demo-JEPA VPP XSkill	Dreamer Predictor Predictive visual representation learning Skill discovery	86	13444	Action Co-Training Action learning Skill transfer	39	8324
Real-world	Demo-JEPA VPP XSkill	Dreamer Predictor Predictive visual representation learning Skill discovery	22	4508	Action Co-Training Action learning Skill transfer	19	3903

differences. Therefore, we introduce an **adaptive goal-updating mechanism**. After executing the planned actions, we compute the latent discrepancy between the newly observed target state z^t_{k+1} and the previously intended goal $\hat{z}^t_{\text{goal}}$ as $D_k = d(z^t_{k+1}, \hat{z}^t_{\text{goal}})$. The system advances to the next source reference pair $(o^s_{i+1}, o^s_{i+1+\Delta})$ only if $D_k < \epsilon$, where ϵ is a predefined distance threshold indicating the sub-goal has been reached. Otherwise, the current goal remains active ($\hat{z}^t_{\text{goal}} \leftarrow \hat{z}^t_{\text{goal}}$), the Dreamer Predictor is bypassed for the current step, and the planner continues optimizing actions toward the unreached goal. This mechanism effectively stabilizes long-horizon tasks by preventing premature temporal advancement. Detail inference pseudocdeo can be found in Algorithm 3.

4 Experiments

We validate Demo-JEPA through cross-embodiment transfer experiments in simulation and real-world environments. Our evaluation assesses its ability to infer executable latent goals from heterogeneous demonstrations without action-level correspondence, and tests its generalization across embodiments, tasks, and domains.

Expert Demonstration. In simulation, we use RLBench James et al. (2020) to collect cross-embodiment demonstrations, acquiring video prompts from a Sawyer arm (source) to guide and evaluate a Franka arm (target) policy. For real-world experiments, we collect visual prompts using a UR5e arm to guide a physical Franka arm. As shown in Figure 3, the target policy takes the source demonstration as input and executes actions in both simulation and real-world deployments. This setup allows us to assess cross-embodiment transfer capabilities in both controlled and practical reality-gap scenarios.

Task Definitions. To evaluate diverse facets of robotic perception and control, we define six representative real-world manipulation tasks: (i) *Lift cup* requires a stable rim-grasp on a pink cup followed by vertical elevation; (ii) *Lift cube* involves the precise picking of a red cube within the operational space; (iii) *Remove plate* entails grasping a plate and relocating it onto a bamboo mat; (iv) *Press button* necessitates the targeted actuation of a small tactile button; (v) *Remove pot lid* involves lifting and clearing a lid from its container; and (vi) *Remove pepper* requires grasping and displacing



Figure 4: **Real-world Tasks.** We visualizes the progression of six real-world manipulation tasks.

a pepper to the workspace's lateral boundary. Collectively, these tasks encompass a wide array of object morphologies and contact dynamics, demanding high-precision grasping, stable manipulation, and robust spatial reasoning. Task progressions are visualized in Figure 4.

Comparative Baselines. We compare Demo-JEPA with two cross-embodiment imitation baselines: VPP Hu et al. (2025), which conditions policy learning on future-aware visual representations from a fine-tuned video prediction model, and XSkill Xu et al. (2023), which executes skill sequences inferred by matching heterogeneous videos to shared skill prototypes.

Training Detail. Action-conditioned training uses Franka execution trajectories (camera observations, robot states, and actions) collected via the RLBench motion planner in simulation and teleoperation in the real world. Dreamer Predictor training utilizes paired visual trajectories (camera observations only) between the source and target (Franka) embodiments. In simulation, temporally matched Sawyer and Franka pairs are generated via retargeted replay of end-effector pose trajectories. For real-world data, where UR5e and Franka demonstrations are independent, we extract progress-aware features using GTCC Donahue & Elhamifar (2024) and align them in the feature space for frame-level correspondence. The predictor is trained on these aligned pairs, leaving held-out prompts for evaluation where the inferred actions are executed by the Franka policy. Table 1 summarizes the two-stage training protocol: Stage I covers representation/transfer pretraining (Dreamer Predictor for Demo-JEPA, predictive visual learning for VPP, and skill discovery for XSkill), and Stage II handles target-embodiment action learning (action co-training for Demo-JEPA, action learning for VPP, and skill transfer for XSkill).

Evaluation Protocol. We evaluate the target Franka policy on held-out source prompt videos, reporting task success rates over **30** simulation and **20** real-world rollouts per scenario under varied initial conditions. Based on the data protocol (Table 1), evaluations span three suites with increasing distribution shifts: **1. Behavior grounding:** Evaluates scenarios from both stages to measure execution under familiar, fully supervised conditions. **2. Cross-embodiment bridging:** Uses Stage I scenarios to test if learned predictions support execution without direct action supervision. **3. Zero-shot generalization:** Assesses the system's ability to follow novel demonstrations under entirely unseen configurations.

4.1 Main Results

Simulation Results. Table 2 reports the simulation results across three evaluation suites. VPP performs best in behavior grounding, suggesting its advantage in in-domain trajectory learning. However, Demo-JEPA shows stronger performance as the distribution shift increases. In cross-embodiment bridging, Demo-JEPA achieves an average success rate of 0.45, outperforming VPP and XSkill by 0.17 and 0.28, respectively. In zero-shot generalization, Demo-JEPA further achieves 0.36, while VPP and XSkill only reach 0.04 and 0.03. These results indicate that the Dreamer

Table 2: **Simulation evaluation results.** We report task success rates across three simulation evaluation suites. The best result for each task or suite average is highlighted in blue, and subsequent result tables follow the same convention.

