# VID2WAM: DISTILLING VIDEO DIFFUSION PRIORS INTO WORLD ACTION MODELS

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

## ABSTRACT

World Action Models (WAMs) improve robot policy learning by jointly modeling future visual dynamics and actions. However, their future prediction targets typically come from recorded robot trajectories, limiting supervision when target-task demonstrations are scarce. Therefore, we ask whether future supervision for WAMs must originate from target-task expert trajectories. In this paper, we propose **Vid2WAM**, an offline distillation framework that transfers visual diffusion priors from a large video foundation model into a compact WAM student. Given an observation and language instruction, Vid2WAM distills supervision through two complementary channels: task-conditioned future rollouts directly supervise the student's future prediction branch, while an inverse dynamics model recovers embodiment-specific pseudo-actions for action learning. To integrate synthetic and real supervision, we introduce *source-aware residual action adaptation* that learns source-specific corrections around a shared action backbone and mitigates interference from noisy pseudo-actions. During inference, both the video teacher and inverse dynamics model are discarded, leaving only the WAM student for efficient deployment. Simulation and real-world experiments show gains under limited expert demonstrations and in target-conditioned adaptation to novel tasks without target-task action trajectories, while preserving low-latency inference.

## 1 INTRODUCTION

Building robot policies that generalize across tasks, objects, and environments remains a central goal of embodied intelligence. Recent progress has followed two complementary directions. Vision–language–action (VLA) models leverage Internet-scale vision–language pretraining to transfer broad semantic and linguistic knowledge into control (Zitkovich et al., 2023; Kim et al., 2025; Black et al., 2024). Meanwhile, video-based policies and World Action Models (WAMs) incorporate future prediction into policy learning, providing spatiotemporal supervision over motion, interaction, and task progression beyond static semantic supervision and instantaneous action labels (Du et al., 2023; Wu et al., 2024; Hu et al., 2025; Xu et al., 2025; Kim et al., 2026a; Ye et al., 2026).

A key advantage of WAMs stems from their future-predictive supervision. Rather than learning only from action labels, they are optimized to anticipate future visual dynamics, providing dense temporal learning signals that improve action representations even when explicit future rollouts are not required during deployment (Li et al., 2025; Xu et al., 2025; Yuan et al., 2026). However, most existing WAMs implicitly share one assumption: *future supervision must be observed in task-specific expert demonstrations*. Both future visual targets and action labels are obtained from paired robot trajectories collected through simulation or teleoperation. Their future supervision is therefore limited by the scale and diversity of recorded robot demonstrations, particularly when target-task expert trajectories are scarce or unavailable.

We thus argue that the assumption above is unnecessarily restrictive: video foundation models can offer another source of future supervision. After adaptation with limited embodiment-specific data, they can be queried with diverse initial observations and language instructions to synthesize task-conditioned rollouts for new tasks and environments (Jang et al., 2025; Wang et al., 2026). This process requires an initial observation rather than a complete target-task trajectory, allowing the conditioning distribution to be expanded using simulator snapshots, real-world observations, or image synthesis. An inverse dynamics model (IDM) can recover embodiment-specific pseudo-actions from generated rollouts (Jang et al., 2025; Ye et al., 2025). However, treating these rollouts only



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.

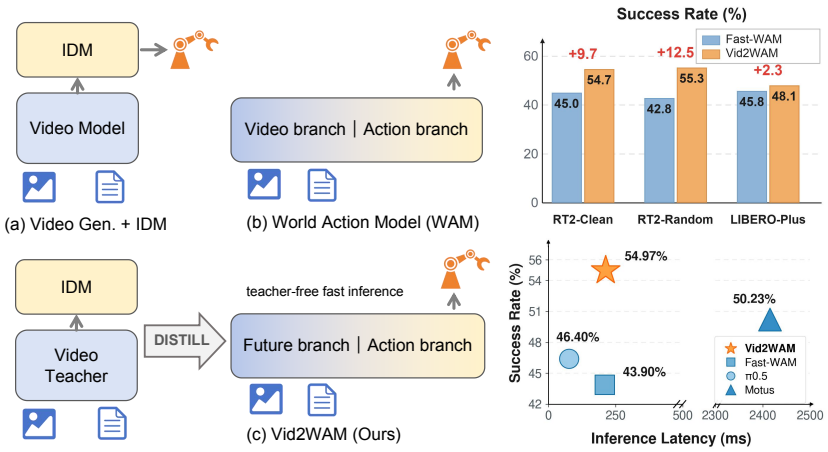


Figure 1: **Left:** Comparison of video-based robot policy paradigms. (a) Video generation plus IDM uses the video model during execution. (b) WAM jointly learns future prediction and action generation from real video–action trajectories. (c) Vid2WAM distills teacher-generated futures and IDM pseudo-actions into a compact WAM for teacher-free inference. **Top right:** RoboTwin clean and randomized bars report held-out-task success; LIBERO-Plus bars report the full-benchmark average (seen and held-out tasks) in the novel regime. **Bottom right:** Vid2WAM achieves the highest novel-task success rate among the evaluated methods while retaining low inference latency.

as pseudo-labeled action data means that the generated futures do not directly supervise the student's future prediction, and action supervision mainly depends on imperfect recovery from IDMs. Ambiguous transitions, generation artifacts, and physically inconsistent motion can further corrupt these pseudo-actions (Wang et al., 2026). This motivates transferring generated rollouts through complementary visual and action channels.

This paper presents **Vid2WAM**, a framework for distilling a large video foundation model into an efficient WAM student. Given an initial observation and language instruction, an embodiment-adapted video teacher generates a future rollout. We decode this rollout into pixels and re-encode it into the student's latent space, providing representation-compatible supervision for the future prediction branch. In parallel, an IDM recovers embodiment-specific pseudo-actions that supervise action learning. To mitigate the interference from pseudo-action noise, we further introduce *source-aware residual action adapters*, which provide source-specific residual corrections around a shared action backbone. At inference time, the video teacher and IDM are removed, leaving only the compact WAM student for direct action generation.

Our contributions are threefold. First, we use generated futures from a video foundation model as direct future-prediction supervision for a WAM beyond recorded target-task expert trajectories. Second, we introduce a dual-channel, source-aware transfer strategy that jointly distills future visual dynamics and IDM-inferred actions, with residual adapters reducing interference from noisy pseudo-actions. Third, we demonstrate gains under limited expert demonstrations and in target-conditioned adaptation to novel tasks without target-task action trajectories, while retaining efficient inference.

## 2 Related Work

**World Models for Robot Control.** Video models provide broad priors over motion, interaction, and temporal task structure. One line of work uses these priors explicitly at deployment, generating visual subgoals or future rollouts and converting them into actions through inverse dynamics, goal-conditioned control, motion tracking, or test-time action selection (Du et al., 2023; 2024; Zhou et al., 2024; Liang et al., 2025; Collins et al., 2025; Zhou et al., 2026). Although effective for generalization, this paradigm incurs generation latency and exposes control to rollout errors. A related line of work jointly models future observations and robot actions, using future prediction to provide temporally structured supervision for policy learning (Wu et al., 2024; Cheang et al., 2024; Hu et al., 2025; Li et al., 2025; Zhu et al., 2025). Fast-WAM shows that video co-training can remain beneficial even when future generation is removed at inference (Yuan et al., 2026), while

Efficient-WAM reduces cost of retaining future imagination (Li et al., 2026). However, these WAMs derive future and action supervision primarily from recorded robot trajectories. Vid2WAM instead leverages a video foundation model to generate task-conditioned future rollouts offline and distills them into a compact WAM without invoking the video teacher at deployment.

**Video-Generated Supervision for Robot Learning**. Early generative augmentation methods edit objects, backgrounds, and other visual factors in recorded demonstrations while reusing their actions (Chen et al., 2023; Yu et al., 2023). More recent work also synthesizes or edits initial observations, expanding the states from which robot experience can be generated (Kim et al., 2026b). DreamGen provides the closest data-centric pipeline: it adapts a video model to a target embodiment, generates task-conditioned rollouts, and recovers pseudo-actions using an inverse dynamics or latent action model (Jang et al., 2025; Ye et al., 2025). Other video-prior policies retain a video model at deployment: mimic-video decodes actions from intermediate video-model features (Pai et al., 2025), Vidar recovers actions from generated videos with a masked IDM (Feng et al., 2025), and Video2Act asynchronously updates video-derived features for a fast action head (Jia et al., 2025). These methods use video-model computation at deployment, with different action-decoding and update paths.

In action-only synthetic-data pipelines, generated frames and recovered actions form synthetic demonstrations, but the generated futures do not directly supervise a student's future-prediction objective. This action supervision also highly depends on imperfect pseudo-labels. Prior work mitigates this issue by improving the executability of generated videos (Wang et al., 2026) or by avoiding synthetic action supervision and routing geometric targets to the visual backbone (Chen et al., 2026). Vid2WAM instead treats each generated rollout as dual supervision: future dynamics directly supervise world modeling, while IDM-derived pseudo-actions provide embodiment-specific action grounding through source-aware residual adaptation.

**Distilling Foundation-Model Priors into Robot Policies.** Knowledge distillation transfers capabilities from a large teacher into a smaller student (Hinton et al., 2015), and subsequent work has extended this principle from output imitation to teacher-generated data and intermediate supervision (Hsieh et al., 2023; Mukherjee et al., 2023; Li et al., 2023). In robotics, prior methods distill planner-generated experience or foundation-model representations into deployable policies (Ha et al., 2023; Shang et al., 2024). More closely related, S-VAM self-distills multi-step generated videos into one-step geometric and semantic foresight, while PFD distills action corrections induced by privileged recorded futures (Yan et al., 2026; Fang et al., 2026). CKT-WAM transfers intermediate context between heterogeneous WAMs, whereas Efficient-WAM compresses the computation required for future imagination (Jiang et al., 2026; Li et al., 2026). Vid2WAM uses an external pretrained video model to generate futures beyond recorded target-task expert trajectories, whereas S-VAM self-distills a WAM's generated foresight and PFD uses privileged recorded futures for action correction. Each Vid2WAM teacher rollout supervises both future prediction and action learning; the video teacher and IDM are discarded during deployment.

# 3 METHOD

**Problem Formulation and Overview** Given a current visual observation $o_0$, proprioceptive state $e_0$, and language instruction $\ell$, we aim to learn a policy that predicts an executable action chunk $a_{1:H}$. Following the World Action Model (WAM) paradigm, the student is jointly trained on an action predictor and an auxiliary future-video predictor through a shared video-action representation.

As illustrated in Fig. 2, Vid2WAM transfers the predictive prior of a large video teacher into the student using both real demonstrations and offline teacher-generated rollouts. Real demonstrations provide expert actions and observed future-video supervision, whereas teacher rollouts provide generated future-video latents and IDM-derived pseudo-actions. The video teacher and IDM are used only offline; at deployment, the teacher, IDM, auxiliary future-video prediction head, and pseudo-source adapters are discarded, leaving the student WAM to efficiently generate action chunks from the current policy condition.

## 3.1 TEACHER-DERIVED DISTILLATION TARGETS

We convert the video teacher's generative prior into two complementary forms of student-compatible supervision: future-video latent targets and IDM-derived pseudo-action targets.

