

CLAP: Cross-Embodiment Video World Models are Zero-Shot Physical Simulators

Kechen Liu*, Ola Shorinwa*

Princeton University

*Equal contribution.

State-of-the-art action-conditioned video models are typically restricted to a single robot embodiment, preventing them from leveraging the vast corpus of heterogeneous video data that contains rich signals for learning generalizable physics. To bridge this gap, we introduce CLAP, a framework for cross-embodiment action-conditioned video generation capable of being trained on diverse, internet-scale videos across human and robotic agents. CLAP is grounded in the insight that universal physical laws govern spatiotemporal dynamics regardless of the actor. However, cross-embodiment learning is non-trivial because action representations vary sharply across robot platforms and are typically absent in human videos. CLAP addresses this fundamental challenge through three core contributions. First, CLAP reconciles disparate action spaces across human and robot morphologies using end-effector poses, natural language instructions, and learned latent action representations. Second, to resolve the individual limitations of each action representation, CLAP introduces a curriculum-based cross-embodiment learning recipe that first learns foundational physical priors across unlabeled video data using latent actions and subsequently grounds them in end-effector action spaces for zero-shot deployment to real-world tasks. Crucially, CLAP approaches or surpasses state-of-the-art single-embodiment video models in challenging environments like DROID, with performance advantages that compound via few-shot adaptation to target embodiments. Through inference-time cross-policy planning and reinforcement-learning-based policy finetuning in video world models, we demonstrate CLAP’s zero-shot generalization to real-world tasks, improving the performance of state-of-the-art robot policies, such as $\pi_{0.5}$ and MolmoAct-2. Ultimately, CLAP delivers the most comprehensive suite of action-conditioned video world models to date — spanning diverse action-conditioning spaces (end-effector, language, and latent) and robot morphologies (including cross-embodiment, DROID, Bridge, bimanual YAM robots, and G1 humanoids). Third, CLAP establishes a novel paradigm for training high-fidelity single-embodiment video world models via sample-efficient, few-shot adaptation of cross-embodiment models to target robot platforms. We open-source all code and models at <https://github.com/omni-CLAP/clap>.

Keywords: Action-Conditioned Video Generation, Cross-Embodiment Learning, Video World Models

Website: omni-clap.github.io

Code: github.com/omni-CLAP/clap



1 Introduction

The defining breakthroughs in Large Language Models (LLMs) were precipitated by web-scale training on diverse text datasets, shattering performance barriers by learning a unified representation of human language [1–4]. Today, as action-conditioned video world models continue to struggle with physical inconsistency, the robotics and vision communities face a critical question: can generative video models learn generalizable physical priors through cross-embodiment data scaling, or does the heterogeneous nature of different robot morphologies demand agent-specific architectures? In this work, we investigate how video world models can bridge the gap between diverse embodiments and analyze the structural bottlenecks that must be overcome.

Prior action-conditioned video models [5, 6] focus exclusively on single robot embodiments, sidestepping the inherent challenges associated with diverse robot morphologies, which forecloses the internet-scale video

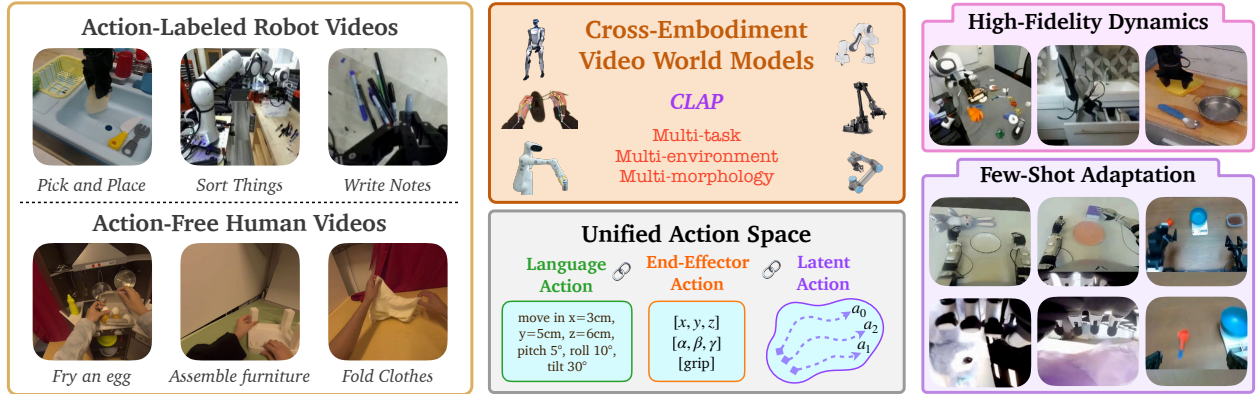


Figure 1 We introduce **CLAP**, a cross-embodiment learning framework that trains action-conditioned video models on diverse human and robot video data to learn fundamental physical priors that are critical for fine-grained dynamics prediction.

data behind recent foundation-model advances. To address these limitations, we introduce **CLAP**, a training framework for cross-embodiment action-conditioned video models designed to learn fundamental physical laws across different agents. CLAP’s core insight lies in the fact that the same universal laws of physics govern all dynamical interactions irrespective of the agent’s embodiment. For example, a bottle tips, a towel folds, and a drawer slides according to the same physics whether the actor is a Franka arm, a WidowX gripper, or a human hand, underscoring that the appropriate response to heterogeneity is not to restrict training to one embodiment.

Translating this physical principle into a scalable framework, however, requires overcoming severe cross-embodiment challenges. Web-scale datasets feature actors with vastly different physical morphologies and action spaces, and crucial subsets (e.g., human videos) are entirely devoid of action labels. CLAP bridges this gap via three fundamental contributions. First, CLAP harmonizes the disparate action spaces of diverse embodiments using end-effector poses, natural language instructions, and learned latent action representations. However, these action representations have native limitations. For example, while latent actions enable learning from unlabeled videos, they do not facilitate zero-shot real-world deployment. Conversely, while end-effector actions address this drawback, they preclude training on unlabeled video data. Second, to address these limitations, CLAP introduces a curriculum-based cross-embodiment learning framework that first learns foundational physical priors from unlabeled video data using latent actions and subsequently refines them in end-effector action spaces for direct, zero-shot deployment to real-world tasks. Third, CLAP establishes a novel paradigm for training single-embodiment video models through sample-efficient adaptation of cross-embodiment models to target embodiments. This adaptation framework facilitates the seamless transfer of physical priors across highly disparate robot morphologies.

Critically, CLAP approaches or surpasses state-of-the-art single-embodiment baselines in challenging environments like DROID [7]. Notably, this performance advantage compounds through sample-efficient adaptation to the target embodiment. Through inference-time cross-policy planning and reinforcement-learning-based policy finetuning in video world models, we demonstrate CLAP’s zero-shot generalization to real-world manipulation tasks, achieving higher success rates compared to the base performance of state-of-the-art policies, e.g., $\pi_{0.5}$ [8] and MolmoAct-2 [9].

Ultimately, CLAP delivers the most comprehensive suite of action-conditioned video world models to date (Fig. 1), which spans diverse action-conditioned spaces (end-effector, language, and latent) and robot morphologies (e.g., cross-embodiment, DROID, Bridge, bimanual YAM robots and G1 humanoids). We open-source all video world models with accompanying code. (See Appendix A for preliminaries and Appendix D for a nuanced summary of the core contributions of CLAP.)

2 Related Work

Video Generation in Robotics. Recent advances in video generation models [10–13] have enabled high-fidelity synthesis of physically consistent video content, motivating their growing adoption as embodied world models in robotics. One line of work leverages video models as data generators, synthesizing robot trajectories paired with pseudo-action labels for downstream policy learning [14, 15]. Other works directly employ video models as policy backbones, decoding actions through tracking, inverse dynamics, or unified video-action prediction [16–20] (see [21] for a review of these video generation models). Most relevant to our work, a growing body of research employs action-conditioned video models as learned simulators for policy evaluation and improvement, rolling out policies in imagination space rather than on real hardware [5, 6, 22–26]. However, all these methods are primarily restricted to a single robot embodiment, hindering them from leveraging diverse cross-embodiment video data. Our work addresses these fundamental limitations.

Cross-Embodiment Robot Learning and Latent Actions. A central challenge in scaling generalist robot policies is the heterogeneity of robot embodiments, which exhibit varying degrees of freedom, kinematic structures, and observation configurations. To bridge this heterogeneity, prior work has explored several strategies for unifying state and action spaces across embodiments. The most direct approaches pad states and actions to a maximum dimensionality or adopt a shared end-effector pose or language representation [27–30], while others learn embodiment-specific action heads or projectors atop a shared backbone [31–34]. Other approaches [35–38] leverage latent actions as a paradigm for unifying heterogeneous action spaces. Genie [39] learns latent actions via a VQ-VAE objective for interactive video generation, while LAPA [40] applies them to VLA policies. However, current latent action models (LAMs) typically require downstream alignment to embodiment-specific commands. To address this challenge, we propose end-effector and language actions as unified action representations that directly leverage ground-truth actions for fine controllability when dense action-labeled video data is available. To combine their complementary strengths, CLAP uses a curriculum-based video pretraining recipe that integrates latent actions with end-effector actions for direct, zero-shot deployment.

3 CLAP: Training Cross-Embodiment Video World Models

Cross-embodiment learning from video data presents a unique challenge: robot and human data is inherently heterogeneous with different observation and action spaces, and internet-scale video data largely lacks action labels. To address these challenges, CLAP introduces a novel recipe for training cross-embodiment video world models, reconciling heterogeneous action spaces using end-effector poses, natural language, and latent action representations (see Secs. 3.1 and 3.2). CLAP facilitates data-efficient, few-shot adaptation to target embodiments (see Sec. 3.3). Fig. 2 summarizes this pipeline.

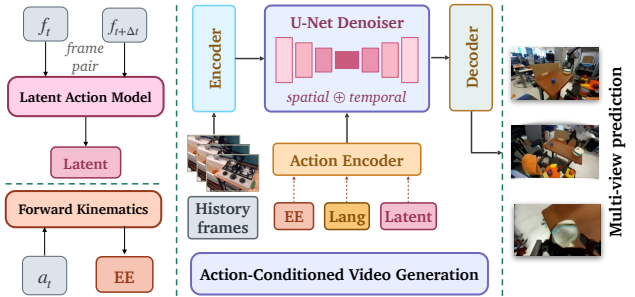


Figure 2 CLAP CLAP reconciles disparate action spaces across human and robot morphologies to train high-fidelity cross-embodiment video models.

3.1 Harmonizing Heterogeneous Action Spaces

Varying robotic degrees of freedom (DoF) dictate highly divergent action spaces; for instance, the ubiquitous WidowX and Franka platforms operate with six and seven DoFs, respectively. Because action-conditioned video models require fixed-dimensional conditioning inputs, generalizing across these diverse morphologies demands a unified action space. To resolve this gap, CLAP establishes unified action representations in end-effector, language, and latent action-spaces and applies critical transformations to bridge the disparate action spaces of diverse robot embodiments. While each action representation offers unique advantages, they also introduce distinct limitations. Below, we analyze these trade-offs and introduce a curriculum-based

action harmonization method that combines the strengths of different action representations to overcome their individual drawbacks. We provide additional details in Appendix B.1.

End-effector Actions. While joint-space control is widely adopted by prior work in robot control, it inherently restricts a video model’s amenability to architectures sharing identical degrees of freedom. To facilitate cross-embodiment learning, we introduce **CLAP-EE**, which adopts end-effector (EE) actions as unified, task-relevant representations that abstract away low-level morphological differences. Concretely, we map joint positions to a 7-DoF operational space via the robot’s forward kinematics. In line with prior work [27], our canonical end-effector action space comprises three translational dimensions and three Euler angles for orientation, alongside a continuous gripper state. This representation is readily compatible with existing large-scale data repositories, such as the Open X-Embodiment (OXE) dataset [41], which natively provide frame-level joint position and kinematic metadata. Since the volume of a robot’s operational space varies widely with its morphology, end-effector actions demand video modeling across larger action spaces, which typically poses an intractable challenge, particularly in absolute-action spaces. To address this limitation, prior work [42] trains single-embodiment video models using relative actions. However, we show that relative-action spaces degrade the prediction fidelity of cross-embodiment video models conditioned on end-effector actions (see Sec. 4.2). To circumvent this degradation, **CLAP-EE** utilizes absolute end-effector actions and normalizes the disparately-sized action spaces of each robot platform into finite bounds $[-1, 1]$, effectively simplifying the learning problem within a shared, uniformly-bounded action space across all robots. Crucially, each robot inherits a unique normalization factor, accounting for its morphological differences. Although these action harmonization architecture overcomes the aforementioned challenges, a central limitation persists: end-effector action conditioning requires access to labeled video data, impeding data scaling to the massive corpus of unlabeled internet videos.

Language Actions. Language actions express control primitives as natural text, offering a versatile abstraction that enables coarse-to-fine controllability with minimal architectural adaptation. Furthermore, recent work [43] suggests that text-based action abstractions are more effective in preserving pre-trained spatiotemporal priors within foundational models. Therefore, we derive **CLAP-LANG**, a recipe for training cross-embodiment models conditioned on language actions. To maintain controllability within a textual interface, we project numerical end-effector actions into the language space by mapping continuous coordinate actions to text. Preliminary experiments revealed that the high-precision format of ground-truth robot end-effector actions often leads to truncation by the CLIP tokenizer. Moreover, existing state-of-the-art text tokenizers are poorly aligned to preserve the uniqueness of numerical values during tokenization, potentially compromising the information content of tokenized actions. **CLAP-LANG** uses concise templated motion primitives (e.g., “ $x=$, $y=$, ...”) to compensate for the limited context length of text tokenizers and the limited resolution of the discrete language-tokenization space. In addition to normalizing the action space, **CLAP-LANG** transforms absolute end-effector actions into relative-action spaces to further narrow the action space, improving the information density of templated language actions. We demonstrate that relative-action spaces improve future-prediction fidelity in cross-embodiment video models in Sec. 4.2. However, compensating for the weaknesses of language tokenization still requires decreasing the precision of the conditioning actions, ultimately limiting the spatial resolution of language-action-conditioned video models compared to their end-effector-action-conditioned counterparts.