Method	Behavior Grounding				Cross-Embodiment Bridging					Zero-Shot Generalization				
	Basketball in Hoop	Change Channel	Close Box	Avg.	Push Button	Pick and Lift	Slide Block to Target	Pick up Cup	Avg.	Phone on Base	Close Laptop Lid	Put Rubbish in Bin	Close Drawer	Avg.
VPP Hu et al. (2025)	0.53	**0.27**	**0.60**	**0.47**	**0.63**	0.27	0.07	0.17	0.28	0.00	0.17	0.00	0.00	0.04
XSkill Xu et al. (2023)	**0.60**	0.13	0.43	0.39	0.53	0.07	0.00	0.07	0.17	0.03	0.00	0.00	0.10	0.03
Demo-JEPA (Ours)	0.33	0.17	0.43	0.31	0.60	**0.43**	**0.37**	**0.40**	**0.45**	**0.20**	**0.60**	**0.33**	**0.30**	**0.36**

Table 3: **Real-world evaluation results.** We report task success rates on physical robot tasks under the same three-suite evaluation protocol as in simulation.

Method	Behavior Grounding			Cross-Embodiment Bridging			Zero-Shot Generalization		
	Lift Cube	Lift Cup	Avg.	Remove Plate	Press Button	Avg.	Remove Pot Lid	Remove Pepper	Avg.
VPP Hu et al. (2025)	**0.55**	**0.75**	**0.65**	0.15	**0.90**	0.53	0.00	0.00	0.00
XSkill Xu et al. (2023)	0.40	0.50	0.45	0.20	0.60	0.40	0.00	0.10	0.05
Demo-JEPA (Ours)	0.25	0.60	0.43	**0.30**	0.80	**0.55**	**0.15**	**0.35**	**0.25**

Table 4: **Trajectory retargeting, goal reference, and component ablations in simulation.** We analyze trajectory retargeting, naive and oracle latent goals, and Demo-JEPA variants without Conv3D or temporal perturbation.

Method	Behavior Grounding				Cross-Embodiment Bridging					Zero-Shot Generalization				
	Basketball in Hoop	Change Channel	Close Box	Avg.	Push Button	Pick and Lift	Slide Block to Target	Pick up Cup	Avg.	Phone on Base	Close Laptop Lid	Put Rubbish in Bin	Close Drawer	Avg.
Trajectory Retargeting	0.80	0.73	0.57	0.70	0.73	0.67	0.83	0.80	0.76	0.50	0.70	0.70	0.73	0.66
V-JEPA 2.1 (Naive)	–	–	–	–	–	–	–	–	–	–	–	–	–	–
V-JEPA 2.1 (Oracle)	0.43	0.10	0.50	0.34	0.83	0.50	0.27	0.60	0.55	0.33	0.60	0.50	0.23	0.42
Demo-JEPA	**0.33**	**0.17**	**0.43**	**0.31**	0.60	**0.43**	**0.37**	**0.40**	**0.45**	0.20	**0.60**	0.33	**0.30**	**0.36**
Demo-JEPA (w/o Conv3D)	**0.33**	0.03	0.27	0.21	0.60	**0.43**	0.33	**0.40**	0.44	**0.23**	0.23	**0.40**	**0.30**	0.29
Demo-JEPA (w/o Temp. Perturb.)	0.20	0.07	0.40	0.22	**0.67**	0.17	0.27	0.23	0.33	0.13	0.40	0.27	0.27	0.27

Table 5: **Trajectory retargeting, goal reference, and architecture ablation in the real world.** We analyze trajectory retargeting, naive and oracle latent goals, and the Demo-JEPA variant without Conv3D on real-world tasks.

Method	Behavior Grounding			Cross-Embodiment Bridging			Zero-Shot Generalization		
	Lift Cube	Lift Cup	Avg.	Remove Plate	Press Button	Avg.	Remove Pot Lid	Remove Pepper	Avg.
Trajectory Retargeting	0.00	0.00	0.00	0.00	0.50	0.25	0.00	0.00	0.00
V-JEPA 2.1 (Naive)	–	–	–	–	–	–	–	–	–
V-JEPA 2.1 (Oracle)	0.35	0.75	0.55	0.40	0.75	0.58	0.15	0.40	0.28
Demo-JEPA	**0.25**	**0.60**	**0.43**	**0.30**	**0.80**	**0.55**	0.15	**0.35**	**0.25**
Demo-JEPA (w/o Conv3D)	0.20	0.50	0.35	0.10	0.70	0.40	**0.20**	0.25	0.23

Predictor can infer target-compatible latent goals beyond direct action supervision, enabling more robust cross-embodiment transfer under unseen task configurations.

Real-World Results. In the real-world experiments, VPP also performs best in behavior grounding, where task-specific trajectory regularities are closer to the training distribution. Under stronger distribution shifts, Demo-JEPA becomes more effective. It achieves the best average success rate of 0.55 in cross-embodiment bridging and reaches 0.25 in zero-shot generalization. These findings validate our hypothesis: interpreting cross-embodiment demonstrations as latent goals is more effective than using them as fixed action sequences or skill prototypes. By mapping UR5e prompts to Franka-compatible latent targets, the Dreamer Predictor successfully bridges the embodiment gap.

