WORLD ACTION MODELING WITH PROGRESSIVE VISUAL SUB-GOALS

Anonymous authors
Paper under double-blind review

ABSTRACT

World action models (WAMs) have emerged as a promising paradigm for robotic control by jointly predicting future visual dynamics and actions from an initial observation and instruction. However, existing WAMs struggle with long-horizon prediction, as generating dense video rollouts is highly inefficient. Some recent WAMs address this by predicting a single future frame without generating the full video, but this approach neglects *how to progress* toward the goal. We present **ProWAM**, a progressive world action model that jointly predicts actions and an ordered sequence of sparse visual sub-goals, providing explicit visual guidance to anchor action generation throughout task execution. This design scales naturally, as sub-goal prediction can be learned from large-scale action-free videos, allowing the video backbone to offload complex visual planning from the action policy. For efficient action generation, ProWAM executes a single video-backbone forward pass to cache sparse sub-goal features, eliminating iterative full-video generation and requiring only lightweight action denoising during replanning. Across extensive evaluations, ProWAM achieves superior out-of-distribution robustness. On simulation benchmarks, it sets new *state-of-the-art* results on LIBERO-Plus (85.8%) and randomized RoboTwin (75.7%), outperforming the strongest baseline with relative gains of up to **+35.9%**. On RoboCasa365, ProWAM achieves a 48.1% success rate and 18.2% on the challenging Composite-Unseen split, ranking **2nd overall**. Crucially, in zero-shot real-world experiments, ProWAM achieves 70.0% success, outperforming the strongest baseline by **+15.0** (from 55.0% to 70.0%, a **+27.3%** relative gain) in novel scenes. These results demonstrate the value of progress-indexed visual foresight for closed-loop control.

1 INTRODUCTION

Vision-language-action (VLA) models primarily inherit semantic knowledge from vision-language pretraining but remain data-hungry due to their reliance on large, costly collections of robot demonstrations (Kim et al., 2024; Black et al., 2024; 2025). In contrast, *world action models* (WAMs) introduce a complementary source of transferable knowledge: visual-dynamics priors learned by video generation models from large-scale video datasets (Blattmann et al., 2023; Yang et al., 2024; Wan et al., 2025). Recent WAMs have shown that incorporating such predictive visual knowledge can improve data efficiency and out-of-distribution generalization in robotic control (Pai et al., 2025; Ye et al., 2026c; Yuan et al., 2026; Zhang et al., 2026; Li et al., 2026).

Existing WAMs differ in how they utilize future visual prediction to guide action generation. *Full-imagination* WAMs (Pai et al., 2025; Ye et al., 2026c; Li et al., 2026) leverage dense future rollouts that provide detailed near-term guidance but are inefficient as they require multiple denoising passes through a large video backbone. Alternatively, *zero-imagination* WAMs (Yuan et al., 2026; Ye et al., 2026a) generate actions without predicting future visual states. Between these extremes, ImageWAM (Zhang et al., 2026) conditions control on an action-horizon endpoint frame without explicitly representing intermediate visual states. These designs reveal a fundamental trade-off: dense visual rollouts provide short-horizon dynamics at high computational cost, whereas other efficient alternatives lack explicit intermediate-progress guidance over the full task.

To reconcile this trade-off, we introduce **ProWAM**, a *progressive world action model* that extends a single fixed-state prediction into an ordered sequence of sparse visual sub-goals. Each sub-goal

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.

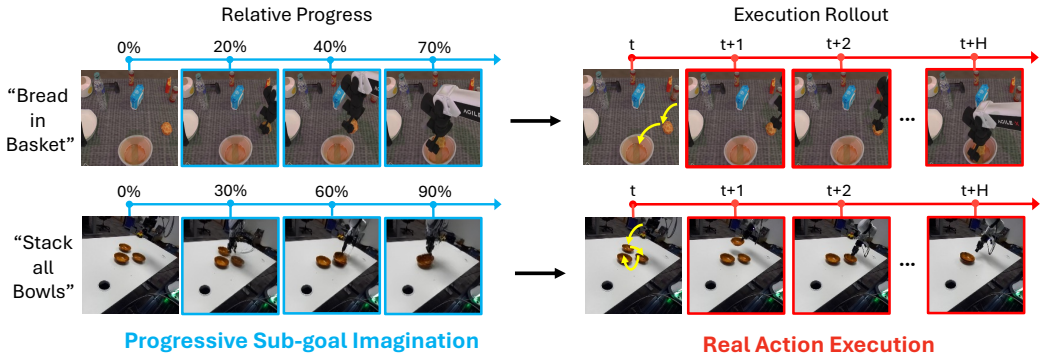


Figure 1: **Progressive Sub-goal Imagination for Long-Horizon Control.** Given the current observation, ProWAM generates sparse visual targets at requested relative progress values (*left*). The resulting sub-goal features directly condition action generation for the corresponding execution rollout (*right*), providing explicit visual guidance without generating dense video rollouts.

is indexed by normalized progress relative to the current state, where $r = 0$ denotes the current observation and $r = 1$ denotes the trajectory endpoint. This relative parameterization dynamically re-anchors sub-goals to the current observation, enabling consistent guidance across arbitrary task stages and varying trajectory lengths. Consequently, these sparse targets efficiently guide action generation without requiring dense video rollouts or separate high-level planning modules.

Specifically, ProWAM adopts a vision-action *mixture-of-transformers* (MoT) architecture consisting of a video generation expert and a lightweight diffusion-based action expert for policy generation. Each visual sub-goal is represented as a latent slot associated with a normalized progress value. The video expert jointly predicts near-term future transitions and progress-conditioned sub-goals, while the action expert attends to current observations and sub-goals to generate low-level action chunks. Asymmetric attention prevents action information from altering the visual prediction pathway. This design preserves the generative prior of the video expert, enabling video-only pretraining on large-scale action-free videos followed by joint fine-tuning on task-specific robot demonstrations to couple visual planning with action generation. At inference, ProWAM reuses cached sub-goal features to eliminate repeated video-backbone computation, enabling efficient closed-loop control. Overall, our contributions are threefold:

- We introduce *progressive sub-goal imagination* (Figure 1), a new visual planning paradigm for WAMs that represents long-horizon futures as an ordered sequence of sparse visual sub-goals indexed by normalized task progress. This representation provides explicit intermediate guidance without requiring dense video rollouts during inference or an external planner.
- We instantiate this paradigm in **ProWAM**, a WAM that natively integrates progress-conditioned visual sub-goal prediction with action generation for long-horizon control. By embedding sub-goal slots into the video pathway and coupling them via asymmetric attention, ProWAM enables scalable action-free video pretraining followed by joint vision-action fine-tuning. At inference, cached sub-goal features guide action denoising without repeated video-expert computation.
- Extensive simulation and real-world evaluations demonstrate that ProWAM achieves exceptional out-of-distribution robustness, setting a new *state-of-the-art* on LIBERO-Plus (Fei et al., 2025) and randomized RoboTwin (Chen et al., 2025) with an average absolute performance gain of **+11.4** over prior WAMs. Furthermore, it ranks as the **second** overall on challenging long-horizon RoboCasa365 (Nasiriany et al., 2024) benchmark, and outperforms DreamZero (Ye et al., 2026c) by **+15.0** on zero-shot real-world tasks in novel scenes.

2 Preliminaries

Diffusion-based Video Generation Pipeline. Prevailing video generation pipelines (Yang et al., 2024; Wan et al., 2025) predominantly adopt a continuous-time denoising process formalized via flow matching (Lipman et al., 2022; Esser et al., 2024). In this work, we build upon the I2V framework of Wan (Wan et al., 2025). Formally, an input video $\mathbf{V}$ is first encoded into a compressed latent representation using a pretrained *variational autoencoder* (VAE): $\mathcal{E}(\mathbf{V}) = \mathbf{X} \in$

$\mathbb{R}^{c\times(f/4)\times(h/8)\times(w/8)}$, where c is the latent channel dimension, and f, h, w represent the frame count, height, and width of the input video, respectively. Subsequently, we employ flow matching to learn a continuous-time generative diffusion process, which models the mapping between the target latent $\mathbf{X}^1$ (i.e., $\mathbf{X}$) and random Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$ via a fixed, linearly sampled timestep process: $\mathbf{X}^t = t\mathbf{X}^1 + (1-t)\epsilon, t \in [0, 1]$. In this way, a θ-parameterized velocity prediction model $\mathcal{V} : \mathbb{R}^{c\times(f/4)\times(h/8)\times(w/8)} \to \mathbb{R}^{c\times(f/4)\times(h/8)\times(w/8)}$, instantiated as a *diffusion transformer* (DiT), is trained using the following *mean squared error* (MSE) objective:

$$\mathcal{L}_{\text{video}} = \mathbb{E}_{\epsilon,\mathbf{X}^1,\boldsymbol{T}_{\text{ref}},\boldsymbol{I}_{\text{ref}},t}\left\|\mathcal{V}(\mathbf{X}^t, t, \boldsymbol{T}_{\text{ref}}, \boldsymbol{I}_{\text{ref}}; \theta) - \frac{d\mathbf{X}^t}{dt}\right\|_2^2, \tag{1}$$

where the target velocity is given by $d\mathbf{X}^t/dt = \mathbf{X}^1 - \epsilon$. Here $\boldsymbol{T}_{\text{ref}}$ and $\boldsymbol{I}_{\text{ref}}$ denote text and image embeddings extracted by off-the-shelf encoders (umT5 (Raffel et al., 2020) and CLIP (Radford et al., 2021), respectively). These embeddings are fused with the original latent representation via cross-attention, providing conditional reference for controllable video generation. During inference, a video can be generated by iteratively denoising ϵ using $\mathcal{V}$, conditioned on the $\boldsymbol{T}_{\text{ref}}$ and $\boldsymbol{I}_{\text{ref}}$.

Tailoring Video Generative Knowledge to Physical Action. To repurpose a pretrained video generator as a control policy, a *world action model* (WAM) augments the video backbone with a lightweight action expert, typically instantiated as a DiT (Black et al., 2025; Ye et al., 2026c). This expert inherits the visual-dynamics prior of the video model while emitting executable controls. Concretely, let $\boldsymbol{A} = [\mathbf{a}_0, \ldots, \mathbf{a}_{H-1}] \in \mathbb{R}^{H\times d_a}$ denote an action chunk of horizon H, where d_a is the action dimension. When observations and actions are temporally aligned, a clip containing f observation frames has $f-1$ transitions, and we therefore set $H = f - 1$. The action and video experts jointly learn a policy that generates the action chunk conditioned on the textual instruction $\boldsymbol{T}_{\text{ref}}$, the observed initial visual state (i.e., the reference image $\boldsymbol{I}_{\text{ref}}$), and the visual conditioning $\mathcal{C}(\mathbf{X})$ extracted from the predicted future video latents $\mathbf{X} \in \mathbb{R}^{c\times(f/4)\times(h/8)\times(w/8)}$, formalizing an *imagine-to- act* objective $\pi_\gamma = p(\boldsymbol{A} \mid \boldsymbol{T}_{\text{ref}}, \mathcal{C}(\mathbf{X}), \boldsymbol{I}_{\text{ref}})$. Mirroring the video branch, the action chunk follows an independent flow-matching formulation: $\boldsymbol{A}^{t'} = t'\boldsymbol{A}^1 + (1-t')\epsilon$, where the action timestep $t' \sim \mathcal{U}[0, 1]$ is sampled independently from the video timestep t. The ϕ-parameterized action velocity expert $\mathcal{A} : \mathbb{R}^{H\times d_a} \to \mathbb{R}^{H\times d_a}$ is trained by minimizing:

$$\mathcal{L}_{\text{act}} \;=\; \mathbb{E}_{t',\boldsymbol{A},\mathbf{X}}\Big[\big\|\mathcal{A}(\boldsymbol{A}^{t'}, t', \mathcal{C}(\mathbf{X}), \boldsymbol{I}_{\text{ref}}, \boldsymbol{T}_{\text{ref}}; \phi) - (\boldsymbol{A}^1 - \epsilon)\big\|_2^2\Big]. \tag{2}$$

The overall training objective combines both experts via a weighted sum:

$$\mathcal{L} \;=\; \lambda_{\text{video}}\mathcal{L}_{\text{video}} \;+\; \lambda_{\text{act}}\mathcal{L}_{\text{act}}. \tag{3}$$

where λ_{act} and λ_{video} are loss weights and the full policy is composed from $\gamma = \{\phi, \theta\}$. During the inference, the model performs chunk-level action generation through the same denoising process used for $\mathcal{V}$, thereby grounding control in the *imagination* prior inherited from video generalists.

