PROWORLD: MULTI-VIEW WORLD MODEL WITH PROPRIOCEPTIVE FEEDBACK FOR CLOSED-LOOP POLICY EVALUATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Evaluating robot manipulation policies through repeated physical trials is costly. We study whether an action-conditioned world model can serve as a closed-loop test environment whose policy success rates track those on real robots. Beyond realistic video, the model must return the camera views and proprioceptive state needed for the policy's next action and preserve failed-action outcomes to avoid overestimating policy capability. We introduce ProWorld, a world model that jointly predicts multi-view video and robot state, then feeds both back to a frozen policy for repeated interaction. ProWorld comprises a shared transformer for joint video–state flow prediction and Shared-Time View Encoding to distinguish cameras while preserving temporal synchronization. To reduce interaction hallucinations that turn missed grasps into successful lifts, we augment training with failed interactions collected in high-fidelity simulation, reducing inflated success rates in the generated environment. ProWorld achieves the highest overall TriWorldBench score among the compared methods. Across three policies and four local manipulation tasks, it attains a Pearson correlation of 0.95 and a Spearman correlation of 0.94 between generated and real success rates. These results support joint visual-proprioceptive feedback and failure-aware training as a basis for world model policy evaluation.

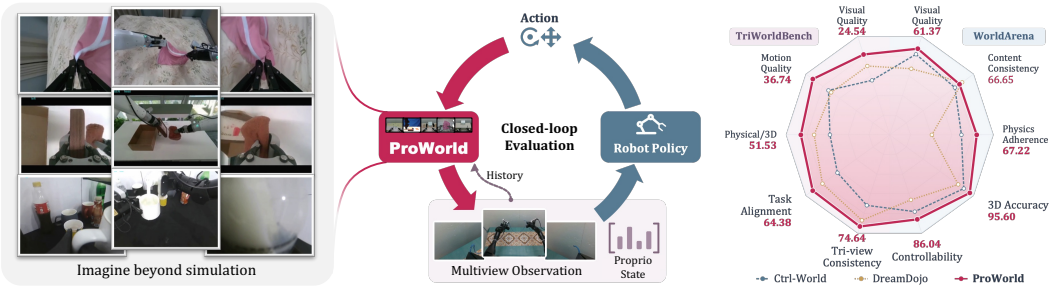


Figure 1: **ProWorld for closed-loop policy evaluation.** Left: ProWorld returns multi-view observations and proprioceptive state to the policy after each action. Right: video-generation performance on TriWorldBench and WorldArena compared with Ctrl-World and DreamDojo.

1 INTRODUCTION

Evaluating robot policies through physical trials is costly, both in terms of robot time and the need for repeated scene resets. With the development of action-conditioned video foundation models (Yang et al., 2024; Zhu et al., 2025; Liao et al., 2025; Gao et al., 2026), recent works have extended these models to policy evaluation (Guo et al., 2025; GigaWorld Team et al., 2026; Jeon et al., 2026; Tseng et al., 2025). These approaches use learned world models as closed-loop evaluation environments: initialized from a real observation, the world model predicts the consequences of actions executed by a fixed policy and generates the observations used for subsequent policy decisions. The goal is to estimate policy success rates through generated closed-loop interactions that faithfully reflect real-robot performance.

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.

However, two requirements that are largely overlooked by existing approaches remain critical for reliable policy evaluation. First, the world model should provide the complete observation space required by the policy, including multiple camera views and proprioceptive states such as joint positions. Second, it must faithfully preserve unsuccessful interactions. For example, if a missed grasp is subsequently rendered as a successful lift, the world model produces an *interaction hallucination*: it predicts a successful outcome that is not supported by the executed action. Such hallucinations can inflate estimated success rates and obscure genuine policy failures. A reliable evaluator therefore requires both complete visual–proprioceptive feedback and faithful modeling of failure outcomes.

We introduce ProWorld, which jointly predicts multi-view videos and robot state from observation history and an action chunk, then feeds both back to a frozen policy. A shared transformer learns both outputs through joint flow matching. State queries read the visual stream through asymmetric attention, encouraging agreement between predicted state and visible motion. The policy thus receives predicted execution feedback rather than its preceding command as the next state.