4.2 Analysis

Trajectory Retargeting Analysis. To analyze the limitations of transferring source execution across embodiments, we evaluate trajectory retargeting by explicitly mapping source trajectories to the target robot. Because this setting uses source trajectory information unavailable to Demo-JEPA, it serves as an analysis rather than a direct baseline. In simulation, retargeting achieves only 70.6% average success despite matched scenes and similar execution strategies (Table 4), showing that embodiment differences alone can make direct trajectory transfer unreliable. In the real world, where demonstrations are collected independently and require different execution strategies, retargeting fails on five of six tasks (Table 5). These results show that source execution cannot be assumed transferable, motivating Demo-JEPA to infer compatible goals and generate actions on the target side.

Table 6: **Imitation learning extension in simulation.** We report simulation success rates for standard DP, Demo-DP, and the original Demo-JEPA.

Method	Behavior Grounding				Cross-Embodiment Bridging					Zero-Shot Generalization				
	Basketball in Hoop	Change Channel	Close Box	Avg.	Push Button	Pick and Lift	Slide Block to Target	Pick up Cup	Avg.	Phone on Base	Close Laptop Lid	Put Rubbish in Bin	Close Drawer	Avg.
DP	0.17	0.33	0.20	0.23	0.67	–	–	–	–	–	–	–	–	–
Demo-DP	**0.33**	**0.37**	0.13	0.28	**0.83**	0.33	0.30	0.30	0.44	0.10	0.40	0.07	0.23	0.18
Demo-JEPA (Ours)	**0.33**	0.17	**0.43**	**0.31**	0.60	**0.43**	**0.37**	**0.40**	**0.45**	**0.20**	**0.60**	**0.33**	**0.30**	**0.36**

Table 7: **Imitation learning extension in the real world.** We report real-world success rates for standard DP, Demo-DP, and the original Demo-JEPA.

Method	Behavior Grounding			Cross-Embodiment Bridging			Zero-Shot Generalization		
	Lift Cube	Lift Cup	Avg.	Remove Plate	Press Button	Avg.	Remove Pot Lid	Remove Pepper	Avg.
DP	0.45	0.60	0.53	–	–	–	–	–	–
Demo-DP	**0.50**	**0.80**	**0.65**	**0.50**	**0.95**	**0.73**	0.10	0.20	0.15
Demo-JEPA (Ours)	0.25	0.60	0.43	0.30	0.80	0.55	**0.15**	**0.35**	**0.25**

Goal Reference Comparison with V-JEPA 2.1. We compare Demo-JEPA with two V-JEPA 2.1 Mur-Labadia et al. (2026) references that use different planning goals. The naive reference directly uses the future latent state from the source demonstration, while the oracle uses target-embodiment ground-truth future trajectories that are unavailable at deployment. This comparison tests whether V-JEPA 2.1 latents are directly transferable across embodiments and how closely Demo-JEPA can approach privileged target-future planning. Tables 4 and 5 show that the naive reference fails across all tasks, indicating that V-JEPA 2.1 alone does not provide cross-embodiment goal compatibility. By contrast, Demo-JEPA closely approaches the oracle using only source demonstrations and current target observations. These results show that the Dreamer Predictor converts heterogeneous demonstrations into target-compatible latent goals for planning.

Ablations. We ablate the Conv3D-based fusion module by replacing it with mean pooling over stacked latent features (Tables 4 and 5). In simulation, Conv3D's impact correlates with motion complexity. While mean pooling suffices for simpler tasks (*Basketball in Hoop*), Conv3D significantly improves performance on tasks requiring structured, non-trivial motions like twisting (*Change Channel*) or articulated closure (*Close Box*), which defeat permutation-invariant averaging. This advantage amplifies in noisy real-world settings with larger embodiment discrepancies; coordination-heavy tasks like *Remove Plate* degrade consistently without Conv3D. Overall, while not strictly required for simple tasks, Conv3D's explicit spatiotemporal modeling is crucial for preserving the structured temporal information needed for complex, articulated, or real-world behaviors. We further ablate temporal perturbation in simulation (Table 4). Without it, cross-embodiment bridging and zero-shot generalization drop from 0.45 to 0.33 and from 0.36 to 0.27, respectively, with performance decreasing on 10 of 11 tasks. This indicates that temporal perturbation improves robustness to imperfect source-target progress alignment.

Imitation Learning Extension. Finally, we test the Dreamer Predictor's utility in standard imitation learning. In **Demo-DP**, the predicted target future representation conditions a Diffusion Policy (DP) head Chi et al. (2025), explicitly providing demonstration-derived future states instead of relying solely on current observations. Tables 6 and 7 show Demo-DP consistently improves over standard DP in both simulation and real-world settings, proving the predictive latent goals are strong conditioning signals. Furthermore, Demo-DP outperforms VPP and XSkill in zero-shot generalization. Comparing Demo-DP with the original Demo-JEPA isolates the impact of the execution module. Demo-DP excels in behavior grounding, as the diffusion policy acts as a strong local action expert under familiar conditions. Conversely, Demo-JEPA is superior in zero-shot generalization, highlighting that while DP is powerful in-domain, planner-based execution remains far more robust under the severe distribution shifts of unseen tasks.

5 Conclusion

We present **Demo-JEPA**, a framework that reframes cross-embodiment imitation as **latent goal planning**. By decoupling task intent from embodiment-specific execution, Demo-JEPA infers target-compatible future goals from source robot demonstrations while generating actions entirely in the target robot's own action space. Experiments in simulation and the real world show that this formulation is particularly effective under cross-embodiment and zero-shot settings with increasing distribution shift. Our results highlight latent goal prediction as a promising alternative to directly transferring source robot execution. Remaining limitations include the fidelity of action-conditioned world models and the need for temporal alignment during training, which we leave for future work.