Existing WAMs span a spectrum of visual contexts $\mathcal{C}(\mathbf{X})$ provided to the action policy: *full-imagination* models (Pai et al., 2025; Ye et al., 2026c) condition actions on complete predicted video clips ($\mathcal{C}(\mathbf{X}) = \mathbf{X}_{1:f}$), whereas *zero-imagination* models (Yuan et al., 2026) omit generated future latents entirely ($\mathcal{C}(\mathbf{X}) = \emptyset$). This presents a clear trade-off: *full-imagination* offers dense spatial guidance but incurs substantial latency and video artifacts, whereas *zero-imagination* enables real-time execution but lacks the visual foresight required for long-horizon tasks.

3 ProWAM: Long-Horizon Control with Progressive Sub-goals

Beyond a Single Goal Guidance. To enable explicit visual planning in WAM, a natural baseline is *final-goal prediction* (FGP)[1]. However, compressing an entire episode into a solitary final frame specifies where the task terminates but provides no guidance on how to get there, leaving the intermediate execution path unspecified. Inspired by hierarchical planning paradigms that decompose extended horizons into structured sub-states/goals (Li et al., 2023; Luo et al., 2026c), we introduce a *progressive sub-goal signal* directly into the WAM architecture. Specifically, we generalize FGP into

[1]Our pilot study finds that *self-generated final goal* improves baselines in most settings (Appendix A).

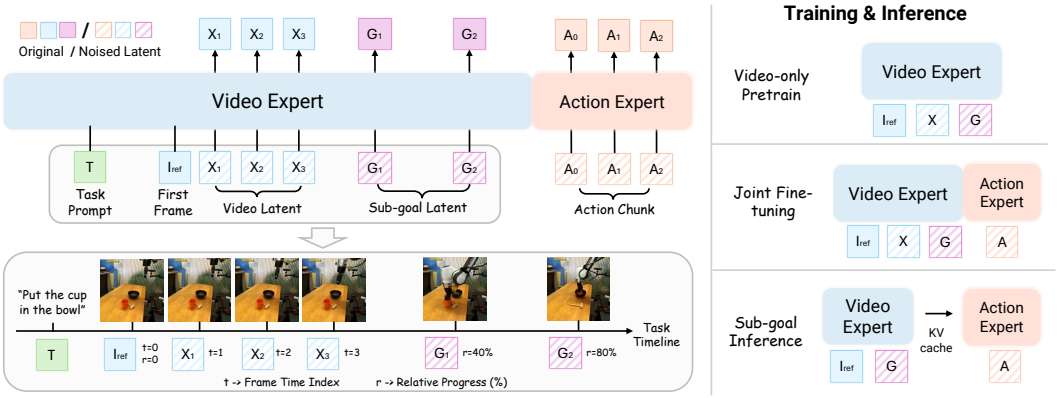


Figure 2: **Overview of ProWAM. Left: Training architecture.** Given a task instruction $\mathbf{T}$ and reference observation $\mathbf{I}_{\text{ref}}$, ProWAM jointly denoises future-video latents $\mathbf{X}$, progressive sub-goal latents $\mathbf{G}$, and action chunks $\mathbf{A}$. The bottom timeline aligns these latent variables with physical execution, with sub-goals providing visual targets at specified relative progress levels. **Right: Training and inference pipeline.** ProWAM employs two-stage training: (1) *Video-only Pretraining*, where the video expert learns visual dynamics and progressive sub-goal prediction from action-free videos; and (2) *Joint Fine-tuning*, where the video and action experts are jointly optimized on robot demonstrations, enabling the action expert to condition action generation on predicted sub-goals. At inference, the video expert predicts sparse sub-goal features instead of a dense video rollout. These features are cached and reused during action denoising to avoid repeated video-expert computation.

a *progressive sequence of sparse k sub-goals* that predicts an ordered sequence of sparse sub-goals at selected relative future positions, yielding our *progressive world action model* (**ProWAM**). Crucially, while classical hierarchical planners (Ai et al., 2026; Shou et al., 2026) depend on dedicated, multi-stage goal-generation modules, ProWAM generates the entire sub-goal trajectory natively and injects their features directly to the action expert (as shown in Figure 2).

Progressive Sub-goal Parameterization. We parameterize intermediate sub-goals using a normalized *progress coordinate* $r \in [0, 1]$, defined as the relative temporal position along the remaining demonstration trajectory from the current observation ($r = 0$) to its endpoint ($r = 1$). This normalization provides a consistent temporal progress scale across trajectories of different lengths. Given k monotonically increasing progress levels $0 < r_1 < \cdots < r_k \leq 1$, the i-th sub-goal corresponds to the frame at

$$\tau(r_i) = \tau_0 + \left\lfloor r_i \big(\tau_{\text{end}} - \tau_0\big) \right\rceil, \qquad i = 1, \ldots, k, \tag{4}$$

where τ_0 and τ_{end} denote the current-frame and demonstration-endpoint indices, respectively, and $\lfloor \cdot \rceil$ denotes nearest-integer rounding. Each selected frame is independently encoded as $\tilde{G}_i = \mathcal{E}(V_{\tau(r_i)}) \in \mathbb{R}^{c \times 1 \times (h/8) \times (w/8)}$. The resulting sequence spans selected future states, from near-term goal at smaller r_i to longer goal as $r_i \to 1$ ($r_1 = 1, k = 1$ in FGP). During training, we sample $k \sim \mathcal{U}\{1, \ldots, K_{\max}\}$, enabling flexible sub-goal learning strategy. At inference, $\mathbf{G}$ is predicted relative to the latest observation, which serves as $r = 0$ at each replanning step.

To incorporate progressive sub-goal guidance directly into the WAM architecture, we append the progressive sub-goal sequence to the video latent sequence, yielding:

$$\mathbf{Z} = \big[\underbrace{\mathbf{X}_0}_{\text{current observation}} \,\big\|\, \underbrace{\mathbf{X}_1, \ldots, \mathbf{X}_{f/4-1}}_{\text{clip transitions}} \,\big\|\, \underbrace{\mathbf{G}_1, \ldots, \mathbf{G}_k}_{\text{progressive sub-goals}} \big], \tag{5}$$

where $\mathbf{X}$ denotes the $f/4$ video latent frames, with $\mathbf{X}_0$ representing the initial observation. During training, $\mathbf{X}_0$ serves as a clean observation latent at every denoising timestep, whereas the transition clip and sub-goal latents follow the same flow-matching process. This formulation preserves full-sequence video prediction while adding the guidance of explicit progress-conditioned sub-goals.

Progress-conditioned Modulation. Sub-goal slots share the same network weights and temporal position encodings as the video latents. To specify the target temporal stage of each sub-goal, we additionally condition every slot on its progress coordinate $r_i \in [0, 1]$. Specifically, r_i is mapped

through a sinusoidal embedding followed by a *multi-layer perceptron* (MLP) to yield an *adaptive LayerNorm* (adaLN) (Peebles & Xie, 2023) modulation offset $\Delta_{\mathrm{mod}}(r_i)$. This offset is added directly to the standard flow-matching timestep modulation:

$$\mathrm{mod}(\mathbf{G}_i) \;=\; \mathrm{mod}(t) \;+\; \Delta_{\mathrm{mod}}(r_i). \tag{6}$$

This modulation anchors each sub-goal to its corresponding task stage relative to the reference observation. Trained under progress conditioning and varying sub-goal counts k, ProWAM natively generalizes to flexible sub-goal schedules at inference without architectural modifications.

Asymmetric Vision-Action Coupling. Following Bi et al. (2026); Yuan et al. (2026), ProWAM implements the video expert $\mathcal{V}$ and action expert $\mathcal{A}$ via a *mixture-of-transformers* (MoT) (Liang et al., 2024) architecture with shared attention heads. This design enables $\mathcal{V}$ to retain its pretrained generative representations while $\mathcal{A}$ specializes in action generation by directly conditioning on predicted visual features. Information flow across the experts is governed by a structured visibility mask:

- **Sub-goal planning** [$\mathbf{G} \leftrightarrow \mathbf{G}, \mathbf{G} \rightarrow \mathbf{X}_0$][2]: $\mathbf{G}$ attend bidirectionally among themselves and to the reference observation $\mathbf{X}_0$. This joint attention enables intermediate milestones to share visual features, allowing information exchange across sub-goal slots rather than k isolated predictions.
- **Transition conditioning** [$\mathbf{X} \leftrightarrow \mathbf{X}, \mathbf{X} \rightarrow \{\mathbf{X}_0, \mathbf{G}\}$]: $\mathbf{X}$ attend bidirectionally among themselves to maintain temporal coherence, while conditioning on $\mathbf{X}_0$ and $\mathbf{G}$ as guidance anchors. This encourages visual consistency between the generated video and the planned sub-goals.
- **Asymmetric vision-action coupling** [$\boldsymbol{A} \leftrightarrow \boldsymbol{A}, \boldsymbol{A} \rightarrow \{\mathbf{X}_0, \mathbf{G}\}, \mathbf{X} \not\rightarrow \boldsymbol{A}$]: $\mathcal{V}$ does not attend to action tokens ($\mathbf{X} \not\rightarrow \boldsymbol{A}$), maintaining an action-independent visual forward pathway of $\mathcal{V}$. Conversely, action tokens attend bidirectionally among themselves and cross-attend to visual anchors $\{\mathbf{X}_0, \mathbf{G}\}$, grounding low-level action execution directly in the predicted sub-goals.

Thus, the joint policy $\pi_\gamma(\boldsymbol{A} \mid \boldsymbol{T}_{\mathrm{ref}}, \mathcal{C}(\mathbf{Z}), \boldsymbol{I}_{\mathrm{ref}})$ (with parameters $\gamma = \{\phi, \theta\}$) is explicitly conditioned on the visual context chain anchored by the current observation:

$$\mathcal{C}(\mathbf{Z}) \;=\; \{\mathbf{X}_0\} \cup \{\mathbf{G}_1, \ldots, \mathbf{G}_k\}. \tag{7}$$

Training Objective. To jointly enable progressive visual imagination and downstream action control, ProWAM is trained via flow matching with independent timesteps for the video expert ($t \sim \mathcal{U}[0,1]$) and action expert ($t' \sim \mathcal{U}[0,1]$), where sub-goals $\mathbf{G}$ share the video timestep t. The full objective combines three distinct velocity-regression terms:

$$\mathcal{L}_{\mathrm{final}} \;=\; \lambda_{\mathrm{video}}\mathcal{L}_{\mathrm{video}} \;+\; \lambda_{\mathrm{sub}}\,\mathcal{L}_{\mathrm{sub}} \;+\; \lambda_{\mathrm{act}}\,\mathcal{L}_{\mathrm{act}}, \tag{8}$$

where $\mathcal{L}_{\mathrm{video}}$ and $\mathcal{L}_{\mathrm{sub}}$ supervise flow-matching velocity predictions for transition latents $\mathbf{X}$ and sub-goal latents $\mathbf{G}$, respectively. Similarly, $\mathcal{L}_{\mathrm{act}}$ trains the action expert to predict velocities for action chunks $\boldsymbol{A}$, conditioned on the visual context $\mathcal{C}(\mathbf{Z})$, reference observation $\boldsymbol{I}_{\mathrm{ref}}$, and task instruction $\boldsymbol{T}_{\mathrm{ref}}$. We optimize this objective across 2 distinct phases to preserve generative priors while learning control: In **Stage 1** (Video-only Training), we pretrain the video expert on action-free video corpora using $\lambda_{\mathrm{video}}\mathcal{L}_{\mathrm{video}} + \lambda_{\mathrm{sub}}\mathcal{L}_{\mathrm{sub}}$ to learn visual dynamics and progress-conditioned sub-goal prediction. In **Stage 2** (Joint Video-Action Fine-tuning), we jointly fine-tune both experts on robot demonstrations using $\mathcal{L}_{\mathrm{final}}$, grounding action execution directly in the planned visual sub-goals. Consequently, ProWAM can leverage vast, action-free video corpora for scalable pre-training, drastically improving sample efficiency when fine-tuning on limited, expensive robot demonstrations.