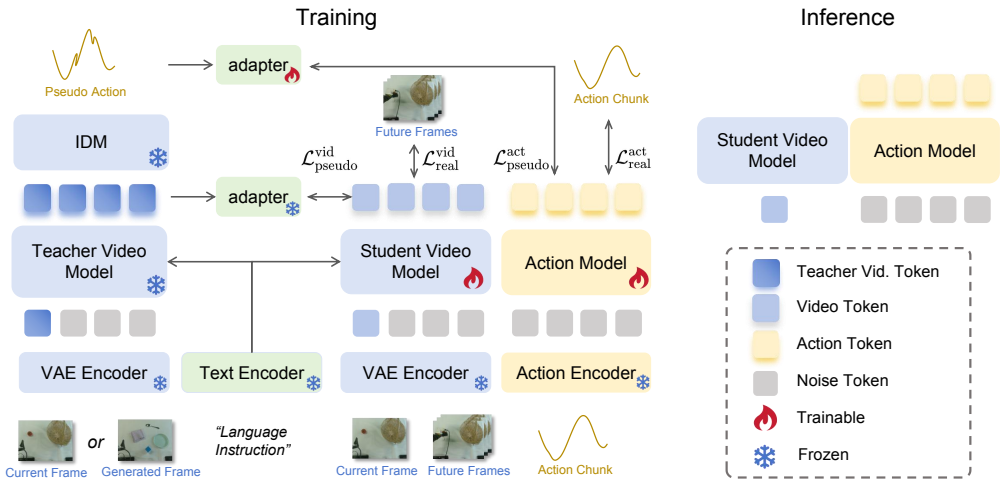


Figure 2: **Overview of Vid2WAM.** Vid2WAM distills future prediction priors from a pretrained video diffusion model into a compact World Action Model (WAM). During training, the teacher generates future videos and IDM-derived pseudo actions supervise the student WAM, while only the student is used for deployment. At inference, a single current-frame pass through the shared video backbone produces latent world features for direct action prediction.

**Video teacher adaptation.** We instantiate the video teacher $T_\psi$ with a large pretrained video diffusion model and fine-tune it using only a limited set of demonstration videos. This lightweight adaptation specializes the teacher to the robot's appearance and embodiment-specific physical constraints, while preserving the motion-prediction and instruction-following capabilities inherited from large-scale pretraining. Consequently, a small amount of in-domain demonstration data is sufficient to elicit robot-aware rollout generation. Given an initial observation $o_0$ and language instruction $\ell$, the adapted teacher generates

$$\hat{v}_{1:T} = T_\psi(o_0, \ell). \tag{1}$$

After adaptation, the teacher is frozen and used exclusively for offline rollout generation.

**Inverse dynamics model training.** We train separate inverse dynamics models (IDMs) for actions and proprioceptive states from visual trajectories:

$$\hat{a}_{1:H} = \mathrm{IDM}_a(\hat{v}_{1:T}), \qquad \hat{e}_{1:H} = \mathrm{IDM}_e(\hat{v}_{1:T}). \tag{2}$$

where $\hat{a}_{1:H}$ and $\hat{e}_{1:H}$ denote the predicted action chunk and robot state sequence respectively. The IDMs are supervised using action-labeled robot trajectories, but do not require language instructions or task-specific semantic annotations. It can therefore be trained using task-agnostic interaction data collected with the same robot embodiment and compatible observation setup, without requiring expert demonstrations for the target tasks. Once trained, the IDM is frozen and used only as a labeler for teacher-generated rollouts.

**Pseudo-data generation and labeling.** Vid2WAM keeps the real demonstration set fixed and collects an additional pool of initial states. The frozen video teacher produces visual rollouts from these initial states, which are re-encoded by the student VAE into student-space future latents $\hat{z}_{1:T}$. The IDM then recovers the corresponding pseudo-action chunks $\hat{a}_{1:H}$ and robot state sequences $\hat{e}_{1:H}$ from the generated rollouts. The resulting tuples $(o_0, \ell, \hat{z}_{1:T}, \hat{a}_{1:H}, \hat{e}_{1:H})$ are stored in an offline distillation buffer. During training, $\hat{e}_{1:H}$ serves as the WAM's proprioceptive input condition, while $\hat{z}_{1:T}$ and $\hat{a}_{1:H}$ provide future-video and action targets respectively.

### 3.2 Source-Aware Student Training

Vid2WAM jointly trains the student on real demonstrations and offline teacher-generated pseudo rollouts. Both domains share the video and action backbones.

**Source-specific residual adaptation.** Directly mixing expert actions with IDM-recovered pseudo-actions can introduce interference because the latter contain errors from video generation and inverse

dynamics. We therefore place lightweight residual adapters before and after the shared action backbone, with separate parameters for the real and pseudo domains. For an action token $x$ from domain $d \in \{\text{real}, \text{pseudo}\}$, an adapter computes

$$\text{Ada}_d(x) = x + \alpha W^d_{\text{up}} \sigma\big(W^d_{\text{down}} \,\text{LN}(x)\big)\,, \tag{3}$$

where $W^d_{\text{down}}$ and $W^d_{\text{up}}$ form a bottleneck projection, $\sigma$ is a nonlinear activation, and $\alpha$ controls the residual contribution. We initialize $W^d_{\text{up}}$ to zero, so each adapter initially implements an identity mapping. The shared representation therefore remains the default path, while the residual branches provide source-dependent corrections. Only the real-domain adapter is retained at inference time.

**Training objective.** We use the flow-matching objective for action chunks and future-video latents (Lipman et al., 2023). Let $y$ denote either target, we sample Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$ and $t \in (0, 1)$, and construct

$$y_t = (1 - t)y + t\epsilon. \tag{4}$$

For each domain $d$, the model predicts the corresponding velocity field with flow-matching loss

$$\mathcal{L}_{\text{FM}}(y, d) = \mathbb{E}_{y,\epsilon,t}\Big[\|f_\theta(y_t, t, c; d) - (\epsilon - y)\|_2^2\Big]\,. \tag{5}$$

We group the four training terms into real- and pseudo-domain objectives:

$$\begin{aligned} \mathcal{L}_{\text{real}} &= \mathcal{L}_{\text{FM}}(a, \text{real}) + \beta\mathcal{L}_{\text{FM}}(z, \text{real}), \\ \mathcal{L}_{\text{pseudo}} &= \mathcal{L}_{\text{FM}}(\hat{a}, \text{pseudo}) + \beta\mathcal{L}_{\text{FM}}(\hat{z}, \text{pseudo}), \end{aligned} \tag{6}$$

where $(a, z)$ are the action and future-video targets from real demonstrations, while $(\hat{a}, \hat{z})$ are the corresponding targets from teacher-generated distillation buffers. The complete training objective is

$$\mathcal{L} = \lambda_{\text{real}}\mathcal{L}_{\text{real}} + \lambda_{\text{pseudo}}\mathcal{L}_{\text{pseudo}}, \tag{7}$$

where $\beta$ balances action and video supervision, and $\lambda_{\text{real}}$ and $\lambda_{\text{pseudo}}$ balance the two domains.

# 4 Experiments

We investigate four questions: (1) how well existing pretrained VLA and WAM policies transfer to novel tasks; (2) whether Vid2WAM improves performance on such tasks without target-task expert trajectories; (3) whether it improves task success and perturbation robustness under a fixed, limited budget of real demonstrations; and (4) how its supervision signals and source-aware components contribute to these gains. We conduct experiments on RoboTwin 2.0 (Chen et al., 2025b), LIBERO (Liu et al., 2023), LIBERO-Plus (Fei et al., 2025), and a real-world bimanual platform.

## 4.1 Experimental Setup

**Simulation Benchmarks.** For RoboTwin 2.0, we evaluate bimanual manipulation for 50 tasks under both clean and randomized conditions, with 100 trials per task for every policy. For LIBERO, we use the Spatial, Object, Goal and Long suites, each with 10 tasks and evaluate for 50 trials per task. LIBERO-Plus stresses robustness by applying perturbations to LIBERO tasks along seven factors: camera, robot initial state, language, lighting, background, sensor noise, and layout. We use the protocol with trials on 500 variants per suite, covering all perturbations.

**Real-World Setup.** Our real-world platform consists of two AgileX Piper arms, one overhead Intel RealSense D435 camera, and two wrist-mounted D435 cameras. The policy takes all three views, arm joint angles and gripper states as input and predicts joint-action chunks for both arms. We evaluate six seen tasks and three held-out novel tasks spanning household and scientific manipulation. Each seen task provides 60 real trajectories collected with variations in object placement, lighting, and distractors. Each held-out task provides no real trajectory and only two true initial observations, one on clean environment with only task-specific objects and one on randomized setup with distractors. We then use GPT-Image-2 to generate 60 diverse initial-state images from the real observations for conditioning the video teacher. Because the generated images lack multi-view alignment, pseudo supervision for held-out tasks is generated from the overhead view only; the complete policy retains multi-view input when real multi-view observations are available. The real-world platform and evaluation setup are detailed in Appendix C.

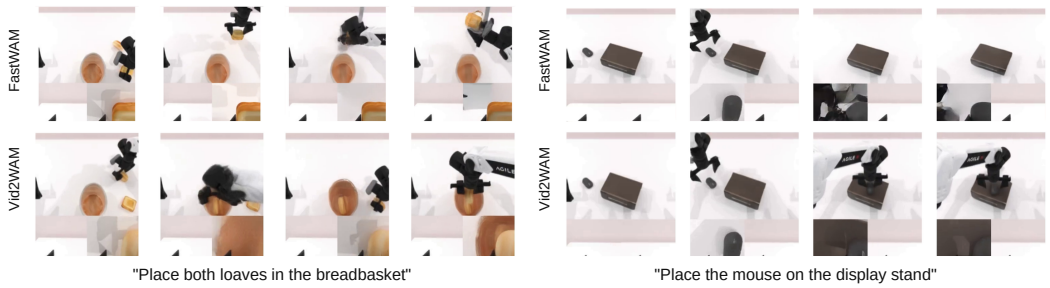


Figure 3: Qualitative future-video predictions for novel tasks. For each task, Fast-WAM (top) and Vid2WAM (bottom) receive the same initial observation and language instruction; frames progress from left to right. Fast-WAM often fails to predict coherent task progression, whereas Vid2WAM predicts future motions that more consistently reach the instructed outcome.

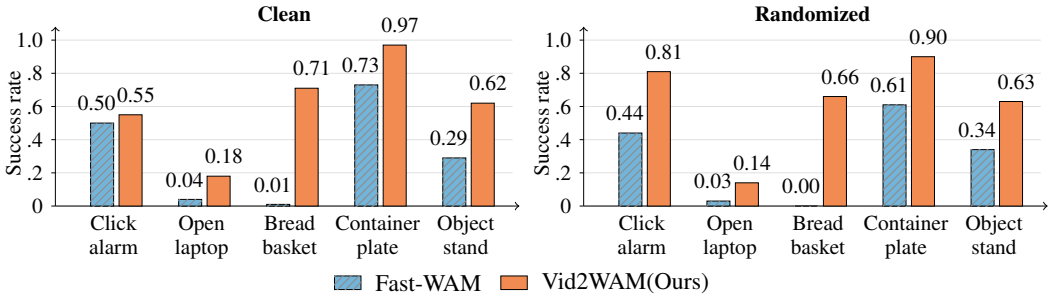


Figure 4: Success rates on representative novel RoboTwin tasks. Vid2WAM consistently outperforms Fast-WAM under clean and randomized evaluation.

**Evaluation Regimes.** Standard manipulation benchmarks typically train policies with expert trajectories from every evaluation task and test them under limited variations in object configurations. Such protocols primarily measure in-distribution execution and may conflate task generalization with interpolation around task-specific demonstrations. To more directly evaluate data efficiency and novel-task generalization, we consider two complementary regimes. In the **low-data regime**, each task in every suite or setting provides only 10 expert trajectories, with Vid2WAM receiving an equal number of offline teacher-generated trajectories. In the **novel-task regime**, we withhold all expert trajectories from selected target tasks: baselines are directly transferred from the remaining tasks, whereas Vid2WAM uses only target-task instructions and initial observations to generate pseudo supervision. Neither the video teacher nor the IDM is trained on target-task trajectories. These regimes respectively evaluate policy learning under scarce real demonstrations and adaptation to novel tasks without target-task expert supervision.

**Baselines.** We compare with $\pi_{0.5}$ (Physical Intelligence et al., 2025), Motus (Bi et al., 2025), and Fast-WAM (Yuan et al., 2026). $\pi_{0.5}$ and Motus assess whether broad robot-data pretraining alone enables direct transfer to the evaluated tasks. Fast-WAM serves as the closest architectural baseline, isolating the effect of our teacher-derived future supervision and source-aware adaptation. Under each evaluation regime, all methods receive the same downstream real action-labeled demonstrations. In the novel-task regime, these baselines use the originally collected seen-task demonstrations for downstream training; target-task observations and instructions enter only at evaluation.