AI USE STATEMENT

In this work, we used generative AI tools to assist with language editing, manuscript restructuring, and LaTeX formatting. Generative AI was not used to generate experimental data, conduct experiments, or independently determine scientific claims or conclusions. All AI-assisted text and code were reviewed, revised, and verified by the authors for correctness and consistency with the experimental results. The authors take full responsibility for the final content of this work, including any text or artifacts produced with the aid of generative AI.

REFERENCES

Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.

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.

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.

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.

Erik Bauer, Elvis Nava, and Robert K Katzschmann. Latent action diffusion for cross-embodiment manipulation. *arXiv preprint arXiv:2506.14608*, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control, 2026. URL https://arxiv.org/abs/2410.24164.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, 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.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control, 2023. *URL https://arxiv. org/abs/2307.15818*, 1:2, 2024.

Returaj Burnwal, Hriday Mehta, Nirav Pravinbhai Bhatt, and Balaraman Ravindran. Learning from observation: A survey of recent advances, 2025. URL https://arxiv.org/abs/2509.19379.

Boyu Chen, Yi Chen, Lu Qiu, Jerry Bai, Yuying Ge, and Yixiao Ge. Unit: Toward a unified physical language for human-to-humanoid policy learning and world modeling, 2026. URL https://arxiv.org/abs/2604.19734.

Lawrence Yunliang Chen, Kush Hari, Karthik Dharmarajan, Chenfeng Xu, Quan Vuong, and Ken Goldberg. Mirage: Cross-embodiment zero-shot policy transfer with cross-painting, 2024. URL https://arxiv.org/abs/2402.19249.

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

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in neural information processing systems*, 31, 2018.

Embodiment Collaboration, Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Andrey Kolobov, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Felipe Vieira Frujeri, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guangwen Yang, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homanga Bharadhwaj, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jay Vakil, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi "Jim" Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Muhammad Zubair Irshad, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R Sanketi, Patrick "Tree" Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Martín-Martín, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham Sonawani, Shubham Tulsiani, Shuran Song, Sichun Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vikash Kumar, Vincent Vanhoucke, Vitor Guizilini, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiangyu Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yansong Pang, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yongqiang Dou, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, Zipeng Fu, and Zipeng Lin. Open x-embodiment: Robotic learning datasets and rt-x models, 2025. URL https://arxiv.org/abs/2310.08864.

Coline Devin, Abhishek Gupta, Trevor Darrell, Pieter Abbeel, and Sergey Levine. Learning modular neural network policies for multi-task and multi-robot transfer. In *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 2169–2176. IEEE, 2017.

Gerard Donahue and Ehsan Elhamifar. Learning to predict activity progress by self-supervised video alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern*

Recognition, pp. 18667–18677, 2024.

Ria Doshi, Homer Walke, Oier Mees, Sudeep Dasari, and Sergey Levine. Scaling cross-embodied learning: One policy for manipulation, navigation, locomotion and aviation. *arXiv preprint arXiv:2408.11812*, 2024.

Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis, 2021. URL https://arxiv.org/abs/2012.09841.

Shichao Fan, Kun Wu, Zhengping Che, Xinhua Wang, Di Wu, Fei Liao, Ning Liu, Yixue Zhang, Zhen Zhao, Zhiyuan Xu, et al. Xr-1: Towards versatile vision-language-action models via learning unified vision-motion representations. *arXiv preprint arXiv:2511.02776*, 2025.

Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion, 2017. URL https://arxiv.org/abs/1610.00696.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. URL https://arxiv.org/abs/1406.2661.

Abhishek Gupta, Coline Devin, YuXuan Liu, Pieter Abbeel, and Sergey Levine. Learning invariant feature spaces to transfer skills with reinforcement learning. *arXiv preprint arXiv:1703.02949*, 2017.

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

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination, 2020. URL https://arxiv.org/abs/1912.01603.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners, 2021. URL https://arxiv.org/abs/2111.06377.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. URL https://arxiv.org/abs/2006.11239.

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, 2025. URL https://arxiv.org/abs/2412.14803.

Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):1–35, 2017.

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

Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.

Kushal Kedia, Prithwish Dan, Angela Chao, Maximus Adrian Pace, and Sanjiban Choudhury. One-shot imitation under mismatched execution, 2025. URL https://arxiv.org/abs/2409.06615.

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.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022. URL https://arxiv.org/abs/1312.6114.

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

Haoyun Li, Ivan Zhang, Runqi Ouyang, Xiaofeng Wang, Zheng Zhu, Zhiqin Yang, Zhentao Zhang, Boyuan Wang, Chaojun Ni, Wenkang Qin, et al. Mimicdreamer: Aligning human and robot demonstrations for scalable vla training. *arXiv preprint arXiv:2509.22199*, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023. URL https://arxiv.org/abs/2210.02747.

Minghuan Liu, Menghui Zhu, and Weinan Zhang. Goal-conditioned reinforcement learning: Problems and solutions, 2022. URL https://arxiv.org/abs/2201.08299.

Lorenzo Mur-Labadia, Matthew Muckley, Amir Bar, Mido Assran, Koustuv Sinha, Mike Rabbat, Yann LeCun, Nicolas Ballas, and Adrien Bardes. V-jepa 2.1: Unlocking dense features in video self-supervised learning, 2026. URL https://arxiv.org/abs/2603.14482.

Reuven Y Rubinstein. Optimization of computer simulation models with rare events. *European Journal of Operational Research*, 99(1):89–112, 1997.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2022. URL https://arxiv.org/abs/2010.02502.