Efficient Inference with Sub-goal Caching. Rather than performing burdensome joint video-action denoising that advances the video and action branches synchronously across T steps (Ye et al., 2026c), we empirically find that sub-goal features obtained from a single video-expert evaluation support comparable policy performance to full joint denoising (Appendix C). At each replanning step, we evaluate $\mathcal{V}$ once to predict the sub-goals and cache their key–value features. The action head then reuses this fixed cache throughout T action-denoising steps. Upon receiving a new observation, ProWAM recomputes the sub-goals and refreshes the cache. This reduces video-expert forward passes from T to 1 ($\approx 10\times$ fewer video-expert evaluations), achieving inference cost comparable to prior cached WAMs (Pai et al., 2025; Yuan et al., 2026; Zhang et al., 2026).

[2]The notation $A \rightarrow B$ ($A \not\rightarrow B$) indicates that queries in A attend (do not attend) to keys/values in B, whereas $A \leftrightarrow A$ denotes bidirectional self-attention within A. Note that $\mathbf{X}_0$ only attends to itself.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETTINGS

Following the evaluation protocols in concurrent works (Black et al., 2025; Bu et al., 2025b; Yuan et al., 2026; Luo et al., 2026a), we adopt several prevailing benchmarks for *in-domain* (ID) and *out-of-domain* (OOD) simulation evaluation, including LIBERO (Liu et al., 2023) & LIBERO-Plus (Fei et al., 2025), RoboTwin (Chen et al., 2025), and RoboCasa365 (Nasiriany et al., 2024), alongside zero-shot real-world robot deployments on Franka. We strictly follow the standard evaluation metrics tailored for each benchmark and report the average success rate for each task. Our ProWAM adopts the Wan2.2-TI2V-5B (Wan et al., 2025) as the video backbone. Since our sub-goal-conditioned prediction relies on a strong visual world prior, we study two paradigms that differ in the strength of that prior: (i) **ProWAM-Base**, initialized directly from the Wan base model (w/o **Stage-1**). (ii) **ProWAM**, initialized from a video-only paradigm pretrained on large-scale robot and egocentric data (see Appendix B for details). The video branch still generates sub-goals as an auxiliary co-training objective, so the shared representation remains sub-goal-aware. At inference, the action expert does not require fully denoised sub-goal latents but conditions on sub-goal key-value features cached from a single video-expert evaluation, yielding a simpler and faster policy. Additional experimental details are deferred to Appendix C & D.

4.2 SIMULATED POLICY EVALUATION

Table 1: **Success rate (%) on LIBERO-series.** Plus* is strictly used for zero-shot evaluation.

Method	LIBERO	Plus*
Vision-Language-Action Models (VLA)		
π_0 (Black et al., 2024)	94.1	53.6
π_0-Fast (Pertsch et al., 2025)	85.5	61.6
StarVLA (Ye et al., 2026b)	96.5	77.0
$\pi_{0.5}$ (Black et al., 2025)	96.9	77.4
ABot-M0 (Yang et al., 2026)	98.6	80.5
World Action Models (WAM)		
VLA-JEPA (Sun et al., 2026)	96.1	79.5
Fast-WAM (Yuan et al., 2026)	97.6	51.5
Image-WAM (Zhang et al., 2026)	98.4	83.1
Being-H0.7 (Luo et al., 2026b)	99.2	82.1
ProWAM-Base	**99.2**	80.3
ProWAM	98.8	**85.8**

Table 2: **Success rate (%) on RoboTwin.** Random* is used for OOD evaluation.

Method	Clean	Random*
Vision-Language-Action Models (VLA)		
DP (Chi et al., 2025)	28.0	0.6
RDT (Liu et al., 2024)	34.5	13.7
π_0 (Black et al., 2024)	46.4	16.3
UP-VLA (Zhang et al., 2025)	52.9	15.2
World Action Models (WAM)		
Fast-WAM (Yuan et al., 2026)	71.9	6.3
BagelVLA (Hu et al., 2026)	75.3	20.5
HALO (Shou et al., 2026)	80.5	26.4
Image-WAM (Zhang et al., 2026)	85.0	37.6
MV-WAM (Chen et al., 2026a)	84.0	55.7
ProWAM-Base	**87.6**	67.4
ProWAM	85.8	**75.7**

LIBERO & LIBERO-Plus. Table 1 provides a comprehensive quantitative comparison between two variants of our model and prior baselines across the LIBERO benchmark suite. Notably, both of our models achieve near-saturation performance on the in-domain LIBERO tasks ($\geq 98.6\%$). As a result, the primary discriminative signal for evaluating generalizability shifts to the challenging zero-shot LIBERO-Plus evaluation setting. Under this regime, our ProWAM achieves a new *state-of-the-art* (SOTA) performance of 85.8%, markedly outperforming previous strong WAM and VLA baselines. Furthermore, ProWAM-Base reveals a sharp performance drop (5.5 lower) on zero-shot tasks, highlighting the importance of sub-goal pretraining for OOD robustness.

RoboTwin. Table 2 reports results on RoboTwin, where vision-action fine-tuning uses *only clean demonstrations* and evaluated under both in-distribution and randomized conditions. ProWAM achieves 75.7% success under scene randomization, outperforming the strongest prior WAM by **+20.0**. Removing video pretraining reduces randomized OOD success from 75.7% to 67.4%, while clean-scene performance remains comparable (85.8% versus 87.6%). These results indicate that video pretraining primarily improves robustness to visual domain shifts.

RoboCasa365. Table 3 reports results on RoboCasa365. ProWAM achieves a **48.1%** overall success rate and ranks **second overall** among the evaluated methods. It also delivers competitive perfor-

Table 3: **Closed-loop performance on RoboCasa365**. Full results are reported in Appendix 4.2.

Ranking	Method	Overall	Atomic-Seen	Composite-Seen	Composite-Unseen
1	Xiaomi-Robotics-1 (Team et al., 2026)	57.4	80.2	57.1	32.1
2	**ProWAM (Ours)**	**48.1**	**76.0**	**46.5**	**18.2**
3	ABot-M0.6 (Chen et al., 2026c)	46.6	79.4	48.3	7.9

mance on long-horizon composite tasks. Notably, ProWAM achieves the **second-highest success rate** on the challenging long horizon *Composite-Unseen* split, substantially outperforming prior world-model baselines. These results support the effectiveness of progressive visual sub-goals for long-horizon execution and generalization to held-out task compositions.

4.3 Zero-shot Real-World Evaluation

Evaluation protocol & Results To evaluate true out-of-the-box generalization without collecting environment-specific demonstrations or retraining baselines, we adopt DROID (Khazatsky et al., 2024) as the unified real-robot training dataset. Table 4 summarizes the real-world evaluation across the 4 physical tasks, which span basic *pick-and-place*, *long-horizon multi-object assembly*, *contact-rich placement*, and *spatially grounded placement* (full task details are given in Appendix D). ProWAM achieves an average success rate of **70.0%**. Notably, ProWAM outperforms DreamZero by a **+15.0** absolute margin, exhibiting particular advantages in spatial placement, where it reaches **100.0%** success against the baselines. These results demonstrate that ProWAM enables direct zero-shot generalization without requiring domain-specific fine-tuning.

Table 4: **Real-world performance.**

Method	**Size**	**SR (%)**
$\pi_{0.5}$	3B	50.0
DreamZero	14B	55.0
ProWAM (Ours)	5B	**70.0**

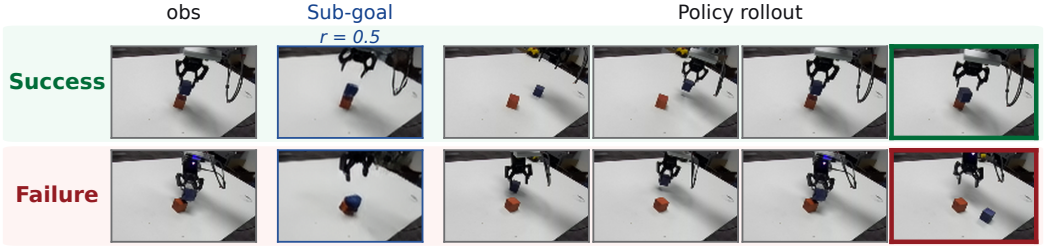


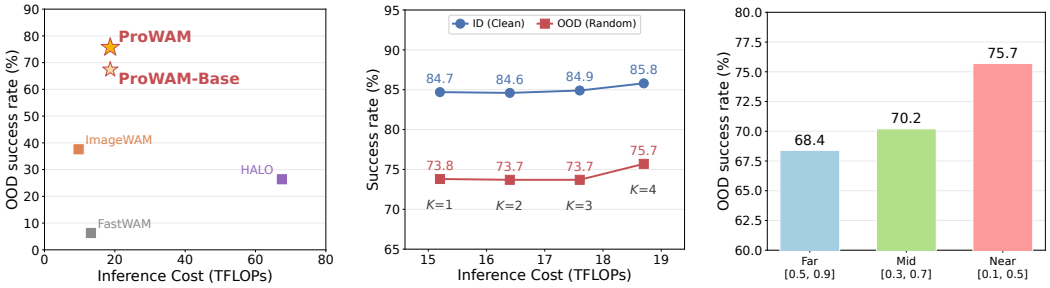
Figure 3: **Failure case on "Cube Stacking."** The failed sub-goal (*bottom*) exhibits more ambiguous contact geometry than the successful one (*top*), and the gripper subsequently opens prematurely.

Failure Case Analysis. Figure 3 compares sub-goals generated immediately before release in successful and failed "Cube Stacking" rollouts. In the successful rollout, the sub-goal depicts a centered placement with a clear contact boundary. In the failure case, the two cubes appear blurred together, obscuring the intended contact geometry, then the subsequent execution releases the cube off-center and drops it. This qualitative correspondence suggests that policy execution can tolerate texture imperfections but may remain sensitive to errors in task-relevant geometry.

4.4 Ablation Studies

We conduct ablation studies on RoboTwin, whose moderate success rates offer a sufficient dynamic range to differentiate design choices. In contrast, LIBERO's near-saturated in-distribution SR ($\sim$ 99%) obscures performance differences within run-to-run noise, making it far less discriminative.

Efficiency & Sub-goals Number. To evaluate computational efficiency, we benchmark inference latency and FLOPs for a 16-action chunk on a single NVIDIA H100 GPU. Compared with naive joint denoising (115.5 TFLOPs, 581 ms), our KV-cache strategy reduces computation to 18.7 TFLOPs (84% reduction) and latency to 313 ms (46% reduction), while achieving 3.3 percentage points higher OOD success in this evaluation. As shown in Figure 4(a), ProWAM achieves a favorable trade-off between computational efficiency and robustness: while in-distribution performance remains similar across methods, ProWAM reaches 75.7% OOD success under domain randomization, outperforming other methods at comparable or lower computational cost.



(a) **Efficiency vs. Performance.** (b) **Effect of Sub-goal Number.** (c) **Effect of Sub-goal Horizon.**

Figure 4: **Efficiency and sub-goal ablations. (a)** OOD success versus inference computation across WAMs. **(b)** Performance and computation across different sub-goal counts, where $k = 4$ yields peak OOD performance with efficient inference. **(c)** Performance across `Near`, `Mid`, and `Far` progress schedules. `Near` achieves the best OOD result while ID performance remains stable.

