# SELF-DISTILLING FUTURE VIDEO REPRESENTATIONS FOR WORLD ACTION MODELS

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

## ABSTRACT

World Action Models (WAMs) have emerged as a promising approach to learning generalist robot policies using rich spatiotemporal priors from pre-trained video diffusion models. A recent line of WAMs improves inference efficiency by conditioning a separate action decoder on intermediate future video representations extracted from a single denoising step, avoiding costly iterative denoising process required for video generation. This design makes action prediction depend directly on the quality of these future representations. In this paper, we introduce *Future REpresentation learning via self-Distillation* (FRED), a framework that trains WAMs to improve the quality of future video representations from noise in a single denoising step. Specifically, FRED uses self-distillation to predict future representations from noise, using representations extracted from ground-truth future observations as targets. We further condition the video model on historical observations, providing temporal context for predicting future representations better. Across diverse real-world manipulation scenarios, FRED consistently outperforms competing WAMs. It achieves particularly large gains on long-horizon and out-of-domain tasks under zero-shot environment generalization, and substantially improves human-to-robot skill transfer, reaching 50.0% task success on tasks absent from the robot training data and demonstrated only in human videos, compared with 19.4% for the best competing WAM. Project page is available.

## 1 INTRODUCTION

World Action Models (WAMs; Pai et al. 2025; Liang et al. 2025; Kim et al. 2026; Ye et al. 2026) have recently emerged as a promising approach to learning generalist robot policies. Built on pre-trained video diffusion models (Wan et al., 2025; Ali et al., 2025), WAMs have shown strong generalization to unseen tasks (Ye et al., 2026; Ma et al., 2026), adaptation to new robot embodiments (Ye et al., 2026), and skill transfer from human videos to robots (Dyna Robotics, 2026). Their core idea is to leverage the rich spatiotemporal priors learned from large-scale in-the-wild video datasets (Chen et al., 2024; Nan et al., 2024) by training a model to jointly predict future frames and corresponding actions. This allows the strong generalization capabilities of pre-trained video models to extend from future video prediction to robot control.

Interestingly, recent works have shown that WAMs can achieve comparable performance with a separate decoupled action decoder that generates actions conditioned on intermediate features obtained from a single denoising step (Ma et al., 2026; Dyna Robotics, 2026; Qiu et al., 2026; Zhao et al., 2026; Mimic Robotics, 2026). By avoiding iterative denoising with large video diffusion backbones, these approaches substantially reduce the inference latency, making WAMs practical for real-time robot control. Yet, it remains unclear why single-step inference works so well, given that the underlying video diffusion models are never explicitly trained for one-step generation.

In this work, we argue that the effectiveness of single-step inference of recent WAMs stems from the implicit learning of future representation generation from a video model with denoising objectives. Specifically, video diffusion models may not generate high-fidelity videos in a single denoising step, yet they can capture high-level semantics (*e.g.*, global robot motion and target object placement). Thus, the resulting intermediate features could serve as useful future representations for action control. However, they are learned only implicitly through the video denoising objective rather than being directly optimized as future representations for action generation.



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

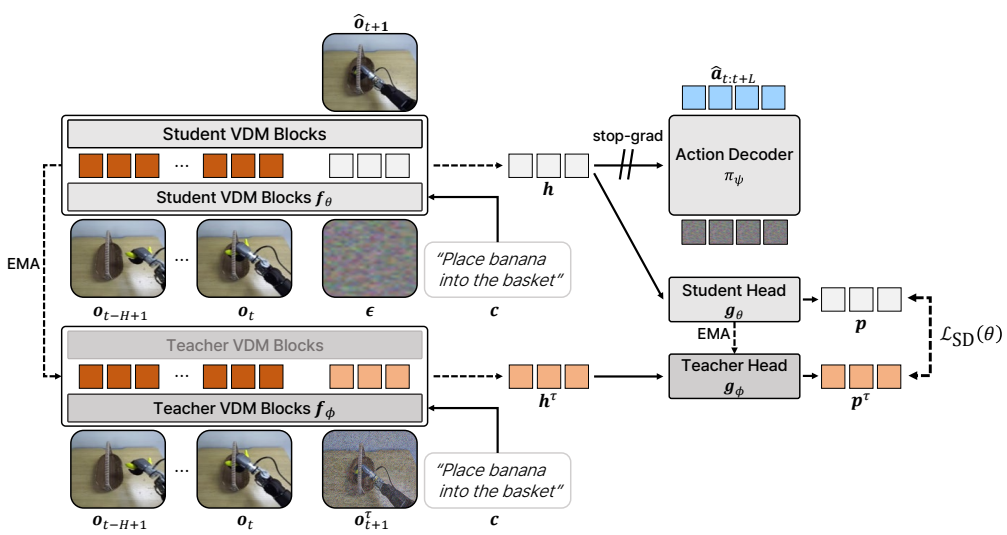



Figure 1: **Overview of FRED.** We design a framework for training World Action Models (WAMs) that explicitly improves the future representations $\mathbf{h}$ from a single video denoising step used for action generation. We interpret the usage of a video diffusion model in WAM as a future representation generator, *i.e.*, $\mathbf{h} = \mathbf{f}_\theta(\boldsymbol{\epsilon}|\mathbf{o}_t, \mathbf{c})$. From this perspective, we improve action generation quality by enhancing $\mathbf{h}$ through self-distillation to predict target representations $\mathbf{h}^\tau$ extracted from noisy future observations $\mathbf{o}^\tau_{t+l}$ by a momentum teacher, and using historical observations $\mathbf{o}_{t-H:t}$ as conditioning input to exploit richer temporal context.

To address this, we introduce *Future REpresentation learning via self-Distillation* (FRED), a simple framework for training WAM that explicitly improves the future representations from a single video denoising step used for action generation via self-distillation. Specifically, FRED first trains the video model to predict target future representations from noise via self-distillation, where the targets are extracted from noisy ground-truth future observations by a momentum teacher of the video model. To provide additional temporal context (*e.g.*, robot motion) for the future representation prediction, we condition the video model on a short history of observations. Notably, FRED significantly improves the quality of future video representations, which translates into better robot control performance (see Figure 3c).

To validate the effectiveness and generality of FRED, we conduct comprehensive experiments across diverse real-world robotic manipulation scenarios. We first evaluate FRED on zero-shot environment generalization using a diverse set of real-world manipulation tasks, including long-horizon manipulation, out-of-domain task execution, and object grounding. Our results show that FRED consistently outperforms competing WAMs across these settings, improving long-horizon performance from 61.7% to 81.7% and out-of-domain task success from 10.4% to 52.1% at the 14B scale. We further evaluate FRED on human-to-robot skill transfer by co-training the video model on human videos without action labels, including tasks absent from the robot training data. We show that FRED substantially outperforms competing WAMs, improving task success from 19.4% to 50.0%, while achieving 97.2% instruction-following performance. These results highlight the effectiveness of explicit future representation learning for improving generalization and transfer in WAMs.

We highlight the main contributions of this paper below:

- We interpret the video diffusion model in WAMs as a future representation generator, whose ability to generate future representations emerges implicitly from the denoising objective.
- We present FRED, a simple framework that explicitly learns future video representations predicted from noise through self-distillation. We additionally condition the video model on observation history to provide temporal context for future prediction.
- We show that FRED consistently outperforms baseline WAMs across diverse real-world manipulation settings: zero-shot environment generalization and human-to-robot knowledge transfer.

## 2 Related work

**World Action Models.** Prior work studies robot control via World Action Models (WAMs), which build on pre-trained video diffusion models and jointly model future observations and actions using robot data, allowing spatiotemporal priors from large-scale video to guide action generation (Shen et al., 2026; Li et al., 2026a; Kim et al., 2026; Ye et al., 2026). These models generate plausible future visual states to support action prediction, but iterative denoising with large video backbones incurs substantial computational cost at inference time. Recent WAMs address this bottleneck by decoupling video and action generation and reducing video denoising to a single step. Within this shared design, some works predict actions without explicitly generating future videos at inference by masking access to future video features (Yuan et al., 2026) or directly decoding actions from video-backbone states (Li et al., 2026b). In contrast, another line of work conditions action prediction on future video representations obtained from noise in a single video denoising step (Pai et al., 2025; Ma et al., 2026; Qiu et al., 2026; Zhao et al., 2026). Our work follows the latter design and focuses on how to train the video diffusion model to generate useful future representations from a single denoising step, rather than relying on the denoising objective alone to induce them implicitly.

**Diffusion Models and Representation Learning.** Prior works have shown that diffusion denoisers learn discriminative representations useful for visual recognition (Xiang et al., 2023; Yang & Wang, 2023; Li et al., 2023). More recent work improves these representations during generative training, where REPA (Yu et al., 2025) aligns noisy diffusion features with clean-image representations from a pre-trained visual encoder, while SRA (Jiang et al., 2025) uses lower-noise representations from the diffusion model itself as targets. This direction has been extended to video diffusion models through alignment with pre-trained video representations (Zhang et al., 2026; Lee et al., 2025; Bhowmik et al., 2026), while Self-Flow (Chefer et al., 2026) incorporates self-supervised representation learning directly into flow matching without an external encoder. Our work studies representation learning in WAMs, where intermediate video representations from a single denoising step directly condition action generation. AGRA (Qiu et al., 2026) adapts representation alignment to this setting by aligning these representations with features from a frozen visual encoder. FRED instead uses representations extracted from observed future frames as targets for representations generated from noise. This explicitly trains the video model to predict the future representation used by the action decoder, rather than aligning its intermediate features to a fixed external representation space.