Latent Actions. Large volumes of web-scale video data do not contain action annotations, particularly human videos, impeding their amenability to end-effector and language actions. To harness the rich diversity of unlabeled video data in training cross-embodiment video models, we leverage latent actions as proxy action representations, which describe the underlying transformation between pairs of video frames using (low-dimensional) action tokens. Latent action models (LAMs) [39, 40] utilize autoencoders (VAEs) to learn pseudo-actions from videos through self-supervision on a reconstruction loss objective. CLAP’s LAM takes in a pair of video frames $(f^t, f^{t+\Delta t})$ ($\Delta \in \mathbb{Z}$) and outputs d -dimensional continuous latent actions (we set $d = 32$). The encoder q_ϕ consists of a spatiotemporal transformer that extracts features from both input video frames, which is mapped to latent action $a_{\text{ln}} \in \mathbb{R}^d$. To reconstruct $f^{t+\Delta t}$, the decoder p_ψ takes in f^t and a_{ln} , incentivizing the model to learn causal relationships. We train the LAM with the reconstruction loss:

$$\mathcal{L}_{\psi, \phi}^{\text{LAM}}(f^{t+\Delta t}) = \mathbb{E}_{q_\phi(a_{\text{ln}} | f^t, f^{t+\Delta t})} \log p_\psi(f^{t+\Delta t} | f^t, a_{\text{ln}}) - \alpha D_{\text{KL}}(q_\phi(a_{\text{ln}} | f^t, f^{t+\Delta t}) || p_l(a_{\text{ln}})), \quad (1)$$

with KL divergence-regularization, where $\alpha \in \mathbb{R}_+$ denotes the relative weight between the two objective function terms, and $p_l(a_{\text{in}})$ represents the prior distribution. During training, we dynamically vary Δt as a form of data augmentation to improve model robustness. We train the video model **CLAP-LAM** using the latent actions extracted from raw (unlabeled) video data for effective cross-embodiment scaling, which can improve prediction accuracy (see [Sec. 4.1](#)). Despite this strength, **CLAP-LAM** requires downstream alignment to a target embodiment’s action space, which can degrade its prediction fidelity, as observed in preliminary experiments.

Curriculum-based Latent-to-EE Actions. CLAP’s end-effector-action, language-action, and latent-action harmonization methods offer valuable unification advantages; however, they all face important challenges. To resolve these limitations, we introduce **CLAP-CURR**, a curriculum-based video pretraining method that first learns foundational physical priors from unlabeled video data using latent actions and subsequently refines these priors conditioned on end-effector actions for zero-shot deployment to real-world tasks. Critically, **CLAP-CURR** combines the unique strengths of latent actions with those of end-effector actions to overcome their individual limitations. This strategy achieves unrestricted cross-embodiment scaling without requiring downstream adaptation to target robot embodiments. Moreover, by eliminating the need for downstream adapters, **CLAP-CURR** circumvents the degradation in prediction fidelity typically introduced by alignment layers. After training the video model with latent actions, CLAP swaps the action head with another that takes in 7-dimensional end-effector actions, while retaining the pretrained weights in the video model backbone. Thereafter, **CLAP-CURR** jointly trains the new action head and the video model on action-labeled video data.

3.2 Training the Cross-Embodiment Video World Model

Model Architecture. CLAP uses the video diffusion [44] paradigm for action-conditioned video generation, which is described by the probability flow ordinary differential equation: $d\mathbf{x} = -\dot{\sigma}(t)\sigma(t)\nabla_{\mathbf{x}} \log p_v(\mathbf{x}; \sigma(t))dt$, given the input video $\mathbf{x}$ sampled from the probability distribution p_v and noise scheduler $\sigma(t)$. We supervise the video model $\mathcal{V}_\theta$ using the loss function:

$$\mathbb{E}_{\sigma, \mathbf{x}_1, \mathbf{x}_0} \left[w(\sigma) \left\| \mathcal{V}_\theta(c_{\text{in}}(\sigma) \cdot \mathbf{x}_c; c_{\text{noise}}(\sigma)) - \frac{1}{c_{\text{out}}(\sigma)} (\mathbf{x}_1 - c_{\text{skip}}(\sigma) \cdot \mathbf{x}_c) \right\|_2^2 \right], \quad (2)$$

with partially denoised video input $\mathbf{x}_c = \mathbf{x}_1 + \sigma\mathbf{x}_0$, ground-truth video $\mathbf{x}_1 \sim p_{\text{data}}$, pure noise $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and scaling and conditioning factors $(c_{\text{in}}, c_{\text{out}}, c_{\text{skip}})$ and c_{noise} , where $w(\sigma) = \lambda(\sigma)c_{\text{out}}(\sigma)^2$ and $\sigma \sim p_d$. In practice, we train latent video diffusion models using continuous VAEs for video tokenization followed by a U-Net for salient spatiotemporal modeling.

History Conditioning and Multi-view Video Prediction. Standard action-conditioning inputs lack the temporal velocity information crucial for high-fidelity future prediction. To provide rich temporal context, we condition the video model on a history of past observations and robot poses across a horizon H , employing causal cross-attention to bridge historical context with future predictions. Furthermore, to enforce geometrically consistent training supervision and enhance downstream robustness, we incorporate multi-view conditioning to generate multi-view future frames. Structurally, we concatenate all views vertically in the latent space, passing the unified frame directly to the model without altering its core architecture.

3.3 Beyond Zero-Shot: Data-Efficient Adaptation to Target Morphologies

Through broad-scale training on diverse robot video data, CLAP’s models learn generalizable priors that deliver high-accuracy dynamics prediction. Nonetheless, their zero-shot prediction accuracy can be improved by finetuning them on target embodiments, spanning pretrained embodiments which have been seen during training and novel embodiments. Via finetuning, these models can refine their learned priors to better align them with target embodiments.

Pretrained Embodiments. CLAP establishes a novel framework for training single-embodiment video world models via few-shot adaptation of cross-embodiment models. CLAP’s models provide foundational physical priors that can be readily adapted to a target embodiment via data-efficient finetuning on the robot’s native end-effector action space. This finetuning process does not require any modifications to the architecture of the underlying cross-embodiment model, since the target embodiment’s action space already matches that of

the base model. Crucially, this paradigm for training high-fidelity single-embodiment video models yields superior world models compared to prevailing approaches that train them from scratch or from less-aligned video model backbones such as WAN [11] and SVD [12].

Novel Embodiments. Beyond pretrained embodiments, CLAP facilitates data-efficient adaptation of cross-embodiment models to novel robot morphologies that are markedly different from those seen during training, such as bimanual robots and humanoids, which typically feature higher-dimensional action spaces. Due to this action-space disparity, model finetuning requires substituting the model’s action head with a new one compatible with the target embodiment. Crucially, CLAP retains all other components of the cross-embodiment video model to preserve rich spatiotemporal priors. During finetuning, CLAP aligns these underlying priors with the target morphology, enabling high-fidelity action-conditioned future prediction.

4 Experiments

Through the lens of the following research questions, we investigate the impact of cross-embodiment scaling on action-conditioned video models and their efficacy in downstream robotics tasks:

[Q1.] Do cross-embodiment video models surpass their single-embodiment counterparts? (Sec. 4.1)

[Q2.] Do relative-action spaces outperform absolute-action spaces in cross-embodiment video modeling? (Sec. 4.2)

[Q3.] Do cross-embodiment video models achieve zero-shot real-world generalization? (Sec. 4.3)

[Q4.] Beyond Zero-Shot: Does CLAP facilitate data-efficient adaptation to target embodiments? (Sec. 4.4)

Evaluation Setup. We train CLAP’s cross-embodiment models on the Open X-Embodiment [41] and EgoDex [45] datasets for 100K steps on eight H100 or H200 GPUs (varying by availability), taking approximately two to three days per training run. We use an effective batch size of 64 across all models, which is kept consistent across all GPU architectures via gradient accumulation. We benchmark the video models with the following perceptual metrics: SSIM [46], PSNR [46], LPIPS [47], FVD [48], and FID [49] using standard implementations. For FVD, we use features extracted by the R3D-18 model, which was pretrained on the Kinetics-400 dataset, available via torchvision. We report aggregate metrics computed on the stacked views for about 100 trajectories per dataset subset. For each dataset, we follow standard procedures, with training splits for training and val/test splits used for evaluation. We provide additional details along with inference and timing results in Appendix C.1 and Appendix C.2.

Baselines. We benchmark CLAP cross-embodiment video models against the state-of-the-art Ctrl-World [5] baseline in the DROID environment. To ensure a fair comparison, we omit baselines with mismatched compute or inference requirements, including models that cannot perform autoregressive generation initialized from a single frame and those that require privileged information like robot URDFs and simulators. Although state-of-the-art baselines exist in the Bridge environment (e.g., WorldGym [6], Cosmos-Predict 2.5 [50]), we found these baselines to underperform our internal Bridge video models. Consequently, we train a new Bridge baseline (**Bridge-Base**) that outperforms existing alternatives (see Sec. 4.1). We train all baselines using the same model architecture and hyperparameters as CLAP to control for confounding variables, noting the significant effect that hyperparameters typically have on performance. This experiment design enables us to isolate the impacts of different components of CLAP on future-prediction fidelity. Comprehensive experiment results are provided in Appendix C.4.

4.1 Do cross-embodiment video models surpass their single-embodiment counterparts?

We benchmark CLAP’s cross-embodiment video models against state-of-the-art single-embodiment baselines across diverse robot datasets, with a focus on dominant platforms like Bridge and DROID (Fig. 3). We structure our evaluation along the following core axes:

Action-space Harmonization. We evaluate how different action-space harmonization schemes impact future-prediction fidelity. In doing so, we identify fine-grained trade-offs between end-effector and language action-spaces, revealing key differences in how well each integrates with diverse data sources. From Fig. 3, end-effector-action spaces enable fine-grained dynamics prediction across multiple robot morphologies, yielding

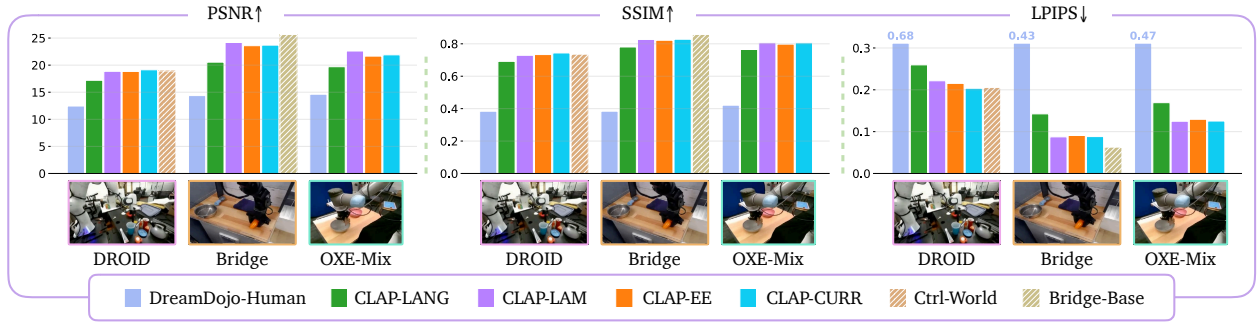


Figure 3 Comparison to SOTA single-embodiment baselines. CLAP approaches or exceeds SOTA performance on the DROID platform, showing only minor fidelity degradation in simpler domains like Bridge. Across the diverse mix of embodiments (OXE-Mix), CLAP-LAM and CLAP-CURR achieve the highest prediction accuracies, closely followed by CLAP-EE.

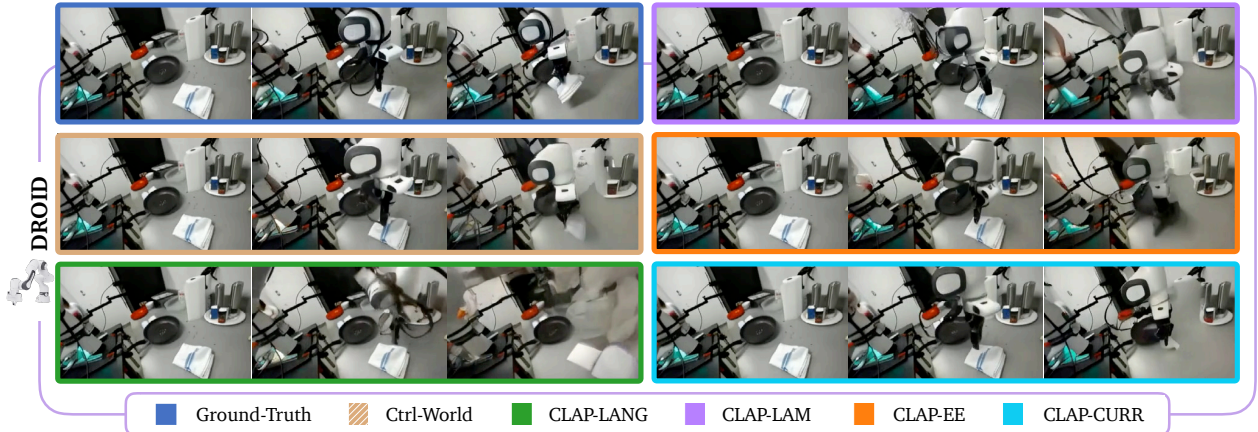


Figure 4 CLAP's performance on DROID. CLAP-CURR combines the strengths of CLAP-LAM and CLAP-EE to achieve high-fidelity dynamics prediction with fine-grained per-frame controllability, outperforming SOTA single-embodiment baselines. Meanwhile, CLAP-LANG delivers comparable performance over shorter horizons, though it is constrained by compounding errors over longer trajectories.

superior prediction fidelity compared to language-action spaces. This divergence aligns with intuition: end-effector-action harmonization offers continuous precision, whereas coarse language actions suffer from the limitations of discrete tokenization. Consequently, end-effector-action-conditioned video world models resolve the granular action signals that are crucial for high-fidelity predictions. Notably, latent actions can match or surpass end-effector actions in future-prediction accuracy, defining the frontier in cross-embodiment video modeling across a diverse mix of robot platforms (OXE-Mix). Latent-action harmonization uniquely unlocks internet-scale training on unlabeled (action-free) video data (e.g., human video data) compared to other action harmonization schemes. However, these breakthroughs come at the expense of downstream deployment because physical robots operate in geometric action spaces, creating a mismatch between training and inference conditions. Addressing these limitations, CLAP-CURR seamlessly combines the strengths of latent actions with those of end-effector actions to achieve high-fidelity cross-embodiment video models capable of zero-shot transfer to real-world deployment use-cases, without impeding data scaling. Figs. 4 and 5 demonstrate these findings. On the more challenging DROID environment, CLAP-CURR closely follows the ground-truth robot trajectory across all frames, improving upon the prediction fidelity of CLAP-LAM and CLAP-EE. On the Bridge platform, all CLAP's models deliver high-accuracy future predictions. CLAP-LANG achieves comparable performance over shorter prediction horizons, but it is ultimately constrained by compounding errors over longer horizons.