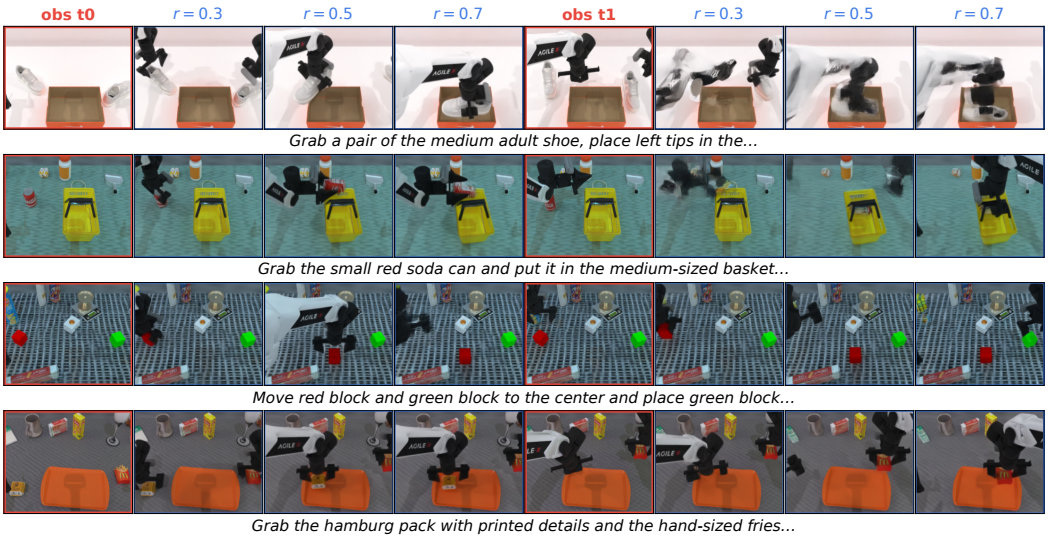


Figure 5: **Closed-loop sub-goal visualization.** For each observation at t_0 and t_1, the model generates sub-goals in a single pass at progress levels $r \in \{0.3, 0.5, 0.7\}$ to guide action generation.

Furthermore, k serves as an inference-time hyperparameter: a single model trained with $k \in \{1, \ldots, 4\}$ supports all four settings (Figure 4(b)). In-domain and OOD performance remains stable across $k \in [1, 3]$, and reaches its highest value of 75.7% at $k = 4$. Because all sub-goals are predicted in one cached forward pass, increasing k from 1 to 4 adds only 3.5 TFLOPs. Even at $k = 1$, ProWAM achieves 73.8% OOD success at 15.2 TFLOPs, substantially outperforming FastWAM's 6.3% at a comparable computational cost of 13.2 TFLOPs.

Sub-goal Planning Horizon. We evaluate three progress schedules under identical training settings: `Near` ($r \in [0.1, 0.5]$), `Mid` ($r \in [0.3, 0.7]$), and `Far` ($r \in [0.5, 0.9]$) (Figure 4(c)). Shorter horizons yield monotonic OOD gains, rising from 68.4% (`Far`) to 70.2% (`Mid`) and 75.7% (`Near`), while ID performance remains stable around 86%. This $+7.3$ improvement suggests that nearer targets offer tighter visual grounding for action generation, whereas distant goals leave larger unguided execution gaps. We thus adopt the `Near` schedule by default.

Sub-goal Generation. Figure 5 visualizes sub-goals generated in a single cached forward pass during closed-loop execution. First, the predicted states advance with increasing r and adapt to the latest observation at each replanning step. Second, while single-step generation introduces minor visual artifacts (e.g., blurred textures), sub-goals preserve salient scene structure and task progression, providing geometric guidance for action generation without requiring pixel-perfect reconstruction.

5 Related Work

World Action Models: Conditioning Control on Future Imagination. Generative video priors (Blattmann et al., 2023; Wan et al., 2025) have spurred the development of *world action models* (WAMs), which leverage visual foresight to condition robotic policies (Wang et al., 2026). Early frameworks adopted cascaded pipelines, first predicting future visual states and subsequently recovering actions via inverse dynamics or learned policies (Du et al., 2023; Bharadhwaj et al., 2024). Modern WAMs tightly unify predictive modeling and action generation within joint diffusion objectives and shared architectures (Kim et al., 2026b; Bi et al., 2026). These approaches form an *imagination spectrum* according to the future visual context used for action generation. *Full-imagination* WAMs (Pai et al., 2025; Ye et al., 2026c; Li et al., 2026) condition actions on dense generated video rollouts, providing detailed short-horizon guidance at high computational cost. In contrast, *zero-imagination* WAMs (Yuan et al., 2026; Ye et al., 2026a) use no predicted future visual states during inference. Between them, Zhang et al. (2026) conditions action generation on a single future target. ProWAM instead predicts a variable-length sequence of future visual states indexed by relative progress, providing guidance at multiple selected stages of the remaining trajectory.

Vision-Language-Action Models: Leveraging Foundation Knowledge. *Vision-language-action* (VLA) models transfer semantic priors from multimodal pretraining to robotic action generation by mapping task instructions and visual observations to actions (Ma et al., 2026). Building on generalist models (Kim et al., 2024; Team et al., 2024; Liu et al., 2024), existing VLAs represent actions using either discrete tokens (Zitkovich et al., 2023; Pertsch et al., 2025) or continuous diffusion and flow-matching objectives (Black et al., 2024; Liu et al., 2024; Chi et al., 2025). For long-horizon tasks, recent hierarchical VLAs additionally introduce textual sub-instructions, multimodal reasoning traces, or visual sub-goals (Belkhale et al., 2024; Ai et al., 2026; Shou et al., 2026; Chen et al., 2026b). While effective, many of these approaches rely on *decoupled* high-level modules or multi-stage inference pipelines, increasing computational complexity and inducing cascading error propagation across stages. ProWAM instead *natively* predicts progressive visual sub-goals within its video-generation pathway and directly couples them to action generation inside a *unified* WAM.

Integrating Intermediate Foresight into End-to-End Control. Directly executing long-horizon manipulation through unconditioned action policies often suffers from compounding trajectory drift. Decomposing complex tasks into intermediate visual sub-goals provides structured foresight, simplifying long-horizon control into manageable visual milestones (Li et al., 2023; Xie et al., 2026). Existing approaches incorporate such foresight either through standalone goal-generation modules (Shou et al., 2026) or via online model-based trajectory search over candidate rollouts (Neary et al., 2025; Guo et al., 2025). Nevertheless, modular goal generators remain vulnerable to error propagation across decoupled stages, while online trajectory search introduces prohibitive computational latency for real-time control. ProWAM instead integrates progress-conditioned sub-goal prediction and action generation through shared attention, reusing cached sub-goal features during iterative action denoising without requiring a separate planner or online trajectory search.

6 Conclusion

In this paper, we introduced ProWAM, a world action model that integrates visual planning and physical control within a unified generative framework. Motivated by hierarchical policies that decompose long-horizon tasks into intermediate sub-goal states, we incorporate progressive visual sub-goal prediction directly into the world-modeling process without requiring a separate high-level planner. Specifically, ProWAM uses a generative video expert to predict visual sub-goals indexed by relative temporal progress and couples their features to low-level action generation, without auxiliary modules or test-time search. Moreover, the unified architecture supports pretraining on large-scale action-free videos followed by joint vision-action fine-tuning, allowing visual dynamics learned without action labels to inform downstream action generation. Evaluations across simulation benchmarks and four physical manipulation tasks show strong performance under the evaluated in-domain, randomized, and held-out long horizon settings. Overall, our results support progressive visual prediction as an effective interface between visual imagination and robotic control.

AI USE STATEMENT

We used a generative AI tool to migrate the LaTeX source to the official ICLR template, check compilation, make formatting and readability edits, and transcribe and summarize existing experimental results from author-provided text and figures. This workflow did not use generative AI to generate datasets, formulate mathematical claims or proofs, design experiments, implement the proposed method, or produce new experimental results. The authors reviewed all AI-assisted changes and take responsibility for the final content, including text, claims, and artifacts produced with generative AI.

REPRODUCIBILITY STATEMENT

The method specification and training objectives are provided in Sections 2–3. Experimental settings, benchmark protocols, and evaluation metrics are described in the main text. The Appendix reports the pilot configurations, video-only pre-training setup, downstream training hyperparameters, ablations, and additional benchmark details needed to reproduce the reported results.

ETHICS STATEMENT

This work studies robot learning on physical manipulation platforms and does not involve human subjects or personally identifiable data. As with other robotic control systems, deployment outside the evaluated conditions may create safety risks. The models should therefore be validated under the intended operating conditions and used with appropriate physical safeguards and human oversight.

REFERENCES

Bo Ai, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Greg Balke, Kevin Black, George Bokinsky, Shihao Cao, Thomas Charbonnier, Vedant Choudhary, Foster Collins, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Maitrayee Dhaka, Jared DiCarlo, Danny Driess, Michael Equi, Adnan Esmail, Yunhao Fang, Chelsea Finn, Catherine Glossop, Thomas Godden, Ivan Goryachev, Lachlan Groom, Haroun Habeeb, Hunter Hancock, Karol Hausman, Gashon Hussein, Victor Hwang, Brian Ichter, Connor Jacobsen, Szymon Jakubczak, Rowan Jen, Tim Jones, Gregg Kammerer, Ben Katz, Liyiming Ke, Mairbek Khadikov, Chandra Kuchi, Marinda Lamb, Devin LeBlanc, Brendon LeCount, Sergey Levine, Xinyu Li, Adrian Li-Bell, Vladislav Lialin, Zhonglin Liang, Wallace Lim, Yao Lu, Enyu Luo, Vishnu Mano, Nandan Marwaha, Aikys Mongush, Liam Murphy, Suraj Nair, Tyler Patterson, Karl Pertsch, Allen Z. Ren, Gavin Schelske, Charvi Sharma, Baifeng Shi, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, Will Stoeckle, Jiaming Tang, Jimmy Tanner, Shalom Tekeste, Marcel Torne, Kyle Vedder, Quan Vuong, Anna Walling, Haohuan Wang, Jason Wang, XuDong Wang, Chris Whalen, Samuel Whitmore, Blake Williams, Charles Xu, Sukwon Yoo, Lili Yu, Wuming Zhang, Zhuoyang Zhang, and Ury Zhilinsky. $\pi_{0.7}$: a steerable generalist robotic foundation model with emergent capabilities, 2026. URL https://arxiv.org/abs/2604.15483.

Suneel Belkhale, Tianli Ding, Ted Xiao, Pierre Sermanet, Quon Vuong, Jonathan Tompson, Yevgen Chebotar, Debidatta Dwibedi, and Dorsa Sadigh. Rt-h: Action hierarchies using language. *arXiv preprint*, 2024.

Homanga Bharadhwaj, Debidatta Dwibedi, Abhinav Gupta, Shubham Tulsiani, Carl Doersch, Ted Xiao, Dhruv Shah, Fei Xia, Dorsa Sadigh, and Sean Kirmani. Gen2act: Human video generation in novel scenarios enables generalizable robot manipulation. *arXiv preprint*, 2024.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, et al. Motus: A unified latent action world model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35101–35113, 2026.

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

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. pi0: A vision-language-action flow model for general robot control. *arXiv preprint*, 2024.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. pi0.5: a vision-language-action model with open-world generalization. *arXiv preprint*, 2025.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint*, 2023.

Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025a.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions. *arXiv preprint*, 2025b.

Jintao Chen, Peidong Jia, Qingpo Wuwu, Jiaming Liu, Mengfei Du, Chun-Kai Fan, Xiaowei Chi, Hao Chen, Chengyu Bai, Zezhong Qian, et al. Mv-wam: Manifold-aware world action model with value augmentation. *arXiv preprint arXiv:2606.21088*, 2026a.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.

Yiye Chen, Yanan Jian, Xiaoyi Dong, Shuxin Cao, Jing Wu, Patricio Vela, Benjamin E. Lundell, and Dongdong Chen. Vista: Enhancing visual conditioning via track-following preference optimization in vision-language-action models, 2026b. URL https://arxiv.org/abs/2602.05049.