## 3 Method

### 3.1 Overview

Let $\mathcal{D} = \{\mathbf{X}_i\}_{i=1}^{N}$ be an expert demonstration dataset, where each demonstration $\mathbf{X}_k = [(\mathbf{o}_t, \mathbf{q}_t, \mathbf{a}_t)_{t=1}^{T_k}, \mathbf{c}]$ consists of a visual observation $\mathbf{o}_t$, proprioceptive state $\mathbf{q}_t$, and robot action $\mathbf{a}_t$ at each timestep $t \in [1, T_k]$, with a task instruction $\mathbf{c}$. We are interested in training a World Action Model (WAM) as a policy (Ye et al., 2026), which jointly predicts a future observation $\mathbf{o}_{t+1}$ and a future action $\mathbf{a}_t$ conditioned on $(\mathbf{o}_t, \mathbf{q}_t, \mathbf{c})$.[1] Typical WAM methods post-train a pre-trained video diffusion model $\mathbf{v}_\theta(\mathbf{o}_{t+1}^\tau | \mathbf{o}_t, \mathbf{c})$ (with diffusion timestep $\tau \in [0, 1]$) to jointly denoise the future observation and action, *i.e.*, $\mathbf{v}_\theta(\mathbf{o}_{t+1}^\tau, \mathbf{a}_t^\tau | \mathbf{o}_t, \mathbf{c})$. This approach has shown strong generalization as it can exploit strong spatiotemporal knowledge learned from large-scale natural video datasets.

Our method is motivated by recent observations and methods in the WAM literature (Ma et al., 2026; Mimic Robotics, 2026; Qiu et al., 2026), mainly proposed to improve computational efficiency. Specifically, instead of updating the large video diffusion model architecture to jointly predict future observations and actions, they keep the video diffusion model architecture unchanged and have a separate smaller diffusion-based action decoder $\pi_\psi$. The action decoder generates $\mathbf{a}_t$ conditioned on $\mathbf{q}_t$ and the intermediate hidden state $\mathbf{h}$ obtained from $\mathbf{v}_\theta$ with inputs $(\mathbf{o}_t, \mathbf{c})$ and *pure Gaussian noise* $\boldsymbol{\epsilon}$, *i.e.*, a single forward operation. By doing so, it removes the iterative forward process of a large video diffusion model and thus can considerably improve the efficiency. We also adopt this decoupled architecture as our method and training scheme.

[1] For simplicity, we describe our method as a policy that predicts a single action and a single future observation. However, in practice, it predicts $L > 1$ future actions and observations at once, which is straightforward to implement within the WAM training framework.

Beyond computational efficiency, this decoupled architecture also offers a useful perspective on the role of the video diffusion model. We interpret its usage as a *future representation generator*: $\mathbf{h} = \mathbf{f}_\theta(\boldsymbol{\epsilon}|\mathbf{o}_t, \mathbf{c})$, where $\mathbf{f}_\theta$ denotes the first layers of $\mathbf{v}_\theta$ that output $\mathbf{h}$. From this perspective, we introduce a simple framework for training WAM that improves action generation quality significantly by explicitly training the video model to improve $\mathbf{h}$ via self-distillation. We additionally provide a short observation history as temporal context for this prediction. We illustrate an overview of our method, FRED, in Figure 1.

### 3.2 FRED

FRED introduces a simple framework for training WAM by explicitly improving the representation generator $\mathbf{f}_\theta$ of a video diffusion model $\mathbf{v}_\theta$ to better generate future video representations $\mathbf{h}$ from noise $\boldsymbol{\epsilon}$. To this end, we first use self-distillation to train the student $\mathbf{f}_\theta$ to predict target future representations from noise, where the targets are extracted by a momentum teacher $\mathbf{f}_\phi$ from noisy ground-truth future observations. To facilitate future representation prediction, we additionally condition the video model on a $H$-frame short history of observations $\mathbf{o}_{t-H:t} := [\mathbf{o}_{t-H+1}, \cdots, \mathbf{o}_t]$, rather than only the current frame $\mathbf{o}_t$. This provides temporal cues (*e.g.*, robot motion) that may help disambiguate the future state.

Formally, given historical observations $\mathbf{o}_{t-H:t}$ and text instruction $\mathbf{c}$ as conditioning inputs, we first process random noise $\boldsymbol{\epsilon}$ through the student (online) video diffusion model encoder $\mathbf{f}_\theta$ to generate intermediate video representations $\mathbf{h}$. Under the same conditioning, we process noisy future observations $\mathbf{o}^\tau_{t+1} := (1-\tau)\mathbf{o}_{t+1} + \tau\boldsymbol{\epsilon}$ with $\tau \in [0, 1]$ through the teacher (momentum) video diffusion model encoder $\mathbf{f}_\phi$ to obtain target representations $\mathbf{h}^\tau$.

$$\mathbf{h} = \mathbf{f}_\theta(\boldsymbol{\epsilon}|\mathbf{o}_{t-H:t}, \mathbf{c}) \quad \text{and} \quad \mathbf{h}^\tau = \mathbf{f}_\phi(\mathbf{o}^\tau_{t+1}|\mathbf{o}_{t-H:t}, \mathbf{c}) \tag{1}$$

We then process these intermediate representations through the patch-wise student and teacher projection heads $\mathbf{g}_\theta, \mathbf{g}_\phi$ and softmax functions to obtain distributions $\mathbf{p}, \mathbf{p}^\tau$, respectively. Specifically, we apply softmax with temperature $T_1 > 0$ to obtain the student distribution $\mathbf{p}$. For the teacher distribution $\mathbf{p}^\tau$, we center the logits by subtracting the mean teacher logits $\mathbf{m}$ and apply softmax with a sharpening temperature $T_2 > 0$ to prevent representation collapse, following DINO (Caron et al., 2021) and iBOT (Zhou et al., 2022).

$$\mathbf{p} = \mathrm{softmax}\left(\mathbf{g}_\theta(\mathbf{h})/T_1\right) \quad \text{and} \quad \mathbf{p}^\tau = \mathrm{softmax}\left((\mathbf{g}_\phi(\mathbf{h}^\tau) - \mathbf{m})/T_2\right) \tag{2}$$

Then, FRED improves $\mathbf{f}_\theta$ to generate better future video representations by minimizing the KL divergence between its output distributions $\mathbf{p}$ and the teacher distributions $\mathbf{p}^\tau$, averaged over all $N$ patches:

$$\mathcal{L}_{\mathrm{SD}}(\theta) = \frac{1}{N}\sum_{i=1}^{N} \mathrm{KL}\left(\mathbf{p}^\tau_i \| \mathbf{p}_i\right) \tag{3}$$

**WAM Training Objective and Inference.** The overall WAM training objective combines the self-distillation loss with the video and action flow-matching losses (Lipman et al., 2022), as follows:

$$\mathcal{L}_{\mathrm{video}}(\theta) = \mathcal{L}_{\mathrm{FM}}(\theta) + \lambda_{\mathrm{SD}}\mathcal{L}_{\mathrm{SD}}(\theta) \quad \text{and} \quad \mathcal{L}_{\mathrm{action}}(\psi) = \mathcal{L}_{\mathrm{FM}}(\psi), \tag{4}$$

where $\lambda_{\mathrm{SD}} > 0$ is a hyperparameter that balances the flow-matching and self-distillation objectives. Here, we do not update $\theta$ with the action flow-matching loss $\mathcal{L}_{\mathrm{FM}}(\psi)$; thus, the representation generator $f_\theta$ is trained only by the video generation and self-distillation objectives. We update the parameters of momentum teacher networks as $\phi \leftarrow m\phi + (1-m)\theta$, where $m \in [0, 1]$ is the momentum coefficient.

At inference, we first obtain the future representation $\mathbf{h}$ from the representation generator $\mathbf{f}_\theta$, and then use the action decoder $\pi_\psi$ to generate $\mathbf{a}_t$ conditioned on $(\mathbf{q}_t, \mathbf{h})$ using the Euler ODE sampling procedure described in Appendix E.

## 4 Experiments

We design experiments to investigate whether explicitly learning to generate future video representations using FRED can improve the control performance of World Action Models (WAMs) in complex real-world environments. In particular, we consider the following questions:

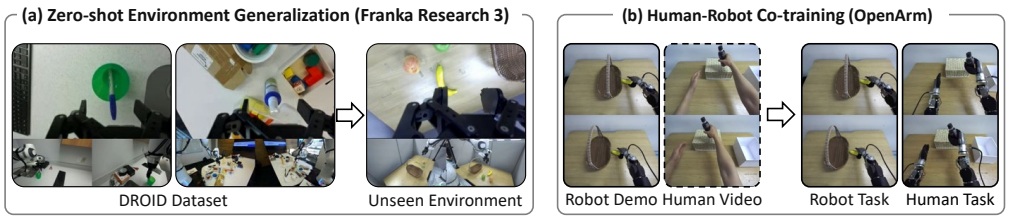



Figure 2: **Visualization of the training and evaluation setup.** (a) For zero-shot environment generalization, we train FRED on the DROID (Khazatsky et al., 2024) dataset and evaluate it on diverse tasks in a new real-world environment without task-specific fine-tuning. (b) For Human-robot co-training, we train FRED either on robot demonstrations alone or jointly on robot demonstrations and human videos, and evaluate it on both in-domain robot tasks and human-to-robot skill transfer tasks.

- Can FRED generalize to unseen environments and tasks? (Table 1)
- Can FRED benefit from human video co-training, allowing human-to-robot knowledge transfer? (Table 2)
- What is the effect of each component of FRED? (Tables 3 to 7 and Figure 3)
- How do future representation and control performance correlate? (Table 4 and Figure 3)

### 4.1 Experimental Setup

**Implementation Details.** For the 5B-scale experiments, we use Wan2.2-TI2V-5B (Wan et al., 2025) as the video diffusion backbone and a 16-layer DiT (Peebles & Xie, 2023) as the action decoder. For self-distillation in FRED, we extract features from layer 18 of the video backbone for both the student and teacher, fix the teacher's noise level at 0.25, and use $\lambda_{\text{SD}}$=0.01 by default. We update the momentum teacher using an exponential moving average (EMA) of the student, with the decay warmed up using inverse decay schedule. We use 4 historical observations with the current observation as input. For our 14B-scale experiments, we utilize the Wan2.1-I2V-14B (Wan et al., 2025) with a 24-layer DiT action decoder and extract features from the layer 23 of the video backbone for self-distillation. Unless otherwise stated, we extract video features from the same layer for self-distillation and action decoder conditioning. We provide further details in Appendix A.