Matthew E Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(7), 2009.

Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. *arXiv preprint arXiv:1805.01954*, 2018.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pp. 1096–1103, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605582054. doi: 10.1145/1390156.1390294. URL https://doi.org/10.1145/1390156.1390294.

Amber Xie, Oleh Rybkin, Dorsa Sadigh, and Chelsea Finn. Latent diffusion planning for imitation learning, 2025. URL https://arxiv.org/abs/2504.16925.

Mengda Xu, Zhenjia Xu, Cheng Chi, Manuela Veloso, and Shuran Song. Xskill: Cross embodiment skill discovery, 2023. URL https://arxiv.org/abs/2307.09955.

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

Kevin Zakka, Andy Zeng, Pete Florence, Jonathan Tompson, Jeannette Bohg, and Debidatta Dwibedi. Xirl: Cross-embodiment inverse reinforcement learning, 2021. URL https://arxiv.org/abs/2106.03911.

Jinliang Zheng, Jianxiong Li, Dongxiu Liu, Yinan Zheng, Zhihao Wang, Zhonghong Ou, Yu Liu, Jingjing Liu, Ya-Qin Zhang, and Xianyuan Zhan. Universal actions for enhanced embodied foundation models, 2025. URL https://arxiv.org/abs/2501.10105.

Table 8: **Additional VPP variants in simulation.** We evaluate VPP with mixed training data and explicit cross-embodiment trajectory supervision. Gray entries denote tasks additionally included in action training for VPP (mix) and VPP (cross), and are therefore not directly comparable under the original evaluation protocol.

Method	Behavior Grounding				Cross-Embodiment Bridging					Zero-Shot Generalization				
	Basketball in Hoop	Change Channel	Close Box	Avg.	Push Button	Pick and Lift	Slide Block to Target	Pick up Cup	Avg.	Phone on Base	Close Laptop Lid	Put Rubbish in Bin	Close Drawer	Avg.
VPP (mix)	0.47	0.27	0.50	0.41	0.40	0.40	0.20	0.20	0.30	0.13	0.07	0.07	0.00	0.07
VPP (cross)	0.53	0.33	0.50	0.46	0.73	0.50	0.47	0.23	0.48	0.13	0.13	0.10	0.00	0.09
VPP	**0.53**	**0.27**	**0.60**	**0.47**	**0.63**	0.27	0.07	0.17	0.28	0.00	0.17	0.00	0.00	0.04
Demo-JEPA	0.33	0.17	0.43	0.31	0.60	**0.43**	**0.37**	**0.40**	**0.45**	**0.20**	**0.60**	**0.33**	**0.30**	**0.36**

Table 9: **Scaling study in simulation.** We report success rates under different data and task scaling ratios. (a) summarizes task-level results and suite averages, while (b) visualizes the average trends under data scaling and task scaling.

Scaling Setting	Ratio	Cross-Embodiment Bridging					Zero-Shot Generalization				
		Push Button	Pick and Lift	Slide Block to Target	Pick up Cup	Avg.	Phone on Base	Close Laptop Lid	Put Rubbish in Bin	Close Drawer	Avg.
Data Scaling	20%	0.50	0.23	0.00	0.33	0.27	0.10	0.37	**0.33**	0.27	0.27
	50%	**0.60**	**0.43**	0.20	0.27	0.38	0.13	0.40	**0.33**	**0.30**	0.29
Task Scaling	20%	0.33	0.00	0.10	0.27	0.18	0.00	0.33	0.17	0.20	0.18
	50%	0.50	0.40	0.13	0.30	0.33	0.13	0.33	**0.33**	0.20	0.25
Full	100%	**0.60**	**0.43**	**0.37**	**0.40**	**0.45**	**0.20**	**0.60**	**0.33**	**0.30**	**0.36**

(a)



(b)

A Additional Experiment

A.1 Target-Data and Cross-Embodiment Supervision

To examine whether the performance of Demo-JEPA can be attributed simply to additional target-side or cross-embodiment training data, we construct two additional VPP variants.

VPP(mix) mixes the data used for predictive visual representation learning and action learning during training.

VPP(cross) further introduces explicit cross-embodiment supervision by taking source robot observations as input and using the corresponding target robot trajectories as action supervision.

For both variants, the first seven tasks are included during action training. Therefore, the four original Cross-Embodiment Bridging tasks become behavior-grounding tasks under this setting; we retain the original suite grouping in Table 8 only for direct comparison with the main evaluation protocol.

As shown in Table 8, adding cross-embodiment trajectory supervision improves VPP(cross) over VPP(mix) on several seen tasks. However, both variants generalize poorly to unseen tasks. On the four Zero-Shot Generalization tasks, VPP(mix) and VPP(cross) achieve average success rates of 6.7% and 9.2%, respectively, compared with 35.8% for Demo-JEPA. These results suggest that additional trajectory supervision improves fitting to the training distribution but does not provide the same level of generalization to unseen tasks.

A.2 Scaling Study

Table 9 investigates how the scale of paired visual trajectories affects the Dreamer Predictor. We isolate two scaling dimensions: data scaling, which reduces the number of paired episodes per task to 20% or 50%, and task scaling, which reduces the number of task categories to 20% or 50%. The results show that task diversity is more important than per-task episode volume. At the 20% scale, data scaling outperforms task scaling in both cross-embodiment bridging (0.27 vs. 0.18) and zero-shot generalization (0.27 vs. 0.18). The same trend holds at the 50% scale, where data scaling achieves higher average success rates than task scaling in cross-embodiment bridging (0.38 vs. 0.33) and zero-shot generalization (0.29 vs. 0.25). These results indicate that exposure to diverse task semantics is the primary driver of transferable source-to-target latent mappings, while increasing intra-task trajectories further improves predictor robustness.