Yuzhi Chen, Ronghan Chen, Dongjie Huo, Yandan Yang, Dekang Qi, Haoyun Liu, Tong Lin, Shuang Zeng, Junjin Xiao, Xinyuan Chang, et al. Abot-physworld: Interactive world foundation model for robotic manipulation with physics alignment. *arXiv preprint arXiv:2603.23376*, 2026c.

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

Yilun Du, Mengjiao Yang, Pete Florence, Fei Xia, Ayzaan Wahid, Brian Ichter, Pierre Sermanet, Tianhe Yu, Pieter Abbeel, Joshua B Tenenbaum, et al. Video language planning. *arXiv preprint*, 2023.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

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.

Wenkai Guo, Guanxing Lu, Haoyuan Deng, Zhenyu Wu, Yansong Tang, and Ziwei Wang. Vla-reasoner: Empowering vision-language-action models with reasoning via online monte carlo tree search. *arXiv preprint arXiv:2509.22643*, 2025.

Yucheng Hu, Jianke Zhang, Yuanfei Luo, Yanjiang Guo, Xiaoyu Chen, Xinshu Sun, Kun Feng, Qingzhou Lu, Sheng Chen, Yangang Zhang, et al. Bagelvla: Enhancing long-horizon manipulation via interleaved vision-language-action generation. *arXiv preprint arXiv:2602.09849*, 2026.

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

Dongyoung Kim, Huiwon Jang, Myungkyu Koo, Suhyeok Jang, Taeyoung Kim, Beomjun Kim, Byungjun Yoon, Changsung Jang, Daewon Choi, Dongsu Han, et al. Rldx-1 technical report. *arXiv preprint arXiv:2605.03269*, 2026a.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint*, 2024.

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

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, et al. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026.

Wenhao Li, Xiangfeng Wang, Bo Jin, and Hongyuan Zha. Hierarchical diffusion for offline decision making. *Proceedings of Machine Learning Research*, 202:19425–19439, 2023.

Weixin Liang, Lili Yu, Liang Luo, Srinivasan Iyer, Ning Dong, Chunting Zhou, Gargi Ghosh, Mike Lewis, Wen-tau Yih, Luke Zettlemoyer, et al. Mixture-of-transformers: A sparse and scalable architecture for multi-modal foundation models. *arXiv preprint arXiv:2411.04996*, 2024.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: benchmarking knowledge transfer for lifelong robot learning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *NeurIPS*, 2023.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint*, 2024.

Qian Long, Yueze Wang, Jiaxi Song, Junbo Zhang, Peiyan Li, Wenxuan Wang, Yuqi Wang, Haoyang Li, Shaoxuan Xie, Guocai Yao, et al. Scaling world model for hierarchical manipulation policies. *arXiv preprint arXiv:2602.10983*, 2026.

Hao Luo, Yicheng Feng, Wanpeng Zhang, Sipeng Zheng, Ye Wang, Haoqi Yuan, Jiazheng Liu, Chaoyi Xu, Qin Jin, and Zongqing Lu. Being-h0: Vision-language-action pretraining from large-scale human videos. In *International Conference on Machine Learning*. PMLR, 2026a.

Hao Luo, Ye Wang, Wanpeng Zhang, Sipeng Zheng, Ziheng Xi, Chaoyi Xu, Haiweng Xu, Haoqi Yuan, Chi Zhang, Yiqing Wang, et al. Being-h0. 5: Scaling human-centric robot learning for cross-embodiment generalization. *arXiv preprint arXiv:2601.12993*, 2026b.

Yunhao Luo, Utkarsh Mishra, Yilun Du, and Danfei Xu. Generative trajectory stitching through diffusion composition. *Advances in Neural Information Processing Systems*, 38:37809–37843, 2026c.

Yueen Ma, Zixing Song, Yuzheng Zhuang, Jianye Hao, and Irwin King. A survey on vision-language-action models for embodied ai. *IEEE Transactions on Neural Networks and Learning Systems*, 2026. doi: 10.1109/TNNLS.2025.3650584.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. *arXiv preprint arXiv:2406.02523*, 2024.

Cyrus Neary, Omar G. Younis, Artur Kuramshin, Ozgur Aslan, and Glen Berseth. Improving pre-trained vision-language-action policies with model-based search. *arXiv preprint arXiv:2508.12211*, 2025.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903. IEEE, 2024.

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

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint*, 2025.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 2020.

Quanxin Shou, Fangqi Zhu, Shawn Chen, Puxin Yan, Zhengyang Yan, Yikun Miao, Xiaoyi Pang, Zicong Hong, Ruikai Shi, Hao Huang, et al. Halo: A unified vision-language-action model for embodied multimodal chain-of-thought reasoning. *arXiv preprint arXiv:2602.21157*, 2026.

Jingwen Sun, Wenyao Zhang, Zekun Qi, Shaojie Ren, Zezhi Liu, Hanxin Zhu, Guangzhong Sun, Xin Jin, and Zhibo Chen. Vla-jepa: Enhancing vision-language-action model with latent world model. *arXiv preprint arXiv:2602.10098*, 2026.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint*, 2024.

Xiaomi Robotics Team, Jun Guo, Piaopiao Jin, Jason Li, Peiyan Li, Yingyan Li, Futeng Liu, Wanli Peng, Optimus Qin, Yifei Su, et al. Xiaomi-robotics-1: Scaling vision-language-action models with over 100k hours of real-world trajectories. *arXiv preprint arXiv:2607.15330*, 2026.

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.

Siyin Wang, Junhao Shi, Zhaoyang Fu, Xinzhe He, Feihong Liu, Chenchen Yang, Yikang Zhou, Zhaoye Fei, Jingjing Gong, Jinlan Fu, et al. World action models: The next frontier in embodied ai. *arXiv preprint arXiv:2605.12090*, 2026.

World Agents. WorldDreamer: Robocasa365 leaderboard submission. `https://github.com/robocasa-benchmark/leaderboard/blob/main/submissions_md/worlddreamer_2026_06_20.md`, June 2026. Evaluated on RoboCasa365 v1.0.1, June 20, 2026.

Zhuoyuan Wu and Jun Gao. Oscar: Omni-embodiment action-conditioned world model for robotics. *arXiv e-prints*, pp. arXiv–2606, 2026.

Xinyi Xie, Zican Hu, Zhanyu Liu, Yicheng Dong, Wenhao Wu, Zhenhong Sun, Haoran Li, Chunlin Chen, Zhi Wang, and Pichao Wang. Look before you leap: Distilling tree search into action evaluation for frozen vla models. *arXiv preprint arXiv:2607.03751*, 2026.

Yandan Yang, Shuang Zeng, Tong Lin, Xinyuan Chang, Dekang Qi, Junjin Xiao, Haoyun Liu, Ronghan Chen, Yuzhi Chen, Dongjie Huo, et al. Abot-m0: Vla foundation model for robotic manipulation with action manifold learning. *arXiv preprint arXiv:2602.11236*, 2026.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint*, 2024.

Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Hao Li, Hengtao Li, Jie Li, Jindi Lv, Jingyu Liu, et al. Gigaworld-policy: An efficient action-centered world–action model. *arXiv preprint arXiv:2603.17240*, 2026a.

Jinhui Ye, Ning Gao, Senqiao Yang, Jinliang Zheng, Zixuan Wang, Yuxin Chen, Pengguang Chen, Yilun Chen, Shu Liu, and Jiaya Jia. Starvla-α: Reducing complexity in vision-language-action systems. *arXiv preprint arXiv:2604.11757*, 2026b.

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

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

Jianke Zhang, Yanjiang Guo, Yucheng Hu, Xiaoyu Chen, Xiang Zhu, and Jianyu Chen. Up-vla: A unified understanding and prediction model for embodied agent. *arXiv preprint*, 2025.

Yuyang Zhang, Wenyao Zhang, Zekun Qi, He Zhang, Haitao Lin, Jingbo Zhang, Yao Mu, Xiaokang Yang, Wenjun Zeng, and Xin Jin. Imagewam: Do world action models really need video generation, or just image editing? *arXiv preprint arXiv:2606.19531*, 2026.

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 T. 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 *CoRL*, 2023.

APPENDIX OVERVIEW

A Pilot Experiment **16**

B Video-only Pretrained Expert **18**

C Simulation Experiments **19**

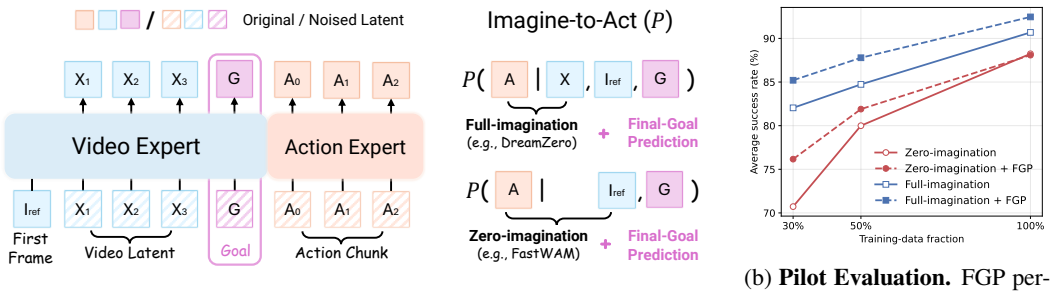
D Real Robotic Experiment **24**

E Simple ABC Experiment **26**

F Limitations and Future Work **27**

In this supplementary document, we provide pilot study analyses, additional experimental details and analysis, failure cases, and extended discussions referenced in the main text.

A PILOT EXPERIMENT



(a) **The joint WAM architecture augmented with FGP.**

(b) **Pilot Evaluation.** FGP performance across three datasets.

Figure 6: **(a) Standard WAMs instantiate the *imagine-then-act* formulation via two effective regimes**: *full-imagination* (e.g., DreamZero (Ye et al., 2026c)), which conditions on the full video trajectory X, and *zero-imagination* (e.g., FastWAM (Yuan et al., 2026)), which conditions solely on the first frame (reference image) $\boldsymbol{I}_{\text{ref}}$. We incorporate FGP into both regimes by co-denoising a task-level final goal, enabling action tokens to attend directly to it. **(b) Success rates across training-data fractions** ($\{0.3, 0.5, 1.0\}$): FGP improves performance in most evaluated settings, with larger gains under reduced training-data regimes.

Backbone and Model Variants This section documents the full configuration behind Figure 6b. All variants are trained and evaluated with the FastWAM codebase, and all four configurations share an identical WAM setting from (Yuan et al., 2026): a MoT that jointly models future video and robot actions on a common flow-matching diffusion backbone built on Wan2.2-TI2V-5B, with a Wan2.1-T2V-1.3B text tokenizer. Each training window contains 33 observation frames, encoded by the Wan VAE into a latent video sequence, together with an action chunk of 32 steps (an action-to-video frequency ratio of 4, i.e. 9 video latent frames per window). Both experts are trained with flow matching (logit-normal timestep shift 5.0, 1000 train timesteps) and share the text conditioning (T5 embeddings, precomputed and cached per task). Language is the only task conditioning; no end-effector or goal image is provided as input unless stated otherwise. The variants below are identical in weights, data, and optimization and differ only in *how the action expert is coupled to the video* (its cross-attention structure and denoising schedule) and in whether an auxiliary goal-prediction target is added. For two baselines, we adopt $\mathcal{L}_{\text{video}}$ and $\mathcal{L}_{\text{act}}$ together during the training, and for FDP, we further add $\mathcal{L}_{\text{sub}}$ with setting $k = 1$ and the sub-goal is treated as the final frame of each task. For inference, here are the differences:

- **Zero-imagination Paradigm.** The model denoises video and actions jointly in a single pass. Action tokens cross-attend directly to the initial observation (the first-frame video latents) without relying on imagined future video, generating actions and video in lockstep.
- **Full-imagination Paradigm.** The model decouples video and action generation into a two-stage pipeline. At inference, Stage 1 first generates the future video trajectory from noise, and Stage 2 infers the action chunk conditioned on this generated video via inverse dynamics.
- **FGP.** Both paradigm can be augmented with a dedicated final-goal prediction. At inference, the goal and actions are jointly denoised from noise, allowing action tokens to co-attend to the evolving goal representation with first frame, explicitly guiding the policy generation.