**Baselines.** We use several recent end-to-end robot foundation models as baselines. Specifically, we first consider recent World Action Models (WAMs): (a) DiT4DiT (Ma et al., 2026), Fast-WAM (Yuan et al., 2026), and AGRA (Qiu et al., 2026) are WAMs with one-step video generation, and (b) DreamZero (Ye et al., 2026) is a WAM with multi-step video generation due to shared video-action modeling within the same architecture. We also consider recent pre-trained Vision-Language-Action models (VLAs): $\pi_{0.5}$ (Intelligence et al., 2025) and GR00T N1.7 (NVIDIA, 2026). We describe more details of baselines in Appendix C.

### 4.2 Zero-shot Environment Generalization

We first investigate whether FRED can generalize to unseen environments and tasks using future video representations generated in a single denoising step. To this end, we train our method on DROID (Khazatsky et al., 2024), one of the largest publicly available robotics datasets, and evaluate it across diverse real-world tasks. We provide detailed task descriptions and training and evaluation protocols in Appendix B, and the full results in Appendix G.1. We categorize the evaluation tasks into four types:

- *In-domain easy tasks* use instruction and object categories seen during training, but are evaluated in a new environment with different object details.
- *In-domain long-horizon task* extends in-domain easy settings to longer-horizon manipulations that require multiple manipulation steps.
- *Out-of-domain tasks* evaluate generalization to manipulation behaviors whose action verbs are absent from the DROID training instructions. They require the robot to execute novel motions.
- *Object grounding tasks* require the robot to select and manipulate the object specified by the instruction in the presence of distractors with similar colors or shapes.

Table 1: **Results on zero-shot generalization to a new real-world environment.** Performance of various policies trained on the DROID (Khazatsky et al., 2024) dataset without environment-specific fine-tuning. We report the average task progress rate (%) for in-domain long-horizon tasks, and average success rate (%) for other task types.

| Method | Backbone | In-domain | | Out-of-domain | Object Grounding |
|---|---|---|---|---|---|
| | | Easy | Long-horizon | | |
| $\pi_{0.5}$ (Intelligence et al., 2025) | - | 58.3 | 53.3 | 16.7 | 16.7 |
| GR00T N1.7 (NVIDIA, 2026) | - | 41.7 | 60.0 | 18.8 | 22.9 |
| Fast-WAM (Yuan et al., 2026) | Wan2.2-5B | 26.4 | 26.7 | 2.1 | 10.4 |
| DiT4DiT (Ma et al., 2026) | Wan2.2-5B | 16.7 | 16.7 | 0.0 | 8.3 |
| AGRA (Qiu et al., 2026) | Wan2.2-5B | 41.7 | 31.7 | 10.4 | 33.3 |
| **FRED (Ours)** | Wan2.2-5B | **47.2** | **46.7** | **37.5** | **37.5** |
| DreamZero (Ye et al., 2026) | Wan2.1-14B | 63.9 | 61.7 | 10.4 | 39.6 |
| **FRED (Ours)** | Wan2.1-14B | **79.2** | **81.7** | **52.1** | **54.2** |

**Results.** As shown in Table 1, we first find that FRED consistently outperforms the other WAM baselines across all task types in the 5B video backbone setup, where all methods generate actions in a single video denoising step. The improvements are particularly large on long-horizon and out-of-domain tasks: compared with the strongest competing WAM baseline, FRED improves performance from 31.7% to 46.7% and from 10.4% to 37.5%, respectively, while also improving object grounding from 33.3% to 37.5%. Compared with DiT4DiT (Ma et al., 2026), which does not explicitly optimize intermediate video representations for action prediction, FRED achieves substantially higher performance on both long-horizon (46.7% vs. 16.7%) and out-of-domain tasks (37.5% vs. 0.0%), suggesting that explicitly learning future representations can be particularly beneficial for WAMs in challenging generalization settings. At the larger 14B scale, FRED further improves across all task types and outperforms DreamZero (Ye et al., 2026) with the same backbone and other state-of-the-art VLAs, including substantial gains on long-horizon (81.7% vs. 61.7%) and out-of-domain tasks (52.1% vs. 10.4%). Overall, these results show that our simple future representation learning remains effective at larger model scales with improved generalization.

### 4.3 Human-robot Co-training

We next investigate whether FRED can leverage human video to improve robot control, as WAMs with separate video and action modules naturally support joint training on robot and human demonstrations by applying the video generation objective to human videos without requiring action labels. To this end, we first collect robot demonstrations using the OpenArm humanoid[2] and human manipulation videos in the same environment. We then compare policies trained on robot demonstrations alone with those jointly trained on robot demonstrations and human videos. We categorize the evaluation tasks as follows:

- *In-domain atomic tasks* evaluate basic manipulation skills present in the robot training data.
- *In-domain composite task* requires following an unseen instruction that combines two manipulation skills seen during robot training.
- *Human skill transfer* covers manipulation tasks involving placements, objects, or behaviors demonstrated only in human videos. We evaluate both task performance and instruction following.

We provide detailed task descriptions, training and evaluation protocols in Appendix B.2, and the full results in Appendix G.2.

**In-domain Results.** In Table 2a, we first find that FRED consistently outperforms WAM baselines on both atomic and composite tasks in the in-domain setting when trained with only robot data. In particular, FRED shows a larger margin over the baselines on the composite task, achieving 70.8% task progress compared with 50.0% for the strongest baseline. This suggests that the benefit of FRED extends beyond executing individual skills to combining them over a longer horizon. Next, we observe that when we additionally add human data to co-train the model, FRED further improves

[2]https://openarm.dev/

Table 2: **Results on humanoid control with and without robot-human co-training.** Performance of policies trained on the OpenArm humanoid using robot demonstrations only or with additional human videos. We report the average success rate (%) on in-domain atomic tasks, the average task progress rate (%) on in-domain composite tasks, and both task success (%) and instruction-following performance (%) on human skill transfer tasks.

(a) Results on robot in-domain tasks.

| | Atomic | | Composite | |
|---|---|---|---|---|
| Method | w/o Human | w/ Human | w/o Human | w/ Human |
| $\pi_{0.5}$ (Intelligence et al., 2025) | 66.7 | - | 50.0 | - |
| GR00T N1.7 (NVIDIA, 2026) | 91.7 | - | 45.8 | - |
| DreamZero (Ye et al., 2026) | 68.1 | 68.1 | 0.0 | 0.0 |
| Fast-WAM (Yuan et al., 2026) | 56.9 | 65.3 | 0.0 | 0.0 |
| DiT4DiT (Ma et al., 2026) | 77.8 | 86.1 | 50.0 | 41.7 |
| AGRA (Qiu et al., 2026) | 81.9 | 75.0 | 8.3 | 12.5 |
| **FRED (Ours)** | **91.7** | **97.2** | **70.8** | **75.0** |

(b) Results on human skill transfer task.

| | Task Success | | Instruction Following | |
|---|---|---|---|---|
| Method | w/o Human | w/ Human | w/o Human | w/ Human |
| $\pi_{0.5}$ (Intelligence et al., 2025) | 0.0 | - | 16.7 | - |
| GR00T N1.7 (NVIDIA, 2026) | 5.6 | - | 33.3 | - |
| DreamZero (Ye et al., 2026) | 0.0 | 0.0 | 0.0 | 13.9 |
| Fast-WAM (Yuan et al., 2026) | 0.0 | 0.0 | 0.0 | 11.1 |
| DiT4DiT (Ma et al., 2026) | 0.0 | 19.4 | 0.0 | 55.6 |
| AGRA (Qiu et al., 2026) | 0.0 | 0.0 | 0.0 | 0.0 |
| **FRED (Ours)** | 0.0 | **50.0** | 0.0 | **97.2** |

from 91.7% to 97.2% on atomic tasks and from 70.8% to 75.0% on composite tasks, suggesting that the video denoising objective on human data itself provides positive transfer to policy performance.

**Human Skill Transfer Results.** In Table 2b, we first observe that the evaluated end-to-end models trained without human videos achieve near-zero task success on tasks demonstrated only in the human-video dataset. The VLA baselines nevertheless show some instruction-following behavior, possibly due to knowledge acquired during VLA pre-training. When human videos are additionally used to train WAMs, we find that the other WAM baselines still show limited transfer; for example, DreamZero and AGRA achieve instruction-following rates of only 13.9% and 0.0%, respectively, whereas FRED improves substantially, achieving success rates of 50.0% and instruction-following rates of 97.2%. Notably, FRED more reliably follows the target skills demonstrated in human videos, whereas most baseline WAMs, except DiT4DiT, often execute behaviors seen in the robot training tasks instead. These results highlight the effectiveness of FRED in transferring skills from human videos to robot control, suggesting that future prediction provides useful motion information for transfer, and that explicitly learning such representations while separating them from robot action optimization may further improve human-to-robot transfer.

### 4.4 Analysis and Ablation Studies

We conduct extensive ablation studies and analyses to investigate how each component affects robot control and whether control performance correlates with the quality of generated future representations. We use the RoboCasa Kitchen (Nasiriany et al., 2024) benchmark for these experiments.

**Component Analysis.** In Table 3, we evaluate the contribution of self-distillation to future representation learning and the role of historical conditioning in providing temporal context for future prediction. We find that both components contribute complementary gains to the performance of FRED, as removing self-distillation or historical conditioning reduces the success rate from 68.8% to 65.1% and 64.8%, respectively, while removing both components further decreases it to 61.4%.