B PRELIMINARY

We briefly review the key building blocks of our approach: action-conditioned world models for modeling robot and environment dynamics, and the cross-entropy method (CEM) for trajectory optimization.

B.1 ACTION-CONDITIONED WORLD MODELS

Action-conditioned world models learn predictive latent dynamics conditioned on robot actions. Given a trajectory dataset

$$\mathcal{D} = \{\tau_k\}_{k=1}^{N},$$

where each trajectory is defined as

$$\tau = \{(o_t, s_t, a_t)\}_{t=1}^{T},$$

o_t denotes the visual observation, s_t denotes the robot state (e.g., proprioceptive state), and a_t denotes the action executed at time step t.

An encoder $E(\cdot)$ first maps observations into latent representations:

$$z_t = E(o_t).$$

A dynamics predictor $F_{wm}(\cdot)$ then models future latent transitions conditioned on the current latent state, robot state, and action:

$$\hat{z}_{t+1} = F_{wm}(z_t, s_t, a_t).$$

To improve long-horizon dynamics modeling, the predictor can be recursively rolled out using previously predicted latent states:

$$\hat{z}_{t+k} = F_{wm}(\hat{z}_{t+k-1}, s_{t+k-1}, a_{t+k-1}), \quad k > 1.$$

B.2 CROSS ENTROPY METHOD

Given a learned action-conditioned world model, we aim to infer an action sequence that drives the predicted future latent state toward a target latent representation z_{gt}. Formally, we seek an action sequence

$$\mathbf{a}^*_{t:t+H-1}$$

that minimizes the latent prediction error:

$$\mathbf{a}^*_{t:t+H-1} = \arg\min_{\mathbf{a}_{t:t+H-1}} d\left(F_{wm}(z_t, s_t, \mathbf{a}_{t:t+H-1}), z_{gt}\right),$$

where $d(\cdot, \cdot)$ denotes a latent distance metric such as the ℓ_1 distance.

Since directly inverting the nonlinear dynamics predictor $F_{wm}(\cdot)$ is intractable, we employ the Cross-Entropy Method (CEM) to iteratively optimize action sequences.

Specifically, CEM maintains a Gaussian sampling distribution parameterized by mean M and standard deviation S. At each iteration, we sample N candidate action sequences:

$$\mathbf{a}_{1:N} \sim \mathcal{N}(M, S).$$

Each candidate action sequence is rolled out through the world model to obtain predicted future latent states:

$$\hat{z}_{1:N} = F_{wm}(z_t, s_t, \mathbf{a}_{1:N}).$$

We then evaluate each candidate using the latent prediction objective:

$$\mathcal{L}_{plan} = d(\hat{z}, z_{gt}).$$

The top-K candidates with the lowest planning loss are selected as elites to update the sampling distribution:

$$M \leftarrow \beta M + (1 - \beta) M_{\text{elite}},$$

$$S \leftarrow \beta S + (1-\beta) S_{\text{elite}},$$

where β denotes the momentum coefficient.

After multiple iterations, the optimized mean action sequence is used as the final planned action. Although iterative planning with CEM introduces additional computational cost, it avoids directly projecting high-level representations into actions. Instead, actions are optimized online through latent dynamics rollout, which provides stronger adaptability under environment and embodiment shifts.

C PSEUDOCODE

We present detailed pseudocode in this appendix to clearly and systematically describe the core algorithms of our framework, including the Cross-Entropy Method (CEM) optimization, as well as the training and inference procedures. The aim is to provide an unambiguous and implementation-oriented specification that complements the high-level descriptions in the main text.

Algorithm 1 Cross-Entropy Method (CEM) for Latent Trajectory Optimization

Input: Current latent state z_t; Current robot state s_t; Target latent z_{gt}; Dynamics predictor $F_{wm}(\cdot)$; Planning horizon H; Iterations L; Population size N; Number of elites K; Momentum β.

Output: Optimized action sequence $\mathbf{a}^*_{t:t+H-1}$.

Initialize mean $M \in \mathbb{R}^{H \times \dim(a)}$ and standard deviation $S \in \mathbb{R}^{H \times \dim(a)}$.
// e.g., $M \leftarrow \mathbf{0}$, $S \leftarrow \mathbf{I}$
for $iter = 1$ **to** L **do**
  Sample N candidate action sequences:
  $\mathcal{A} = \{\mathbf{a}_1, \ldots, \mathbf{a}_N\}$, where $\mathbf{a}_i \sim \mathcal{N}(M, \text{diag}(S^2))$
  for each candidate $\mathbf{a}_i \in \mathcal{A}$ **do**
    Rollout the world model to predict future latent state:
    $\hat{z}^{(i)}_{t+H} = F_{wm}(z_t, s_t, \mathbf{a}_i)$ ▷ Recursive prediction using Equation B.1
    Evaluate planning loss:
    $\mathcal{L}_i = d(\hat{z}^{(i)}_{t+H}, z_{gt})$ ▷ Using ℓ_1 or ℓ_2 distance
  end for
  Identify indices of K sequences with lowest loss (elites):
  $\mathcal{E} = \text{arg-top-}K \text{ smallest } \{\mathcal{L}_1, \ldots, \mathcal{L}_N\}$
  Compute statistics of elite sequences:
  $M_{\text{elite}} = \frac{1}{K} \sum_{i \in \mathcal{E}} \mathbf{a}_i$
  $S_{\text{elite}} = \sqrt{\frac{1}{K} \sum_{i \in \mathcal{E}} (\mathbf{a}_i - M_{\text{elite}})^2}$
  Update sampling distribution with momentum:
  $M \leftarrow \beta M + (1-\beta) M_{\text{elite}}$
  $S \leftarrow \beta S + (1-\beta) S_{\text{elite}}$
end for
return $\mathbf{a}^* = M$