Benchmarks and Observation Spaces. We evaluate our method across three simulation benchmarks spanning single-arm, bimanual, and long-horizon kitchen manipulation: LIBERO, RoboTwin, and RoboCasa (Nasiriany et al., 2024). Table 5 summarizes their respective observation and action spaces. For our pilot validation, we evaluate on a representative subset of five RoboCasa tasks (*CloseFridge*, *OpenCabinet*, *OpenDrawer*, *TurnOnMicrowave*, and *TurnOnSinkFaucet*). All benchmarks share a unified configuration with 33 observation frames, an action-to-video ratio of 4, and a maximum text context length of 128, differing only in camera count, video resolution, and action/state dimensionality as listed in Table 5, where we mostly follow Yuan et al. (2026).

Table 5: **Benchmark observation and action spaces.** All use a 33-frame window and language conditioning. "Concat" = per-camera frames resized then tiled into a single video canvas.

Benchmark	Tasks	Cameras	Video (Concat)	Action	State
LIBERO (4 suites)	4×10	2 (agent, wrist)	224×448 (horiz.)	7-D	8-D
RoboTwin 2.0	50	3 (head, L/R wrist)	384×320	14-D	14-D
RoboCasa (5-task)	5	3 (L/R agent, eye-in-hand)	384×320	12-D	16-D

Training Configuration. The data-ratio sweep sub-samples the *training* set. Given a ratio $R \in \{0.3, 0.5, 1.0\}$, we deterministically shuffle the training episodes with a fixed seed and retain the first R fraction. The resulting subsets are therefore nested ($\mathcal{D}_{0.3} \subset \mathcal{D}_{0.5} \subset \mathcal{D}_{1.0}$), so larger data budgets are strict supersets of smaller ones with no samples dropped. All runs share the optimization hyperparameters. LIBERO-based models are trained on 8 GPUs, while RoboTwin and RoboCasa use up to 64 GPUs. With fixed epochs, total optimizer steps scale linearly with the data ratio.

Unifying Distal Planning and Control via Native Goal Prediction. Existing WAMs differ fundamentally in *how much generated future context conditions the action expert*, denoted by $\mathcal{C}(\mathbf{X})$. This spectrum spans from *full-imagination* models that condition on full-horizon latent clips ($\mathcal{C}(\mathbf{X}) = \mathbf{X}_{1:f}$) to leverage dynamic priors (Black et al., 2025; Ye et al., 2026c), down to *zero-imagination* baselines that bypass visual generation entirely during inference ($\mathcal{C}(\mathbf{X}) = \emptyset$) (Yuan et al., 2026). However, regardless of the choice of $\mathcal{C}(\mathbf{X})$, control is ultimately anchored on a compact, short-horizon rollout (video clip $\mathbf{V}$) starting from the current observation. While such short-sighted conditioning captures local dynamics effectively, it provides minimal long horizon guidance toward the ultimate destination, constraining sample efficiency and generalization under data scarcity. While hierarchical VLAs attempt to mitigate this myopia by training dedicated goal experts (Ai et al., 2026; Shou et al., 2026; Long et al., 2026), as a generative model over future latents, we show that WAM natively handles long horizon planning without extra overhead, enabling the video expert to synthesize distal targets that guide the action expert with long-horizon context.

Table 6: **Training hyperparameters for the pilot experiments.**

Hyperparameter	Value
Optimizer	AdamW (β_1=0.9, β_2=0.95)
Peak learning rate	1×10^{-4}
LR schedule	cosine
Epochs	10-LIBERO, 5-RoboTwin, 20-RoboCasa
Global batch size	128-LIBERO, 1024-(RoboTwin, RoboCasa)
Gradient accumulation	1
Weight decay	1×10^{-2}
Max gradient norm	1.0
Flow-matching timesteps	1000 (logit-normal shift 5.0)
Loss weights	λ_{act}=1.0, λ_{sub}=1.0 (FGP)

Table 7: **Pilot evaluation of FGP.** Average SR (%) under (30%, 50%, 100%) training-data regimes.

Method	LIBERO	RoboTwin	RoboCasa (5 Tasks)
	30% / 50% / 100%	30% / 50% / 100%	30% / 50% / 100%
Zero-Imagination	82.35 / 90.95 / **97.40**	79.82 / 88.64 / **91.71**	50.00 / 60.40 / 75.60
Zero-Imagination + FGP	**88.72** / **94.05** / 97.00	**82.57** / **89.21** / 91.25	**57.20** / **62.43** / **76.00**
Full-Imagination	93.55 / 95.75 / 98.25	84.18 / 88.44 / 91.41	68.40 / 70.00 / 82.40
Full-Imagination + FGP	**94.50** / **97.85** / **98.65**	**87.11** / **89.10** / **91.54**	**74.00** / **76.40** / **87.20**

B VIDEO-ONLY PRETRAINED EXPERT

Data Recipe. The goal of this stage is to endow the policy with a strong, embodiment-aligned visual dynamics prior before any action learning, so that the subsequent action head can be grounded in an already-capable video predictor rather than learned from scratch. To this end, ProWAM is obtained through a *video-only* pretraining stage that adapts a large-scale video prior to the target robot embodiment before any action learning. Similar to training strategy in Luo et al. (2026a); Wu & Gao (2026), we build on the Wan2.2-TI2V-5B video diffusion transformer, initialized from a checkpoint pretrained on large-scale robot video data, a mixture of DROID (Ye et al., 2026c), AgiBott-Beta (Bu et al., 2025a) filtered from (Wu & Gao, 2026), and 57 tasks from OpenX-Embodiment (O'Neill et al., 2024) with a ratio of $60\%/30\%/10\%$. This endows general-purpose Wan with a task-level sub-goal prior for robotic manipulation. To monitor generalization we hold out a fixed, seed-deterministic set of 400 episodes (200 DROID + 200 AgiBot), excluded from training, and evaluate the forward-only flow-matching loss on them every 500 steps.

Resolution and Multi-View Composition. Each training clip contains 25 frames rendered at 512×512. Because the source datasets differ in the number of available camera views, we render every episode into a *single composite frame* whose layout is selected automatically from the dataset's native view count (depth streams excluded, and capped at four views): a single view fills the whole frame; two views are placed left|right (each $\frac{W}{2} \times H$); three views use a top-1/bottom-2 layout (view 0 spanning the top half at $W \times \frac{H}{2}$, the other two tiled below, each $\frac{W}{2} \times \frac{H}{2}$); and four views form a 2×2 grid (each cell $\frac{W}{2} \times \frac{H}{2}$). Each view is letterboxed into its cell with aspect ratio preserved (rather than stretched), avoiding spatial distortion. This yields a fixed-size observation interface that handles single- and multi-view episodes uniformly, and lets a single model learn across heterogeneous camera rigs without per-dataset architectural changes.

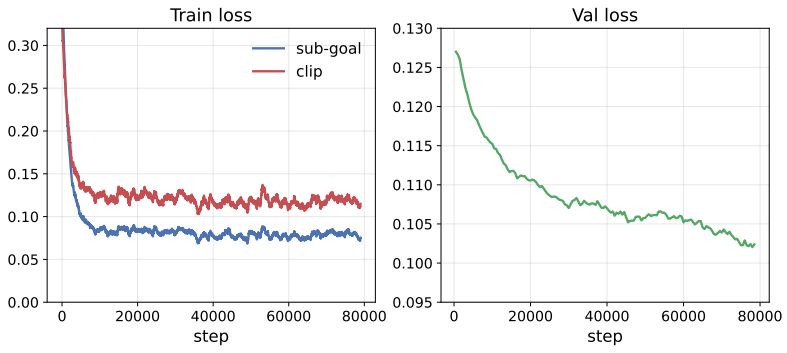


Figure 7: **Pre-training curves for the sub-goal video expert.** Both training losses (sub-goal and clip) and the validation (Val) loss decrease overall throughout pre-training.

Video-only sub-goal objective. At this stage no action head is attached, so the model is trained purely as a flow-matching video predictor. The key ingredient is *progressive sub-goal conditioning*: for each clip we sample up to $K{=}8$ future sub-goal frames from the same trajectory at relative progress positions (via coverage sampling, which spreads sub-goals across the horizon rather than

clustering them), and condition the predictor on them jointly with the current observation. Sub-goals are injected with a zeroed rotary position and referenced relative to the first frame, and the clip loss is computed bidirectionally. The sub-goal and clip losses are weighted equally, both set to 1. Training the video model to forecast frames toward sampled sub-goals is precisely what instills the task-level sub-goal prior that ProWAM later reuses for action prediction. We set $\lambda_{\text{video}} = \lambda_{\text{sub}} = 1$.

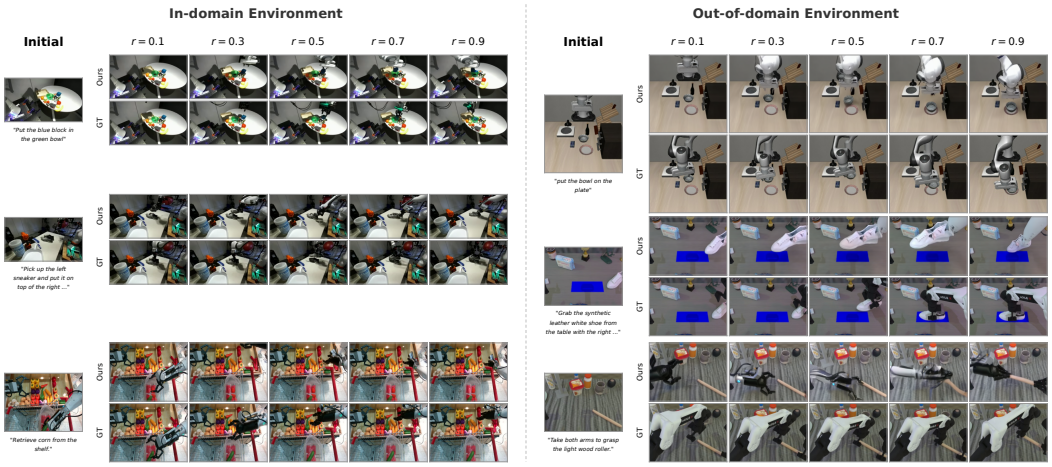


Figure 8: **Visualized sub-goal prediction of our pretrained video expert**. Each example shows the initial frame (Initial) and predicted sub-goals at five progress values $r \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, with our prediction (Ours, top) above the ground truth (GT, bottom). Left: in-domain training embodiments (DROID, AgiBot). Right: out-of-domain environments unseen during pre-training (LIBERO, RoboTwin). The selected examples qualitatively illustrate transfer to environments not used during video pretraining.

Optimization. We train only the diffusion transformer (the T5 text encoder and the VAE are kept frozen), on 128 NVIDIA H100 GPUs using DeepSpeed ZeRO-2 sharding with bf16 mixed precision. We use AdamW (β_1=0.9, β_2=0.999, weight decay 10^{-2}) with a constant learning rate of 1×10^{-5} and a global batch size of 256 for 8 epochs ($\sim$79k steps, i.e. roughly 2×10^7 video clips seen).

Loss Analysis. Figure 7 summarizes the pretraining dynamics. Both the clip and sub-goal training losses decrease rapidly during the first $\sim$10,000 steps before approaching stable plateaus. Over the reported 79k-step schedule, the held-out flow-matching loss decreases overall from 0.127 to 0.102, with no sustained increase. We therefore use the final checkpoint for downstream fine-tuning.