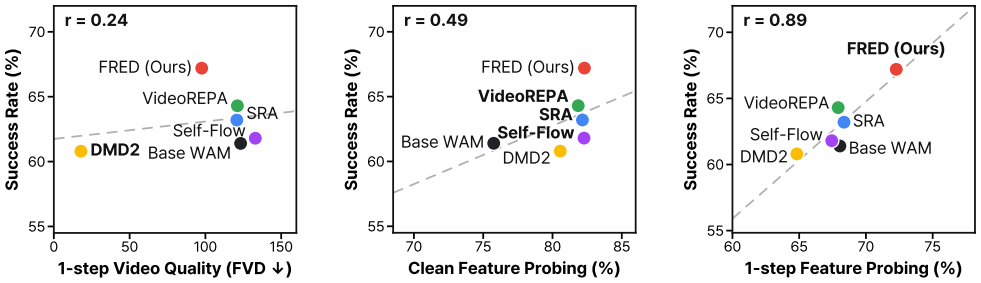
**Comparison with Representation Learning Methods.** To verify the effectiveness of FRED as a representation learning method for WAM, we compare FRED with recent representation learning approaches for diffusion models, *i.e.*, VideoREPA (Zhang et al., 2026), SRA (Jiang et al., 2025),

Table 3: **Component ablation study** of FRED on RoboCasa Kitchen benchmark.

| Method | SR(%) |
|---|---|
| **FRED (Ours)** | **68.8** |
| w/o Self-distillation | 65.1 |
| w/o History | 64.8 |
| Base WAM (DiT4DiT) | 61.4 |

Table 4: **Comparison of representation learning methods on simulation benchmarks.** For each benchmark, all models are trained on its training dataset for 100K iterations with a batch size of 64. We report averaged success rate (%) on each benchmark.

| Method | RoboCasa | GR-1 Tabletop | LIBERO |
|---|---|---|---|
| DiT4DiT (Ma et al., 2026) | 61.4 | 50.4 | 97.4 |
| w/ VideoREPA (Zhang et al., 2026) | 64.3 | 51.6 | 97.3 |
| w/ SRA (Jiang et al., 2025) | 62.9 | 52.1 | 97.6 |
| w/ Self-Flow (Chefer et al., 2026) | 61.8 | 50.9 | 97.9 |
| **w/ FRED (Ours)** | **68.8** | **54.8** | **98.2** |




(a) One-step video generation quality (b) Representation quality from observed futures (c) Representation quality from one-step generation

Figure 3: **Video generation and representation quality versus policy performance.** Using DiT4DiT as the base WAM, we compare variants trained with different video-model objectives in terms of task success rate, video generation quality, and representation quality on RoboCasa Kitchen (Nasiriany et al., 2024). We find that task success rates are not strongly correlated with (a) better one-step video generation and (b) representation quality on clean future observations, but more strongly correlated with (c) the quality of one-step generated future representations from noise.

and Self-Flow (Chefer et al., 2026), on RoboCasa Kitchen (Nasiriany et al., 2024), GR-1 Tabletop (Bjorck et al., 2025), and LIBERO (Liu et al., 2023) benchmarks. In Table 4, we find that VideoREPA, SRA, and Self-Flow show inconsistent gains over the base WAM, whereas FRED consistently improves performance across all benchmarks. This suggests that directly learning the future representations generated from noise is particularly effective for improving action control in WAMs.

**Future Representation Quality and Robot Control.** Since the action decoder directly conditions on future video representations generated from noise, their quality may be closely correlated with robot control performance. To investigate this, we use DiT4DiT (Ma et al., 2026) as the base WAM and compare variants trained with different video model objectives along these three dimensions: one-step video generation quality, representation quality on observed future videos, and the quality of future representations generated from noise. The WAM variants include an objective that explicitly improves one-step video generation through distillation (DMD2; Yin et al. 2024) and representation-learning objectives that improve intermediate video representations. We measure video generation quality using Fréchet Video Distance (FVD; Unterthiner et al. 2018). For the representation quality, we follow prior works on evaluating visual representation with frozen-backbone probing (Yu et al., 2025; Assran et al., 2025). In particular, we train a lightweight attentive probe on the video representations to classify each video into predefined motion primitive classes (see Appendix D for more detailed setup).

In Figures 3a and 3b, we find that task success is only weakly correlated with one-step video generation quality ($r = 0.24$) and representation quality on observed future videos ($r = 0.49$). In contrast, in Figure 3c, we find that the quality of future representations generated from noise shows a stronger correlation with task success ($r = 0.89$). This supports our view of the video diffusion model as a future representation generator and suggests that improving the representations directly used for action generation may be particularly relevant to control performance.

**Self-distillation Design Choices.** We analyze several design choices of FRED: teacher noise level, feature extraction layer used for both self-distillation and action conditioning, self-distillation coefficient, and gradient flow from the action loss to the video model. In Table 5a, we find that a

Table 5: **Analysis of design choices.** We examine the effects of (a) teacher noise level, (b) feature extraction layer, (c) self-distillation coefficient, and (d) blocking action gradients to the video backbone on RoboCasa Kitchen. Default settings are highlighted in gray.

(a) Teacher noise

| Teacher $\tau$ | SR (%) |
|---|---|
| 0.00 | 65.3 |
| 0.25 | **68.8** |
| 0.50 | 67.8 |
| 0.75 | 67.6 |

(b) Feature extraction layer

| Layer | SR (%) |
|---|---|
| 10 | 57.5 |
| 14 | 63.0 |
| 18 | **68.8** |
| 22 | 66.8 |
| 26 | 64.7 |

(c) Loss coefficient

| $\lambda_{\text{SD}}$ | SR (%) |
|---|---|
| 0.1 | 68.1 |
| 0.01 | **68.8** |
| 0.001 | 62.7 |

(d) Stop-gradient

| Stop-grad | SR (%) |
|---|---|
| ✗ | 65.6 |
| ✓ | **68.8** |

Table 6: **Generalization across video backbones.** We evaluate FRED with different video diffusion backbones on RoboCasa Kitchen.

| Method | SR (%) |
|---|---|
| DiT4DiT (Wan2.2-5B) | 61.4 |
| **w/ FRED (Ours)** | **68.8** |
| DiT4DiT (Cosmos-Predict2.5-2B) | 56.7 |
| **w/ FRED (Ours)** | **62.1** |

Table 7: **Inference cost.** Average inference latency in the real-world Franka Research 3 setup on an NVIDIA A100 80GB GPU.

| Method | Model Size | # Frames | Time (s, ↓) |
|---|---|---|---|
| $\pi_{0.5}$ | 3B | 1 | 0.261 |
| GR00T N1.7 | 3B | 1 | 0.121 |
| DiT4DiT | 5B | 1 | 0.111 |
| **FRED (Ours)** | 5B | 5 | 0.136 |
| DreamZero | 14B | 5 | 3.255 |
| **FRED (Ours)** | 14B | 5 | 0.406 |

moderately noisy teacher input with $\tau = 0.25$ performs best, outperforming both clean future observations and more heavily corrupted inputs. We hypothesize that moderate corruption provides more informative target representations for self-distillation, consistent with prior observations that diffusion representations can be strongest at intermediate noise levels (Yu et al., 2025; Vélez et al., 2025). In Table 5b, we find that performance peaks at layer 18, while using either shallower or deeper layers degrades performance, which aligns with prior findings in WAMs that intermediate video-model layers provide more effective representations for action conditioning (Ma et al., 2026). In Table 5c, we find that $\lambda_{\text{SD}} = 0.01$ performs best among the evaluated coefficients, while a larger coefficient of 0.1 achieves comparable performance. Finally, in Table 5d, we find that stopping gradients from the action loss to the video model improves the success rate from 65.6% to 68.8%, suggesting that optimizing future video representations directly for action prediction is not necessarily beneficial.

**Generality across Video Backbones.** In Table 6, we evaluate whether the gains of FRED generalize across different video diffusion backbones. We find that FRED consistently improves the success rates of DiT4DiT with both the Wan2.2-5B (Wan et al., 2025) and Cosmos-Predict2.5-2B (NVIDIA et al., 2025) backbones, demonstrating that its gains persist across different video backbones.

**Inference Cost.** In Table 7, we measure the average inference latency of each policy. At the 5B scale, we observe that FRED achieves inference latency comparable to state-of-the-art VLAs, despite conditioning on 5 frames by default, with only a 1.2× increase over 1-frame input. Moreover, FRED is approximately 8× faster than DreamZero while achieving higher success rates, demonstrating the practical efficiency of using a single video denoising step for action generation.

## 5 Conclusion

In this paper, we have presented FRED, a simple framework for training World Action Models (WAMs) that explicitly learns the future representations used for action generation from a single video denoising step via self-distillation. Our key insight is to view the video diffusion model as a future representation generator and explicitly improve these representations through self-distillation, while using observation history as additional temporal context. Our experiments show that FRED consistently improves robot control in complex real-world scenarios, including zero-shot environment generalization and human-to-robot skill transfer. We hope our work on the connection between representation learning and WAMs encourages further work toward efficient and generalizable robot foundation models that better leverage large-scale pre-trained video models. We provide more discussion in Appendix F.

## AI USE STATEMENT

In this work, we used generative AI tools to improve the clarity of the manuscript and to remove backgrounds from photographs used for visualization in Figures 4 and 6. We did not use generative AI tools to generate experimental results, analyses, or scientific claims. We reviewed all AI-assisted text and verified its technical accuracy and consistency with our experiments and cited sources. The authors made all final decisions regarding the manuscript content and presentation. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

## ETHICS STATEMENT

This paper presents a method to advance robot manipulation with World Action Models. Its potential societal impact falls within the established considerations for robot learning, particularly the safe and responsible use of robots in physical environments. Although video generation may raise concerns about hallucinated visual content, our approach obtains future representations for action control through a single denoising step without generating a video. We therefore do not anticipate ethical concerns specific to this method beyond those generally associated with robot learning.

## REPRODUCIBILITY STATEMENT

We provide the main implementation details of FRED in section 4.1 and further details in section A. In addition, we will open-source the source code with the model checkpoint.

## REFERENCES

Michael Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 2025.

Arslan Ali, Junjie Bai, Maciej Bala, Yogesh Balaji, Aaron Blakeman, Tiffany Cai, Jiaxin Cao, Tianshi Cao, Elizabeth Cha, Yu-Wei Chao, et al. World simulation with video foundation models for physical ai. *arXiv preprint arXiv:2511.00062*, 2025.

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

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Transactions on Machine Learning Research*, 2024.

Aritra Bhowmik, Denis Korzhenkov, Cees G Snoek, Amirhossein Habibian, and Mohsen Ghafoorian. Moalign: Motion-centric representation alignment for video diffusion models. In *International Conference on Learning Representations*, 2026.

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

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *IEEE International Conference on Computer Vision*, 2021.