Algorithm 2 Two-Stage Training of Demo-JEPA

Input: Source dataset $\mathcal{D}^s$, Target dataset $\mathcal{D}^t$; Pretrained encoder E, world model F_{wm}; Perturbation radius r, epochs E_1, E_2.
Output: Trained Dreamer Predictor $\mathcal{P}_\theta$, fine-tuned F_{wm}.

Stage 1: Train Dreamer Predictor
for $epoch = 1$ to E_1 **do**
  Sample target o^t_k, o^t_{k+n} and source o^s_k, o^s_{k+n}
  Sample perturbation $\delta \sim \mathcal{U}(-r, r)$, set $i = k + \delta$, $j = k + n$
  Encode: $z^t_k = E(o^t_k)$, $z^s_i = E(o^s_i)$, $z^s_j = E(o^s_j)$
  Compute embodiment cross-attention: $f_{\text{emb}} = \text{Attn}(z^t_k, z^s_i, z^s_i)$
  Compute motion cross-attention: $f_{\text{mot}} = \text{Attn}(z^s_j, z^s_i, z^s_i)$
  Fuse: $f_{\text{fused}} = \phi([z^t_k \oplus f_{\text{emb}} \oplus f_{\text{mot}}])$
  Predict latent goal: $\hat{z}_{\text{goal}} = \mathcal{P}_\theta(f_{\text{fused}})$
  Compute loss: $\mathcal{L}_{\text{pred}} = \|\hat{z}_{\text{goal}} - E(o^t_{k+n})\|^2_2$
  Update $\mathcal{P}_\theta$ to minimize $\mathcal{L}_{\text{pred}}$
end for
Stage 2: Action Co-Training
Freeze $\mathcal{P}_\theta$
for $epoch = 1$ to E_2 **do**
  Compute $\hat{z}_{\text{goal}}$ using frozen $\mathcal{P}_\theta$ (same as Stage 1)
  Sample action sequence $\mathbf{a} = a^t_{k:k+n-1}$
  $\tilde{z} = F_{wm}(z^t_k, s^t_k, \mathbf{a})$
  Compute planning loss: $\mathcal{L}_{\text{plan}} = \|\tilde{z} - \hat{z}_{\text{goal}}\|^2_2$
  Update F_{wm} to minimize $\mathcal{L}_{\text{plan}}$
end for
return $\mathcal{P}_\theta$, F_{wm}

Algorithm 3 Inference with Adaptive Goal Updating

Input: Source demonstration $\tau^s = \{o^s_1, \ldots, o^s_N\}$; Initial target observation o^t_0; Encoder E; Dreamer Predictor $\mathcal{P}_\theta$; World model F_{wm}; CEM planner (Algorithm 1); Temporal offset Δ; threshold ϵ.
Output: Executed action sequence for task completion.

$i \leftarrow 1$
$z^t \leftarrow E(o^t_0)$
Encode source pair: $z^s_{\text{cur}} \leftarrow E(o^s_i)$, $z^s_{\text{fut}} \leftarrow E(o^s_{i+\Delta})$
Compute initial latent goal:
  $f_{\text{emb}} = \text{Attn}(z^t, z^s_{\text{cur}}, z^s_{\text{cur}})$
  $f_{\text{mot}} = \text{Attn}(z^s_{\text{fut}}, z^s_{\text{cur}}, z^s_{\text{cur}})$
  $f_{\text{fused}} = \phi([z^t \oplus f_{\text{emb}} \oplus f_{\text{mot}}])$
  $\hat{z}_{\text{goal}} = \mathcal{P}_\theta(f_{\text{fused}})$
while $i + \Delta \leq N$ **do**
  Optimize action sequence via CEM:
    $\mathbf{a}^* \leftarrow \text{CEM}(F_{wm}, z^t, s^t, \hat{z}_{\text{goal}})$
  Execute first action a^*_0, observe new o^t_{next}, s^t_{next}
  $z^t_{\text{next}} \leftarrow E(o^t_{\text{next}})$
  Compute discrepancy $D = d(z^t_{\text{next}}, \hat{z}_{\text{goal}})$
  if $D < \epsilon$ **then**
    $i \leftarrow i + 1$
    if $i + \Delta \leq N$ **then**
      Update source pair: $z^s_{\text{cur}} \leftarrow E(o^s_i)$, $z^s_{\text{fut}} \leftarrow E(o^s_{i+\Delta})$
      Recompute $\hat{z}_{\text{goal}}$ (same as initialization)
    end if
  end if
  $z^t \leftarrow z^t_{\text{next}}$, $s^t \leftarrow s^t_{\text{next}}$
end while
return success

D TRAINING DETAILS

We present the training details of Demo-JEPA and Demo-DP. To improve clarity, we separate shared configurations from stage-specific settings.

D.1 SHARED TRAINING CONFIGURATION

All stages use identical data preprocessing, augmentation, and optimization settings unless otherwise specified.

Table 10: Shared training configuration across all stages.