Sub-goal Generation. As shown in Figure 8, the pretrained video expert predicts sub-goals that advance coherently with the progress value r. As r increases from 0.1 to 0.9, the generated frames depict the manipulation progressing toward task completion, such as approaching, grasping, and transporting the target object, and qualitatively align with the corresponding ground-truth frames on the training embodiments. Similar behavior is observed on LIBERO and RoboTwin, which are not included during pretraining: the predicted sub-goals remain instruction-aligned and capture major object interactions and arm motions. These qualitative results suggest that progress-conditioned sub-goal prediction transfers to unseen embodiments and scenes, providing useful visual context for the downstream action expert.

C Simulation Experiments

Training. During downstream adaptation, all the simulation benchmarks adopt a two-stage training paradigm on top of the pretrained sub-goal video expert. **Stage 1 (Cold-start)** continues training only the video expert or base Wan on the task-specific downstream video data (no action head), keeping the progressive-sub-goal conditioning, so the model adapts to the target visual domain while preserving its learned future-prediction prior. **Stage 2 (Joint SFT)** attaches the action DiT and co-trains video and action with a mixed-attention coupling, where the action tokens attend to the current

frame and the predicted sub-goal(s). We set $\lambda_{\text{video}} = \lambda_{\text{sub}} = \lambda_{\text{act}} = 1$. The simulation benchmarks share this recipe and differ only in data and clip length, all hyper-parameters are listed in Table 8-10:

- **LIBERO & LIBERO-Plus.** Standard LIBERO (Liu et al., 2023) consists of four single-arm tabletop manipulation suites (Spatial, Object, Goal, and Long), each featuring 10 tasks and 500 expert demonstrations. We train our policy across all four standard suites and report the *average success rate* (ASR) over 50 evaluation trials per task, yielding a total of 2000 trials across 40 tasks. LIBERO-Plus (Fei et al., 2025) introduces a more challenging setting built upon the original LIBERO tasks, incorporating diverse perturbations (camera, robot, language, lighting, background, sensor noise, and layout). We use only the original LIBERO demonstrations for training and do not incorporate any LIBERO-Plus data, which is used exclusively for zero-shot generalization evaluation. We evaluate under the LIBERO-Plus protocol with ASR. We use all four LIBERO suites, with the original 50 demonstrations per task. Observations are the agent-view and wrist cameras composited side-by-side at 224×448. Stage 1 performs video-only adaptation from the pretrained expert for 9 epochs, the resulting checkpoint initializes the video expert for Stage 2, which co-trains the action head for another 10 epochs. Unless otherwise noted, our best model uses a single sub-goal (k=1) at progress r=0.7.

- **RoboTwin.** RoboTwin (Chen et al., 2025) is a large-scale simulated benchmark for bimanual robot manipulation, which covers 50 tasks and requires policies to coordinate under diverse object layouts and scene conditions. Specifically, it provides 2,500 trajectories collected in clean scenes and 25,000 trajectories collected with heavy scene randomization. In contrast to prior work that trains on both (Yuan et al., 2026; Zhang et al., 2026; Li et al., 2026), we train our policy exclusively on the 2,500 clean-scene demonstrations (Clean) and treat the randomized scenes (Random) as an out-of-distribution test to scrutinize the model's cross-domain generalization, reporting ASR over 100 trials per task. We use all 50 RoboTwin tasks under the multi-task setting, training on the 50 clean demonstrations per task. The randomized split evaluates control generalization without randomized-scene action supervision. Following (Chen et al., 2026a; Hu et al., 2026), the video expert is trained with action-free videos from both clean and randomized scenes, whereas vision-action fine-tuning uses only clean demonstrations. The randomized split therefore evaluates whether visual knowledge learned without action labels can support action generation under substantial scene variation. Observations are the head and two wrist cameras composited into a single frame at 384×320 (head on the top two-thirds, left/right wrist on the bottom). We follow the same two-stage recipe as LIBERO, with three differences: (i) a three-camera RoboTwin composite instead of two, (ii) a small cold-start (RoboTwin is roughly $24\times$ larger than LIBERO, so video-only adaptation runs about 1 epoch), and (iii) the best sub-goal schedule is near-biased with a higher k (K=4, progress $0.1, 0.2, 0.35, 0.5$). Stage 2 co-trains the action head for 5 epochs on the clean split with the same differential learning rate.

- **RoboCasa365.** RoboCasa365 (Nasiriany et al., 2024) is a large-scale benchmark for generalist robot policies, spanning 365 everyday manipulation tasks across 2,500 diverse kitchen environments. Following the official multi-task protocol, policies are trained on the Human300 dataset, which contains human-teleoperated demonstrations for 300 tasks, and evaluated in closed loop on 50 target tasks. The evaluation comprises 18 *Atomic-Seen*, 16 *Composite-Seen*, and 16 *Composite-Unseen* tasks. The first two splits contain tasks observed during pretraining, whereas *Composite-Unseen* evaluates zero-shot generalization to held-out task compositions. We report success rates for each split and the mean success rate over all 50 tasks as the overall score. We follow the same two-stage recipe used for LIBERO (Liu et al., 2023) and RoboTwin (Chen et al., 2025). As shown in Table 10, in Stage 1, we pretrain the video expert on the RoboCasa demonstrations with the progressive sub-goal objective; in Stage 2, we jointly fine-tune the video expert and the action expert. The three simulated views (one top, two bottom) are composited into a single 448×448 image (top view across the upper region, wrist/side views below). Each clip spans 49 raw frames sampled at stride 3 (17 latent frames). Actions ($d_a = 12$) and proprioception ($d_p = 16$) are normalized using dataset statistics, and the action expert predicts a horizon of $H = 32$. Sub-goals use a progressive-K schedule with $K_{\max} = 6$ and progress levels $r \in \{0.3, 0.4, 0.5, 0.65, 0.8, 0.9\}$. We train at a global batch size of 1024 with a cosine schedule, learning rate 1×10^{-4} for the action expert and 5×10^{-5} for the video expert. We run the official RoboCasa closed-loop evaluator on three splits: *atomic-seen* (18 tasks), *composite-seen* (16 tasks), and *composite-unseen* (16 tasks), with 25 independent rollouts per task (450/400/400

trials respectively). At inference, the video expert is evaluated once per replanning step to obtain and cache the sub-goal key-value features. The action expert then reuses this cache over 10 action-denoising steps. The cache is refreshed when a new observation triggers the next replanning step, and executes the *full* predicted horizon before re-planning (H=32, replan every 32 steps). We report the per-task success rate averaged over trials for each split.

Efficient Inference. A naive world-action policy denoises the imagined sub-goal and action jointly for the full T=10 diffusion steps, invoking the video expert at every step. We find that this repeated video-expert computation is unnecessary in our experiments. We first inspect the generated sub-goals after different numbers of denoising steps (Fig. 9). Additional denoising improves visual detail, consistent with observations in FastWAM (Yuan et al., 2026) and ImageWAM (Zhang et al., 2026) that photorealistic predictions are not required to condition the policy.

We therefore decouple the two branches at inference. At each replanning step, the video expert performs a single forward pass to predict the sub-goals and cache their key–value features. The action head then reuses this fixed cache throughout T action-denoising steps, with no additional video-expert forwards. This reduces video-expert evaluations from T to 1 ($\approx 10\times$ fewer evaluations), yielding an inference cost comparable to FastWAM and ImageWAM. As shown in Table 12, KV-1 and Joint-10 achieve the same observed OOD success rate (85.8%), while KV-1 reduces computation by 86.7%. This suggests that sub-goal features from a single video-expert evaluation provide sufficient conditioning for action denoising in this setting.

Beyond Explicit Sub-goal Conditioning To test whether the action head can rely on the visual knowledge encoded by the pretrained video backbone without explicit sub-goal conditioning, we introduce **ProWAM-Light**, inspired by FastWAM's fast-inference setting. ProWAM-Light masks sub-goal tokens during both training and inference, such that the action head conditions only on the current observation. At inference, a single video-expert forward pass constructs a current-observation key–value cache that is reused throughout action denoising. As shown in Table 12, ProWAM-Light maintains high in-domain performance, but its OOD success decreases from 85.8% to 80.3% despite only a modest compute reduction (15.1 to 13.8 TFLOPs), demonstrating the benefit of explicit sub-goal conditioning for generalization.

Table 8: **Downstream two-stage training hyperparameters (LIBERO).**

	Stage 1 (Cold Start)	**Stage 2 (Joint SFT)**
Trainable	video expert	video expert + action head
Action horizon	–	16
Subgoals (k)	Random $k \sim \mathcal{U}\{1, \ldots, 8\}$	$k = 1$ ($\{0.7\}$)
Learning rate	1×10^{-4}	1×10^{-4} (action), 5×10^{-5} (video)
Scheduler	constant	cosine, 5% warm-up
Global batch	64	128
Optimizer	AdamW ($\beta_2 = 0.95$, weight decay 10^{-2})	
Grad clip	1.0	
Epochs	10	

Table 9: **Downstream two-stage training hyperparameters (RoboTwin).**

	Stage 1 (Cold Start)	**Stage 2 (Joint SFT)**
Trainable	video expert	video expert + action head
Action horizon	–	16
Subgoals (k)	Random $k \sim \mathcal{U}\{1, \ldots, 8\}$	$k = 4$ ($\{0.1, 0.2, 0.35, 0.5\}$)
Learning rate	5×10^{-5}	1×10^{-4} (action), 5×10^{-5} (video)
Scheduler	constant	cosine, 5% warm-up
Epochs	1	5
Optimizer	AdamW ($\beta_2 = 0.95$, wd 10^{-2})	
Grad clip	1.0	
Global batch	1024	

Table 10: **Downstream two-stage training hyperparameters (RoboCasa365).**

	Stage 1 (Cold Start)	**Stage 2 (Joint SFT)**
Trainable	video expert	video expert + action head
Action horizon	–	32
Subgoals (k)	Random $K \sim \mathcal{U}\{1, \ldots, 8\}$	$k = 6$ ($\{0.3, 0.4, 0.5, 0.65, 0.8, 0.9\}$)
Learning rate	1×10^{-5}	1×10^{-4} (action), 5×10^{-5} (video)
Scheduler	constant	cosine, no warm-up
Epochs	0.8 (cold start)	1.5
Global batch	64	1024
Resolution	448×448, 3-view composite (top-1 / bottom-2)	
Frames	49 raw @ stride 3 ($\rightarrow$ 17 latent frames)	
Optimizer	AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$, wd 10^{-2})	
Grad clip	1.0	

Table 11: **Closed-loop SR (%) on RoboCasa365**. ProWAM ranks **2nd overall** on the benchmark (updated at 08/15/2026), outperforming most existing policies especially on unseen-composite tasks.