Hila Chefer, Patrick Esser, Dominik Lorenz, Dustin Podell, Vikash Raja, Vinh Tong, Antonio Torralba, and Robin Rombach. Self-supervised flow matching for scalable multi-modal synthesis. In *International Conference on Machine Learning*, 2026.

Tsai-Shien Chen, Aliaksandr Siarohin, Willi Menapace, Ekaterina Deyneka, Hsiang-wei Chao, Byung Eun Jeon, Yuwei Fang, Hsin-Ying Lee, Jian Ren, Ming-Hsuan Yang, et al. Panda-70m: Captioning 70m videos with multiple cross-modality teachers. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2024.

Dyna Robotics. Dyna-2: A 1-million-hour scaling law for world-action models. August 2026. URL https://dyna.co/dyna-2.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

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.

Dengyang Jiang, Mengmeng Wang, Liuzhuozheng Li, Lei Zhang, Haoyu Wang, Wei Wei, Guang Dai, Yanning Zhang, and Jingdong Wang. No other representation component is needed: Diffusion transformers can provide representation guidance by themselves. In *International Conference on Learning Representations*, 2025.

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*, 2022.

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

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, et al. Cosmos policy: Fine-tuning video models for visuomotor control and planning. *arXiv preprint arXiv:2601.16163*, 2026.

Dohun Lee, Hyeonho Jeong, Jiwook Kim, Duygu Ceylan, and Jong Chul Ye. Improving video diffusion transformer training by multi-feature fusion and alignment from self-supervised vision encoders. *arXiv preprint arXiv:2509.09547*, 2025.

Daiqing Li, Huan Ling, Amlan Kar, David Acuna, Seung Wook Kim, Karsten Kreis, Antonio Torralba, and Sanja Fidler. Dreamteacher: Pretraining image backbones with deep generative models. In *IEEE International Conference on Computer Vision*, 2023.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, et al. Causal world modeling for robot control. In *Robotics: Science and Systems*, 2026a.

Ziang Li, Dongzhou Cheng, Yibin Wang, Shiyue Wang, Xiaoyang Xu, Lingxuan Weng, Juan Wang, and Jiaqi Wang. Light-wam: Efficient world action models with state-fusion action decoding. *arXiv preprint arXiv:2606.08242*, 2026b.

Junbang Liang, Pavel Tokmakov, Ruoshi Liu, Sruthi Sudhakar, Paarth Shah, Rares Ambrus, and Carl Vondrick. Video generators are robot policies. *arXiv preprint arXiv:2508.00795*, 2025.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2022.

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

Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*. Springer, 2024.

Teli Ma, Jia Zheng, Zifan Wang, Chunli Jiang, Andy Cui, Junwei Liang, and Shuo Yang. Dit4dit: Jointly modeling video dynamics and actions for generalizable robot control. *arXiv preprint arXiv:2603.10448*, 2026.

Mimic Robotics. Introducing flux-mimic: Scaling video-action models for general purpose dexterity. July 2026. URL https://www.mimicrobotics.com/blog/introducing-flux-mimic.

Kepan Nan, Rui Xie, Penghao Zhou, Tiehan Fan, Zhenheng Yang, Zhijie Chen, Xiang Li, Jian Yang, and Ying Tai. Openvid-1m: A large-scale high-quality dataset for text-to-video generation. *arXiv preprint arXiv:2407.02371*, 2024.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. In *Robotics: Science and Systems*, 2024.

Weili Nie, Julius Berner, Chao Liu, and Arash Vahdat. Nvidia fastgen: Fast generation from diffusion models, 2026. URL https://github.com/NVlabs/FastGen.

NVIDIA. NVIDIA Isaac GR00T N1.7: A foundation model for generalist robots. https://github.com/NVIDIA/Isaac-GR00T, 2026. GitHub repository.

NVIDIA, Arslan Ali, Junjie Bai, Maciej Bala, Yogesh Balaji, Aaron Blakeman, Tiffany Cai, Jiaxin Cao, Tianshi Cao, Elizabeth Cha, Yu-Wei Chao, Prithvijit Chattopadhyay, Mike Chen, Yongxin Chen, Yu Chen, Shuai Cheng, Yin Cui, Jenna Diamond, Yifan Ding, Jiaojiao Fan, Linxi Fan, Liang Feng, Francesco Ferroni, Sanja Fidler, Xiao Fu, Ruiyuan Gao, Yunhao Ge, Jinwei Gu, Aryaman Gupta, Siddharth Gururani, Imad El Hanafi, Ali Hassani, Zekun Hao, Jacob Huffman, Joel Jang, Pooya Jannaty, Jan Kautz, Grace Lam, Xuan Li, Zhaoshuo Li, Maosheng Liao, Chen-Hsuan Lin, Tsung-Yi Lin, Yen-Chen Lin, Huan Ling, Ming-Yu Liu, Xian Liu, Yifan Lu, Alice Luo, Qianli Ma, Hanzi Mao, Kaichun Mo, Seungjun Nah, Yashraj Narang, Abhijeet Panaskar, Lindsey Pavao, Trung Pham, Morteza Ramezanali, Fitsum Reda, Scott Reed, Xuanchi Ren, Haonan Shao, Yue Shen, Stella Shi, Shuran Song, Bartosz Stefaniak, Shangkun Sun, Shitao Tang, Sameena Tasmeen, Lyne Tchapmi, Wei-Cheng Tseng, Jibin Varghese, Andrew Z. Wang, Hao Wang, Haoxiang Wang, Heng Wang, Ting-Chun Wang, Fangyin Wei, Jiashu Xu, Dinghao Yang, Xiaodong Yang, Haotian Ye, Seonghyeon Ye, Xiaohui Zeng, Jing Zhang, Qinsheng Zhang, Kaiwen Zheng, Andrew Zhu, and Yuke Zhu. World simulation with video foundation models for physical ai. *arXiv preprint arXiv:2511.00062*, 2025.

Jonas Pai, Liam Achenbach, Victoriano Montesinos, Benedek Forrai, Oier Mees, and Elvis Nava. mimic-video: Video-action models for generalizable robot control beyond vlas. *arXiv preprint arXiv:2512.15692*, 2025.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE International Conference on Computer Vision*. IEEE, 2023.

Lu Qiu, Yizhuo Li, Yi Chen, Yuying Ge, Yixiao Ge, and Xihui Liu. Making foresight actionable: Repurposing representation alignment in world action models. *arXiv preprint arXiv:2606.12217*, 2026.

Yichao Shen, Fangyun Wei, Zhiying Du, Yaobo Liang, Yan Lu, Jiaolong Yang, Nanning Zheng, and Baining Guo. Videovla: Video generators can be generalizable robot manipulators. In *Advances in Neural Information Processing Systems*, 2026.

Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

Pedro Vélez, Luisa F Polanía, Yi Yang, Chuhan Zhang, Rishabh Kabra, Anurag Arnab, and Mehdi SM Sajjadi. From image to video: An empirical study of diffusion representations. In *IEEE International Conference on Computer Vision*, 2025.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. Videomae v2: Scaling video masked autoencoders with dual masking. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2023.

Weilai Xiang, Hongyu Yang, Di Huang, and Yunhong Wang. Denoising diffusion autoencoders are unified self-supervised learners. In *IEEE International Conference on Computer Vision*. IEEE, 2023.

Xingyi Yang and Xinchao Wang. Diffusion model as representation learner. In *IEEE International Conference on Computer Vision*. IEEE, 2023.

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

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T Freeman. Improved distribution matching distillation for fast image synthesis. In *Advances in Neural Information Processing Systems*, 2024.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *International Conference on Learning Representations*, 2025.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-wam: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026.

Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning. *arXiv preprint arXiv:2407.08693*, 2024.

Xiangdong Zhang, Jiaqi Liao, Shaofeng Zhang, Fanqing Meng, Xiangpeng Wan, Junchi Yan, and Yu Cheng. Videorepa: Learning physics for video generation through relational alignment with foundation models. In *Advances in Neural Information Processing Systems*, 2026.

Weiheng Zhao, Haoyi Jiang, Xin Shi, Liu Liu, Fan Huang, Zhizhong Su, Wei Sui, and Xinggang Wang. Faster-wam: Efficient inference-time future conditioning for robust world action models. *arXiv preprint arXiv:2608.04404*, 2026.

Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. In *International Conference on Learning Representations*, 2022.

# A Implementation Details

**Implementation Details for Real-world Experiments.** For the Franka Research 3 experiments, we use Wan2.2-TI2V-5B and Wan2.1-I2V-14B as the 5B and 14B video diffusion backbones, respectively, while all OpenArm experiments use Wan2.2-TI2V-5B. We condition the video model on 4 historical frames with the current frame, and predict 8 future frames, with a temporal stride of 3. For FRED, we use the same intermediate layer for self-distillation and action conditioning (layer 18 for Wan2.2-5B and layer 23 for Wan2.1-14B). Across all experiments, we use $T_1 = 0.1$, $T_2 = 0.04$, teacher noise level $\tau = 0.25$, and $\lambda_{\text{SD}} = 0.01$. We update the teacher with an exponential moving average (EMA) using the $1 - (1 + t)^{-3/4}$ warmup decay schedule implemented in Diffusers. We normalize actions and proprioceptive states using Q01-Q99 normalization. We train the Franka 5B and 14B models for 150K and 100K iterations with batch size 128. OpenArm models are trained for 50K iterations; robot-only training uses batch size 32, while robot-human co-training uses 32 robot and 32 human samples per batch. All models are optimized with AdamW and cosine learning rate decay. For the 14B FRED model trained on DROID, we set the video-model learning rate to $10^{-5}$, following DreamZero. For all other settings, we use a learning rate of $10^{-4}$ for both the video model and action decoder.

During training, we sample video diffusion timesteps from a shifted-uniform distribution, following the official Wan2.1 and Wan2.2 implementations (Wan et al., 2025), and sample action diffusion timesteps from a $\text{Beta}(1.5, 1.0)$ distribution. At inference, we generate actions using 4 denoising steps.