Human Videos vs. Multi-morphology Videos. We investigate if human video data can function as a replacement for multi-morphology robot data in training cross-embodiment video models via the DreamDojo-Human

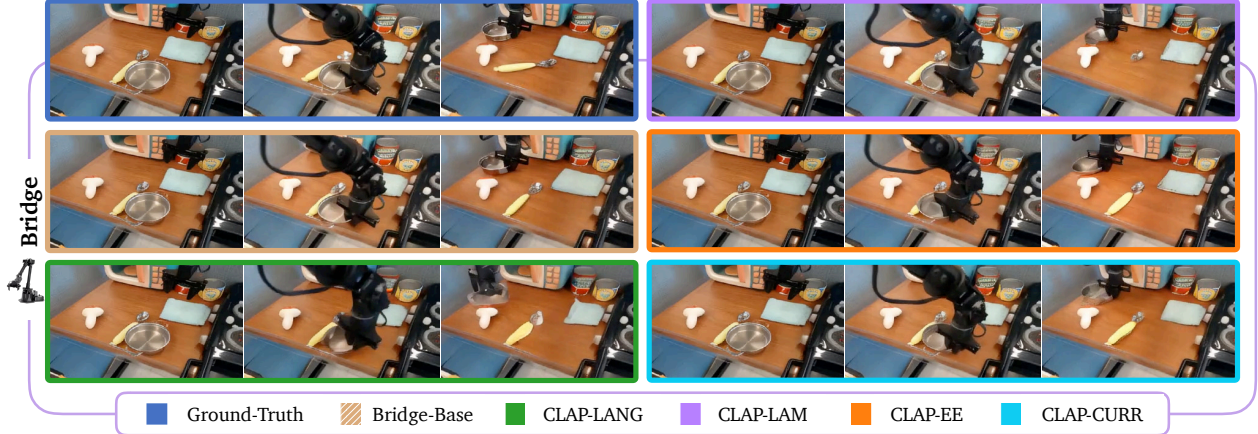


Figure 5 CLAP’s performance on Bridge. Relative to the SOTA baseline, CLAP achieves high-accuracy future prediction on the Bridge platform; however, the performance of **CLAP-LANG** degrades over extended prediction horizons.

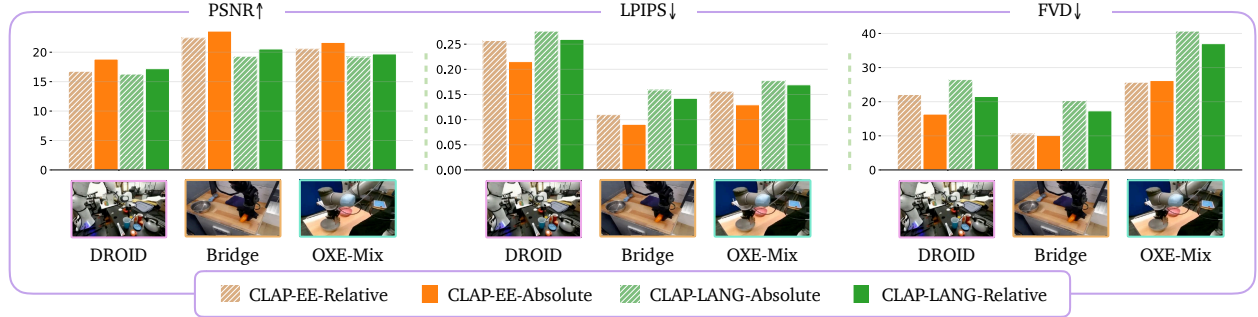


Figure 6 Comparison between relative-action and absolute-action spaces. Action-conditioned video generation in absolute-action spaces yields superior prediction fidelity; however, these gains are limited to future prediction tasks conditioned on end-effector-actions. In contrast, relative actions achieve higher prediction fidelity for language-action-conditioned video models.

baseline. We use the DreamDojo IDM trained exclusively on human videos to extract latent actions which serve as action-conditioning inputs for the video model. **Fig. 4** shows that robot video data is crucial for high-fidelity cross-embodiment video modeling, underscored by the significantly lower prediction accuracies achieved by **DreamDojo-Human**. For example, CLAP’s video models improve the **DreamDojo-Human**’s LPIPS score by at least 61% on the DROID platform, with similar gains on all other perceptual metrics and robot environments. These findings underscore that although cross-embodiment video models can learn foundational physical priors from human video data, robot data is essential for effective transfer of these priors to robot morphologies, establishing it as crucial ingredient for training generalizable cross-embodiment video models.

Cross-Embodiment vs. Single-Embodiment. Single-embodiment video models have set the state-of-the-art (SOTA) in action-conditioned video generation; however, can cross-embodiment video models match their performance? Here, we compare CLAP’s cross-embodiment video models to the SOTA baselines. As shown in **Fig. 3**, CLAP’s video models (e.g., **CLAP-LAM**, **CLAP-EE**) match the performance of the SOTA baselines in the more challenging DROID environment. This finding challenges prevailing wisdom for two core reasons. First, CLAP’s cross-embodiment video models have the same number of model parameters and architecture as the single-embodiment video models. Given that the single-embodiment baselines are purpose-built-and-trained for future prediction on a specific robot platform, one would expect them to outperform CLAP’s cross-embodiment models. Second, to match the performance of single-embodiment baselines, cross-embodiment models typically require training on the union of all single-embodiment data alongside a significantly increased number of training steps. Surprisingly, CLAP’s cross-embodiment models match their single-embodiment counterparts despite utilizing effectively fewer DROID samples with the same training setup. These findings suggest that CLAP learns generalizable physical priors from cross-embodiment data, enabling high-fidelity predictions even

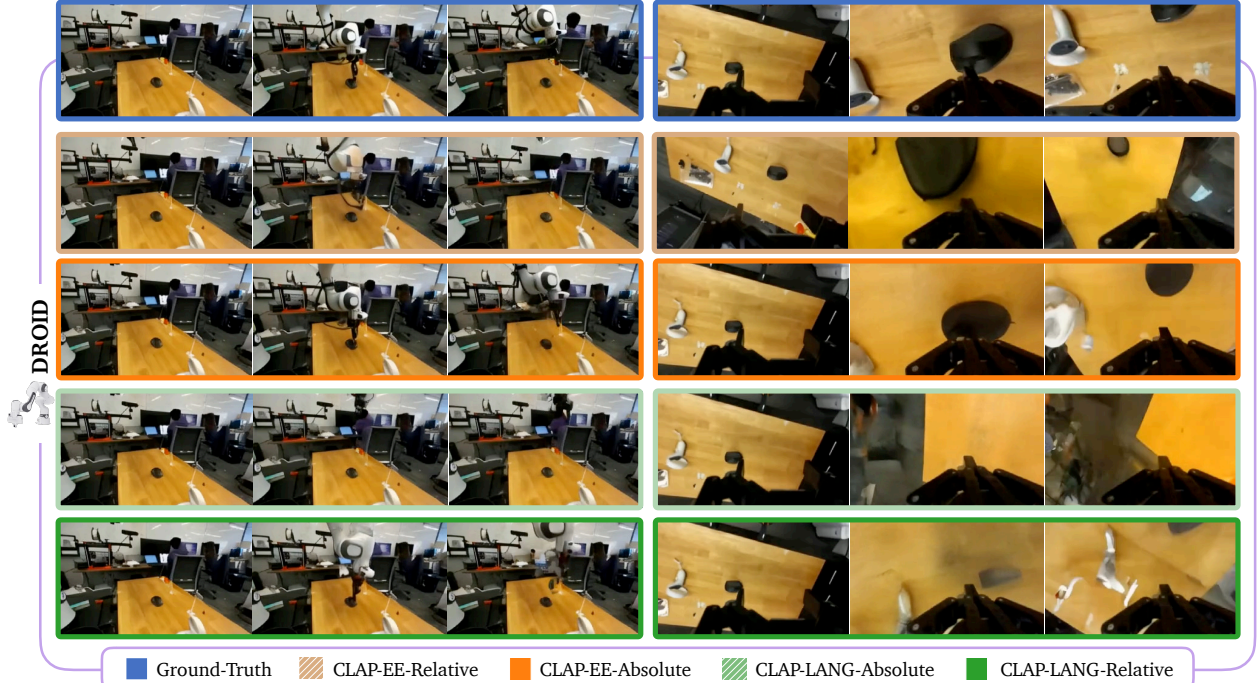


Figure 7 Comparison between relative-action and absolute-action spaces in DROID. In end-effector-action-conditioned video generation, absolute-action spaces enable fine-grained controllability over extended prediction horizons, an advantage that is not observed with relative-action spaces. Conversely, language-action-conditioned video models achieve higher-accuracy future prediction with relative actions.

with limited domain-specific supervision. This paradigm has the potential to transform how action-conditioned video models are trained, mirroring recent breakthroughs in multi-modal foundation models for vision and language generation. In the simpler Bridge environment, CLAP delivers high-accuracy dynamics predictions but does not precisely match our Bridge baseline (**Bridge-Base**), which is itself a new state-of-the-art baseline. (Prior video models achieve PSNR scores below 25 on the Bridge dataset.) The qualitative differences in the model’s performance on the Bridge platform are mostly marginal, as shown in Fig. 5.

4.2 Do relative-action spaces outperform absolute-action spaces in cross-embodiment video modeling?

Recent work on video world models utilize relative-action spaces in action-conditioned video models, motivated by prevailing assumptions that relative actions offer a narrower action space with reduced modeling complexity. Our investigation of the impacts of relative-action and absolute-action spaces on cross-embodiment video modeling reveals nuanced findings, discussed in the following subsections:

End-effector-action Conditioning. From Fig. 6, relative-action spaces underperform absolute-action spaces in future prediction conditioned on end-effector actions across all perceptual metrics and robot environments, e.g., by about 14.6% in LPIPS in the DROID environment. This finding contradicts standard assumptions on the superiority of relative actions. Although relative actions might provide a less complex action space, they remain highly susceptible to compounding errors that degrade prediction fidelity, especially over extended horizons. In contrast, absolute-action spaces provide stronger robustness to compounding errors, yielding higher prediction accuracies, enabled by the action-space normalization strategies discussed in Sec. 3.1. These findings are demonstrated in Fig. 7, where CLAP-EE-Absolute generates future frames that are more closely aligned with the ground-truth compared to CLAP-EE-Relative. These performance gains hold over both shorter and longer prediction horizons.

Language-action Conditioning. In language-action-conditioned video modeling, relative-action spaces achieve superior fidelity in dynamics prediction compared to absolute actions (see Fig. 6). These findings likely

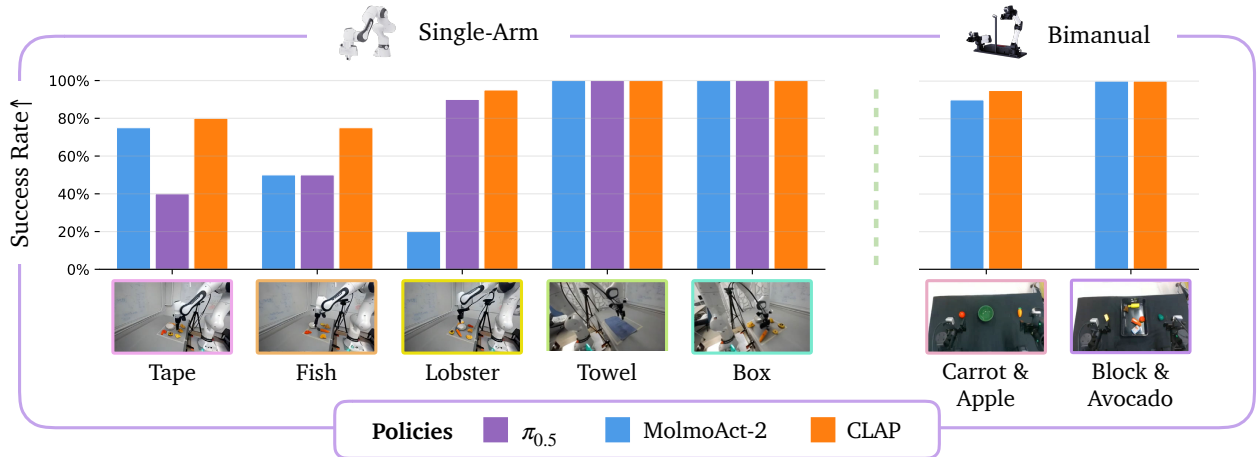


Figure 8 Inference-time cross-policy planning with CLAP. In contrast with prior work, CLAP optimizes over action proposals from multiple SOTA policies for robust robot manipulation across diverse tasks, boosting the success rates of $\pi_{0.5}$ and MolmoAct-2. Moreover, we stress-test CLAP in bimanual robot manipulation beyond its standard operational domain, demonstrating improvements in success rates on a bimanual YAM robot.

stem from the limited resolution of discrete tokenization spaces. Under these constraints, the compact bounds of relative actions utilize the tokenization space more effectively, yielding higher prediction accuracies. In contrast, the wider input distribution of absolute-action spaces limits prediction fidelity due to these bottlenecks. Fig. 7 visualizes these challenges. Unlike CLAP-LANG-Absolute, CLAP-LANG-Relative yields higher-fidelity predictions that approximately match the ground-truth.