Ranking	Method	Overall	Atomic-Seen	Composite-Seen	Composite-Unseen
1	Xiaomi-Robotics-1 (Team et al., 2026)	57.4	80.2	57.1	32.1
2	**ProWAM (Ours)**	**48.1**	**76.0**	**46.5**	**18.2**
3	ABot-M0.6 (Chen et al., 2026c)	46.6	79.4	48.3	7.9
4	ABot-M0.5 (Chen et al., 2026c)	40.3	75.6	37.7	3.3
5	RLDX-1 (Kim et al., 2026a)	36.0	67.6	27.9	8.5
6	WorldDreamer (World Agents, 2026)	35.3	66.3	26.7	9.0
7	GR00T N1.5 (Bjorck et al., 2025)	23.9	50.7	14.8	2.7
8	GR00T N1.6 (Bjorck et al., 2025)	21.9	51.1	9.4	1.7
9	GigaWorld-Policy 0.1 (Ye et al., 2026a)	20.7	44.4	11.8	2.9
10	$\pi_{0.5}$ (Black et al., 2025)	16.9	39.6	7.1	1.2
11	π_0 (Black et al., 2024)	14.8	34.6	6.1	1.1

Table 12: **Impact of KV-cached inference and explicit sub-goal conditioning on LIBERO.**

Method	Explicit Sub-goal	ID	OOD	TFLOPs
ProWAM (Joint-10)	✓	**99.0**	85.8	113.4
ProWAM (KV-3)	✓	98.8	85.6	20.6
ProWAM (KV-1)	✓	98.8	**85.8**	15.1
ProWAM-Light	×	98.6	80.3	**13.8**

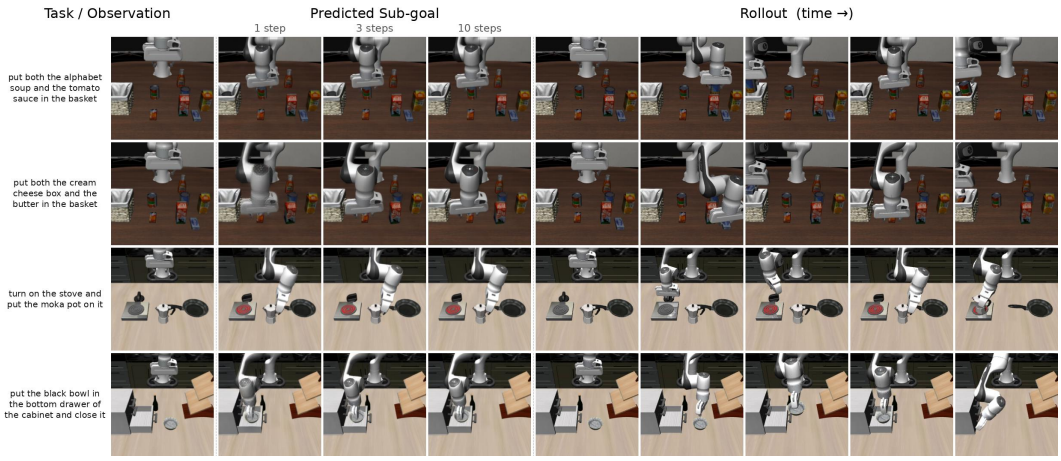


Figure 9: **Qualitative results on LIBERO**. Left: the first frame observation + instruction. Middle: imagined sub-goal after $1/3/10$ denoising steps. Right: executed rollout.

D REAL ROBOTIC EXPERIMENT

Implementation Details. To test whether ProWAM transfers from DROID demonstrations to unseen real-world tasks and scene configurations, we train on DROID (Khazatsky et al., 2024), so every number in this section measures cross-domain generalization rather than in-domain fitting, e.g., LIBERO and RoboTwin, where training and evaluation share a domain. We follow the same two-stage recipe used in those experiments. As shown in Table 13, in Stage 1 we pretrain the video expert on DROID with the progressive sub-goal objective, in Stage 2 we jointly fine-tune the video expert and the action expert under a differential learning rate. The three DROID views (left wrist, two exterior) are composited into a single 576×640 image, with the wrist view spanning the upper two-thirds and the two exterior views tiled below. Each clip spans 33 frames at stride 1. Actions and proprioception are represented in absolute joint space and use identity normalization. At each replanning step, the policy predicts an action chunk of $H = 24$ absolute joint-position targets, executes the first $H_{\text{ol}} = 24$ targets open-loop, and then replans from a new observation. The executed targets are sent directly to the robot's joint-position controller at 15 Hz. Sub-goals follow a balanced near-range ladder with $K_{\max} = 6$: K is drawn uniformly from 1 to 6 and the sub-goal positions for each$K$ are placed on a grid spanning $r \in [0.1, 0.5]$. Other settings are similar to previous ones.

Table 13: **Downstream two-stage training hyperparameters on DROID.**

	Stage 1 (Cold Start)	**Stage 2 (Joint SFT)**
Trainable	video expert	video expert + action head
Action horizon	–	24
Sub-goals (k)	Random $k \sim \mathcal{U}\{1, \ldots, 8\}$	k = 6 ($\{0.1, 0.18, 0.26, 0.34, 0.42, 0.5\}$)
Learning rate	1×10^{-4}	video expert 5×10^{-5} / action head 1×10^{-4}
Epochs	0.5	2
Resolution	576×640, 3-view composite (wrist top / 2 exterior bottom)	
Optimizer	AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$, wd 10^{-2})	
Grad clip	1.0	
Scheduler	cosine, no warm-up	
Global batch	256	

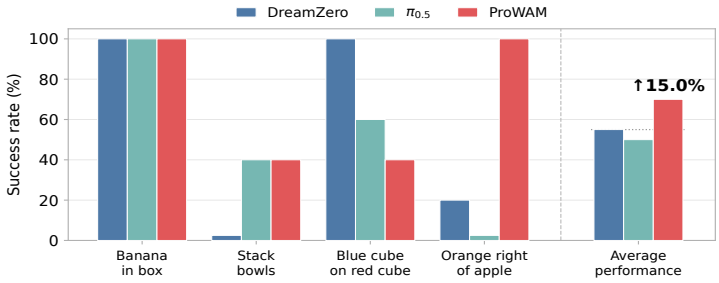
Table 14: **Real-world task suite and success criteria.**

Task	**Instruction**	**Success criterion**
Banana in box	*Put the banana in the box.*	The banana comes to rest inside the box.
Stack bowls	*Stack the bowls.*	All three bowls are nested into a single stack, with no bowl knocked over or displaced from the stack at any point during execution.
Blue cube on red cube	*Put the blue cube on the red cube.*	The blue cube rests unsupported on the red cube. A release at any stage after the grasp that does not land the cube on the red cube counts as a failure.
Orange right of apple	*Put the orange to the right side of the apple.*	The orange is released to the right of the apple within a reasonable offset. Position is scored at the first release location, subsequent corrections do not count.

Real-world Experimental Settings & Analysis. We deploy the policy on a hardware and software setup that mirrors the original DROID setup (Khazatsky et al., 2024). We use a single Franka FR3 arm with a Zed Mini wrist-mounted camera and two Zed 2i exterior cameras. Monocular images used for inference are sourced from the left-rectified sensor from each camera to match the original DROID dataset. We evaluate on a Franka FR3 under randomized initial object configurations, with 5 trials per task. The four tasks are designed independently of the RoboLab suite and span basic



(a) Physical Setup.



(b) **Real-world Experiment Performance.**

Figure 10: **Zero-shot real-world evaluation on Franka FR3. (a) The physical evaluation environment setup:** It features a clean tabletop workspace and 3 recorded camera views. **(b) Success rate across** 4 **physical tasks:** ProWAM achieves the highest average success rate of **70.0%**, outperforming the strongest baseline by an absolute margin of **15.0**.

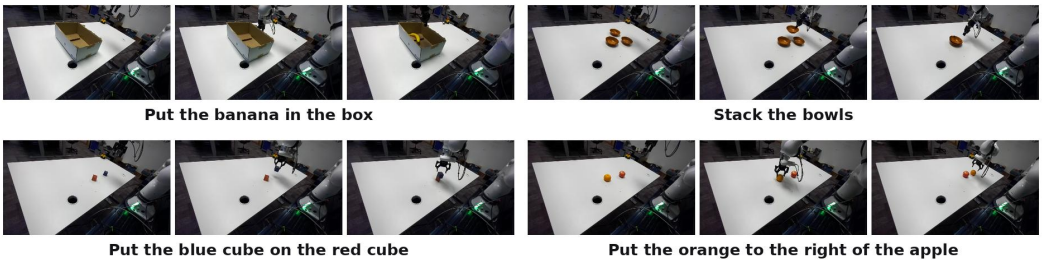


Figure 11: **Real-world physical evaluation tasks.** Multi-camera observations (2 exterior and 1 wrist-mounted camera, only 1 exterior view is shown here to anonymize individuals) across 4 real-world tasks: *Banana in box*, *Stack bowls*, *Blue cube on red cube*, and *Orange right of apple*.

pick-and-place, long-horizon multi-object assembly, contact-rich placement, and spatially grounded placement. All success criteria are fixed prior to evaluation and applied identically to every method, detailed evaluation criterion is in Table 14. Figure 12 illustrates closed-loop visual sub-goal generation in our real-world experiments. Given the current observation, our model produces a temporally ordered visual plan across multiple task-progress levels, which guides subsequent action generation and enables more effective policy execution. While the generated sub-goals place less emphasis on fine-grained textures and photorealistic details, they accurately capture the task-relevant motion, object interactions, and spatial relationships required to complete the manipulation tasks.



Figure 12: **Closed-loop visual sub-goal generation in real-world experiments.** Given observations (t_0, t_1), our model generates progress-conditioned visual sub-goals in a single pass at $r \in \{0.1, 0.26, 0.5\}$. Although visually coarse, the generated sub-goals illustrate task-relevant motion and spatial transitions.

E Simple ABC Experiment

Implementation Details. We additionally evaluate ProWAM on ABC, a simulated bimanual manipulation benchmark that models a realistic dual-arm setup with two 7-DoF arms and head- and wrist-mounted cameras. Its open-source closed-loop evaluation protocol (`bottles-75k`) enables reproducible testing of long-horizon, precision-sensitive manipulation. To this end, we follow a similar training strategy to LIBERO (Liu et al., 2023) and RoboTwin (Chen et al., 2025) for training on ABC. For the bottles task, we train on the dataset's native mixture of teleoperated real-robot ($\approx 43\%$) and simulation ($\approx 57\%$) demonstrations provided by ABC. The three cameras (head + two wrists) are composited into a single 384×320 image (head across the top two-thirds, wrists on the bottom third). Actions and proprioception are z-score normalized using dataset statistics. We use one sub-goal $k{=}1$ at progress-$r{=}0.7$.

Evaluation protocol. We perform the closed-loop evaluation using the official ABC evaluator. Each run covers 20 independent worlds, and each rollout is 120 action chunks of 15 executed steps. We report the strict AVG (all six bottles placed in the bin) and the mean maximum number of bottles placed (out of 6). Both our policy and the official policy are evaluated on the same 20 seed-matched worlds, and both face the same training and evaluation renderer gap. Notably, ABC does not publish a simulation success number in either the paper or the code release, we obtained the reference by running their own checkpoint through the same harness.

Results. Table 15 reports the main comparison. Our policy succeeds in 7 of 20 evaluated worlds, compared with 1 of 20 for the official ABC policy, and achieves a higher mean bottle count (4.85 versus 4.10). The absolute numbers are modest because the strict 6/6 criterion is demanding and a training-to-eval renderer gap depresses all policies. Figure 13 contrasts the two policies on the same evaluation world: our policy places all bottles (bottom row), whereas the official policy stalls with 2

bottles unplaced (top row). The dominant failure mode across policies lies in precision bottlenecks on the *last* bottle, where the end-effector approaches but fails to secure it.

Table 15: Performance on ABC put-bottles-in-bin Task.

Method	ASR	Mean /6 (↑)
ABC official	0.05	4.10
Ours	**0.35**	**4.85**

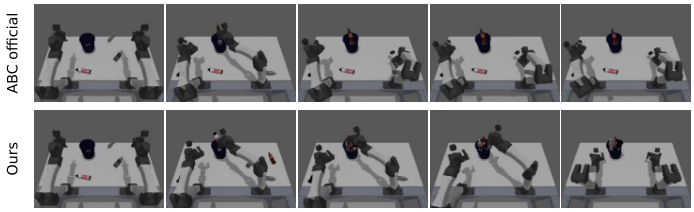


Figure 13: **Visual Comparison between ProWAM and Baseline.** Top shows the baseline policy stalls early, leaving bottles unplaced (4/6). Bottom shows that our policy clears the entire table (6/6).

F Limitations and Future Work

While ProWAM demonstrates strong performance across diverse benchmarks, our study has several limitations that open promising avenues for future work:

- **Manually Specified Inference Schedule.** Although ProWAM supports variable sub-goal counts and progress values, the inference schedule is manually specified rather than inferred from the current task state or complexity. Learning to select the number and placement of sub-goals adaptively is a promising direction for future work.
- **Model and Pre-training Scale:** We build on a moderately sized video expert and a compact action expert. Due to compute constraints, we do not explore how scaling up the backbone size or pre-training corpora impacts performance, which we leave for future exploration.
- **Task Diversity:** Although our evaluation includes several multi-stage tasks, the tested horizons remain limited compared with extended real-world deployments, and evaluation on more dexterous tasks remains an important direction for future work.