For baseline comparisons, we follow the official training recipes whenever training is required. Specifically, we train DiT4DiT and Fast-WAM using their official implementations, while AGRA is reproduced in our codebase. We match the number of training iterations and batch size to our corresponding experiments for all three methods. For $\pi_{0.5}$, GR00T N1.7, and DreamZero, we use the publicly released checkpoints. We provide the detailed training configurations of FRED in Table 8.

Table 8: Hyperparameter details for training FRED.

| | Franka Research 3 (Table 1) | | OpenArm (Table 2) | |
|---|---|---|---|---|
| | Wan2.2-5B | Wan2.1-14B | w/o human data | w/ human data |
| **Video Architecture** | | | | |
| Video backbone | Wan2.2-TI2V-5B | Wan2.1-I2V-14B | Wan2.2-TI2V-5B | Wan2.2-TI2V-5B |
| History / future frames | 5 / 8 (stride 3) | 5 / 8 (stride 3) | 5 / 8 (stride 3) | 5 / 8 (stride 3) |
| Input resolution | $288 \times 320$ | $288 \times 320$ | $384 \times 256$ | $384 \times 256$ |
| **Action Architecture** | | | | |
| Num. layers | 16 | 24 | 16 | 16 |
| Hidden dim. | 768 | 1024 | 768 | 768 |
| Num. heads | 12 | 16 | 12 | 12 |
| Action horizon | 24 | 24 | 24 | 24 |
| **Optimization** | | | | |
| Training iterations | 150K | 100K | 50K | 50K |
| Warmup iterations | 5K | 5K | 2.5K | 2.5K |
| Global batch size | 128 | 128 | 32 | 64 (robot 32 + human 32) |
| Optimizer | AdamW | AdamW | AdamW | AdamW |
| Learning rate (video / action) | $10^{-4}$ / $10^{-4}$ | $10^{-5}$ / $10^{-4}$ | $10^{-4}$ / $10^{-4}$ | $10^{-4}$ / $10^{-4}$ |
| $(\beta_1, \beta_2)$ | $(0.9, 0.95)$ | $(0.95, 0.999)$ | $(0.9, 0.95)$ | $(0.9, 0.95)$ |
| Weight decay | $10^{-2}$ | $10^{-5}$ | $10^{-2}$ | $10^{-2}$ |
| Learning rate schedule | Cosine decay | Cosine decay | Cosine decay | Cosine decay |
| Minimum learning rate ratio | 0.1 | 0.1 | 0.1 | 0.1 |

**Implementation Details for Simulation Experiments.** For each simulation benchmark in Table 4, we train all methods with the same Wan2.2-TI2V-5B (Wan et al., 2025) backbone and 16-layer DiT action decoder for 100K iterations with a global batch size of 64. We use three concatenated views (two external and one wrist) at $384 \times 256$ for RoboCasa Kitchen (Nasiriany et al., 2024), one egocentric view at $256 \times 256$ for GR-1 Tabletop (Bjorck et al., 2025), and two concatenated views (one external and one wrist) at $256 \times 512$ for LIBERO (Liu et al., 2023). We fix the action prediction horizon at 16 across all benchmarks, and we normalize proprioceptive states and actions using min-max normalization for RoboCasa Kitchen and LIBERO, and Q01-Q99 normalization for GR-1 Tabletop. We condition the video model on 4 historical frames and the current frame, and predict 8 future frames, with a temporal stride of 2.

All models are optimized with AdamW using cosine learning rate decay and a 5K iteration warmup. We set the peak learning rate to $10^{-4}$ for both the video model and action decoder, while the minimum learning rate is $1\%$ of the peak across all benchmarks. For the remaining FRED-specific hyperparameters and training and inference details, we follow those used in the real-world experiments.

For baseline comparisons, we follow the official implementation of each method if it is available. For VideoREPA (Zhang et al., 2026), we follow its official implementation with a VideoMAEv2 ViT-B encoder (Wang et al., 2023) to produce the alignment target, modifying only the encoder input to match our input resolution and prediction horizon. For SRA (Jiang et al., 2025), we follow its official SiT-XL/2 (Ma et al., 2024) implementation for video representation alignment. For Self-Flow (Chefer et al., 2026), since there is no available official implementation for training, we reproduce the alignment procedure and layer selection described in the original paper.

**Augmentations.** We summarize the visual augmentations used in our experiments in Table 9.

Table 9: PyTorch-like descriptions of image augmentations for FRED.

| Experiment | Descriptions |
|---|---|
| LIBERO | `Resize((H, W))`<br>`RandomResizedCrop((H, W), scale=(0.9, 0.9), ratio=(W/H, W/H))`<br>`RandomBrightness(scale=(0.8, 1.2))`<br>`RandomContrast(scale=(0.8, 1.2))`<br>`RandomSaturation(scale=(0.8, 1.2))`<br>`RandomHue(scale=(-0.05, 0.05))` |
| All others | `Resize((H, W))`<br>`RandomResizedCrop((H, W), scale=(0.9, 0.9), ratio=(W/H, W/H))`<br>`RandomRotate(scale=(-5°, 5°))`<br>`RandomBrightness(scale=(0.8, 1.2))`<br>`RandomContrast(scale=(0.8, 1.2))`<br>`RandomSaturation(scale=(0.8, 1.2))`<br>`RandomHue(scale=(-0.05, 0.05))` |

# B BENCHMARK DETAILS

## B.1 ZERO-SHOT ENVIRONMENT GENERALIZATION TASKS ON DROID

**Hardware Setup.** We evaluate models trained on DROID in a new environment without additional training on data from the evaluation environment. The evaluation setup differs from the training data in location, background, and objects, and includes tasks with instructions not present in DROID. We conduct these experiments using a Franka Research 3 robot arm equipped with a Robotiq 2F-85 gripper. For $\pi_{0.5}$ and GR00T N1.7, we follow the official implementations that use the wrist camera and the left exterior camera. The action space consists of relative joint angles (7D) and an absolute gripper opening (1D). For the remaining models, we use both the left and right exterior cameras. We show the hardware setup in Figure 4.




Figure 4: **DROID evaluation setup.** The setup consists of a Franka Research 3 robot arm with a Robotiq 2F-85 gripper, a wrist camera, and left and right exterior cameras.

**Evaluation Details.** We evaluate four task sets: (a) *in-domain easy tasks*, (b) *in-domain long-horizon tasks*, (c) *out-of-domain easy tasks*, and (d) *object grounding tasks*. We consider a task in-domain if its instruction is present in DROID, and out-of-domain otherwise. The object grounding tasks test whether the policy can distinguish objects with similar colors or shapes. We evaluate each task over 12 trials and report the success rates with partial credit (*e.g.*, 0.5) for partial task completion (*e.g.*, pick only). We limit each trial to 1,500 action steps for long-horizon tasks and 500 action steps for all other tasks (see Figure 5 for visualization of the tasks).

- **(In-domain Easy) PnP Banana.** The policy must pick up the banana and place it in the basket. We place the basket on either the left or the right. We place an apple and a peach near the banana as distractors. We use the instruction *"Put the banana in the basket"* for the 14B model and *"Pick up the banana and place it in the basket"* for the 5B model.
- **(In-domain Easy) Open/Close Microwave.** The policy must open or close the microwave door according to the instruction. We vary the requested action and place the microwave at the left, center, or right of the workspace.
- **(In-domain Easy) PnP Cube.** The policy must move the specified cube to the specified plate. We vary the cube color between green and blue and the target plate between left and right. The instruction is *"Move the (green/blue) cube to the (left/right) plate"*.
- **(In-domain Long-horizon) Clear Table.** The policy must pick up all 5 objects on the table and place them in the box. We use the instruction *"Pick up all the objects on the table and put them in the box"*. Full credit requires placing every object in the box.
- **(Out-of-domain) Reveal.** The policy must lift the mug that covers an object and expose the object underneath. We place distractor objects to the left and to the right of the mug across trials. We use the instruction *"Reveal the object under the mug"*.
- **(Out-of-domain) Stamp.** The policy must pick up the stamp and press it onto the paper to leave a mark. We use the instruction *"Pick up the stamp and stamp on the paper"*.
- **(Object Grounding) PnP Cup.** The policy must select a cup of the instructed color from four identically sized cups (white, gray, blue, and green) and place it on the plate. We vary the target color using the instruction *"Pick up the (white/gray/blue/green) cup and place it on the plate"*.
- **(Object Grounding) PnP Cutlery.** The policy must select a plastic spoon, knife, or fork and place it on the plate. Selecting the correct piece requires distinguishing fine-grained visual differences, such as the shape of its head. We vary the target piece using the instruction *"Pick up the (spoon/knife/fork) and place it on the plate"*.

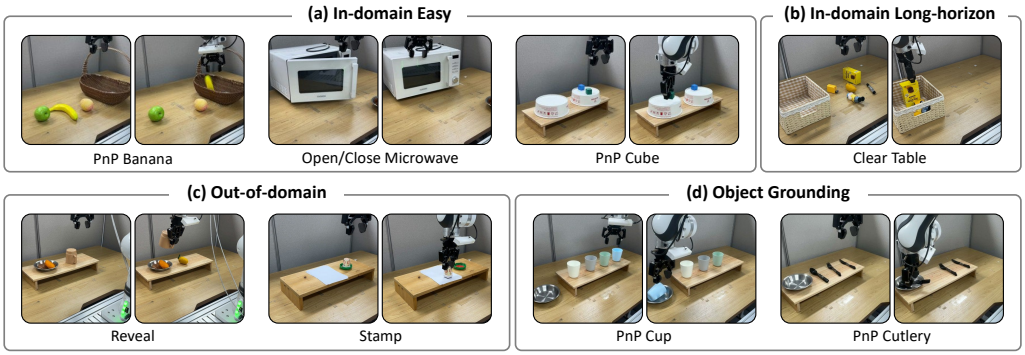



Figure 5: **DROID evaluation tasks.** Task visualization for four task suites: (a) in-domain easy, (b) in-domain long-horizon, (c) out-of-domain, and (d) object grounding. *Clear Table* repeats the pick-and-place step until the table is empty.