**Implementation Details.** We initialize our video teacher from the LVP checkpoint (Chen et al., 2025a), a Wan2.1-14B model (Wan et al., 2025) pretrained on robot manipulation data, and further fine-tune it on our training set. Generated frames are encoded by the student VAE so that the teacher-generated targets and student predictions lie in the same latent space. The IDM uses a ResNet-50 backbone (He et al., 2016) with action and proprioception prediction heads. To reduce the influence of long-horizon generation errors, pseudo supervision is restricted to the first eight seconds of each generated rollout in simulation. For real-world novel-task pseudo data, the spatial video loss is computed only over the valid overhead-view region, while pseudo-action supervision remains un-

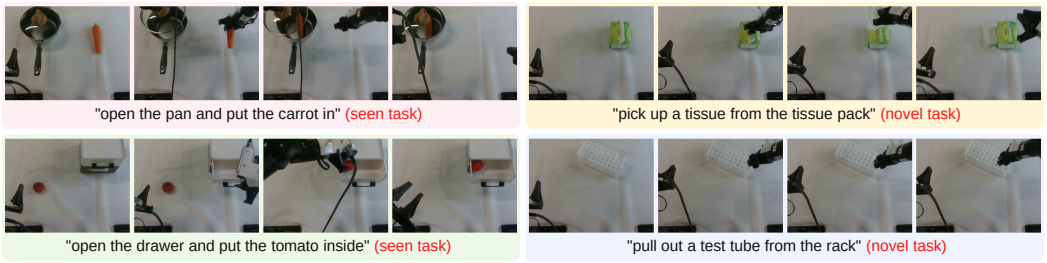


Figure 5: Vid2WAM real-world executions for representative tasks. Frames progress left to right.

changed. Training schedules and multi-view input layouts are described in Appendix A, and spatial and temporal masking details are provided in Appendix E.

### 4.2 IDM Analysis

Our framework relies on IDM pseudo-actions to provide offline supervision for the student policy. We evaluate IDM performance on validation sets of simulation and real-world data, and further analyze its reliability on teacher-generated rollouts under LIBERO low-data regime.

As shown in Table S4, our IDM achieves high validation accuracies: 85.8% on LIBERO/LIBERO-Plus, 81.9% on RoboTwin, and 98.1% on real-world data. Fig. S2 shows that the IDM can generate plausible actions, but will also produce non-executable pseudo actions when the teacher hallucinates or generates visual anomaly. These cases motivate source-aware adaptation and the eight-second simulation cutoff. The detailed setup and full analysis appear in Appendix A.2.

### 4.3 Simulation Results

| Method | Low-data | | Novel regime | | Novel subset | |
|---|---|---|---|---|---|---|
| | Clean | Rand. | Clean | Rand. | Clean | Rand. |
| $\pi_{0.5}$ | 48.7 | 42.8 | 67.9 | 66.5 | 46.5 | 46.3 |
| Motus | 49.9 | 43.6 | 75.1 | 76.3 | 48.9 | 51.5 |
| Fast-WAM | 52.1 | 39.1 | 75.7 | 74.7 | 45.0 | 42.8 |
| Ours | **55.4** | **45.4** | **78.3** | **78.5** | **54.7** | **55.3** |

Table 1: Overall success rates (%) on RoboTwin 2.0. Novel regime averages over 35 seen and 15 unseen tasks, while Novel subset reports the average over 15 unseen tasks only.

**RoboTwin 2.0.** Table 1 reports results under clean and randomized conditions on RoboTwin 2.0, including averages over all 50 tasks and over the 15 novel tasks in the novel regime. The results reveal that both pretrained VLA policies and WAMs struggle to transfer to novel tasks, with all baselines dropping substantially on the 15-task novel subset. This degradation is particularly pronounced for Fast-WAM, whose success rate decreases from 75.7% to 45.0% under clean evaluation and from 74.7% to 42.8% under randomization. By distilling the pretrained video teacher's predictive prior, Vid2WAM improves novel-subset success to 54.7% and 55.3%, outperforming Fast-WAM by 9.7 and 12.5 points. Vid2WAM also achieves the best performance in the low-data regime, demonstrating consistent benefits for both novel-task transfer and learning from limited real demonstrations.

Figure 4 presents representative novel tasks for which Vid2WAM clearly improves over Fast-WAM. The teacher-generated future trajectories provide useful information about task completion and guide the policy toward the intended outcome. Figure 3 further compares the predicted future videos produced from identical initial observations and language instructions. Fast-WAM often stops after intermediate interactions or loses visual coherence, while Vid2WAM continues toward the requested placement. The difference is most visible after the policy has completed an opening or grasping step, suggesting that the distilled future information supports longer-horizon task progression.

**LIBERO and LIBERO-Plus.** Table 3 reports evaluation results on LIBERO under both the low-data and novel regimes. In the low-data regime, Vid2WAM achieves an average success rate of

| Regime | Model | Suite | | | | Perturbation | | | | | | | Overall |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Spatial | Object | Goal | Long | Cam. | Robot | Lang. | Light | BG | Noise | Layout | |
| Low-data | Motus | 34.2 | 35.4 | **29.4** | 12.4 | 2.5 | 29.5 | 53.3 | 38.1 | 31.0 | 6.0 | 38.2 | 27.9 |
| | Fast-WAM | 39.8 | 54.0 | 14.6 | **25.8** | **8.5** | 22.5 | 49.5 | 57.8 | 32.1 | 23.3 | 45.6 | 33.6 |
| | Ours | **50.0** | **55.0** | 27.8 | 23.0 | 8.2 | **31.2** | **57.9** | **71.6** | **36.8** | **23.7** | **48.8** | **39.0** |
| Novel | Motus | 52.2 | 44.0 | 31.8 | **39.6** | 4.7 | **51.6** | 66.3 | 63.8 | **46.2** | 18.7 | 47.7 | 41.9 |
| | Fast-WAM | 48.0 | 59.6 | 37.0 | 38.6 | 24.1 | 42.5 | 64.2 | 67.2 | 39.0 | 34.6 | **52.7** | 45.8 |
| | Ours | **52.4** | **60.2** | **41.4** | 38.6 | **28.5** | 42.8 | **67.7** | **67.5** | 39.7 | **42.4** | 51.6 | **48.1** |

Table 2: Average task success rates (%) on LIBERO-Plus under the low-data and novel regimes, evaluated across four suites and seven perturbation types.

| Method/Regime | Spatial | Object | Goal | Long | **Avg.** |
|---|---|---|---|---|---|
| **Low-data** | | | | | |
| $\pi_{0.5}$ | 91.2 | 95.6 | 84.4 | 76.4 | 86.9 |
| Motus | **93.0** | 94.4 | 85.0 | 78.4 | 87.7 |
| Fast-WAM | 89.6 | 97.8 | 82.8 | 78.8 | 87.3 |
| Ours | **93.0** | **98.6** | **87.6** | **79.4** | **89.7** |
| **Novel** | | | | | |
| $\pi_{0.5}$ | 77.2 | **80.2** | 78.4 | **75.8** | 77.9 |
| Motus | 77.4 | 72.2 | 68.8 | 69.8 | 72.1 |
| Fast-WAM | 76.8 | 79.0 | 77.8 | 73.0 | 76.7 |
| Ours | **79.4** | 79.4 | **79.2** | 75.2 | **78.3** |

Table 3: Average task success rates (%) across LIBERO suites under the low-data and novel regimes.

$89.7\%$, obtaining the best or tied-best result on every suite. In the novel regime, it also achieves the highest average success rate among the four evaluated methods. These results show that Vid2WAM can improve performance both when expert demonstrations are limited and when no real trajectories are available for the target tasks.

Table 2 further evaluates robustness under the perturbations introduced by LIBERO-Plus. Vid2WAM achieves the highest overall performance on both regimes, with particularly clear gains under robot-state, language, and lighting perturbations on the low-data regime. We further evaluate whether the novel regime results depend on the selected target tasks. We consider three LIBERO or LIBERO-Plus splits that withhold novel task IDs $0/1$, $3/6$, or $8/9$ from each suite, where the $8/9$ split corresponds to the settings in Table 2 and Table 3. Tables S11 and S12 show that Vid2WAM achieves a higher average success rate than Fast-WAM on all splits in both benchmarks. Tables S13 and S14 isolate the withheld tasks in LIBERO and LIBERO-Plus; Vid2WAM again outperforms Fast-WAM on every split, showing that the aggregate improvements result from better adaptation to novel tasks rather than selection of specific tasks.

### 4.4 Real-World Experiments

We evaluate real-world performance on nine contact-rich manipulation tasks, covering both seen and novel settings. Seen tasks include *Click Bell*, *Item Handover*, *Carry Basket*, *Open Pan Place*, *Open Drawer Place* and *Insert Test Tube*. They use real trajectories during training. Novel tasks include *Pick Test Tube*, *Take Tissue* and *Tomato Basket*. They are designed to test behavior- or object-level novelty, requiring novel manipulation actions or objects that are absent from real expert trajectories. These tasks use initial observations (true and synthetic) and teacher-generated supervision rather than real demonstrations. Detailed real-world task descriptions are given in Appendix C.1.

Table 4 reports success rates on 20 real-robot trials for all models on each task. Among all tasks, Vid2WAM achieves the best or competitive performance across the four compared methods. Figure 5 further visualizes representative real-world executions of Vid2WAM on seen and novel tasks. As shown in the novel task, the robot can extract a deformable tissue or pull out a transparent test tube despite having no real action-labeled trajectories.

| Regime | Task | $\pi_{0.5}$ | Motus | Fast-WAM | Ours |
|---|---|---|---|---|---|
| Seen | Click Bell | 30% | **70%** | 20% | 65% |
| | Item Handover | 25% | **60%** | 25% | **60%** |
| | Carry Basket | 45% | 45% | 20% | **50%** |
| | Open Pan Place | 5% | **40%** | 10% | **40%** |
| | Open Drawer Place | 60% | 55% | 30% | **65%** |
| | Insert Test Tube | 15% | 0% | 10% | **25%** |
| Novel | Pick Test Tube | 0% | 10% | 5% | **30%** |
| | Take Tissue | 0% | 0% | 5% | **15%** |
| | Tomato Basket | 0% | 20% | 25% | **30%** |

Table 4: Real-world bimanual task success rates.

### 4.5 Ablation Studies

Table 5 evaluates on different variants of our Vid2WAM under LIBERO low-data regime. *Dual Action DiT* variant uses separate action DiTs for ground-truth and teacher-generated samples while sharing the video DiT; *No Adapter* mixes both sources in shared action and video DiTs. *Pseudo Action Only* uses DreamGen-style video-generation-plus-IDM synthetic action supervision (Jang et al., 2025); it retains real-video co-training and removes only the pseudo-video loss. *Future Latent Only* removes pseudo-action supervision while retaining future-latent training.

Results show that among the dual-channel variants, independent action DiTs outperform fully shared training, while the full adapter design (Ours) performs best on average. Removing only pseudo-video supervision (*Pseudo Action Only*) lowers the average from $89.7$ to $86.4$, isolating the generated-future loss in this low-data ablation. *Future Latent Only* reaches $88.2$, suggesting that pseudo-actions also provide complementary supervision.