4.3 Do cross-embodiment video models achieve zero-shot real-world generalization?

We investigate the ability of CLAP’s cross-embodiment models to generalize to real-world robot manipulation tasks without any additional data (i.e., in zero-shot). Prior work [51, 52] has demonstrated that single-embodiment video models are effective in test-time planning in robot manipulation tasks given a set of action proposals. However, generating a set of informative, yet diverse action proposals remains difficult. This challenge has spurred the design of heuristic-based strategies that perturb nominal actions from a single policy with random sampling and noise augmentation. However, these strategies suffer from lower execution quality and require high sample complexity for adequate coverage. To address these limitations, we deploy CLAP as a world modeling backbone in inference-time planning across multiple robot policies, generating a set of diverse candidate actions that typically contains optimal trajectories. We use CLAP-LAM as the base video model with a lightweight adapter to map end-effector actions to latent actions. This experiment design enables us to assess the lower bounds of performance among CLAP’s non-language-action models, given the more significant mismatch between end-effector and latent actions. We generate candidate actions from the SOTA policies $\pi_{0.5}$ [8] and MolmoAct-2 [9] across five tasks on the single-arm Franka Panda robot in a DROID environment. To examine the robustness of CLAP’s models beyond their normal operational boundaries (i.e., in single-arm robot manipulation), we stress-test them in bimanual robot manipulation. We discuss these results in the following subsections and provide additional experiments in finetuning robot manipulation policies with video-model-based reinforcement learning in Appendix C.4.

Single-arm Manipulation. We assess the success rate of $\pi_{0.5}$ and MolmoAct-2 across five tasks, spanning pick-and-place — with target objects *tape*, *fish*, *box*, *lobster* — and a towel-folding task. At inference time, CLAP’s planner predicts future frames conditioned on k sampled actions from each policy and scores each future trajectory using a VLM given the initial and final frames (consisting of left-camera and wrist-camera views). We set $k = 2$ with GPT-5 mini as the VLM, given its superior performance on the RoboRewardBench evaluation benchmark [53]. Thereafter, the robot executes the trajectory with the maximum predicted reward. Fig. 8 reports the success rates of CLAP’s planner compared to the baseline policies across 20 trials per policy on each task, demonstrating that CLAP matches or improves the base success rates of all policies across all tasks. Notably, the base performance of each policy varies significantly with the target object. This variation

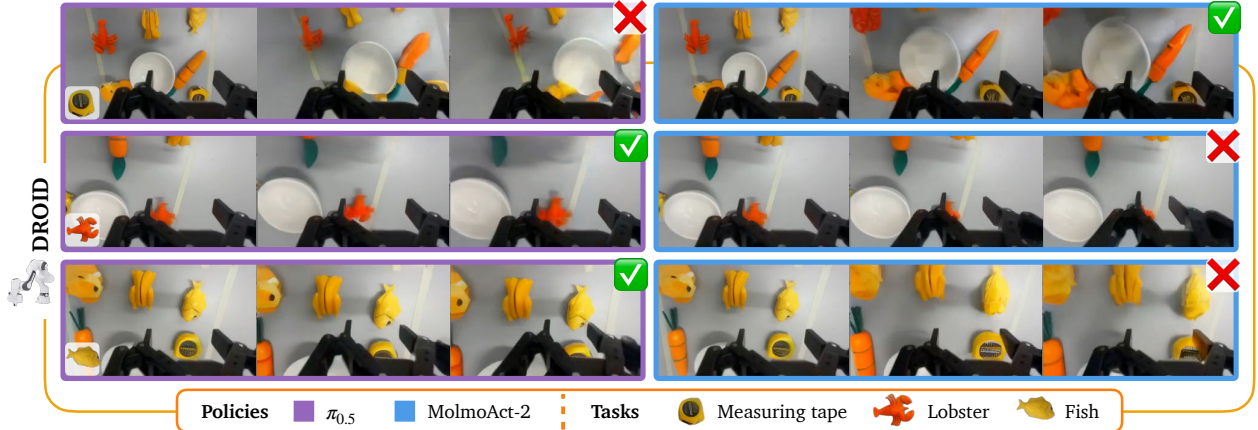


Figure 9 Inference-time cross-policy planning on single-arm Franka robot (DROID). CLAP predicts future frames conditioned on robot action proposals generated by multiple policies to maximize task-aligned rewards. (First row) In the *tape* task, CLAP identifies potential semantic confusion in $\pi_{0.5}$ and thus selects the more consistent action proposals from MolmoAct-2 for execution. (Second row) In the *lobster* task, CLAP detects a failure in MolmoAct-2’s semantic reasoning and thus executes actions from $\pi_{0.5}$. (Third row) In the *fish* task, CLAP selects the more promising action from $\pi_{0.5}$, although both policies’ action proposals are likely to succeed based on CLAP’s predictions. (The first frame per block is the initial frame; all other frames are future frames generated by CLAP.)

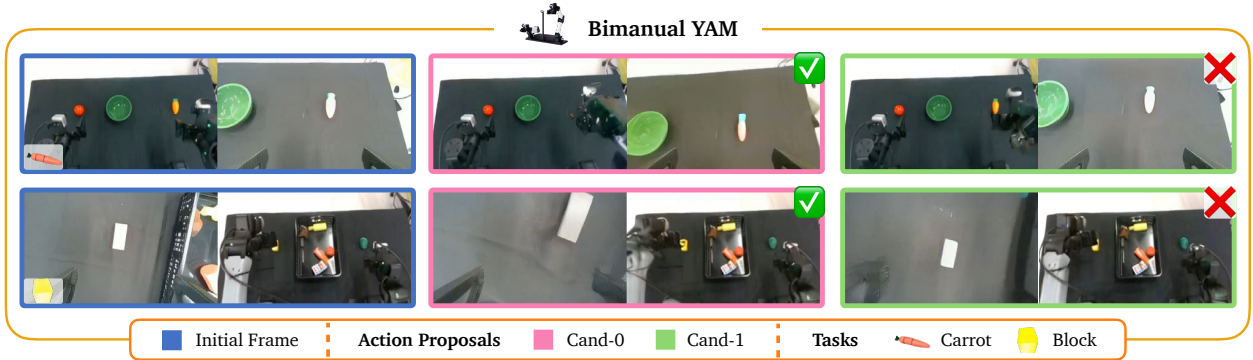


Figure 10 Inference-time cross-policy planning on bimanual YAM. Although CLAP was only trained on single-arm robot data, CLAP generates sufficiently accurate future frames conditioned on candidate robot actions for planning with bimanual robots. (First row) CLAP selects the candidate action that brings the right arm closer to the target object (*carrot*). (Second row) CLAP selects the action that makes greater progress towards the target object (*block*).

can be attributed to the brittle semantic reasoning capabilities of VLAs, especially in tasks involving relatively uncommon target objects. CLAP addresses this core limitation by combining the strengths of multiple policies to compensate for their individual weaknesses. This phenomenon is evident in the *tape*, *fish*, and *lobster* tasks. While MolmoAct-2 performs well on the *tape* task, it struggles on the *lobster* task; in contrast, $\pi_{0.5}$ performs well on the *lobster* task, but struggles on the *tape* task. Notably, even when both policies struggle on a common task, CLAP still achieves higher success rates, because planning is executed sequentially in a receding-horizon fashion, which ultimately improves robustness. Fig. 9 shows the predicted future trajectories generated by CLAP conditioned on candidate actions from both policies. The first frame per block is the initial frame; all other frames are future frames generated by CLAP. In the *tape* task (first row), CLAP detects potential semantic confusion in $\pi_{0.5}$ based on its sampled actions, and selects the actions proposed by MolmoAct-2 to maximize task-aligned rewards. Conversely, in the *lobster* task, CLAP identifies potential semantic failure in MolmoAct-2 but stronger task-alignment in $\pi_{0.5}$ ’s proposed actions. In the *fish* task, CLAP selects $\pi_{0.5}$ actions, which are more promising since they make better progress towards completing task.

Bimanual Manipulation. We evaluate the success rate of CLAP in inference-time planning with the bimanual YAM robot using MolmoAct-2 bimanual-YAM as the base policy. We consider two multi-stage tasks: “put

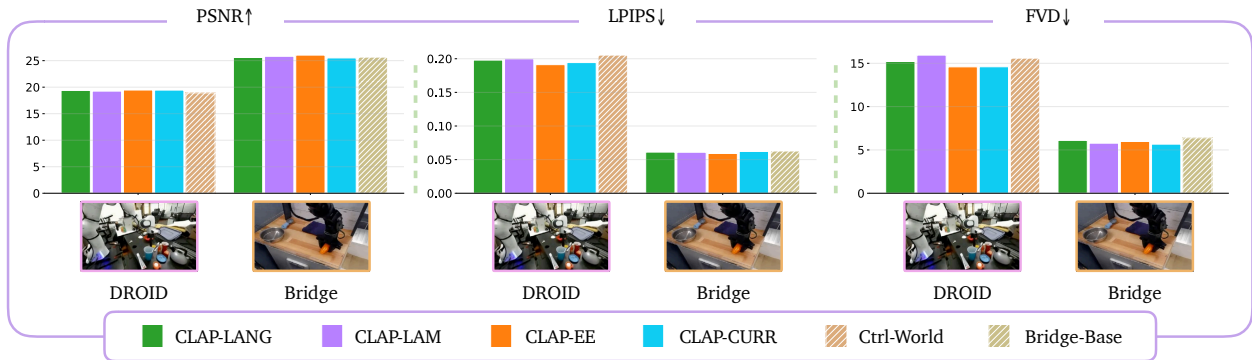


Figure 11 Data-efficient adaptation to pretrained embodiments. CLAP’s cross-embodiment models provide a crucial foundation for data-efficient adaptation to pretrained embodiments. With CLAP, all post-trained models surpass SOTA baselines on nearly all perceptual metrics.

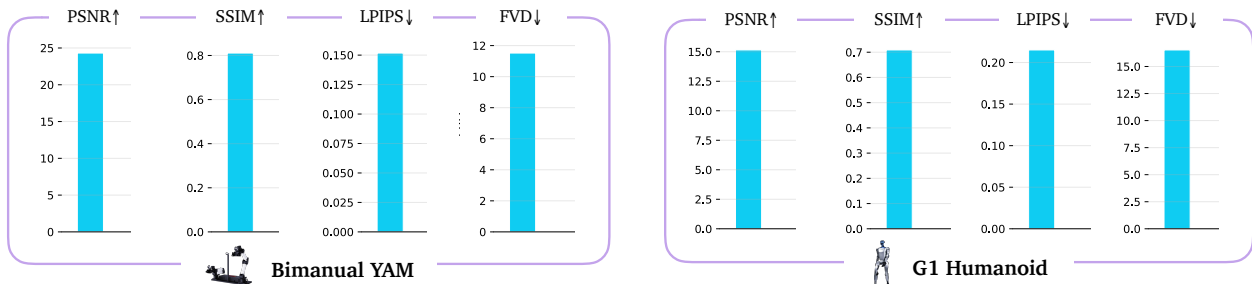


Figure 12 Few-shot adaptation to novel embodiments (bimanual YAM and G1 humanoid). CLAP facilitates few-shot adaptation of cross-embodiment models to the 14-dimensional and 26-dimensional action spaces of the bimanual YAM robot and G1 humanoid, respectively, for high-fidelity future prediction.

the <object A> in the bowl/bin, then put the <object B> in the bowl/bin,” with the following object pairs: {*carrot*, *apple*}, {*block*, *avocado*}. We alternate the order of each object per pair to assess both semantic understanding and the manipulation capability of the base policy. To address the mismatch in the action spaces of the 14-dimensional bimanual robot and 7-dimensional CLAP’s inputs, we dynamically select between the two arms, prioritizing the arm whose action chunk has a greater magnitude. The video model uses this information, along with other temporal and semantic cues, to resolve robot-arm motion in the generated videos. Fig. 8 summarizes the success rates of each method with 10 trials per task. Despite the gap between the CLAP’s normal operational domain and the test-time conditions, CLAP still achieves higher success rates compared to the base policy, with the only failure occurring when the planner dropped the second target object next to, but outside the bowl. From Fig. 10, CLAP generates future predictions accurate enough to distinguish promising action proposals from lower-quality candidates, driving its superior performance. We discuss model adaptation strategies for fine-grained bimanual manipulation in Sec. 4.4.

4.4 Beyond Zero-Shot: Does CLAP facilitate data-efficient adaptation to target embodiments?

We evaluate the effectiveness of CLAP as a foundation for few-shot adaptation of video world models to target embodiments. We ground our evaluations along two primary axes: pretrained embodiments and novel embodiments. In each setting, we drop incompatible components of the action head to match the native end-effector action space of the target embodiment but retain all other parameters.

Pretrained Embodiments. We finetune CLAP’s cross-embodiment models on the Bridge and DROID environments and benchmark their performance against SOTA single-embodiment baselines to assess the ability of the finetuned models to leverage physical priors from the cross-embodiment models for higher-fidelity dynamics prediction. Fig. 11 supports this hypothesis. Notably, all post-trained models surpass the SOTA baselines across nearly all perceptual metrics, suggesting a successful transfer of physical priors from the

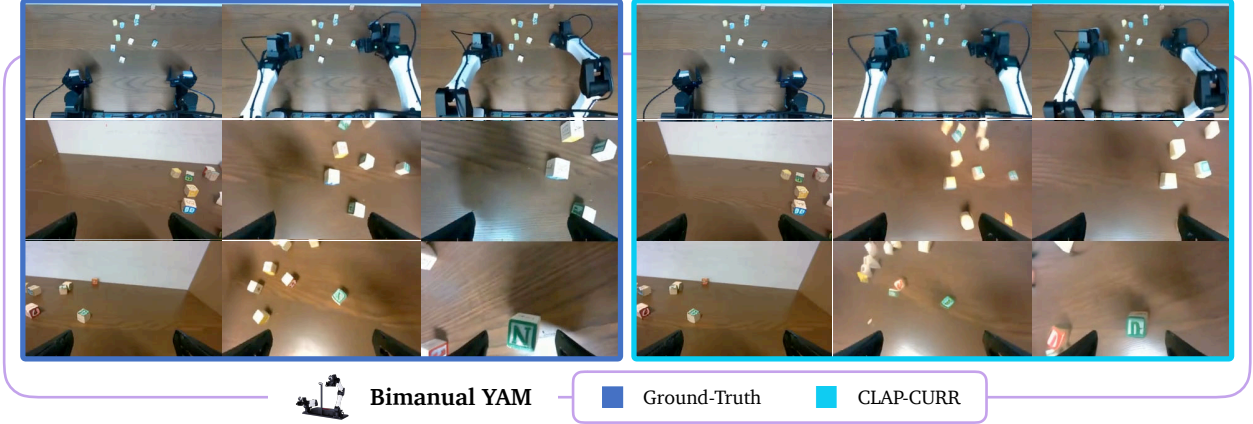


Figure 13 Few-shot adaptation to bimanual YAM. After few-shot finetuning, CLAP achieves high-fidelity future prediction on the bimanual YAM robot and accurately resolves each arm’s motion spatially and temporally.