### B.2 HUMAN-ROBOT CO-TRAINING ON OPENARM

**Hardware Setup.** We use the OpenArm humanoid[3], which consists of two 7-DoF arms. Each arm is equipped with a 6-DoF Inspire RH56F1 hand, while the head is mounted on a 2-DoF neck supporting pitch and yaw motions. We use relative joint actions for the arms, absolute actions for the hands, and absolute actions for the neck. A ZED 2i stereo camera is mounted on the neck, and we use its left and right egocentric views as visual inputs in all experiments. For data collection, we teleoperate the two arms using ExoArm-7, a wearable exoskeleton compatible with OpenArm. Finger motions are captured using Manus Pro gloves, which provide fingertip positions that are converted to finger joint angles through inverse kinematics. We visualize the hardware setup in Figure 6.




Figure 6: **OpenArm evaluation setup.** It consists of the two 7-DoF OpenArm arms and a ZED 2i stereo camera mounted on the 2-DoF neck, together with the objects used in the evaluation tasks.

[3]https://openarm.dev/

**Training Dataset.** For the robot dataset, we collect 553 teleoperated demonstrations on the OpenArm platform, comprising 192K frames. The dataset covers 53 language instructions across 4 skills and 6 object categories; see Figure 7 for the distributions over skills and object categories.

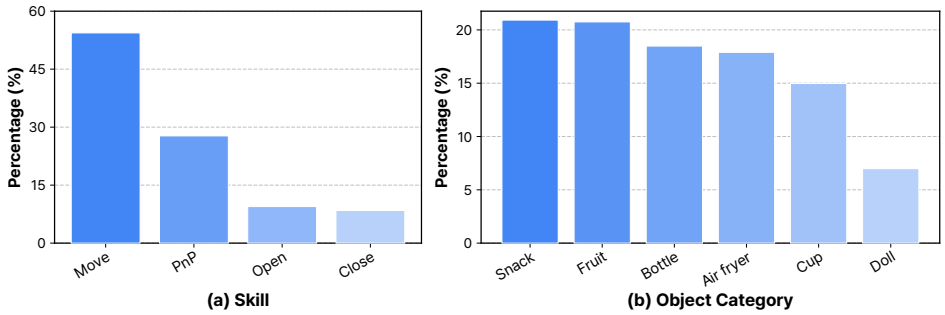



Figure 7: **Distribution of skills and object categories in the OpenArm robot dataset.**

For human co-training, we additionally collect 1,187 egocentric human videos comprising 321K frames, covering 7 skills and 7 object categories; see Figure 8 for the distributions over skills and object categories.

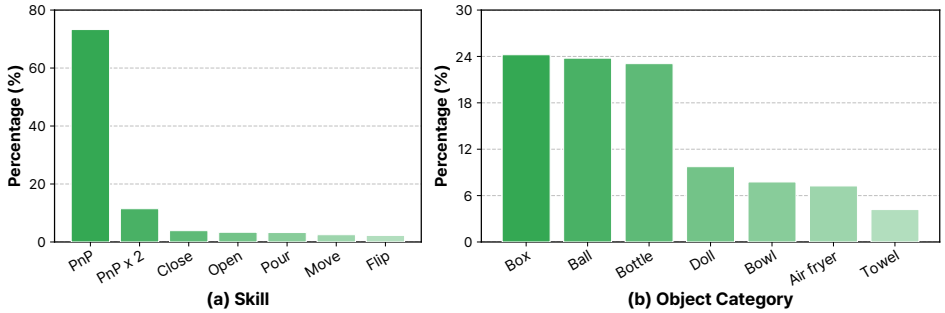



Figure 8: **Distribution of skills and object categories in the human video dataset.**

**Evaluation Details.** We evaluate seven tasks on OpenArm, with 12 trials per task. We report mean task scores as percentages, including partial credit for partial completion. The task configurations are described below (see Figure 9 for visualization of the tasks).

- **(In-domain Atomic) PnP Banana.** The policy must place the banana in the basket. We use six initial positions with two orientations each. We use the instruction *"Place the banana into the basket."*
- **(In-domain Atomic) Open/Close Air Fryer.** The policy must open or close the air fryer. We conduct six opening trials and six closing trials. We use the instruction *"(Open/Close) the air fryer"*.
- **(In-domain Atomic) PnP Apple/Mango.** The policy must pick up an apple or a red mango and place it in the air fryer. We conduct six trials per fruit. We use the instruction *"Pick up the (apple/red mango) and place it in the air fryer"*.
- **(In-domain Composite) Open Air Fryer and Place Apple.** The policy must open the air fryer and then place an apple inside. We use the instruction *"Open the air fryer, pick up the apple and place it in the air fryer"*.
- **(Human Skill Transfer) Place Bottle on Basket.** The policy must place the bottle on the basket. We use the instruction *"Pick up the bottle and put it on the basket"*.
- **(Human Skill Transfer) PnP Novel Objects.** The policy must place a bottle, box, ball, or doll in the white container. We conduct three trials per object. We use the instruction *"Pick up the (bottle/box/ball/doll) and place it in the white container"*.
- **(Human Skill Transfer) Pour Coffee.** The policy must pour coffee from the bottle into the red cup. Full credit requires holding the bottle stably and pouring into the cup. We use the instruction *"Pour the coffee from the bottle into the red cup"*.

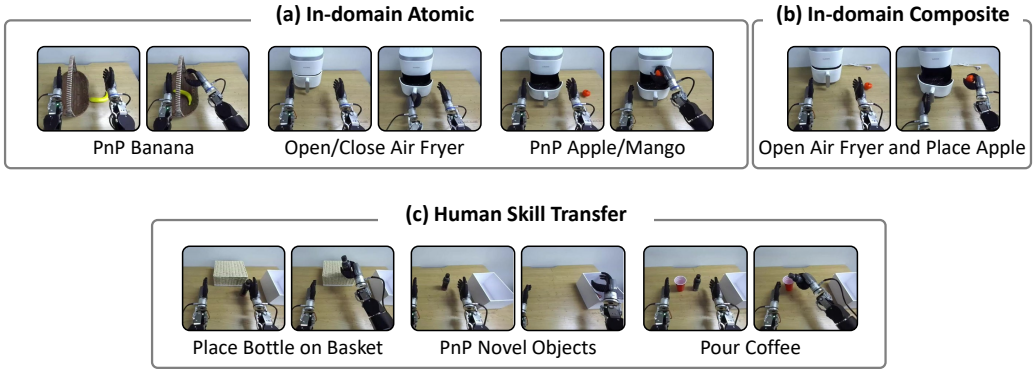



Figure 9: **OpenArm evaluation tasks.** The seven tasks, grouped into (a) in-domain atomic, (b) in-domain composite, and (c) human skill transfer. Each task is shown by the initial and final egocentric frames of one episode.

## C BASELINES

We consider vision-language-action models and video diffusion-based models.

- $\pi_{0.5}$ (Intelligence et al., 2025) is a diffusion-based Vision-Language-Action model (VLA) that combines a vision-language backbone with a flow-matching action transformer. It shares self-attention layers between VLM and diffusion transformer.
- **GR00T N1.7** (NVIDIA, 2026) is also a diffusion-based VLA, but it feeds the intermediate hidden states of the VLM into the cross-attention layer of the diffusion transformer.
- **DreamZero** (Ye et al., 2026) is a World Action Model (WAM) built on a pretrained video diffusion model that jointly predicts future videos and robot actions through a single backbone.
- **DiT4DiT** (Ma et al., 2026) is a WAM that uses separate video and action diffusion transformers, where intermediate denoising features from the video transformer condition the action transformer through cross-attention without decoding future frames.
- **Fast-WAM** (Yuan et al., 2026) is also a WAM that uses separate video and action diffusion transformers, while using attention masking to prevent action prediction from depending on future video tokens, thereby omitting future video prediction at inference.
- **AGRA** (Qiu et al., 2026) extends separate video-action architecture with an action-grounded representation alignment objective that aligns intermediate video diffusion features with spatially coherent features from a frozen visual encoder, while jointly updating the video backbone with the robot action objective.

# D Details of Analysis Setup

We describe how we evaluate one-step video generation quality and design an attentive probing protocol used to evaluate the quality of video representations extracted from observed futures and generated from noise in Figures 3a to 3c.

## D.1 Video Generation

**Video Generation Quality Evaluation and DMD2 Implementation.** We evaluate one-step video generation quality using Fréchet Video Distance (FVD; Unterthiner et al. 2018) between 2,048 generated videos and 2,048 randomly sampled clips from the RoboCasa Kitchen training set (Nasiriany et al., 2024). For DMD2 (Yin et al., 2024), we train a one-step student in two stages. First, we fine-tune the Wan2.2-TI2V-5B model (Wan et al., 2025) on RoboCasa Kitchen (Nasiriany et al., 2024) for 100K iterations with a global batch size of 64. We then distill the fine-tuned model's EMA teacher into a one-step student, following the NVIDIA FastGen implementation (Nie et al., 2026), for 5k iterations with a global batch size of 16, without classifier-free guidance (Ho & Salimans, 2022) and with a GAN loss weight of 0.1.

## D.2 Attentive Probe

**Motion-primitive Labeling.** We partition each trajectory into non-overlapping 16-frame clips and categorize each clip into one of 15 motion-primitive labels, following Zawalski et al. (2024). We determine the label from the dominant accumulated end-effector motion over the clip, with gripper state changes taking priority (see Table 10 for the class distribution).

Table 10: Distribution of motion-primitive labels in the probing validation dataset. CW and CCW denote clockwise and counterclockwise rotation, respectively.

| Primitive | (%) | Primitive | (%) | Primitive | (%) |
|---|---|---|---|---|---|
| Forward | 21.82 | Down | 13.48 | Left | 12.91 |
| Right | 12.27 | Backward | 10.22 | Up | 9.64 |
| Stop | 5.16 | Close gripper | 4.17 | Open gripper | 3.11 |
| Roll right | 2.39 | Tilt up | 1.26 | Rotate CW | 1.32 |
| Roll left | 0.76 | Tilt down | 0.70 | Rotate CCW | 0.80 |

**Attentive Probe Architecture.** Following recent self-supervised learning literature, we use a lightweight attentive probe on top of the frozen video representations (Bardes et al., 2024; Assran et al., 2025) to evaluate the quality of video representation. Specifically, it consists of a single learnable query token cross-attending to the video tokens through a 12-head cross-attention block, followed by an MLP and a linear classifier for 15-way motion primitive classification.