Component	Value	Description
Crop size	256	Input image resolution
Patch size	16	ViT patch size
Tubelet size	2	Temporal grouping of frames
FPS	5	Video sampling rate
Context frames	8	Sequence length per sample
Optimizer	AdamW	Default optimizer
Base learning rate	4.25e-4	Peak learning rate
Warmup epochs	15	Linear warmup schedule
Total epochs	315	Training duration
Weight decay	0.04	L2 regularization
Augmentation	Disabled	No stochastic augmentation

D.2 DEMO-JEPA

D.2.1 STAGE 0: ACTION-CONDITIONED WORLD MODEL

We follow the training setup of VJEPA2 with minor modifications.

Table 11: Stage-specific parameters for AC world model.

Parameter	Value	Description
Batch size	24	Per-GPU batch size
Predictor depth	24	Transformer layers
Embed dim	1024	Hidden dimension
Attention heads	16	Multi-head attention
Frame-causal	True	Temporal causal masking
Modality embedding	True	Image/video separation

D.2.2 STAGE 1: LATENT GOAL PREDICTOR TRAINING

Table 12: Stage-specific parameters for latent goal predictor.

Parameter	Value	Description
Batch size	16	Per-GPU batch size
Fusion type	Conv3D	Feature fusion mechanism
Num self-attn blocks	4	Transformer refinement layers
MLP ratio	4.0	Expansion ratio
Up dim	64	Projection dimension
Norm layer	LayerNorm	Normalization type
Init std	0.02	Weight initialization scale

D.2.3 STAGE 2: ACTION CO-TRAINING

Table 13: Stage-specific parameters for action co-training.

Parameter	Value	Description
Batch size	16	Per-GPU batch size
Predictor depth	24	Transformer layers
Embed dim	1024	Hidden dimension
Attention heads	16	Multi-head attention
Frame-causal	True	Temporal causal masking
Modality embedding	True	Image/video separation

D.3 DEMO-DP

D.3.1 ACTION HEAD FINE-TUNING

Table 14: Diffusion head configuration for Demo-DP.

Parameter	Value	Description
Batch size	16	Per-GPU batch size
Condition dim	1408	Encoder output dimension
Horizon	1	Action prediction steps
Condition steps	256	Context tokens
Diffusion steps	100	Noise schedule steps
Beta schedule	squaredcos_cap_v2	Noise schedule type
Prediction type	epsilon	Noise prediction target
Layers	12	Transformer depth
Condition layers	8	Conditioning blocks
Heads	4	Attention heads
Embed dim	384	Hidden dimension
Dropout (attn)	0.3	Attention dropout
Causal attention	True	Autoregressive masking
Time conditioning	True	Timestep embedding

D.4 COMPUTE BUDGET

We summarize the compute budget in Table 15.

Table 15: Compute budget of Demo-JEPA and Demo-DP training stages.

Stage	GPU	Training Time
Demo-JEPA		
Stage 0: World model	8×A100	7 days
Stage 1: Goal predictor	8×A100	2.5 days
Stage 2: Co-training	8×A100	1 day
Demo-DP		
Action head fine-tuning	8×A100	1 day

E REAL WORLD SETUP

The physical deployment of Demo-JEPA is designed to evaluate cross-embodiment imitation under a minimal and consistent perception setting across heterogeneous robotic platforms.

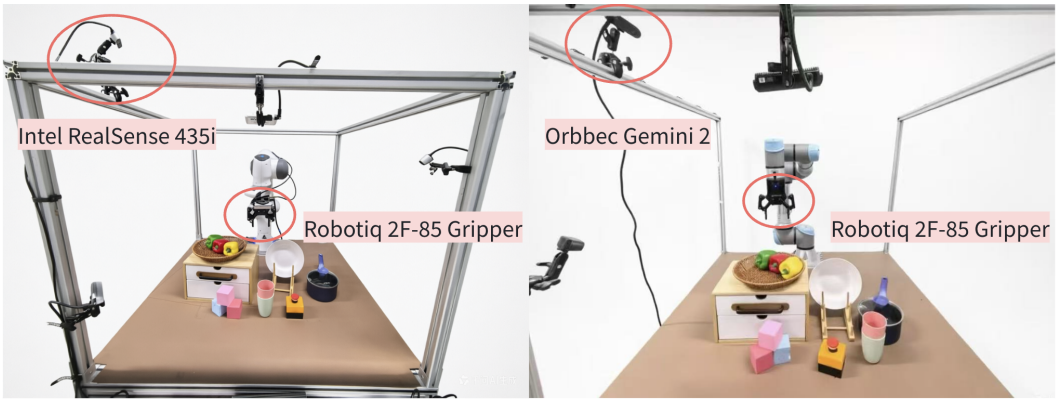

Figure 5: **Real world experiment environment setup.** Franka workspace(left) and UR5e workspace(right).

As shown in Figure 5, we consider two embodiments: a Franka emika manipulator as the target embodiment and a UR arm as the source embodiment. Both systems are equipped with a Robotiq 2F-85 parallel gripper.

For perception, we use a single fixed third-person RGB camera mounted at a right-shoulder viewpoint relative to the workspace. Specifically, the Franka setup uses an Intel RealSense D435i, while the UR setup uses an Orbbec Gemini 2. Notably, the two cameras are not explicitly calibrated across embodiments; instead, they are roughly placed to ensure that the field of view covers both the robot manipulator and the operation plane, without enforcing precise geometric alignment. There are no wrist-mounted or additional ego-centric cameras are used; both FR3 and UR policies operate solely under third-person visual input.