We additionally evaluate an online *Teacher Policy* that composes the video teacher and IDM at inference time without student distillation. It underperforms Fast-WAM and all distilled variants, showing that the gains come from offline future supervision combined with ground-truth action anchoring rather than a stronger online teacher controller. This result is also consistent with generation and IDM errors accumulating under task execution. Together with the *No Adapter* ablation, it suggests that source-dependent residual corrections can mitigate the influence of noisy pseudo-actions.

| Method | Spatial | Object | Goal | Long | **Avg.** |
|---|---|---|---|---|---|
| Fast-WAM | 89.6 | 97.8 | 82.8 | 78.8 | 87.3 |
| Ours | 93.0 | **98.6** | **87.6** | 79.4 | **89.7** |
| Dual Action DiT | 92.2 | 97.8 | 85.8 | **80.6** | 89.1 |
| No Adapter | **93.8** | 97.6 | 84.2 | 78.4 | 88.5 |
| Pseudo Action Only | 90.4 | 96.0 | 85.0 | 74.0 | 86.4 |
| Future Latent Only | 92.8 | 98.2 | 86.4 | 75.2 | 88.2 |
| Teacher Policy | 76.0 | 85.2 | 65.6 | 58.2 | 71.3 |

Table 5: Ablation study on LIBERO benchmark (low-data regime).

| Method | Params | Mean | Std. | Min | Max |
|---|---|---|---|---|---|
| Fast-WAM | 6B | 209.1 | 4.7 | 204.8 | 220.2 |
| Ours | 6B | 212.5 | 11.7 | 199.0 | 227.8 |
| Teacher Policy | 14B | 4894.5 | 106.6 | 4785.8 | 4998.9 |
| $\pi_{0.5}$ | 3.3B | 76.7 | 0.4 | 76.2 | 77.6 |
| Motus | 8B | 2414.9 | 22.8 | 2380.0 | 2456.3 |

Table 6: Inference latency (ms) on a single RTX 4090 GPU.

### 4.6 Inference Latency Analysis

Table 6 reports chunk-level inference latency on a single RTX 4090 GPU, excluding cold-start costs. Results show that our Vid2WAM approximately matches Fast-WAM in latency, and is substantially faster than Motus and the Teacher Policy, though slower than $\pi_{0.5}$.

## 5 Conclusion

We present Vid2WAM, a framework for transferring the predictive diffusion priors of a large video model into a compact world-action policy. By using teacher-generated futures and IDM-inferred actions as offline supervision, Vid2WAM enables the student to acquire behavioral knowledge beyond the available expert trajectories, while source-aware adaptation limits the impact of noisy synthetic action labels. Across simulation and real-world manipulation, Vid2WAM improves data efficiency and generalization to novel tasks without introducing teacher inference at deployment. Our findings support a broader paradigm in which large pretrained models serve as offline knowledge sources for efficient robot policies. Video generators, VLMs, and other multimodal foundation models could supervise compact policies through predicted subgoals, future representations, or outcomes, offering a scalable path for transferring their predictive capabilities to efficient robot policies.

## AI USE STATEMENT

Generative AI tools were used to aid and polish the writing of this manuscript. All AI-assisted edits were reviewed and verified by the authors, who take responsibility for the final content of this work.

## REFERENCES

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, Hongyan Zhao, Hanyu Liu, Zhizhong Su, Lei Ma, Hang Su, and Jun Zhu. Motus: A unified latent action world model, 2025. URL https://arxiv.org/abs/2512.13030.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. $\pi_0$: A vision-language-action flow model for general robot control, 2024. URL https://arxiv.org/abs/2410.24164.

Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, Hanbo Zhang, and Minzhao Zhu. GR-2: A generative video-language-action model with web-scale knowledge for robot manipulation, 2024. URL https://arxiv.org/abs/2410.06158.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024.

Boyuan Chen, Tianyuan Zhang, Haoran Geng, Caiyi Zhang, Peihao Li, Kiwhan Song, William T Freeman, Jitendra Malik, Pieter Abbeel, Russ Tedrake, et al. Large video planner enables generalizable robot control. *arXiv preprint arXiv:2512.15840*, 2025a.

Danze Chen, Yanzhe Chen, Qiming Huang, Zhijun Cao, Chen Gao, and Mike Zheng Shou. Supervise what survives: Geometry-guided VLA adaptation from synthetic robot videos, 2026. URL https://arxiv.org/abs/2606.24448.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, Weiliang Deng, Yubin Guo, Tian Nian, Xuanbing Xie, Qiangyu Chen, Kailun Su, Tianling Xu, Guodong Liu, Mengkang Hu, Huan-ang Gao, Kaixuan Wang, Zhixuan Liang, Yusen Qin, Xiaokang Yang, Ping Luo, and Yao Mu. RoboTwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation, 2025b. URL https://arxiv.org/abs/2506.18088.

Zoey Chen, Sho Kiami, Abhishek Gupta, and Vikash Kumar. GenAug: Retargeting behaviors to unseen situations via generative augmentation, 2023. URL https://arxiv.org/abs/2302.06671.

Jeremy A. Collins, Loránd Cheng, Kunal Aneja, Albert Wilcox, Benjamin Joffe, and Animesh Garg. AMPLIFY: Actionless motion priors for robot learning from videos, 2025. URL https://arxiv.org/abs/2506.14198.

Yilun Du, Mengjiao Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Joshua B. Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 9156–9172, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1d5b9233ad716a43be5c0d3023cb82d0-Abstract-Conference.html.

Yilun Du, Mengjiao Yang, Pete Florence, Fei Xia, Ayzaan Wahid, Brian Ichter, Pierre Sermanet, Tianhe Yu, Pieter Abbeel, Joshua B. Tenenbaum, Leslie Pack Kaelbling, Andy Zeng, and Jonathan Tompson. Video language planning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=9pKtcJcMP3.

Pengcheng Fang, Hongli Chen, and Xiaohao Cai. Privileged foresight distillation: Zero-cost future correction for world action models, 2026. URL https://arxiv.org/abs/2604.25859.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. Libero-plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025.

Yao Feng, Hengkai Tan, Xinyi Mao, Chendong Xiang, Guodong Liu, Shuhe Huang, Hang Su, and Jun Zhu. Vidar: Embodied video diffusion model for generalist manipulation, 2025. URL https://arxiv.org/abs/2507.12898.

Huy Ha, Pete Florence, and Shuran Song. Scaling up and distilling down: Language-guided robot skill acquisition. In *Proceedings of the 2023 Conference on Robot Learning*, 2023. URL https://arxiv.org/abs/2307.14535.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90. URL https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 8003–8017, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.507. URL https://aclanthology.org/2023.findings-acl.507/.

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*, volume 267 of *Proceedings of Machine Learning Research*, pp. 24328–24346. PMLR, 2025. URL https://proceedings.mlr.press/v267/hu25g.html.

Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, Yen-Chen Lin, Loïc 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*, volume 305 of *Proceedings of Machine Learning Research*, pp. 5170–5194. PMLR, 2025. URL https://proceedings.mlr.press/v305/jang25a.html.

Yueru Jia, Jiaming Liu, Shengbang Liu, Rui Zhou, Wanhe Yu, Yuyang Yan, Xiaowei Chi, Yandong Guo, Boxin Shi, and Shanghang Zhang. Video2Act: A dual-system video diffusion policy with robotic spatio-motional modeling, 2025. URL https://arxiv.org/abs/2512.03044.

Yuhua Jiang, Yijun Guo, Hongbing Yang, Guojun Lei, Nuo Chen, Yinuo Zhang, Shaoqiang Yan, Bo Lin, Feifei Gao, and Biqing Qi. CKT-WAM: Parameter-efficient context knowledge transfer between world action models, 2026. URL https://arxiv.org/abs/2605.06247.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P. Foster, Pannag R. 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*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713. PMLR, 2025. URL https://proceedings.mlr.press/v270/kim25c.html.

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. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=wPEIStHxYH.

Seungku Kim, Suhyeok Jang, Byungjun Yoon, Dongyoung Kim, John Won, and Jinwoo Shin. RoboCurate: Harnessing diversity with action-verified neural trajectory for robot learning, 2026b. URL https://arxiv.org/abs/2602.18742.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16698–16708, October 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Li_DreamTeacher_Pretraining_Image_Backbones_with_Deep_Generative_Models_ICCV_2023_paper.html.

Jiajun Li, Tiecheng Guo, Yifan Ye, Rongyu Zhang, Xiaowei Chi, Qianpu Sun, Ying Li, Yunfan Lou, Yan Huang, Zhihe Lu, Meng Guo, and Shanghang Zhang. Efficient-WAM: A 1b-parameter world-action model with low-cost future imagination, 2026. URL https://arxiv.org/abs/2606.10040.

Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model, 2025. URL https://arxiv.org/abs/2503.00200.

Junbang Liang, Ruoshi Liu, Ege Ozguroglu, Sruthi Sudhakar, Achal Dave, Pavel Tokmakov, Shuran Song, and Carl Vondrick. Dreamitate: Real-world visuomotor policy learning via video generation. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 3943–3960. PMLR, 2025. URL https://proceedings.mlr.press/v270/liang25b.html.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The eleventh international conference on learning representations*, 2023.

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*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/8c3c666820ea055a77726d66fc7d447f-Abstract-Datasets_and_Benchmarks.html.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. Orca: Progressive learning from complex explanation traces of GPT-4, 2023. URL https://arxiv.org/abs/2306.02707.

Jonas Pai, Liam Achenbach, Victoriano Montesinos, Benedek Forrai, Oier Mees, and Elvis Nava. mimic-video: Video-action models for generalizable robot control beyond VLAs, 2025. URL https://arxiv.org/abs/2512.15692.

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

Jinghuan Shang, Karl Schmeckpeper, Brandon B. May, Maria Vittoria Minniti, Tarik Kelestemur, David Watkins, and Laura Herlant. Theia: Distilling diverse vision foundation models for robot learning. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=ylZHvlwUcI.

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.

Ruixiang Wang, Qingming Liu, Yueci Deng, Guiliang Liu, Zhen Liu, and Kui Jia. EVA: Aligning video world models with executable robot actions via inverse dynamics rewards, 2026. URL https://arxiv.org/abs/2603.17808.

Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=NxoFmGgWC9.

Huilin Xu, Jian Ding, Jiakun Xu, Ruixiang Wang, Jun Chen, Jinjie Mai, Yanwei Fu, Bernard Ghanem, Feng Xu, and Mohamed Elhoseiny. Diffusion-based imaginative coordination for bimanual manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11469–11479, 2025. doi: 10.1109/ICCV51701.2025.01067. URL https://openaccess.thecvf.com/content/ICCV2025/html/Xu_Diffusion-Based_Imaginative_Coordination_for_Bimanual_Manipulation_ICCV_2025_paper.html.

Haodong Yan, Zhide Zhong, Jiaguan Zhu, Junjie He, Weilin Yuan, Wenxuan Song, Xin Gong, Yingjie Cai, Guanyi Zhao, Xu Yan, Bingbing Liu, Ying-Cong Chen, and Haoang Li. S-VAM: Shortcut video-action model by self-distilling geometric and semantic foresight, 2026. URL https://arxiv.org/abs/2603.16195.

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 *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=VYOe2eBQeh.

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

Tianhe Yu, Ted Xiao, Austin Stone, Jonathan Tompson, Anthony Brohan, Su Wang, Jaspiar Singh, Clayton Tan, Dee M, Jodilyn Peralta, Brian Ichter, Karol Hausman, and Fei Xia. Scaling robot learning with semantically imagined experience. In *Robotics: Science and Systems XIX*, 2023. doi: 10.15607/RSS.2023.XIX.027. URL https://doi.org/10.15607/RSS.2023.XIX.027.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-WAM: Do world action models need test-time future imagination?, 2026. URL https://arxiv.org/abs/2603.16666.

Pengfei Zhou, Shengcong Chen, Di Chen, Jiaxu Wang, Rongjun Jin, Bingwen Zhu, Yike Pan, Songen Gu, Kuanning Wang, Shufeng Nan, Xingyu Qiu, Chenhao Qiu, Pu Yang, Yunuo Cai, Jianxiong Gao, Yifan Li, Yanwei Fu, Xiangyu Yue, Zhi Chen, and Jianlan Luo. $\tau_0$-WM: A unified video-action world model for robotic manipulation, 2026. URL https://arxiv.org/abs/2606.01027.

Siyuan Zhou, Yilun Du, Jiaben Chen, Yandong Li, Dit-Yan Yeung, and Chuang Gan. RoboDreamer: Learning compositional world models for robot imagination, 2024. URL https://arxiv.org/abs/2404.12377.