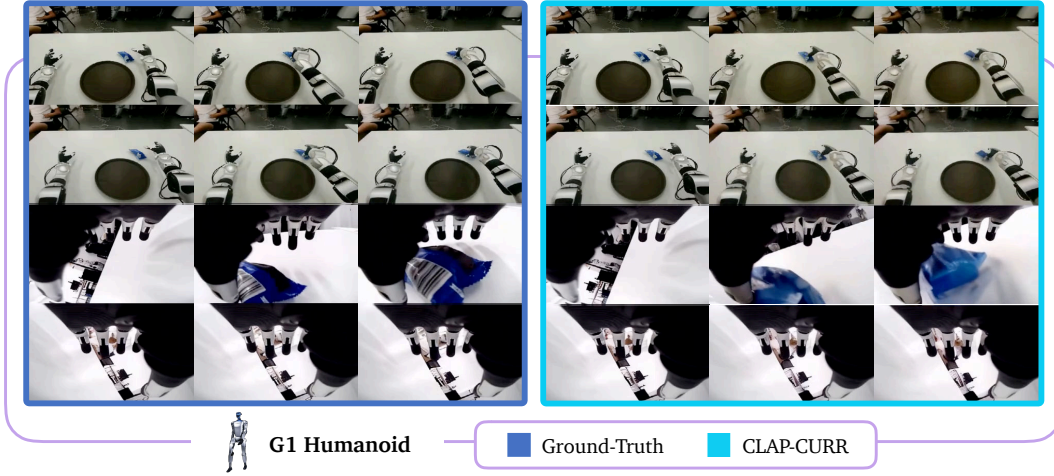


Figure 14 Few-shot adaptation to G1 humanoid. Using CLAP-CURR as the backbone, the finetuned video model achieves high-accuracy dynamics prediction on the G1 humanoid.

base models. In the DROID environment, finetuning bridges the gap between CLAP-LANG and other methods, highlighting the benefits of higher-resolution action-conditioning schemes for prediction fidelity. Similarly, in the Bridge environment, finetuning enables all cross-embodiment models to close the gap with and ultimately outperform the single-embodiment baseline. These findings establish a novel paradigm for training single-embodiment video world models. Concretely, our experiments show that finetuning cross-embodiment video models into single-embodiment variants is a more promising path towards high-fidelity dynamics prediction than the prevailing paradigm of training them from scratch or from less-aligned backbones like SVD [12] or WAN [11].

Novel Embodiments. Motivated by the generalizable physical priors encoded in cross-embodiment video models, we investigate their few-shot adaptation to novel embodiments, including bimanual robots and humanoids. Following the same finetuning framework, we adapt CLAP-CURR on the 14-dimensional action space of the bimanual YAM robot (7-DoF per arm) and the 26-dimensional action space of the G1 humanoid (7-DoF per arm and 6-DoF per hand). Fig. 12 summarizes the performance of each model. The bimanual YAM video model achieves significantly higher perceptual scores because the YAM dataset features less challenging, characterized by slower motion and smoother frame-to-frame transitions, compared to the more visually demanding G1 humanoid environment. In Fig. 13, the bimanual YAM video model accurately predicts the motion of each arm spatially and temporally, producing generated frames that closely match the ground-truth. Similarly, Fig. 14 demonstrates that the G1 humanoid video model achieves high-fidelity future prediction

of both arm and hand movements. These results underscore the effectiveness of CLAP as a foundational backbone for few-shot adaptation to novel embodiments.

5 Conclusion, Limitations, and Future Work

We present CLAP, a cross-embodiment action-conditioned video modeling framework that learns generalizable physical priors from heterogeneous human and robot videos. CLAP bridges the disparate action spaces of diverse embodiments with end-effector poses, language instructions, and learned latent action representations. To address the inherent limitations of these action representations, CLAP introduces a curriculum-based recipe that combines the strengths of latent actions and end-effector actions for data scaling with unlabeled video data while achieving zero action-space mismatch in real-world deployments. These core contributions yield cross-embodiment video models that approach or surpass state-of-the-art single-embodiment baselines. Further, CLAP establishes a novel framework for training single-embodiment action-conditioned video models through sample-efficient adaptation of cross-embodiment models. Crucially, we demonstrate CLAP’s zero-shot generalization to real-world tasks in inference-time cross-policy planning and reinforcement-learning-based policy finetuning in video world models.

Despite its high prediction fidelity, CLAP is still prone to hallucinations, like other video world models. Mitigating these hallucinations is critical for trustworthy integration in diverse robotics applications, such as planning, policy evaluation, and policy finetuning, presenting an exciting direction for future work on hallucination detection and mitigation, e.g., via uncertainty quantification [54]. Additionally, CLAP is primarily trained on single-embodiment robot data and human video data, extending this pipeline to bimanual and humanoid robot data would further scale the training data diversity, which could ultimately boost future-prediction accuracy and zero-shot generalization, constituting a promising path forward. Moreover, greater diversity in the training data could improve CLAP’s adaptation efficiency to novel embodiments. While model and data scaling provide immediate pathways for continuous improvement, they incur significant training and inference costs. Consequently, future research into efficient model architectures will be critical.

Ultimately, our work establishes cross-embodiment learning as a catalyst for breakthroughs in video world modeling, mirroring the paradigm shifts observed in large language models.

Safety, Data Privacy, and Consent

This paper presents work advancing the foundations of artificial intelligence (AI) in robotics. Because this work is primarily computational and theoretical, it does not involve human participants, live user studies, or personally identifiable information (PII); consequently, Institutional Review Board (IRB) approval was not required. All experiments were conducted using publicly available benchmarks and open-source datasets in accordance with their respective licenses. While broader applications of AI in robotics carry potential safety and societal implications, we do not foresee any direct, malicious, or high-risk negative societal consequences uniquely tied to this fundamental algorithmic research.

Acknowledgments

The authors would like to thank Prof. Anirudha Majumdar, Mingtong Zhang, and members of the Princeton IRoM lab for their patience, insightful discussions, and support.

References

- [1] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. URL <https://arxiv.org/abs/2302.13971>.
- [2] Aakanksha Chowdhery, Sharan Narang, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023. URL <https://arxiv.org/abs/2204.02311>.
- [3] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. URL <https://arxiv.org/abs/2001.08361>.
- [4] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 1877–1901, 2020. URL <https://arxiv.org/abs/2005.14165>.
- [5] Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-world: A controllable generative world model for robot manipulation. In *International Conference on Learning Representations (ICLR)*, 2026. URL <https://arxiv.org/abs/2510.10125>.
- [6] Julian Quevedo, Ansh Kumar Sharma, Yixiang Sun, Varad Suryavanshi, Percy Liang, and Sherry Yang. WorldGym: World model as an environment for policy evaluation. *arXiv preprint arXiv:2506.00613*, 2025.
- [7] Alexander Khazatsky, Karl Pertsch, Suraj Nair, et al. Droid: A large-scale in-the-wild robot manipulation dataset. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024. URL <https://arxiv.org/abs/2403.12945>.
- [8] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.
- [9] Haoquan Fang, Jiafei Duan, Donovan Clay, Sam Wang, Shuo Liu, Weikai Huang, Xiang Fan, Wei-Chuan Tsai, Shirui Chen, Yi Ru Wang, Shanli Xing, Jaemin Cho, Jae Sung Park, Ainaz Eftekhari, Peter Sushko, Karen Farley, Angad Wadhwa, Cole Harrison, Winson Han, Ying-Chun Lee, Eli VanderBilt, Rose Hendrix, Suveen Ellawela, Lucas Ngoo, Joyce Chai, Zhongzheng Ren, Ali Farhadi, Dieter Fox, and Ranjay Krishna. Molmoact2: Action reasoning models for real-world deployment. *arXiv preprint arXiv:2605.02881*, 2026. URL <https://arxiv.org/abs/2605.02881>.
- [10] NVIDIA. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025. URL <https://arxiv.org/abs/2501.03575>.
- [11] Team Wan. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. URL <https://arxiv.org/abs/2503.20314>.
- [12] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. URL <https://arxiv.org/abs/2311.15127>.
- [13] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. *OpenAI Technical Report*, 2024.
- [14] Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, Yen-Chen Lin, Loic Magne, Ajay Mandlekar, Avnish Narayan, You Liang Tan, Guanzhi Wang, Jing Wang, Qi Wang, Yinzhen Xu, Xiaohui Zeng, Kaiyuan Zheng, Ruijie Zheng, Ming-Yu Liu, Luke Zettlemoyer, Dieter Fox, Jan Kautz, Scott Reed, Yuke Zhu, and Linxi Fan. Dreamgen: Unlocking generalization in robot learning through video world models. In *Proceedings of The 9th Conference on Robot Learning (CoRL)*, volume 305 of *Proceedings of Machine Learning Research*, pages 5170–5194, 2025. URL <https://arxiv.org/abs/2505.12705>.
- [15] Homanga Bharadhwaj, Debidatta Dwibedi, Abhinav Gupta, Shubham Tulsiani, Carl Doersch, Ted Xiao, Dhruv Shah, Fei Xia, Dorsa Sadigh, and Sean Kirmani. Gen2act: Human video generation in novel scenarios enables

- generalizable robot manipulation. *arXiv preprint arXiv:2409.16283*, 2024. URL <https://arxiv.org/abs/2409.16283>.
- [16] Yilun Du, Mengjiao Yang, Pete Florence, Fei Xia, Ayzaan Wahid, Brian Ichter, Pierre Sermanet, Tianhe Yu, Pieter Abbeel, Joshua B. Tenenbaum, Leslie Kaelbling, Andy Zeng, and Jonathan Tompson. Video language planning. In *International Conference on Learning Representations (ICLR)*, 2024. URL <https://arxiv.org/abs/2310.10625>.
 - [17] Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pages 24328–24346, 2025. URL <https://arxiv.org/abs/2412.14803>.
 - [18] Yue Liao, Pengfei Zhou, Siyuan Huang, Donglin Yang, Shengcong Chen, Yuxin Jiang, Yue Hu, Jingbin Cai, Si Liu, Jianlan Luo, Liliang Chen, Shuicheng Yan, Maoqing Yao, and Guanghui Ren. Genie envisioner: A unified world foundation platform for robotic manipulation. *arXiv preprint arXiv:2508.05635*, 2025. URL <https://arxiv.org/abs/2508.05635>.
 - [19] Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. In *Proceedings of Robotics: Science and Systems (RSS)*, 2025.
 - [20] Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. In *Proceedings of Robotics: Science and Systems (RSS)*, 2025.
 - [21] Zhiting Mei, Tenny Yin, Ola Shorinwa, Apurva Badithela, Zhonghe Zheng, Joseph Bruno, Madison Bland, Lihan Zha, Asher Hancock, Jaime Fernández Fisac, et al. Video generation models in robotics-applications, research challenges, future directions. *arXiv preprint arXiv:2601.07823*, 2026.
 - [22] Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. IRASim: Learning interactive real-robot action simulators. *arXiv preprint arXiv:2406.14540*, 2024.
 - [23] Wei-Cheng Tseng, Jinwei Gu, Qinsheng Zhang, Hanzi Mao, Ming-Yu Liu, Florian Shkurti, and Lin Yen-Chen. Scalable policy evaluation with video world models. *arXiv preprint arXiv:2511.11520*, 2025.
 - [24] Gemini Robotics Team, Coline Devin, Yilun Du, Debidatta Dwivedi, Ruiqi Gao, Abhishek Jindal, Thomas Kipf, Sean Kirmani, Fangchen Liu, Anirudha Majumdar, et al. Evaluating Gemini robotics policies in a Veo world simulator. *arXiv preprint arXiv:2512.10675*, 2025.
 - [25] Arnav Kumar Jain, Yilin Wu, Jesse Farebrother, Gokul Swamy, and Andrea Bajcsy. Weaver, better, faster, longer: An effective world model for robotic manipulation. *arXiv preprint arXiv:2606.13672*, 2026. URL <https://arxiv.org/abs/2606.13672>.
 - [26] Tenny Yin, Zhiting Mei, Zhonghe Zheng, Miyu Yamane, David Wang, Jade Sceats, Samuel M. Bateman, Lihan Zha, Apurva Badithela, Ola Shorinwa, and Anirudha Majumdar. Playworld: Learning robot world models from autonomous play, 2026. URL <https://arxiv.org/abs/2603.09030>.
 - [27] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. In *Proceedings of The 8th Conference on Robot Learning (CoRL)*, volume 270 of *Proceedings of Machine Learning Research*, pages 2679–2713, 2024. URL <https://arxiv.org/abs/2406.09246>.
 - [28] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0 : A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.
 - [29] Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, Ya-Qin Zhang, Jiangmiao Pang, Jingjing Liu, Tai Wang, and Xianyu Zhan. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. In *International Conference on Learning Representations (ICLR)*, 2026. URL <https://arxiv.org/abs/2510.10274>.
 - [30] Lihan Zha, Asher J. Hancock, Mingtong Zhang, Tenny Yin, Yixuan Huang, Dhruv Shah, Allen Z. Ren, and Anirudha Majumdar. Lap: Language-action pre-training enables zero-shot cross-embodiment transfer. *arXiv preprint arXiv:2602.10556*, 2026. URL <https://arxiv.org/abs/2602.10556>.
 - [31] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi,

- Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems (RSS)*, Delft, Netherlands, 2024. URL <https://arxiv.org/abs/2405.12213>.
- [32] Lirui Wang, Xinlei Chen, Jialiang Zhao, and Kaiming He. Scaling proprioceptive-visual learning with heterogeneous pre-trained transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, 2024. URL <https://arxiv.org/abs/2409.20537>. Spotlight.
- [33] NVIDIA. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025. URL <https://arxiv.org/abs/2503.14734>.
- [34] Ria Doshi, Homer Walke, Oier Mees, Sudeep Dasari, and Sergey Levine. Scaling cross-embodied learning: One policy for manipulation, navigation, locomotion and aviation. In *Proceedings of The 8th Conference on Robot Learning (CoRL)*, volume 270 of *Proceedings of Machine Learning Research*, pages 496–512, 2024. URL <https://arxiv.org/abs/2408.11812>.
- [35] Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions. In *Proceedings of Robotics: Science and Systems (RSS)*, 2025. URL <https://arxiv.org/abs/2505.06111>.
- [36] Jinliang Zheng, Jianxiong Li, Dongxiu Liu, Yanan Zheng, Zhihao Wang, Zhonghong Ou, Yu Liu, Jingjing Liu, Ya-Qin Zhang, and Xianyu Zhan. Universal actions for enhanced embodied foundation models. *arXiv preprint arXiv:2501.10105*, 2025. URL <https://arxiv.org/abs/2501.10105>.
- [37] Anthony Liang, Pavel Czempin, Matthew Hong, Yutai Zhou, Erdem Biyik, and Stephen Tu. Clam: Continuous latent action models for robot learning from unlabeled demonstrations. *arXiv preprint arXiv:2505.04999*, 2025. URL <https://arxiv.org/abs/2505.04999>.
- [38] Xiaoyu Chen, Hangxing Wei, Pushi Zhang, Chuheng Zhang, Kaixin Wang, Yanjiang Guo, Rushuai Yang, Yucen Wang, Xinquan Xiao, Li Zhao, Jianyu Chen, and Jiang Bian. villa-x: Enhancing latent action modeling in vision-language-action models. *arXiv preprint arXiv:2507.23682*, 2025. URL <https://arxiv.org/abs/2507.23682>.
- [39] Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Bechtle, Feryal Behbahani, Stephanie Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando de Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pages 4603–4623, 2024. URL <https://arxiv.org/abs/2402.15391>. Best Paper Award.
- [40] Seonghyeon Ye, Joel Jang, Byeongguk Jeon, SeJune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, Lars Liden, Kimin Lee, Jianfeng Gao, Luke Zettlemoyer, Dieter Fox, and Minjoon Seo. Latent action pretraining from videos. In *International Conference on Learning Representations (ICLR)*, 2025. URL <https://arxiv.org/abs/2410.11758>.
- [41] Abby O’Neill, Abdul Rehman, Abhiram Maddukuri, et al. Open x-embodiment: Robotic learning datasets and rt-x models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903, 2024. doi: 10.1109/ICRA57147.2024.10611477.
- [42] Shenyuan Gao, William Liang, Kaiyuan Zheng, Ayaan Malik, Seonghyeon Ye, Sihyun Yu, Wei-Cheng Tseng, Yuzhu Dong, Kaichun Mo, Chen-Hsuan Lin, Qianli Ma, Seungjun Nah, Loic Magne, Jiannan Xiang, Yuqi Xie, Ruijie Zheng, Dantong Niu, You Liang Tan, K. R. Zentner, George Kurian, Suneel Indupuru, Pooya Jannaty, Jinwei Gu, Jun Zhang, Jitendra Malik, Pieter Abbeel, Ming-Yu Liu, Yuke Zhu, Joel Jang, and Linxi ”Jim” Fan. Dreamdojo: A generalist robot world model from large-scale human videos. In *International Conference on Machine Learning (ICML)*, 2026. URL <https://arxiv.org/abs/2602.06949>.
- [43] Asher J. Hancock, Kindi Wu, Lihan Zha, Olga Russakovsky, and Anirudha Majumdar. Actions as language: Fine-tuning vlms into vlms without catastrophic forgetting. In *International Conference on Learning Representations (ICLR)*, 2026. URL <https://arxiv.org/abs/2509.22195>.
- [44] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 2022. URL <https://arxiv.org/abs/2206.00364>.

- [45] Ryan Hoque, Peide Huang, David J. Yoon, Mouli Sivapurapu, and Jian Zhang. Egodex: Learning dexterous manipulation from large-scale egocentric video. In *International Conference on Learning Representations (ICLR)*, 2026. URL <https://arxiv.org/abs/2505.11709>.
- [46] Alain Horé and Djemel Ziou. Image quality metrics: Psnr vs. ssim. *2010 20th International Conference on Pattern Recognition*, pages 2366–2369, 2010. URL <https://api.semanticscholar.org/CorpusID:9506273>.
- [47] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018. URL <https://arxiv.org/abs/1801.03924>.
- [48] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. In *International Conference on Learning Representations (ICLR) Workshop*, 2019. URL <https://arxiv.org/abs/1812.01717>.
- [49] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017. URL <https://arxiv.org/abs/1706.08500>.
- [50] NVIDIA. World simulation with video foundation models for physical ai. *arXiv preprint arXiv:2511.00062*, 2026. URL <https://arxiv.org/abs/2511.00062>.
- [51] Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, and Jinwei Gu. Cosmos policy: Fine-tuning video models for visuomotor control and planning. *arXiv preprint arXiv:2601.16163*, 2026. URL <https://arxiv.org/abs/2601.16163>.
- [52] Zhiting Mei, Yanbo Xu, Tenny Yin, Ola Sho, and Anirudha Majumdar. Inference-time planning with action-conditioned video models for generalizable robot manipulation. In *The 2nd Workshop on Foundation Models Meet Embodied Agents at CVPR 2026*, 2026. URL <https://openreview.net/forum?id=gkSuORauJ4>.
- [53] Tony Lee, Andrew Wagenmaker, Karl Pertsch, Percy Liang, Sergey Levine, and Chelsea Finn. Roboreward: General-purpose vision-language reward models for robotics. *arXiv preprint arXiv:2601.00675*, 2026. URL <https://arxiv.org/abs/2601.00675>.
- [54] Zhiting Mei, Tenny Yin, Micah Baker, Ola Shorinwa, and Anirudha Majumdar. World models that know when they don’t know - controllable video generation with calibrated uncertainty. *arXiv preprint arXiv:2512.05927*, 2026. URL <https://arxiv.org/abs/2512.05927>.
- [55] Andrew Wagenmaker, Mitsuhiro Nakamoto, Yunchu Zhang, Seohong Park, Waleed Yagoub, Anusha Nagabandi, Abhishek Gupta, and Sergey Levine. Steering your diffusion policy with latent space reinforcement learning. *arXiv preprint arXiv:2506.15799*, 2025. URL <https://arxiv.org/abs/2506.15799>.

Appendix

Table of Contents

A Preliminaries	20
B Method	20
B.1 Action Representations	20
B.2 Model Architecture	20
C Experiments	21
C.1 Evaluation Setup	21
C.2 Inference and Timing Results	22
C.3 Additional Results on Latent Action Models	22
C.4 Additional Results on Cross-Embodiment Video Modeling	23
D Nuanced Summary	27

A Preliminaries

Video world models predict future outcomes conditioned on per-frame actions, starting from an initial camera observation. Furthermore, to capture temporal context such as robot velocities that dictate future evolution, video world models are often conditioned on a history of past observations.

In our work, the video world model $\mathcal{V}_\theta$ takes as input a language instruction l , a sequence of observations spanning past history and the current frame $\{o_\tau\}_{\tau=t-H}^t$, and a sequence of future per-frame actions $\{a_\tau\}_{\tau=t}^{t+P}$. From these inputs, $\mathcal{V}_\theta$ predicts the resulting sequence of future frames $\{o_\tau\}_{\tau=t+1}^{t+P}$ via:

$$\{o_\tau\}_{\tau=t+1}^{t+P} \sim \mathcal{V}_\theta(\cdot \mid l, \{o_\tau\}_{\tau=t-H}^t, \{a_\tau\}_{\tau=t}^{t+P}), \quad (3)$$

where H and P denote the history and prediction horizons, respectively.

B Method

B.1 Action Representations

End-effector Actions. Each demonstration is stored as a trajectory of end-effector states aligned to the video frames. A single state is a 7-dimensional vector $s_t = [x_t, y_t, z_t, \phi_t, \theta_t, \psi_t, g_t]$, where (x_t, y_t, z_t) is the gripper position in meters in the robot base frame, $(\phi_t, \theta_t, \psi_t)$ are the roll-pitch-yaw orientation Euler angles in radians, and g_t is the raw gripper signal. Depending on the evaluation setup, end-effector actions are computed either in absolute coordinates or relative to the initial frame of the history window. CLAP-EE directly encodes s_t into a 1024-dimensional vector which is passed into the video model as a conditioning signal.

Language Actions.. CLAP-LANG maps per-frame end-effector actions to natural language, producing a single natural-language string per timestep via the following template:

Language Action Format

x= $\langle \rangle$, y= $\langle \rangle$, z= $\langle \rangle$, roll= $\langle \rangle$, pitch= $\langle \rangle$, yaw= $\langle \rangle$, gripper $\langle g \rangle$

We express the gripper position numerically, normalized between zero and one. Tab. 1 summarizes how we compute the normalized gripper position g_t from the provided gripper signal $\bar{g}_t$ in each dataset. Depending on the action-conditioning space, language actions are expressed either in absolute coordinates or relative to the current frame.

Latent Actions. We infer latent actions directly from the observed video. The latent action model (see Appendix B.2) encodes the transition between consecutive frames into a compact continuous latent vector, yielding a trajectory of T latent actions $z_t \in \mathbb{R}^{32}$, stored as a per-episode array aligned with the video frames. Because these actions are learned purely from observation, they are embodiment- and sensor-agnostic, providing a uniform action signal that is defined even for datasets whose proprioceptive action labels are noisy, inconsistent, or unavailable.

B.2 Model Architecture

Latent Action Model. We model latent actions with a variational autoencoder over consecutive video frames, following the latent-action formulation of Genie [39]. Each training example is a clip of $T = 2$ RGB frames at 240×320 resolution, with pixel values normalized to $[0, 1]$. Frames are split into non-overlapping 16×16 patches, yielding a $15 \times 20 = 300$ -token grid per frame, where each token is the flattened patch of dimension $3 \cdot 16^2 = 768$. We denote the resulting patch tensor $\mathbf{P} \in \mathbb{R}^{B \times T \times N \times d_p}$ with $N = 300$ and $d_p = 768$. The flattened patch is fed into the encoder, which is a factorized spatio-temporal transformer of $L_{\text{enc}} = 24$ blocks operating at model width $d = 1024$ with 16 attention heads. The encoder produces a d -dimensional action-prompt token at the future frame(s).

Table 1 Per-dataset gripper normalization to a common gripper position signal $g_t \in [0, 1]$. The width scheme uses the maximum gripper width $w_{\max} = 0.08$ m.

Scheme	Raw signal $\bar{g}_t$	Gripper position g_t	Datasets
Normal	normalized $[0, 1]$, high = open	$g_t = \bar{g}_t$	bridge, fractal, droid
Flipped	normalized $[0, 1]$, high = closed	$g_t = 1 - \bar{g}_t$	bc.z, fmb
Width	gripper width (m), wide = open	$g_t = \bar{g}_t / w_{\max}$	furniture.bench, taco_play, austin_sailor, stanford_hydra, utaustin_mutex
Command	signed open/close events $\{-1, 0, +1\}$	integrate from open: $+1 \rightarrow \text{close}, -1 \rightarrow \text{open}$	berkeley_autolab_ur5

The model is trained with a standard β -VAE objective, combining pixel reconstruction and a KL prior-matching term:

$$\mathcal{L} = \underbrace{\|x_{t+1} - \hat{x}_{t+1}\|_2^2}_{\text{MSE reconstruction}} + \beta \cdot D_{\text{KL}}(\mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\sigma}_t^2) \parallel \mathcal{N}(0, I)), \quad (4)$$

with a small $\beta = 10^{-6}$, which prioritizes reconstruction fidelity while still regularizing the latent space toward a unit Gaussian.

We optimize with AdamW (learning rate $\text{lr} = 5 \times 10^{-6}$, weight decay 10^{-2} , gradient clipping at 0.3) under mixed-precision (fp16) on 8 GPUs with the distributed data parallel framework, at a global batch size of 64. The model is fine-tuned from a checkpoint pre-trained for 400K steps. As reported in Tab. 2, the training data is a weighted mixture of 10 robot- and egocentric-manipulation video datasets sampled with dataset-specific mixture weights and per-dataset frame-skip and multi-view stacking settings to normalize frame rate and camera layout across sources.

Video Model. CLAP uses the SVD spatio-temporal UNet as its video backbone, interleaving spatial and temporal attention across four resolution stages with cross-attention dimension 1024 and input channel dimension of 8: four for the noised target latents and four for a per-frame conditioning latent. The frozen VAE encoder $\mathcal{E}$ maps each RGB frame to a latent $z_t \in \mathbb{R}^{4 \times H/8 \times W/8}$, and all generation occurs in this latent space. We split a window of T frames into $T_h = 6$ history frames and $T_f = 5$ future frames, and train the model to denoise the future latents given the history $z_{1:T_h}$, the current observation z_{T_h} , and actions $a_{1:T}$. We use EDM noise sampling [44], with $\log \sigma \sim \mathcal{N}(P_{\text{mean}}, P_{\text{std}}^2)$, $P_{\text{mean}}=0.7$, $P_{\text{std}}=1.6$.

A single interface maps any per-frame control signal to the 1024-dimensional token space of the UNet cross-attention layers. End-effector and latent actions are encoded by a three-layer MLP with SiLU activations (with input dimensions 7 and 32 for end-effector actions and latent actions, respectively); per-frame language actions are encoded by a frozen CLIP text encoder. An optional task-level instruction embedding is added to the per-frame tokens, when available in the dataset. Critically, conditioning is applied at the *frame level*: each generated frame attends to its own action token rather than a shared global token, which is what makes the model controllable. The action context is dropped with probability 5% during training to enable classifier-free guidance.

During training, the VAE and CLIP encoder stay frozen; only the UNet is trained, using AdamW at learning rate 10^{-5} , gradient clipping 1.0, and mixed precision. At inference, we use the EDM sampler with 50 steps, frame-wise classifier-free guidance, and autoregressive chunked rollout.

C Experiments

C.1 Evaluation Setup

CLAP trains cross-embodiment video models on diverse datasets using the sampling ratios in Tab. 3, proportional to the complexity and quality of the individual datasets. Multi-view camera images are vertically stacked into a single fixed-size frame (576×320) after resizing individual frames to 192×320 . For uniformity, we repeat single-view camera inputs in datasets without multi-view images, although preliminary experiments

Table 2 Training data mixture and per-dataset sampling weights for LAM.