**Attentive Probe Training and Evaluation.** We train the attentive probe using 300 demonstrations per task used for policy training on RoboCasa Kitchen (Nasiriany et al., 2024), and evaluate it on 50 newly generated demonstrations per task. We extract the intermediate future-token representations from layer 18 of the frozen video backbone and train the probe for 10 epochs with AdamW optimizer using a batch size of 128, a learning rate of $10^{-3}$, and weight decay of 0.05. We report the best validation top-1 accuracy over the 10 epochs.

## E Flow Matching Process used in FRED

We briefly describe conditional flow matching (Lipman et al., 2022; Albergo et al., 2025) as a common formulation for video and action generation.

Let $\mathbf{x}$ denote a clean target (*e.g.*, future video latents or an action sequence) and $\mathbf{c}$ denote the conditioning inputs (*e.g.*, task instructions) that define the conditional target distribution $p(\mathbf{x}|\mathbf{c})$. We construct a noisy input $\mathbf{x}^\tau$ by linearly interpolating between $\mathbf{x}$ and Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$:

$$\mathbf{x}^\tau = (1-\tau)\mathbf{x} + \tau\boldsymbol{\epsilon}, \quad \tau \in [0,1], \tag{5}$$

where $\tau$ denotes the diffusion timestep, with $\tau = 0$ corresponding to clean data and $\tau = 1$ to pure noise. Our video and action generation models are each parameterized by a Diffusion Transformer (DiT; Peebles & Xie 2023), which serves as a conditional velocity network $\mathbf{v}_\theta(\mathbf{x}^\tau, \tau \mid \mathbf{c})$ trained to reconstruct the target velocity $\boldsymbol{\epsilon} - \mathbf{x}$:

$$\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}_{(\mathbf{x},\mathbf{c}),\boldsymbol{\epsilon},\tau}\left[\left\|\mathbf{v}_\theta(\mathbf{x}^\tau, \tau|\mathbf{c}) - (\boldsymbol{\epsilon} - \mathbf{x})\right\|_2^2\right]. \tag{6}$$

At inference, samples are generated from Gaussian noise by integrating the learned velocity from $\tau = 1$ to $\tau = 0$. An Euler sampler (Karras et al., 2022) performs this integration through repeated updates,

$$\mathbf{x}^{\tau-\Delta\tau} = \mathbf{x}^\tau - \Delta\tau\, \mathbf{v}_\theta(\mathbf{x}^\tau, \tau|\mathbf{c}), \tag{7}$$

where $\Delta\tau > 0$ denotes the step size. In our one-step generation setting, we perform a single Euler step with $\Delta\tau = 1$, *i.e.*, the velocity prediction $\mathbf{v}_\theta(\mathbf{x}^\tau, \tau \mid \mathbf{c})$ computed from a single forward pass with $\mathbf{x}^{\tau=1} = \boldsymbol{\epsilon}$. During this forward pass, the velocity network's intermediate hidden states $\mathbf{h}$ are extracted to condition action generation.

## F Limitations and Future Work

A potential limitation of FRED is its reliance on the language understanding capabilities of the underlying video diffusion backbone, as our training objective focuses on future representation learning and does not provide additional supervision for language grounding. As shown in Table 2b, without human-video co-training, FRED achieves 0% instruction-following performance on the human skill transfer tasks, while pre-trained VLA baselines retain some instruction-following ability. Co-training with human videos substantially improves this performance, suggesting that stronger language grounding may further benefit generalization to unseen instructions. Future work could combine the spatiotemporal priors of video diffusion models with the stronger language grounding of pre-trained vision-language models.

Another limitation is that human-to-robot transfer in FRED remains implicit: human and robot videos jointly train the video model, but we do not explicitly learn correspondences between their representations across embodiments. Although human-video co-training substantially improves instruction following, task success remains moderate. Future work could explore explicit cross-embodiment representation learning to better transfer human-demonstrated skills to robot control.

As future work, we believe it would be interesting to extend our approach to more scalable video diffusion models with stronger robot-domain priors. Current pre-trained video diffusion models are trained predominantly on human videos, often requiring substantial adaptation for downstream robotic tasks. Larger robot-pre-trained video backbones may provide a better starting point for future representation learning, consistent with the larger gains observed at the 14B scale in Table 1.

# G FULL RESULTS

## G.1 DROID

We report the full per-task results of the DROID experiments in Table 11.

Table 11: **Per-task results on zero-shot generalization to a new real-world environment.** Performance of policies trained on the DROID dataset without environment-specific fine-tuning. We report the average task progress rate (%) for in-domain long-horizon tasks, and the average success rate (%) for other task types.

| | | In-domain easy | | | Long-horizon | Out-of-domain | | Object grounding | |
|---|---|---|---|---|---|---|---|---|---|
| Method | Backbone | Banana | Microwave | Cube | Clear Table | Reveal | Stamp | Cup | Cutlery |
| $\pi_{0.5}$ (Intelligence et al., 2025) | – | 100.0 | 66.7 | 8.3 | 53.3 | 16.7 | 16.7 | 25.0 | 8.3 |
| GR00T N1.7 (NVIDIA, 2026) | – | 91.7 | 25.0 | 8.3 | 60.0 | 8.3 | 29.2 | 25.0 | 20.8 |
| Fast-WAM (Yuan et al., 2026) | Wan2.2-5B | 41.7 | 33.3 | 4.2 | 26.7 | 4.2 | 0.0 | 8.3 | 12.5 |
| DiT4DiT (Ma et al., 2026) | Wan2.2-5B | 33.3 | 8.3 | 8.3 | 16.7 | 0.0 | 0.0 | 8.3 | 8.3 |
| AGRA (Qiu et al., 2026) | Wan2.2-5B | 100.0 | 25.0 | 0.0 | 31.7 | 12.5 | 8.3 | 25.0 | 41.7 |
| **FRED (Ours)** | Wan2.2-5B | 87.5 | 29.2 | 25.0 | 46.7 | 41.7 | 33.3 | 41.7 | 33.3 |
| DreamZero (Ye et al., 2026) | Wan2.1-14B | 83.3 | 50.0 | 58.3 | 61.7 | 0.0 | 20.8 | 54.2 | 25.0 |
| **FRED (Ours)** | Wan2.1-14B | 91.7 | 70.8 | 75.0 | 81.7 | 33.3 | 70.8 | 50.0 | 58.3 |

### G.2 OpenArm

We report the full per-task results of the OpenArm experiments in Table 12.

Table 12: **Per-task results on humanoid control with and without robot-human co-training.** Performance of policies trained on the OpenArm humanoid using robot demonstrations only or with additional human videos. We report the average success rate (%) on in-domain atomic tasks, the average task progress rate (%) on in-domain composite tasks, and both task success (%) and instruction-following (IF) performance (%) on human skill transfer tasks.

(a) In-domain tasks.

| | | Atomic | | | Composite |
|---|---|---|---|---|---|
| Method | Training Data | PnP Banana | Air Fryer | PnP Apple/Mango | Air Fryer + Apple |
| $\pi_{0.5}$ (Intelligence et al., 2025) | – | 33.3 | 100.0 | 66.7 | 50.0 |
| GR00T N1.7 (NVIDIA, 2026) | – | 91.7 | 100.0 | 83.3 | 45.8 |
| DreamZero (Ye et al., 2026) | Robot only | 58.3 | 100.0 | 45.8 | 0.0 |
| Fast-WAM (Yuan et al., 2026) | Robot only | 29.2 | 66.7 | 75.0 | 0.0 |
| DiT4DiT (Ma et al., 2026) | Robot only | 83.3 | 100.0 | 50.0 | 50.0 |
| AGRA (Qiu et al., 2026) | Robot only | 79.2 | 91.7 | 75.0 | 8.3 |
| **FRED (Ours)** | Robot only | 100.0 | 100.0 | 75.0 | 70.8 |
| DreamZero (Ye et al., 2026) | Robot + human | 70.8 | 100.0 | 33.3 | 0.0 |
| Fast-WAM (Yuan et al., 2026) | Robot + human | 29.2 | 91.7 | 75.0 | 0.0 |
| DiT4DiT (Ma et al., 2026) | Robot + human | 83.3 | 100.0 | 75.0 | 41.7 |
| AGRA (Qiu et al., 2026) | Robot + human | 50.0 | 83.3 | 91.7 | 12.5 |
| **FRED (Ours)** | Robot + human | 100.0 | 100.0 | 91.7 | 75.0 |

(b) Human skill transfer tasks.

| | | Bottle on basket | | Novel Objects | | Pour Coffee | |
|---|---|---|---|---|---|---|---|
| Method | Training Data | Success | IF | Success | IF | Success | IF |
| $\pi_{0.5}$ (Intelligence et al., 2025) | – | 0.0 | 25.0 | 0.0 | 0.0 | 0.0 | 25.0 |
| GR00T N1.7 (NVIDIA, 2026) | – | 0.0 | 0.0 | 16.7 | 100.0 | 0.0 | 0.0 |
| DreamZero (Ye et al., 2026) | Robot only | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| Fast-WAM (Yuan et al., 2026) | Robot only | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| DiT4DiT (Ma et al., 2026) | Robot only | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| AGRA (Qiu et al., 2026) | Robot only | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| **FRED (Ours)** | Robot only | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| DreamZero (Ye et al., 2026) | Robot + human | 0.0 | 33.3 | 0.0 | 8.3 | 0.0 | 0.0 |
| Fast-WAM (Yuan et al., 2026) | Robot + human | 0.0 | 16.7 | 0.0 | 0.0 | 0.0 | 16.7 |
| DiT4DiT (Ma et al., 2026) | Robot + human | 0.0 | 0.0 | 33.3 | 75.0 | 66.7 | 100.0 |
| AGRA (Qiu et al., 2026) | Robot + human | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| **FRED (Ours)** | Robot + human | 25.0 | 91.7 | 66.7 | 100.0 | 58.3 | 100.0 |