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, 2025. URL https://arxiv.org/abs/2504.02792.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pp. 2165–2183, 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

# A Implementation Details

For all models including video teacher, IDM and Vid2WAM, on LIBERO and LIBERO-Plus, the head and wrist camera views are resized to $224 \times 224$ and concatenated horizontally into a $224 \times 448$ frame. For RoboTwin 2.0, the top head camera is resized to $256 \times 320$, while the two wrist views are resized to $128 \times 160$, concatenated horizontally, and placed below to form a $384 \times 320$ canvas. For real-world experiments, the video teacher generates overhead-view rollouts at its native $480 \times 640$ resolution. For student training, these frames are resized to occupy the upper $224 \times 448$ region of a $448 \times 448$ composite, while the two unavailable $224 \times 224$ wrist-view slots in the lower half are black-padded. For real multi-view samples, the true wrist views occupy these lower slots.

## A.1 Video Teacher

**Video teacher training.** The teacher is an image-to-video diffusion model initialized from the Large Video Planner checkpoint (Chen et al., 2025a), which adopts the Wan2.1-I2V-14B architecture (Wan et al., 2025). We fine-tune the diffusion transformer while keeping the UMT5 text encoder, CLIP image encoder, and video VAE frozen. All video teachers are trained for around 500 epochs. Each training sample consists of 49 video frames, which are encoded into 16-channel latent representations using the frozen Wan VAE.

Given a clean video latent $z$, Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$, and $t \sim \mathcal{U}(0, 1)$, we apply the time shift

$$\tau = \frac{st}{1 + (s-1)t}, \qquad s = 3, \tag{S1}$$

and construct

$$z_\tau = (1 - \tau)z + \tau\epsilon. \tag{S2}$$

The teacher predicts the flow target $v^\star = \epsilon - z$ and is optimized with

$$\mathcal{L}_{\text{teacher}} = \mathbb{E}_{z,\epsilon,t}\left[\|v_\Theta(z_\tau, \tau, c) - (\epsilon - z)\|_2^2\right], \tag{S3}$$

where $c$ contains the language embedding and first-frame image condition. We additionally use diffusion forcing (Chen et al., 2024) during training. In particular, one or two latent history frames are sampled with probabilities $0.9$ and $0.1$, respectively, and the history is kept clean with probability $0.7$.

| Hyperparameter | Value |
|---|---|
| Initialization | Wan2.1-I2V-14B-480P with LVP 14B weights |
| Layers | 40 |
| Hidden dimension | 5120 |
| Feed-forward dimension | 13824 |
| Attention heads | 40 |
| Training frames | 49 |
| VAE latent channels | 16 |
| VAE temporal/spatial stride | $4 \times 8 \times 8$ |
| Continuous-time shift | 3.0 |
| Training-time scale | 1000 |
| Optimizer | AdamW |
| Learning rate | $8 \times 10^{-6}$ |
| AdamW betas | $(0.9, 0.95)$ |
| Weight decay | $5 \times 10^{-2}$ |
| Warm-up steps | 100 |
| Batch size | 4 |
| Gradient accumulation | 4 |
| Numerical precision | BF16 |
| Maximum text length | 512 |

Table S1: Shared hyperparameters used to fine-tune the video teacher.

**Teacher-generated rollout examples.** Figure S1 shows representative teacher rollouts across LIBERO, RoboTwin, and real-world settings. The examples cover both low-data and novel-task regimes, as well as clean and randomized real-world initial states.




Figure S1: Representative teacher-generated rollouts across LIBERO, RoboTwin, and real-world settings. Each row presents six temporally ordered keyframes conditioned on the displayed language instruction. The simulation examples include low-data and novel-task regimes, while the real-world examples cover clean and randomized initial states.

Across the six examples, the teacher predicts task-relevant object motion and multi-step manipulation progress while maintaining the scene context needed for the instruction. The novel-task and randomized examples further illustrate useful future-progress cues for held-out instructions and perturbed initial states.

| Domain | Resolution | FPS | Frames | Steps | Language CFG | History CFG |
|---|---|---|---|---|---|---|
| LIBERO | $224 \times 448$ | 15 | 49 | 40 | 3.0 | 1.0 |
| RoboTwin | $384 \times 320$ | 15 | 49 | 40 | 3.0 | 1.0 |
| Real-world | $480 \times 640$ | 10 | 49 | 40 | 3.0 | 2.0 |

Table S2: Domain-specific video-teacher rollout configurations used for constructing the pseudo-data buffers.

### A.2 IDM

**Inverse dynamics model.** The inverse dynamics model (IDM) maps a short video window to the action at its center frame. Given three observations separated by a domain-dependent stride $\delta$, it predicts

$$\widehat{a}_t = q_\psi \left( I_{t-\delta}, I_t, I_{t+\delta} \right). \tag{S4}$$

We use an ImageNet-pretrained ResNet-50 (He et al., 2016) whose first convolution is expanded from three to nine input channels to jointly process the three RGB frames. The pretrained convolutional filters are repeated across the temporal dimension and divided by three. The final ResNet feature map is passed through a learnable spatial-softmax layer and a lightweight MLP action head.

The action mean $\mu_a$ and standard deviation $\sigma_a$ are computed from the training split. The IDM is trained with a Smooth L1 loss in the normalized action space:

$$\mathcal{L}_{\text{IDM}} = \mathbb{E}\left[ \text{SmoothL1}\left( \frac{\widehat{a}_t - \mu_a}{\sigma_a}, \frac{a_t - \mu_a}{\sigma_a} \right) \right]. \tag{S5}$$

The prediction is converted back to the original action space during inference.

| Hyperparameter | Value |
|---|---|
| Visual backbone | ResNet-50 |
| Number of input frames | 3 |
| Input channels | 9 |
| Feature pooling | Spatial softmax |
| Hidden activation | GELU |
| Dropout | 0.1 |
| Training loss | Smooth L1 |
| Optimizer | AdamW |
| Learning rate | $3 \times 10^{-4}$ |
| AdamW betas | $(0.9, 0.999)$ |
| Weight decay | $10^{-2}$ |
| Training iterations | 200,000 |
| Batch size | 32 |
| Learning-rate schedule | Cosine |
| Warm-up iterations | 5,000 |
| Final learning-rate ratio | 0 |
| Numerical precision | BF16 |

Table S3: Shared training hyperparameters for the action IDM.

**IDM training and validation.** For LIBERO/LIBERO-Plus, RoboTwin and real-world, we train separate ResNet-50 IDMs for action and proprioceptive-state targets. Each IDM is trained for 200,000 steps using compatible action-labeled trajectories; withheld-task trajectories are excluded. The frozen IDMs then label pseudo-actions from teacher-generated videos.

For validation, we randomly partition the samples with size $N$. The validation-set size is

$$N_{\text{val}} = \min\left( \lfloor 0.05N \rfloor, 2000 \right), \tag{S6}$$

with at least one validation sample. Target normalization statistics are computed exclusively from the resulting training split. We evaluate the IDM every 5,000 optimization steps and retain the checkpoint with the lowest validation Smooth L1 loss. We further define validation accuracy as the proportion of samples for which all predicted dimensions fall within their corresponding absolute-error tolerances:

$$\text{ValAcc} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\left( \bigcap_{j=1}^{D} \left\{ |\hat{y}_{i,j} - y_{i,j}| < \tau_j \right\} \right), \tag{S7}$$

where $N$ is the number of validation samples, $D$ is the target dimensionality, $\hat{y}_{i,j}$ and $y_{i,j}$ denote the prediction and ground-truth value for dimension $j$ of sample $i$ respectively, and $\tau_j = 0.05$ for every dimension. Consequently, a sample is counted as correct only if every predicted dimension satisfies the tolerance. The rules are identical for action and state IDM training.

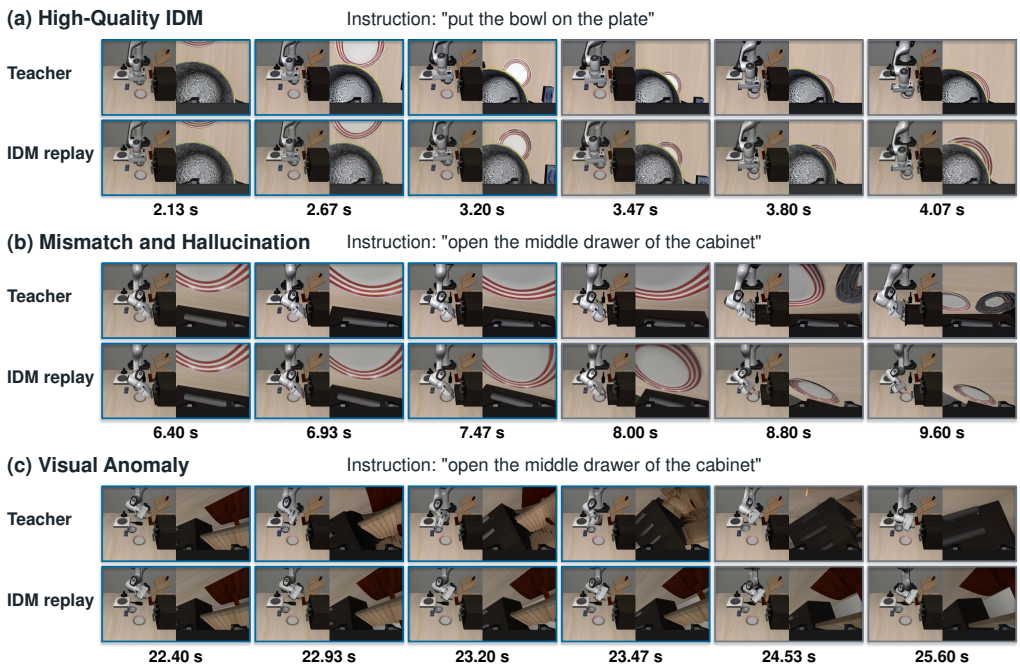


Figure S2: Qualitative teacher-video and IDM-action diagnostics. Each row shows a LIBERO task case with frames progress left to right, which compare the generated teacher video (upper strip) with a diagnostic open-loop simulator replay of IDM-derived actions (lower strip).

| Dataset | IDM Target | Val Loss ↓ | Val Acc ↑ | Val L1 ↓ |
|---|---|---|---|---|
| LIBERO / LIBERO-Plus | Action | 0.0054 | 85.8% | 0.0112 |
| LIBERO / LIBERO-Plus | State | 0.0000 | 100.0% | 0.0020 |
| RoboTwin | Action | 0.0009 | 81.9% | 0.0100 |
| RoboTwin | State | 0.0008 | 83.1% | 0.0095 |
| Real-world | Action | 0.0002 | 98.1% | 0.0044 |
| Real-world | State | 0.0002 | 98.6% | 0.0043 |

Table S4: Best validation performance of the trained IDM models.

All trained IDMs achieve validation accuracy above 80%, with both real-world heads exceeding 98% accuracy.

**Teacher-video and IDM-action analysis.** Figure S2 aligns generated teacher rollouts with simulator replays of the corresponding IDM-derived pseudo-actions under the LIBERO low-data regime. The three cases show a successful replay, an action-grounding mismatch with a slightly hallucinated teacher video, and a visual anomaly in a teacher rollout. They probe distinct ways in which generated teacher videos and pseudo-actions can differ from executable task progress.

The first put-bowl case begins at 2.13 s and succeeds in the diagnostic replay progress to 4.07 s. For the second drawer task, the case beginning at 6.40 s shows an action mismatch: the teacher depicts a plausible drawer interaction at first, but it is clear that from the frame at 8.80 s onward, the wrist view of the teacher video exhibits a visual hallucination with another plate. The replay then does not open the target middle drawer and instead moves the lower drawer. The third case begins at 22.40 s (which is after the 8-second cutoff). The teacher views exhibit localized smearing and stretched scene structure within the task progress, and the corresponding IDM pseudo-action replay shows some bias and does not correctly open the middle drawer.

The first case demonstrates that our IDM can generate precise actions according to the teacher video, while the second and third cases illustrate misalignment between the teacher video and the IDM pseudo-actions. The second case shows that even when the teacher video appears visually plausible at first with slight hallucinations, the IDM may produce noisy actions, leading to an imperfect replay. The third case highlights that visual anomalies in the teacher video can also result in incorrect pseudo-actions. The last two cases suggest the need for a source-aware action adapter to mitigate the impact of noisy IDM-derived pseudo-actions. Meanwhile, they also motivate the first-8 s heuristic discussed in Appendix E, which excludes harmful frames such as those in the 8.80 s onward in the second case and those in the third case.