Training Dataset	Ratio
EgoDex [45]	30.0%
Bridge [41]	14.8%
Fractal [41]	14.2%
DROID [41]	11.1%
BC-Z [41]	7.5%
FMB [41]	7.1%
Language Table [41]	4.4%
Taco Play [41]	3.0%
Furniture Bench [41]	2.4%
RoboTurk [41]	2.3%

Table 3 Training data mixture and per-dataset sampling weights for the video models.

Training Dataset	Ratio
EgoDex [45]	2.5%
Bridge [41]	15.0%
Fractal [41]	1.50%
DROID [41]	75.0%
BC-Z [41]	1.50%
FMB [41]	1.50%
Taco Play [41]	1.50%
Furniture Bench [41]	1.50%

revealed that such repetition is unnecessary. In real-world experiments that directly utilize **CLAP-LAM**, we train a lightweight adapter that maps end-effector actions into the 32-dimensional latent-action space, which serves as conditioning inputs for the video model. Its output head is zero-initialized; while the adapter is trained, the world model is kept frozen. All models are trained on the Open X-Embodiment (OXE) datasets, with **CLAP-LAM** also incorporating the EgoDex dataset. The OXE-Mix training split includes the Bridge, DROID, FMB, Furniture Bench, and Taco Play datasets. We evaluate the trained models on the held-out test or validation splits with about 100 trajectories each, and additionally report their average performance across the Bridge, DROID, and Taco Play datasets. Real-world table-top manipulation experiments are performed on the Franka Emika robot in the DROID configuration with two side cameras and a wrist camera and a Bimanual YAM robot setup with a top-view camera and a wrist-camera view per arm.

C.2 Inference and Timing Results

CLAP requires under 12 GB of VRAM at inference and can comfortably fit on consumer hardware like the RTX 3060, with each inference call consuming approximately 9.7 GB (verified on the A100 and H200 GPU nodes). All timing measurements are evaluated with a nominal prediction of 11 total frames and 25 denoising steps, measured across 20 trials. At initialization, the timing for the first inference call depends on the status of the cuDNN kernel-autotuning cache: if a previous process has already initialized the kernel, the first call takes about 3 seconds, but takes roughly 15 seconds otherwise. Subsequent inference times vary by GPU architecture, averaging $3.24s \pm 0.02s$ on the A100-PCIe-40GB, $2.88s \pm 0.00s$ on the A100-SXM4-80GB, and $1.49s \pm 0.00s$ on the H200.

C.3 Additional Results on Latent Action Models

We train latent action models (LAMs) on our cross-embodiment dataset and individual single-embodiment subdatasets (Bridge and DROID) and evaluate the consistency of the proxy actions relative to the ground-truth trajectory. Using LAMs as inverse-dynamics models (IDMs), we extract latent actions for unseen trajectories, and reconstruct the ground-truth trajectories using the computed latent actions. We benchmark these models against the SOTA DreamDojo IDM baseline on the val/test splits of the in-domains datasets (i.e., datasets seen during training, e.g., Bridge, DROID, and OXE-Mix) and on the following held-out datasets (i.e., unseen data): “austin.sailor,” “utaustin_mutex,” “berkeley_ur5,” and “stanford_hydra,” which include robot morphologies (e.g., the UR5 arm) that were not seen during training.

We evaluate the frame-to-frame prediction accuracy (Fig. 15) and the open-loop rollout accuracy with a skip of two frames (Fig. 16). While single-embodiment IDMs perform the best on their target datasets, their performance degrades sharply outside of these datasets, which limits their generalization. In contrast, CLAP’s IDM generalizes across all datasets and achieves the highest or second-highest perceptual scores, e.g., on the held-out dataset and OXE-mix. These findings hold for both the frame-to-frame reconstruction and open-loop rollout results. In essence, CLAP’s IDM offers better generalization without compromising in-domain performance, compared to all other methods.

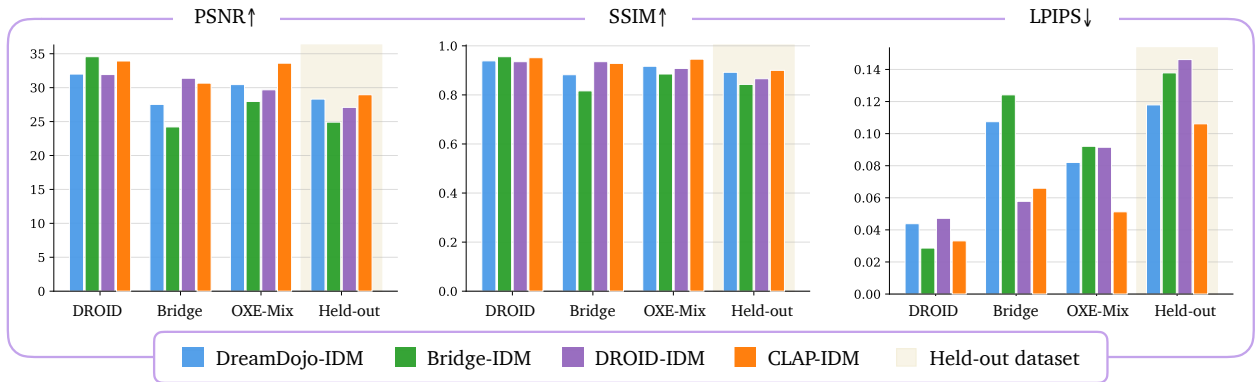


Figure 15 Frame-to-frame reconstruction quality. Each LAM is used as an inverse-dynamics model to extract a latent action between two consecutive ground-truth frames, which is then decoded to reconstruct the next frame. While single-embodiment IDMs perform best on their native distributions, their accuracy degrades sharply out-of-domain. In contrast, CLAP’s IDM consistently outperforms these off-domain baselines while matching or closely trailing single-embodiment models on their target datasets.

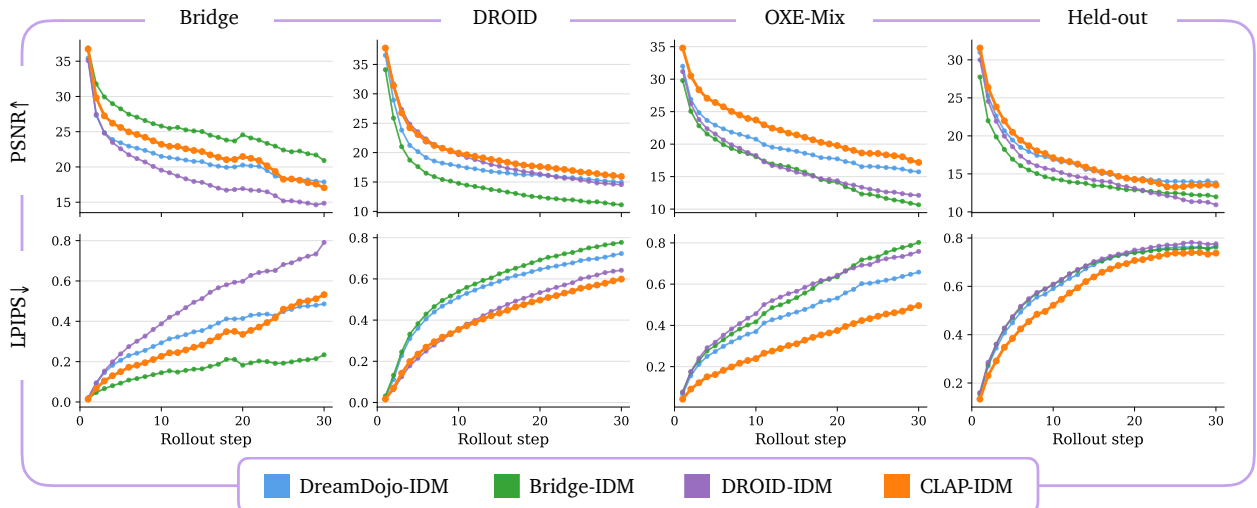


Figure 16 LAMs’ autoregressive rollouts. Starting from a single ground-truth frame, we rollout each LAM for 30 steps using its own decoded predictions. Compared to all other methods, CLAP’s IDM achieves superior generalization beyond the domain of the training datasets.

C.4 Additional Results on Cross-Embodiment Video Modeling

Finetuning Robot Policies via Video Model-based Reinforcement Learning. CLAP facilitates policy finetuning in video world models with reinforcement learning (RL). We use the Diffusion Steering via Reinforcement Learning (DSRL) framework [55] to finetune diffusion policies within CLAP’s video models and evaluate their success rates in the following real-world tasks: (i) put a carrot in a bowl, and (ii) fold a towel. We predict task progress using a simple convolutional neural network to estimate dense per-frame reward signals in RL. We evaluate the base and finetuned policies across 25 and 20 trials in the *carrot* and *towel* tasks, respectively. From Fig. 17, CLAP improves the success rate of the base diffusion policy on the *carrot* task via RL in the video model (from 80% to 88%), without degrading its performance on the *towel* task. These results demonstrate CLAP’s potential for cost-effective policy finetuning, circumventing the significant hardware and setup costs associated with alternative pipelines, such as real-world or simulation-based RL. Exploring these directions constitute an exciting avenue for future work.

Comprehensive Experiment Results. Tabs. 4 to 15 report the results for all experiment.

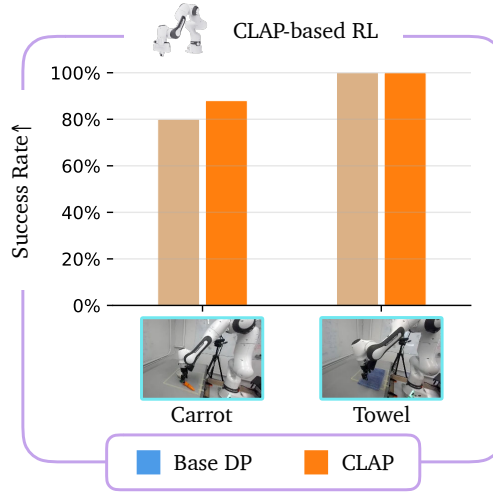


Figure 17 Policy Finetuning via RL in CLAP. CLAP facilitates policy finetuning with reinforcement learning (RL) in video world models. By finetuning base diffusion policies (Base DP), CLAP achieves higher success rates in the *carrot* task, while matching the base policy’s success rate in the *towel* task.

Table 4 Performance of CLAP’s cross-embodiment video world models on DROID.

Method	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	FVD $\downarrow$	FID $\downarrow$
DreamDojo-Human	12.450 $\pm$ 1.691	0.384 $\pm$ 0.080	0.681 $\pm$ 0.089	116.009	241.056
CLAP-LANG	17.200 $\pm$ 2.740	0.692 $\pm$ 0.087	0.260 $\pm$ 0.069	21.517	39.168
CLAP-LAM	18.859 $\pm$ 2.770	0.729 $\pm$ 0.081	0.222 $\pm$ 0.068	19.059	36.863
CLAP-EE	18.835 $\pm$ 2.837	0.734 $\pm$ 0.080	0.215 $\pm$ 0.069	16.370	33.075
CLAP-CURR	19.138 $\pm$ 2.671	0.744 $\pm$ 0.078	0.204 $\pm$ 0.064	16.139	32.484
Ctrl-World	18.928 $\pm$ 2.760	0.736 $\pm$ 0.081	0.205 $\pm$ 0.067	15.591	30.543

Table 5 Performance of CLAP’s cross-embodiment video world models on Bridge.

Method	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	FVD $\downarrow$	FID $\downarrow$
DreamDojo-Human	14.376 $\pm$ 1.220	0.384 $\pm$ 0.079	0.426 $\pm$ 0.089	73.442	75.060
CLAP-LANG	20.548 $\pm$ 2.291	0.780 $\pm$ 0.049	0.142 $\pm$ 0.052	17.322	25.761
CLAP-LAM	24.179 $\pm$ 2.032	0.827 $\pm$ 0.039	0.087 $\pm$ 0.035	9.207	20.755
CLAP-EE	23.589 $\pm$ 2.442	0.822 $\pm$ 0.043	0.091 $\pm$ 0.041	10.119	19.698
CLAP-CURR	23.692 $\pm$ 2.415	0.828 $\pm$ 0.041	0.088 $\pm$ 0.040	9.614	19.205
Bridge-Base	25.660 $\pm$ 2.294	0.858 $\pm$ 0.038	0.063 $\pm$ 0.031	6.402	13.307

Table 6 Performance of CLAP’s cross-embodiment video world models on OXE-Mix.

Method	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	FVD $\downarrow$	FID $\downarrow$
DreamDojo-Human	14.629 $\pm$ 2.793	0.421 $\pm$ 0.109	0.473 $\pm$ 0.189	102.601	159.744
CLAP-LANG	19.705 $\pm$ 2.825	0.765 $\pm$ 0.075	0.169 $\pm$ 0.077	37.010	50.749
CLAP-LAM	22.605 $\pm$ 3.391	0.806 $\pm$ 0.071	0.125 $\pm$ 0.075	20.030	43.397
CLAP-EE	21.665 $\pm$ 2.955	0.798 $\pm$ 0.065	0.129 $\pm$ 0.068	26.199	41.854
CLAP-CURR	21.898 $\pm$ 2.879	0.806 $\pm$ 0.062	0.125 $\pm$ 0.064	23.202	41.218

Table 7 Effects of relative and absolute action-space conditioning on cross-embodiment video modeling (DROID).

Method	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	FVD $\downarrow$
CLAP-EE-Rel	16.758 $\pm$ 2.628	0.697 $\pm$ 0.074	0.257 $\pm$ 0.066	22.113
CLAP-EE-Abs	18.835 $\pm$ 2.837	0.734 $\pm$ 0.080	0.215 $\pm$ 0.069	16.370
CLAP-LANG-Abs	16.290 $\pm$ 2.384	0.691 $\pm$ 0.075	0.276 $\pm$ 0.061	26.485
CLAP-LANG-Rel	17.200 $\pm$ 2.740	0.692 $\pm$ 0.087	0.260 $\pm$ 0.069	21.517

Table 8 Effects of relative and absolute action-space conditioning on cross-embodiment video modeling (Bridge).