### A.3 Data Budget and Model Exposure

Table S5 reports the data budget. For all experiment setups, the video teachers and IDMs are fine-tuned on its regime-specific dataset: the low-data regime uses corresponding expert demonstrations, while the novel regime excludes held-out-task trajectories.

| Benchmark/regime | Tasks | | Expert trajectories (seen tasks only) | Teacher rollouts | Held-out-task exposure | |
|---|---|---|---|---|---|---|
| | Seen | Novel | | | Teacher | IDM |
| LIBERO low-data | 40 | 0 | 400 (10 per task) | 400 | / | / |
| LIBERO novel | 32 | 8 | 1,366 | 312 | No | No |
| RoboTwin low-data | 50 | 0 | 1000 (10 clean + 10 randomized per task) | 1000 | / | / |
| RoboTwin novel | 35 | 15 | 19250 (50 clean +500 randomized per seen-task) | 8250 | No | No |
| Real-world | 6 | 3 | 360 (60 per seen-task) | 180 | No | No |

Table S5: All data budgets and exposures. "Teacher exposure" and "IDM exposure" indicate whether the corresponding model was trained on trajectories from the evaluated held-out target tasks; this is distinct from using a trajectory's first frame to seed a rollout.

**Pseudo-data construction.** After training, the video teacher and IDM are used only as offline data generators. Table S2 lists the rollout parameters used to construct each distillation buffer. Rollout generation takes only the source episode's first visual observation together with the language instruction, after which the future observations are synthesized by the video teacher and the pseudo actions are recovered by the IDM.

Because the Wan2.1 teacher VAE uses 16 latent channels while the Vid2WAM VAE uses 48, teacher rollouts are decoded to RGB and re-encoded by the frozen student VAE before storage.

The 49 frames in Table S2 denote one generation chunk, not a complete pseudo-rollout. Successive chunks are assembled into longer videos before the simulation cutoff and pseudo-data construction.

**Offline teacher-data compute and demonstration trade-off.** RoboTwin novel has the largest distillation buffer with 8,250 teacher rollouts. Generating them with the Wan2.1-14B teacher took approximately four hours on eight NVIDIA H100 GPUs, corresponding to 32 H100 GPU-hours. Since collecting expert trajectories requires successful robot executions and reliable action labels, some additional offline compute can be traded against that expert demonstration collection burden.

### A.4 VID2WAM TRAINING DETAILS

| Hyperparameter | Value |
|---|---|
| Video backbone initialization | Wan2.2-TI2V-5B |
| Number of transformer layers | 30 |
| Number of attention heads | 24 |
| Attention head dimension | 128 |
| Video hidden dimension | 3072 |
| Video FFN dimension | 14336 |
| Action hidden dimension $D$ | 1024 |
| Action FFN dimension | 4096 |
| Text embedding dimension | 4096 |
| Maximum text length | 128 |
| Video patch size | $(1, 2, 2)$ |
| Bottleneck dimension $D_r$ | 128 |
| Adapter activation | GELU |
| Adapter dropout | 0.0 |
| Adapter residual scale $\alpha$ | 1.0 |
| Up-projection initialization | Zero |
| Flow-matching training steps | 1000 |
| Video noise-schedule shift | 5.0 |
| Action noise-schedule shift | 5.0 |
| Balance coefficient $\beta$ | 1.0 |
| $\lambda_{\text{real}}$ | 1.0 |
| $\lambda_{\text{pseudo}}$ | 0.25 |
| Global batch per domain | 128 |
| Gradient accumulation | 1 |
| Learning rate | LIBERO/RoboTwin: $1 \times 10^{-4}$; real-world: $5 \times 10^{-5}$ |
| LR schedule | Cosine |
| Warmup ratio | 5% |
| Minimum LR ratio | 0.01 |
| Optimizer | AdamW |
| AdamW $\beta_1, \beta_2$ | $(0.9, 0.95)$ |
| Weight decay | 0.01 |
| Maximum gradient norm | 1.0 |
| Numerical precision | BF16 |

Table S6: Model and action-adapter hyperparameters in Vid2WAM.

Each iteration processes one real batch and one pseudo batch before a single parameter update, so the global batch size of 128 per domain denotes 128 real and 128 pseudo samples per update. We train Vid2WAM for 10 epochs on LIBERO/LIBERO-Plus and real-world data, and for 5 epochs on RoboTwin 2.0.

**Parameter updates by data source.** Table S7 specifies which student modules receive gradients from each data source under the default objective in Eq. 7.

| Batch | Real adapters | Pseudo adapters | Shared Action DiT | Video backbone |
|---|---|---|---|---|
| Real | Yes | – | Yes | Yes |
| Pseudo | – | Yes | Yes | Yes |

Table S7: Student parameter updates from real and pseudo batches. "Adapters" includes the source-specific input and output residual adapters; "Yes" denotes a gradient contribution.

The two batches contribute gradients to the shared Action DiT and video backbone, while each updates only its own adapters. Thus the pseudo adapters provide a separate path for source-specific corrections without directly updating the real adapters; pseudo-batch gradients still reach the shared modules. The real and pseudo video losses both train the video backbone.

**Details on ablation variants.** Details on the Vid2WAM-based ablation variants discussed in the main paper are shown in Table S8. All Vid2WAM-based variants share the same teacher and IDM

models and corresponding distillation buffers. Training hyperparameters also remain the same. Specifically, *Pseudo Action Only* optimizes $\lambda_{\text{real}}\mathcal{L}_{\text{real}} + \lambda_{\text{pseudo}}\mathcal{L}_{\text{FM}}(\hat{a}, \text{pseudo})$: the real-data action and video losses are unchanged, and only the pseudo-video term is removed.

For the Teacher Policy described in the main paper, we also use the same teacher and IDM models as Vid2WAM across all regimes and benchmarks. This policy uses the teacher to generate task rollouts and the IDM to infer action trajectories for direct execution.

| Variant | Structural or objective change | Training-time parameter change | Inference path |
|---|---|---|---|
| Vid2WAM (default) | Shared video expert and shared ActionDiT; separate input/output adapters for both domains. | around $+1$M | Shared ActionDiT with the real-domain adapters. |
| Dual Action DiT | Shared video expert, but independent real and pseudo Action DiTs. | around $+1$B | Real ActionDiT only |
| No Adapter | One shared video expert and one shared Action DiT, with no domain-specific adapter modules. Both domains update the same path. | 0 | The shared Action DiT path. |
| Pseudo-Action Only | Remove only the pseudo-video loss; retain real-video co-training and pseudo-action loss. | Same as Vid2WAM | Same as Vid2WAM |
| Future-Latent Only | Discard the pseudo-action flow-matching loss. | Same as Vid2WAM | Same as Vid2WAM |

Table S8: Exact implementation of the distillation ablations.

# B Sensitivity to Novel-Task Selection

The split reported in the main paper withholds task IDs $8/9$ from every LIBERO and LIBERO-Plus suite; we additionally evaluate $0/1$ and $3/6$. Tables S11 and S12 report full-benchmark performance, whereas Tables S13 and S14 evaluate only the withheld tasks.

## B.1 Task-ID Mapping

Task IDs are suite-local, and LIBERO-Plus uses the same mapping as LIBERO. Tables S9 and S10 list the task-ID mappings for all four suites.

| ID | LIBERO-Spatial instruction | ID | LIBERO-Object instruction |
|---|---|---|---|
| 0 | pick up the black bowl between the plate and the ramekin and place it on the plate | 0 | pick up the alphabet soup and place it in the basket |
| 1 | pick up the black bowl from table center and place it on the plate | 1 | pick up the bbq sauce and place it in the basket |
| 2 | pick up the black bowl in the top drawer of the wooden cabinet and place it on the plate | 2 | pick up the butter and place it in the basket |
| 3 | pick up the black bowl next to the cookie box and place it on the plate | 3 | pick up the chocolate pudding and place it in the basket |
| 4 | pick up the black bowl next to the plate and place it on the plate | 4 | pick up the cream cheese and place it in the basket |
| 5 | pick up the black bowl next to the ramekin and place it on the plate | 5 | pick up the ketchup and place it in the basket |
| 6 | pick up the black bowl on the cookie box and place it on the plate | 6 | pick up the milk and place it in the basket |
| 7 | pick up the black bowl on the ramekin and place it on the plate | 7 | pick up the orange juice and place it in the basket |
| 8 | pick up the black bowl on the stove and place it on the plate | 8 | pick up the salad dressing and place it in the basket |
| 9 | pick up the black bowl on the wooden cabinet and place it on the plate | 9 | pick up the tomato sauce and place it in the basket |

Table S9: Suite-local LIBERO-Spatial and Object task-ID mappings, shared by LIBERO and LIBERO-Plus.

| ID | LIBERO-Goal instruction | ID | LIBERO-Long instruction |
|---|---|---|---|
| 0 | open the middle drawer of the cabinet | 0 | turn on the stove and put the moka pot on it |
| 1 | open the top drawer and put the bowl inside | 1 | put the black bowl in the bottom drawer of the cabinet and close it |
| 2 | push the plate to the front of the stove | 2 | put the yellow and white mug in the microwave and close it |
| 3 | put the bowl on the plate | 3 | put both moka pots on the stove |
| 4 | put the bowl on the stove | 4 | put both the alphabet soup and the cream cheese box in the basket |
| 5 | put the bowl on top of the cabinet | 5 | put both the alphabet soup and the tomato sauce in the basket |
| 6 | put the cream cheese in the bowl | 6 | put both the cream cheese box and the butter in the basket |
| 7 | put the wine bottle on the rack | 7 | put the white mug on the left plate and put the yellow and white mug on the right plate |
| 8 | put the wine bottle on top of the cabinet | 8 | put the white mug on the plate and put the chocolate pudding to the right of the plate |
| 9 | turn on the stove | 9 | pick up the book and place it in the back compartment of the caddy |

Table S10: Suite-local LIBERO-Goal and Long task-ID mappings, shared by LIBERO and LIBERO-Plus.

A split withholds the corresponding tasks with the specified IDs within each suite, so the same split definition applies consistently across all four suites.

### B.2 Results Across Task Splits

| Novel IDs | Method | Spatial | Object | Goal | Long | **Avg.** |
|---|---|---|---|---|---|---|
| $0/1$ | Fast-WAM | 82.4 | 79.8 | **76.4** | **75.6** | 78.6 |
| | Vid2WAM | **94.2** | **92.6** | 76.2 | 71.8 | **83.7** |
| $3/6$ | Fast-WAM | 73.0 | 79.2 | **77.6** | **76.4** | 76.6 |
| | Vid2WAM | **75.6** | **84.4** | 76.2 | 74.2 | **77.6** |
| $8/9$ | Fast-WAM | 76.8 | 79.0 | 77.8 | 73.0 | 76.7 |
| | Vid2WAM | **79.4** | **79.4** | **79.2** | **75.2** | **78.3** |

Table S11: Sensitivity to novel-task selection on LIBERO. Best results within each split are bolded.

Vid2WAM likewise improves the average LIBERO success rate for all three task splits, supporting the same conclusion across both benchmark variants.

| Novel IDs | Model | Suite | | | | Perturbation | | | | | | | Overall |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Spatial | Object | Goal | Long | Cam. | Robot | Lang. | Light | BG | Noise | Layout | |
| $0/1$ | Fast-WAM | 58.4 | 61.2 | **39.4** | 37.4 | **28.2** | 36.1 | 60.0 | 70.5 | **54.5** | 39.6 | **58.7** | 49.1 |
| | Vid2WAM | **60.2** | **63.2** | 37.2 | **39.8** | 24.8 | **40.0** | **66.3** | **76.5** | 48.4 | **40.6** | **58.7** | **50.1** |
| $3/6$ | Fast-WAM | 40.6 | **56.6** | 30.2 | **36.4** | 13.2 | 37.5 | 52.3 | 64.2 | **43.0** | **32.9** | 48.4 | 41.0 |
| | Vid2WAM | **45.0** | 53.6 | **40.2** | 35.0 | **15.1** | **38.3** | **63.2** | **70.9** | 40.1 | 31.8 | **49.8** | **43.5** |
| $8/9$ | Fast-WAM | 48.0 | 59.6 | 37.0 | **38.6** | 24.1 | 42.5 | 64.2 | 67.2 | 39.0 | 34.6 | **52.7** | 45.8 |
| | Vid2WAM | **52.4** | **60.2** | **41.4** | **38.6** | **28.5** | **42.8** | **67.7** | **67.5** | **39.7** | **42.4** | 51.6 | **48.1** |

Table S12: Sensitivity to novel-task selection on LIBERO-Plus. Each split treats the listed two task IDs in every suite as novel; $8/9$ is the main novel setting reported in the main paper. Best results within each split are bolded.

Vid2WAM improves the average LIBERO-Plus success rate for all three splits, showing that its gain over Fast-WAM in the novel regime does not depend on the selected target tasks.

| Novel IDs | Method | Spatial | Object | Goal | Long | **Avg.** |
|---|---|---|---|---|---|---|
| 0/1 | Fast-WAM | 29.0 | 1.0 | 0.0 | 0.0 | 7.5 |
| | Vid2WAM | **90.0** | **63.0** | 0.0 | 0.0 | **38.3** |
| 3/6 | Fast-WAM | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| | Vid2WAM | **1.0** | **26.0** | **1.0** | 0.0 | **7.0** |
| 8/9 | Fast-WAM | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| | Vid2WAM | **2.0** | **2.0** | **4.0** | **1.0** | **2.3** |

Table S13: Success rates (%) evaluated only on the withheld LIBERO target tasks. Each split removes the listed two task IDs from every suite.

Vid2WAM improves average withheld-task success for all three LIBERO splits. The "zero to one" improvements on most suites indicate that our distillation framework can guide the task process correctly.

| Novel IDs | Method | Spatial | Object | Goal | Long | **Avg.** |
|---|---|---|---|---|---|---|
| 0/1 | Fast-WAM | 51.0 | 1.0 | 0.0 | 0.0 | 13.0 |
| | Vid2WAM | **69.0** | **34.0** | 0.0 | 0.0 | **25.8** |
| 3/6 | Fast-WAM | 0.0 | 0.0 | 1.0 | 0.0 | 0.3 |
| | Vid2WAM | **2.0** | **9.0** | **2.0** | 0.0 | **3.3** |
| 8/9 | Fast-WAM | 1.0 | 2.0 | 1.0 | 0.0 | 1.0 |
| | Vid2WAM | **2.0** | **12.0** | **5.0** | 0.0 | **4.8** |

Table S14: Success rates (%) evaluated only on the withheld LIBERO-Plus target tasks. Each split removes the listed two task IDs from every suite.

Vid2WAM also improves average withheld-target success for all three LIBERO-Plus splits. Low absolute success on several Goal and Long targets nevertheless shows that adaptation without target-task action trajectories remains difficult.

## C Real-World Experiment Details

Figure S3 illustrates the real-world platform used for data collection and evaluation. A camera-equipped dual-arm robot is operated through the paired teleoperation interface. The top-down camera, studio lighting, and reconfigurable tabletop provide controlled variation in viewpoint, illumination, texture, object layout, and task composition.




Figure S3: Real-world bimanual platform, including the teleoperation interface, camera-equipped dual-arm robot, top-down camera, studio lighting, and reconfigurable tabletop with task objects.

### C.1 Detailed Task Description

The evaluation includes six seen tasks with expert trajectories and three novel tasks trained only from true or synthetic initial observations, language instructions, and offline teacher supervision. The novel tasks introduce behaviors or object–action compositions absent from the labeled trajectories.

**Seen tasks.**

- **Click Bell:** the right gripper closes and presses a tabletop bell.
- **Item Handover:** the right arm lifts a pen upright, transfers it to the left arm, and the left arm drops it into a pen holder.
- **Carry Basket:** the left arm places a milk carton into a basket, after which the right arm lifts and relocates the basket by its handle.
- **Open Pan Place:** the left arm opens the pan lid, the right arm places a carrot inside, and the left arm closes the lid.
- **Open Drawer Place:** the right arm opens a handled drawer while the left arm places a tomato inside.
- **Insert Test Tube:** the arms hand over a transparent test tube before vertically inserting it into a rack.

**Novel tasks.**

- **Pick Test Tube:** the right arm grasps a test tube already inserted in the rack and pulls it out.
- **Take Tissue:** the right arm pinches and extracts a tissue from a soft tissue pack.
- **Tomato Basket:** the left arm picks up a tomato and place it into the basket.

### C.2 Initial-State Augmentation

As shown in Figure S4, for each novel task, GPT-Image-2 generates 30 synthetic images from a collected overhead observation in clean setting with only task-specific objects, and 30 images from an observation with different distractors. The generated images vary object placement and illumination without requiring additional real frames.

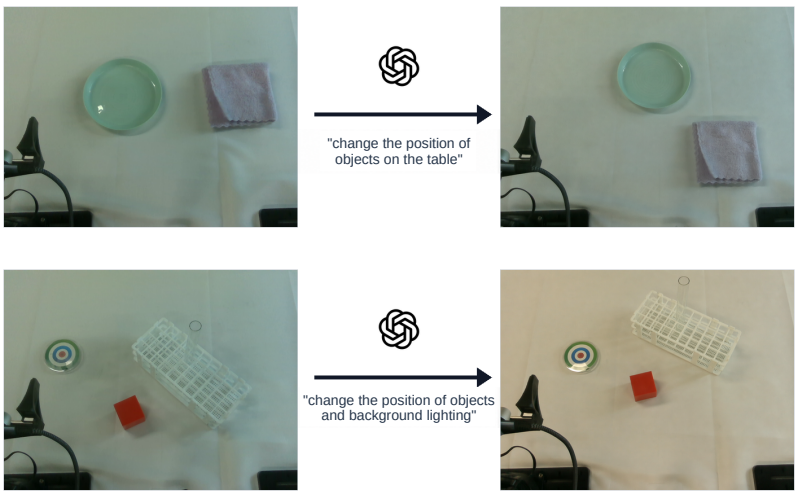


Figure S4: Real-world initial-state augmentation with GPT-Image-2. In each row, the collected overhead observation (left) is edited into a new conditioning image (right). The top example changes object placement, while the bottom example changes both object placement and illumination.

## D FULL ROBOTWIN TASK-LEVEL RESULTS

Tables S15 and S16 expand the main paper's aggregate RoboTwin results to all 50 tasks under clean and randomized evaluation settings.

| Task | Vid2WAM | | Fast-WAM | | $\pi_{0.5}$ | | Motus | |
|---|---|---|---|---|---|---|---|---|
| | Clean | Random | Clean | Random | Clean | Random | Clean | Random |
| adjust_bottle | 0.95 | **0.90** | **0.99** | 0.86 | 0.94 | 0.88 | 0.85 | 0.87 |
| beat_block_hammer | **0.84** | **0.66** | 0.69 | 0.52 | 0.75 | 0.64 | 0.48 | 0.29 |
| blocks_ranking_rgb | **0.42** | **0.16** | 0.39 | 0.13 | 0.13 | 0.15 | 0.27 | **0.16** |
| blocks_ranking_size | **0.34** | 0.11 | 0.16 | 0.08 | 0.10 | 0.09 | 0.19 | **0.17** |
| click_alarmclock | 0.82 | 0.69 | 0.78 | 0.51 | 0.79 | 0.84 | **0.93** | **0.94** |
| click_bell | 0.96 | 0.84 | 0.95 | 0.85 | 0.92 | **0.94** | **0.97** | 0.88 |
| dump_bin_bigbin | **0.90** | **0.77** | 0.88 | 0.74 | 0.69 | 0.47 | 0.62 | 0.51 |
| grab_roller | **0.99** | **0.91** | **0.99** | 0.89 | 0.92 | 0.74 | 0.98 | 0.73 |
| handover_block | **0.30** | **0.20** | 0.13 | 0.05 | 0.05 | 0.04 | 0.09 | 0.02 |
| handover_mic | **0.96** | **0.63** | 0.95 | 0.62 | 0.32 | 0.20 | 0.46 | 0.44 |
| lift_pot | 0.14 | 0.17 | 0.18 | 0.20 | **0.59** | **0.39** | 0.14 | 0.09 |
| move_can_pot | 0.49 | **0.45** | 0.57 | 0.33 | **0.60** | 0.41 | 0.08 | 0.25 |
| move_playingcard_away | 0.78 | 0.68 | 0.73 | 0.59 | 0.81 | 0.86 | **0.92** | **0.87** |
| move_stapler_pad | 0.03 | 0.02 | 0.01 | 0.00 | **0.18** | **0.11** | 0.08 | 0.04 |
| hanging_mug | 0.07 | **0.11** | 0.09 | 0.10 | 0.02 | 0.03 | **0.17** | 0.06 |
| open_laptop | 0.82 | 0.69 | **0.84** | 0.63 | 0.71 | **0.72** | 0.70 | **0.72** |
| open_microwave | 0.30 | 0.18 | 0.24 | 0.18 | 0.05 | 0.09 | **0.57** | **0.40** |
| pick_diverse_bottles | 0.25 | 0.11 | 0.20 | 0.03 | 0.44 | 0.23 | **0.45** | **0.33** |
| pick_dual_bottles | 0.29 | 0.15 | 0.25 | 0.12 | **0.59** | **0.51** | 0.58 | 0.38 |
| place_a2b_left | 0.30 | 0.26 | 0.24 | 0.09 | 0.49 | 0.42 | **0.77** | **0.60** |
| place_a2b_right | 0.34 | 0.33 | 0.21 | 0.14 | 0.45 | 0.37 | **0.77** | **0.57** |
| place_bread_basket | **0.70** | 0.56 | 0.60 | 0.51 | 0.61 | **0.58** | 0.54 | 0.54 |
| place_bread_skillet | 0.45 | 0.29 | 0.55 | 0.23 | 0.36 | 0.24 | **0.57** | **0.44** |
| place_can_basket | 0.68 | 0.52 | **0.79** | 0.51 | 0.62 | **0.54** | 0.29 | 0.32 |
| place_cans_plasticbox | **0.88** | **0.69** | **0.88** | **0.69** | 0.52 | 0.42 | 0.28 | 0.18 |
| place_container_plate | 0.96 | **0.92** | **0.98** | 0.87 | 0.95 | 0.82 | 0.91 | 0.79 |
| place_dual_shoes | 0.15 | 0.11 | **0.22** | **0.13** | 0.12 | 0.06 | 0.19 | 0.02 |
| place_empty_cup | 0.82 | 0.64 | 0.69 | 0.48 | 0.77 | 0.80 | **0.88** | **0.86** |
| place_fan | 0.18 | 0.13 | **0.30** | 0.13 | 0.08 | 0.14 | 0.18 | **0.15** |
| place_burger_fries | **0.84** | 0.77 | 0.79 | 0.69 | 0.81 | **0.82** | 0.57 | 0.58 |
| place_mouse_pad | 0.18 | 0.07 | 0.09 | 0.03 | **0.28** | 0.14 | 0.18 | **0.29** |
| place_object_basket | 0.70 | 0.59 | **0.72** | **0.69** | 0.31 | 0.17 | 0.59 | 0.61 |
| place_object_scale | 0.36 | 0.28 | 0.24 | 0.13 | 0.45 | **0.42** | **0.49** | 0.35 |
| place_object_stand | **0.88** | **0.80** | 0.81 | 0.52 | 0.75 | 0.61 | 0.60 | 0.56 |
| place_phone_stand | **0.60** | **0.60** | 0.51 | 0.36 | 0.47 | 0.47 | 0.38 | 0.20 |
| move_pillbottle_pad | 0.37 | 0.22 | 0.31 | 0.20 | **0.54** | **0.58** | 0.36 | 0.15 |
| place_shoe | 0.37 | 0.21 | 0.37 | 0.34 | 0.24 | 0.21 | **0.57** | **0.61** |
| press_stapler | 0.92 | 0.81 | 0.87 | 0.68 | 0.80 | 0.71 | **1.00** | **0.91** |
| put_bottles_dustbin | **0.13** | **0.15** | 0.08 | 0.04 | 0.03 | 0.01 | 0.07 | 0.00 |
| put_object_cabinet | **0.34** | **0.22** | 0.29 | 0.20 | 0.09 | 0.06 | 0.10 | 0.19 |
| rotate_qrcode | 0.53 | **0.33** | **0.54** | 0.22 | 0.32 | 0.21 | 0.24 | 0.22 |
| scan_object | 0.31 | 0.21 | 0.26 | 0.24 | 0.28 | 0.25 | **0.59** | **0.49** |
| shake_bottle | **1.00** | **1.00** | **1.00** | 0.98 | 0.96 | 0.97 | 0.94 | 0.94 |
| shake_bottle_horizontally | **1.00** | **1.00** | **1.00** | 0.97 | 0.97 | 0.97 | 0.93 | 0.91 |
| stack_blocks_three | **0.30** | **0.18** | 0.18 | 0.10 | 0.05 | 0.07 | 0.00 | 0.03 |
| stack_blocks_two | 0.66 | **0.54** | 0.68 | 0.43 | 0.60 | 0.42 | **0.70** | **0.54** |
| stack_bowls_three | **0.62** | **0.54** | 0.55 | 0.39 | 0.27 | 0.13 | 0.31 | 0.22 |
| stack_bowls_two | **0.89** | **0.85** | 0.81 | 0.69 | 0.73 | 0.71 | 0.79 | 0.80 |
| stamp_seal | 0.35 | 0.25 | 0.25 | 0.16 | **0.41** | **0.43** | 0.10 | 0.09 |
| turn_switch | 0.22 | 0.18 | 0.24 | 0.26 | 0.40 | 0.31 | **0.54** | **0.49** |
| **Overall** | **0.5536** | **0.4536** | 0.5214 | 0.3906 | 0.4866 | 0.4280 | 0.4992 | 0.4360 |