Method	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	FVD $\downarrow$
CLAP-EE-Rel	22.459 $\pm$ 2.399	0.841 $\pm$ 0.036	0.110 $\pm$ 0.038	10.545
CLAP-EE-Abs	23.589 $\pm$ 2.442	0.822 $\pm$ 0.043	0.091 $\pm$ 0.041	10.119
CLAP-LANG-Abs	19.317 $\pm$ 2.170	0.770 $\pm$ 0.047	0.159 $\pm$ 0.050	20.356
CLAP-LANG-Rel	20.548 $\pm$ 2.291	0.780 $\pm$ 0.049	0.142 $\pm$ 0.052	17.322

Table 9 Effects of relative and absolute action-space conditioning on cross-embodiment video modeling (OXE-Mix).

Method	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	FVD $\downarrow$
CLAP-EE-Rel	20.593 $\pm$ 3.795	0.798 $\pm$ 0.084	0.157 $\pm$ 0.083	25.707
CLAP-EE-Abs	21.665 $\pm$ 2.955	0.798 $\pm$ 0.065	0.129 $\pm$ 0.068	26.199
CLAP-LANG-Abs	19.197 $\pm$ 2.783	0.765 $\pm$ 0.068	0.178 $\pm$ 0.074	40.696
CLAP-LANG-Rel	19.705 $\pm$ 2.825	0.765 $\pm$ 0.075	0.169 $\pm$ 0.077	37.010

Table 10 Success rates (%) in inference-time cross-policy planning in single-arm manipulation (DROID).

Task	MolmoAct-2	$\pi_{0.5}$	CLAP
Measuring Tape	75.0	40.0	80.0
Fish	50.0	50.0	75.0
Red Lobster	20.0	90.0	95.0
Box	100.0	100.0	100.0
Towel	100.0	100.0	100.0

Table 11 Success rates (%) in inference-time cross-policy planning in bimanual manipulation (bimanual YAM).

Task	MolmoAct-2	CLAP
Pick Carrot & Apple	90.0	95.0
Pick Block & Avocado	100.0	100.0

Table 12 Performance after few-shot adaptation to pretrained embodiments (DROID).

Method	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	FVD $\downarrow$	FID $\downarrow$
CLAP-LANG	19.381 $\pm$ 2.780	0.745 $\pm$ 0.080	0.198 $\pm$ 0.068	15.208	30.605
CLAP-LAM	19.257 $\pm$ 2.552	0.742 $\pm$ 0.077	0.200 $\pm$ 0.065	15.945	31.023
CLAP-EE	19.461 $\pm$ 2.871	0.745 $\pm$ 0.082	0.191 $\pm$ 0.066	14.590	27.708
CLAP-CURR	19.450 $\pm$ 2.732	0.747 $\pm$ 0.078	0.194 $\pm$ 0.064	14.598	30.619
Ctrl-World	18.928 $\pm$ 2.760	0.736 $\pm$ 0.081	0.205 $\pm$ 0.067	15.591	30.543

Table 13 Performance after few-shot adaptation to pretrained embodiments (Bridge).

Method	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	FVD $\downarrow$	FID $\downarrow$
CLAP-LANG	25.595 $\pm$ 2.165	0.860 $\pm$ 0.035	0.061 $\pm$ 0.029	6.101	12.564
CLAP-LAM	25.821 $\pm$ 2.220	0.862 $\pm$ 0.036	0.061 $\pm$ 0.029	5.779	12.822
CLAP-EE	26.044 $\pm$ 2.189	0.865 $\pm$ 0.037	0.059 $\pm$ 0.028	5.975	12.510
CLAP-CURR	25.511 $\pm$ 2.332	0.858 $\pm$ 0.039	0.062 $\pm$ 0.031	5.667	12.435
Bridge-Base	25.660 $\pm$ 2.294	0.858 $\pm$ 0.038	0.063 $\pm$ 0.031	6.402	13.307

Table 14 Performance after few-shot adaptation to novel embodiments.

Embodiment	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	FVD $\downarrow$	FID $\downarrow$
Bimanual YAM	24.310 $\pm$ 3.840	0.811 $\pm$ 0.073	0.152 $\pm$ 0.072	11.519	28.564
G1 humanoid	15.151 $\pm$ 1.120	0.709 $\pm$ 0.053	0.215 $\pm$ 0.050	16.527	19.599

Table 15 Success rates (%) after RL-based policy finetuning with CLAP.

Task	Base DP	CLAP
Put Carrot in Bowl	80.0	88.0
Fold Towel	100.0	100.0

D Nuanced Summary

This section discusses the nuances surrounding CLAP’s novelty, broader potential impact, and scope of claims.

List of Questions

Q1: CLAP uses an old video backbone (SVD, 2023), a pretrained language encoder (CLIP), simple MLP action encoders, and does not introduce a new dataset. Is it novel?	28
Q2: Why does CLAP use SVD and not newer backbones like Wan 2.2, and does it generalize to these backbones?	28
Q3: What motivated the choice of diffusion over alternative frameworks like flow-matching?	28
Q4: Can CLAP run on consumer-grade GPUs?	28
Q5: Does CLAP run in real-time (at least 10 Hz)?	28
Q6: Is the training scale of CLAP comparable to GPT, Claude, or Gemini, and can it realistically handle internet-scale video data?	28
Q7: Do the benefits of cross-embodiment learning plateau?	28
Q8: Does CLAP outperform physics-based simulators, and is its output high-fidelity?	28
Q9: Does CLAP consistently outperform all single-embodiment models?	28
Q10: Will CLAP work effectively with all robots, e.g., mobile robots?	29
Q11: Does CLAP hallucinate?	29
Q12: Which action representation yields optimal performance?	29
Q13: What is the actual utility of learned latent actions?	29
Q14: Is it a fair comparison between language, end-effector poses, and latent actions?	29
Q15: Are language actions fundamentally poor at video prediction?	29
Q16: Is it fair to compare single-embodiment models to post-trained models?	29
Q17: Does CLAP always beat $\pi_{0.5}$ and MolmoAct-2 via inference-time planning?	29
Q18: Why not use object-centric metrics like object masks and poses extracted with an inverse dynamics model (IDM) or a pose tracker instead of perceptual metrics?	29
Q19: Why not compare against more video model baselines (backbones)?	29
Q20: Why not compare against other video models in the planning and RL experiments?	30
Q21: Why not compare against other baselines when adapting to novel embodiments?	30
Q22: Why not use action reconstruction as the metric when comparing the IDMs instead of future-frame reconstruction?	30
Q23: Why did you use a simple GPT-based reward model in the RL experiments?	30

Detailed Q&A

Q1: CLAP uses an old video backbone (SVD, 2023), a pretrained language encoder (CLIP), simple MLP action encoders, and does not introduce a new dataset. Is it novel?

Answer: Not quite. By that same token, current state-of-the-art action-conditioned video models are largely derivative. CLAP’s novelty lies in introducing a new paradigm for action-conditioned video modeling rooted specifically in cross-embodiment learning.

Q2: Why does CLAP use SVD and not newer backbones like Wan 2.2, and does it generalize to these backbones?

Answer: CLAP adopts an SVD backbone to enable thorough experimentation across action, image, and history conditioning schemes and data mixtures at a minimal cost. While newer backbones (such as Wan 2.2) yield higher-quality video, they require roughly 10 days of training per run on an H200 node compared to SVD’s 2 days, representing prohibitive costs for all but industry-tier budgets. Moreover, SVD enables inference on budget-friendly consumer-grade hardware unlike many newer backbones. Crucially, CLAP generalizes seamlessly to these newer backbones because it is not tied to any architecture-specific features.

Q3: What motivated the choice of diffusion over alternative frameworks like flow-matching?

Answer: CLAP builds directly on SVD’s native diffusion framework for stable integration and rapid iteration. However, future work will adopt flow matching for accelerated inference.

Q4: Can CLAP run on consumer-grade GPUs?

Answer: Yes, CLAP requires under 12 GB of VRAM at inference, which easily fits on consumer hardware like the RTX 3060.

Q5: Does CLAP run in real-time (at least 10 Hz)?

Answer: No, CLAP falls short of 10 Hz. Its inference latency is hardware-dependent, averaging 3.24s on the A100-PCIe-40GB, 2.88s on the A100-SXM4-80GB, and 1.49s on the H200 for a standard configuration with 11 frames and 25 denoising steps, averaged over 20 trials.

Q6: Is the training scale of CLAP comparable to GPT, Claude, or Gemini, and can it realistically handle internet-scale video data?

Answer: No, CLAP is trained on significantly smaller datasets than foundation models like GPT, Claude, or Gemini. However, CLAP’s architecture natively supports internet-scale data, though training at that scale remains resource-intensive. We welcome collaborations to scale CLAP to internet-scale datasets. In preliminary experiments, we observe that CLAP scales favorably with both model size and data volume.

Q7: Do the benefits of cross-embodiment learning plateau?

Answer: Ultimately, but we see no evidence to suggest that this plateau is near.

Q8: Does CLAP outperform physics-based simulators, and is its output high-fidelity?

Answer: Like other video models, CLAP is not yet a replacement for well-tuned physics-based simulators, but it offers strong potential to eventually supersede them. Moreover, CLAP’s predictions achieve a level of fidelity that approaches or surpasses existing state-of-the-art single-embodiment video models in challenging environments.

Q9: Does CLAP consistently outperform all single-embodiment models?

Answer: No, CLAP does not universally outperform single-embodiment models. Like other cross-embodiment or foundation models (e.g., GPT), it can occasionally underperform domain-specific models tailored to a single embodiment.

Q10: Will CLAP work effectively with all robots, e.g., mobile robots?

Answer: Not universally. Performance depends on CLAP’s scale and distribution coverage, but accuracy improves as target robot platforms grow closer to the training distribution. Moreover, further adaptation to floating reference frames may be required.

Q11: Does CLAP hallucinate?

Answer: Yes, like existing SOTA video models, CLAP is prone to hallucinations at the boundaries of its training distribution, presenting an exciting direction for future work on uncertainty quantification and hallucination mitigation.

Q12: Which action representation yields optimal performance?

Answer: No single action representation universally outperforms the others; performance depends on a nuanced trade-off where language actions minimize the domain gap for foundation models but lack precision, end-effector poses offer high precision but require labeled video data, and latent actions unlock unlabeled video data usage while introducing a deployment domain gap.

Q13: What is the actual utility of learned latent actions?

Answer: Learned latent actions enable cross-embodiment learning on unlabeled internet-scale videos by bypassing the need for manual action annotations.

Q14: Is it a fair comparison between language, end-effector poses, and latent actions?

Answer: Whether a direct comparison is fair depends on the goal, as CLAP focuses on cross-embodiment modeling rather than benchmarking representations. Forcing a data-matched comparison would require training all action representations on the same data, which would block the use of unlabeled (internet-scale) data and defeat the framework’s core objective.

Q15: Are language actions fundamentally poor at video prediction?

Answer: Language actions are not inherently poor at video prediction; rather, existing language encoders are simply not optimized for processing numerical data, creating an artificial gap that limits their overall expressiveness.

Q16: Is it fair to compare single-embodiment models to post-trained models?

Answer: Yes, this comparison is fair given CLAP’s goal of providing a straightforward way to adapt cross-embodiment models to a target embodiment. This ensures that both the single-embodiment and post-trained models operate under the same effective user-side training budget.

Q17: Does CLAP always beat $\pi_{0.5}$ and MolmoAct-2 via inference-time planning?

Answer: Not always. While inference-time planning allows CLAP to explore different action proposals from various policies to boost performance, it can still fail due to issues like video or reward prediction hallucinations.

Q18: Why not use object-centric metrics like object masks and poses extracted with an inverse dynamics model (IDM) or a pose tracker instead of perceptual metrics?

Answer: While object-centric metrics can offer fine-grained assessments, state-of-the-art IDMs and pose trackers are prone to hallucinating outputs across diverse tasks, which can compromise evaluation integrity. To avoid this, CLAP uses standard perceptual metrics common in state-of-the-art video models, leaving improved evaluation metrics as an open area of research.

Q19: Why not compare against more video model baselines (backbones)?

Answer: First, no off-the-shelf cross-embodiment action-conditioned video models exist. To eliminate confounding variables such as model size or pretraining data, all models are trained on the same

backbone derived from the state-of-the-art DROID model. Furthermore, in the Bridge environment, this baseline already outperforms existing state-of-the-art methods.

Q20: Why not compare against other video models in the planning and RL experiments?

Answer: The primary goal of the planning and RL experiments is to demonstrate real-world zero-shot generalization, making additional video model comparisons unnecessary for a few reasons. First, applying video models to planning or RL is already well-established, predating current state-of-the-art video models. More importantly, performance differences between state-of-the-art video models rarely manifest in real-world deployment, typically appearing only in contrived edge cases that current robot policies are not even capable of operating within.

Q21: Why not compare against other baselines when adapting to novel embodiments?

Answer: These experiments focus on demonstrating seamless adaptation to novel embodiments. While fine-tuning already proves that CLAP provides strong priors for in-domain settings, cross-domain training (e.g., from a robot-data-free SVD backbone to robot video prediction) incurs no performance penalty compared to training from scratch. In fact, within a single gradient step, our adapted models generate meaningful video predictions, unlike models trained from raw SVD backbones. Additionally, no open-source video model baselines currently exist for bimanual YAM and the G1 Humanoid.

Q22: Why not use action reconstruction as the metric when comparing the IDMs instead of future-frame reconstruction?

Answer: The main goal of the IDM is to extract actions from unlabeled videos. Because ground-truth actions are unavailable in these datasets, CLAP relies on future-frame reconstruction to assess the accuracy of the latent actions.

Q23: Why did you use a simple GPT-based reward model in the RL experiments?

Answer: As noted earlier, ablating the reward model is not central to the goal of the RL experiments. In practice, any feasible reward model — even a human operator — can be swapped into the deployment pipeline. The GPT-based reward model was simply sufficient for our setup.

Q24: Is evaluating single-arm CLAP models on bimanual robots informative?

Answer: It is informative as a stress test, pushing the single-arm CLAP model outside its standard operational domain by introducing two robots into a single scene. By managing the action-conditioning gap (accounting for the difference between the 7-dimensional model and the 14-dimensional bimanual setup), we can effectively deploy and evaluate the model. Naturally, for optimal performance, the model should be adapted to the target embodiment, as demonstrated in our experiments.