Table S15: Success rates on the RoboTwin benchmark (low-data regime).

Vid2WAM achieves the best aggregate performance in the low-data regime under both clean and randomized evaluation conditions.

| Task | Vid2WAM | | Fast-WAM | | $\pi_{0.5}$ | | Motus | |
|---|---|---|---|---|---|---|---|---|
| | Clean | Random | Clean | Random | Clean | Random | Clean | Random |
| adjust_bottle | **1.00** | **1.00** | **1.00** | 0.99 | 0.99 | 0.98 | **1.00** | **1.00** |
| beat_block_hammer | 0.99 | 0.96 | **1.00** | **0.99** | 0.89 | 0.89 | 0.83 | 0.85 |
| **blocks_ranking_rgb** | **0.08** | 0.02 | 0.06 | **0.08** | 0.03 | 0.04 | 0.04 | **0.08** |
| blocks_ranking_size | 0.83 | 0.84 | **0.85** | 0.82 | 0.69 | 0.42 | 0.79 | **0.87** |
| **click_alarmclock** | 0.55 | **0.81** | 0.50 | 0.44 | **0.58** | 0.53 | 0.49 | 0.79 |
| click_bell | **1.00** | **1.00** | **1.00** | **1.00** | 0.91 | 0.79 | **1.00** | **1.00** |
| dump_bin_bigbin | **0.98** | **0.94** | **0.98** | 0.93 | 0.88 | 0.89 | 0.94 | **0.94** |
| grab_roller | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | 0.99 | 0.99 |
| handover_block | **0.95** | 0.86 | **0.95** | 0.85 | 0.78 | 0.72 | 0.92 | **0.89** |
| handover_mic | **1.00** | **1.00** | 0.99 | 0.99 | 0.97 | 0.83 | **1.00** | **1.00** |
| lift_pot | 0.98 | 0.99 | **0.99** | **1.00** | 0.80 | 0.86 | 0.97 | 0.96 |
| move_can_pot | 0.94 | 0.98 | **0.98** | **0.99** | 0.48 | 0.56 | 0.93 | 0.97 |
| **move_playingcard_away** | 0.04 | 0.02 | **0.11** | 0.00 | 0.08 | **0.08** | 0.06 | 0.01 |
| move_stapler_pad | 0.72 | **0.73** | 0.67 | 0.65 | 0.63 | 0.55 | **0.77** | 0.71 |
| hanging_mug | **0.66** | **0.51** | 0.49 | 0.47 | 0.08 | 0.12 | 0.51 | 0.44 |
| **open_laptop** | **0.18** | **0.14** | 0.04 | 0.03 | 0.12 | **0.14** | 0.11 | 0.10 |
| open_microwave | 0.47 | 0.48 | 0.55 | 0.55 | 0.53 | 0.65 | **0.77** | **0.72** |
| pick_diverse_bottles | 0.68 | 0.76 | **0.78** | **0.86** | 0.74 | 0.66 | 0.48 | 0.61 |
| **pick_dual_bottles** | 0.72 | 0.76 | **0.99** | **0.91** | 0.79 | 0.81 | 0.65 | 0.72 |
| place_a2b_left | **0.96** | 0.93 | 0.94 | **0.94** | 0.90 | 0.87 | 0.94 | 0.92 |
| **place_a2b_right** | 0.10 | **0.17** | 0.03 | 0.10 | **0.16** | 0.15 | 0.15 | 0.15 |
| **place_bread_basket** | **0.71** | **0.66** | 0.01 | 0.00 | 0.19 | 0.21 | 0.29 | 0.23 |
| place_bread_skillet | **0.96** | 0.85 | 0.95 | **0.95** | 0.78 | 0.79 | 0.89 | 0.87 |
| place_can_basket | 0.67 | 0.70 | 0.71 | 0.58 | **0.74** | **0.73** | 0.64 | 0.72 |
| place_cans_plasticbox | **0.99** | **0.98** | 0.98 | 0.96 | 0.82 | 0.75 | 0.98 | 0.97 |
| **place_container_plate** | **0.97** | 0.90 | 0.73 | 0.61 | 0.81 | 0.86 | 0.95 | **0.92** |
| place_dual_shoes | 0.88 | **0.93** | **0.92** | 0.91 | 0.68 | 0.71 | 0.81 | 0.85 |
| place_empty_cup | **1.00** | **1.00** | 0.99 | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** |
| place_fan | **0.97** | **0.96** | **0.97** | 0.95 | 0.83 | 0.85 | **0.97** | 0.94 |
| **place_burger_fries** | **0.17** | 0.17 | 0.02 | 0.00 | 0.03 | 0.02 | 0.16 | **0.21** |
| place_mouse_pad | 0.90 | 0.86 | **0.94** | **0.93** | 0.67 | 0.74 | 0.89 | 0.92 |
| place_object_basket | 0.74 | 0.86 | 0.81 | 0.81 | 0.78 | 0.82 | **0.90** | **0.91** |
| place_object_scale | 0.90 | 0.94 | 0.89 | **0.96** | 0.77 | 0.75 | **0.92** | 0.93 |
| **place_object_stand** | **0.62** | **0.63** | 0.29 | 0.34 | 0.24 | 0.22 | 0.41 | 0.52 |
| place_phone_stand | **0.98** | **0.99** | **0.98** | **0.99** | 0.83 | 0.87 | 0.89 | 0.90 |
| **move_pillbottle_pad** | 0.11 | 0.05 | 0.01 | 0.01 | **0.18** | **0.15** | 0.05 | 0.04 |
| place_shoe | 0.94 | **1.00** | 0.97 | 0.99 | 0.93 | 0.94 | **0.98** | 0.99 |
| **press_stapler** | 0.99 | **1.00** | **1.00** | 0.97 | 0.94 | 0.93 | **1.00** | **1.00** |
| put_bottles_dustbin | **0.83** | **0.93** | 0.80 | 0.89 | 0.49 | 0.42 | 0.56 | 0.72 |
| put_object_cabinet | 0.93 | **0.91** | **0.94** | **0.91** | 0.76 | 0.77 | 0.88 | 0.86 |
| rotate_qrcode | 0.93 | 0.85 | **0.96** | **0.90** | 0.81 | 0.84 | 0.89 | 0.85 |
| scan_object | **0.94** | 0.89 | 0.91 | **0.95** | 0.73 | 0.64 | 0.82 | 0.91 |
| shake_bottle | **1.00** | **1.00** | **1.00** | 0.99 | 0.99 | 0.97 | **1.00** | **1.00** |
| **shake_bottle_horizontally** | **1.00** | 0.99 | **1.00** | 0.99 | **1.00** | **1.00** | **1.00** | **1.00** |
| stack_blocks_three | **0.97** | 0.95 | 0.96 | **0.96** | 0.86 | 0.84 | 0.94 | 0.91 |
| **stack_blocks_two** | **1.00** | **1.00** | **1.00** | 0.97 | 0.97 | 0.98 | **1.00** | **1.00** |
| stack_bowls_three | 0.81 | **0.82** | 0.77 | 0.80 | **0.84** | 0.66 | 0.83 | 0.74 |
| **stack_bowls_two** | 0.96 | **0.97** | 0.96 | **0.97** | 0.86 | 0.82 | **0.98** | 0.96 |
| stamp_seal | 0.77 | **0.86** | **0.84** | 0.77 | 0.71 | 0.73 | 0.71 | 0.65 |
| turn_switch | 0.68 | 0.69 | 0.65 | 0.72 | 0.69 | 0.69 | **0.86** | **0.90** |
| **Overall** | **0.7830** | **0.7848** | 0.7572 | 0.7474 | 0.6792 | 0.6648 | 0.7508 | 0.7628 |
| **Novel subset** | **0.5467** | **0.5527** | 0.4500 | 0.4280 | 0.4653 | 0.4627 | 0.4893 | 0.5153 |

Table S16: Success rates on the RoboTwin benchmark (novel regime). Novel target tasks are marked in bold.

Vid2WAM leads both full-benchmark and novel-subset aggregates under clean and randomized settings, with larger gains on the novel subset.

# E SPATIAL AND TEMPORAL MASKING

For building real-world novel task pseudo-data, image synthesis is restricted to the overhead view because independently generated views do not reliably preserve cross-view alignment. To fit the model, the generated image occupies the upper camera slot, while unavailable wrist-camera regions are zero-filled, and a spatial latent mask limits pseudo-video loss to the valid region. Ground-truth multi-view and action supervision remain unmasked.

**Spatial masking.** In the single-view construction, the overhead image is resized to the top $224 \times 448$ region of a $448 \times 448$ composite image, while the bottom $224 \times 448$ region is a black placeholder. The student VAE has a spatial downsampling factor of 16, so this image is represented on a $28 \times 28$ latent grid. Consequently, latent rows $h \in [0, 14)$ and all columns $w \in [0, 28)$ are valid, whereas rows $h \in [14, 28)$ are ignored. Multi-view samples that do not contain an explicit spatial mask are assigned an all-one mask and therefore retain the original full-image loss.

**Temporal masking.** Video clips are padded to a fixed training length before VAE encoding. Because the student VAE downsamples time by a factor of four, frame-level padding flags are converted into a latent-time validity mask. A latent step is ignored only when all contributing video frames are padding, and the video loss is averaged over the remaining valid latent steps. The first frame, which provides the image condition, is handled separately. This temporal masking is the same for single-view and multi-view samples; single-view pseudo videos additionally use the spatial mask described above. Action chunks are fitted to the student horizon, with padded action positions excluded from supervision.

**Rollout-duration cutoff.** Simulation pseudo-rollouts are truncated to the first 8 seconds of video before constructing pseudo-data to mitigate long-horizon generation errors and hallucinations. Real-world pseudo-rollouts use all frames from the 20-second generated video and do not apply this temporal cutoff, which keeps the full additional supervision and improves real-world task success rates. This duration cutoff is separate from the padding-based temporal mask above.