To preserve camera identity and synchronization, we develop *Shared-Time View Encoding* (STVE). Independently encoded camera streams receive shared time coordinates and explicit view coordinates. This avoids pixel-tiling boundaries and separates camera identity from elapsed time. Compatible with pretrained video backbones, STVE improves visual prediction over pixel tiling and time concatenation methods.

To reduce interaction hallucinations, we augment demonstrations and robot-policy rollouts with failed interactions collected in high-fidelity simulation. Successful demonstrations provide limited evidence of what happens when actions fail. For example, a missed grasp may leave the object in place, while an unintended push may knock it over. Simulated failures supply this missing supervision. In our local evaluation, this training reduces hallucinated successes and preserves failed interactions.

We assess policy-outcome agreement and video prediction quality. Across three policies and four local tasks, ProWorld achieves a Pearson correlation of 0.95 and a Spearman correlation of 0.94 with real success rates. We also evaluate two policies on six RoboChallenge tasks. On TriWorldBench, ProWorld scores 53.68, exceeding the strongest compared baseline's 51.72.

Our contributions are threefold: **(1) Joint video–state prediction** that supplies visual observations and learned proprioceptive feedback for closed-loop policy interaction; **(2) Shared-Time View Encoding**, which preserves camera identity and synchronization while improving visual prediction over pixel tiling and time concatenation; and **(3) Failure-aware training** that uses simulated failures to reduce interaction hallucinations and success-rate inflation.

2 Related Work

Interactive robot world models. Generative video models have progressed from visually plausible rollouts to action-conditioned simulators (Yang et al., 2024; Zhu et al., 2025; Liao et al., 2025; Gao et al., 2026; Guo et al., 2025; Qiu et al., 2026; GigaWorld Team et al., 2026; NVIDIA et al., 2026). Recent systems explicitly use such simulators to assess policies, including scalable video-model evaluation, RoboWorld, and an interactive simulator for long-horizon robot learning (Tseng et al., 2025; Jeon et al., 2026; Wang et al., 2026). DW0.5 further uses generated successes and failures for policy post-training (Dexmal, 2026). Table 1 summarizes these systems. ProWorld combines learned state feedback with calibration-free long-horizon multi-view rollout.

Multi-view, multimodal, and world–action models. Multiple cameras expose complementary evidence about contact and motion. Multi-View Masked World Models learns visual representations from multiple viewpoints (Seo et al., 2023), while recent generative systems study joint multi-view prediction and cross-embodiment synthesis (Guo et al., 2025; Yuan et al., 2026a; Li et al., 2026). Multi-view streams can be organized as pixel canvases (NVIDIA et al., 2026; GigaWorld Team et al., 2026) or concatenated frame sequences (NVIDIA et al., 2025; Bai et al., 2025). Geometry-aware rotary embeddings incorporate camera geometry into attention (Xiang et al., 2026); ProWorld instead assigns camera identity a dedicated rotary coordinate alongside shared time, without requiring calibration (Section 3.3). World–action models (WAMs) use predictive world representations to produce executable actions: DreamZero jointly models future video and actions, and Motus unifies

Table 1: Capabilities of robot world models. ProWorld brings together long-horizon, multi-view generation and proprioceptive feedback without camera calibration or forward kinematics (FK).

Model	Minute-long generation	Multi-view generation	Calibration-free	FK-free	State output	Closed-loop evaluation	Multi-embodiment adaptation
IRASim	×	×	✓	✓	×	×	✓
Ctrl-World	×	✓	✓	×	×	✓	×
DreamDojo	✓	×	✓	✓	×	×	✓
GE-Sim 2.0	✓	✓	×	×	✓	✓	×
GigaWorld-1	✓	✓	×	×	×	✓	✓
DW0.5	×	✓	✓	✓	×	×	✓
Xiaomi-Robotics-U0	×	✓	×	×	×	×	✓
ProWorld (ours)	✓	✓	✓	✓	✓	✓	✓

world modeling, control, and inverse dynamics (Ye et al., 2026; Bi et al., 2025). ProWorld instead supplies the observations through which an external policy is evaluated, including its proprioceptive feedback.

Long-horizon generation and evaluation. Diffusion transformers and flow matching provide the modeling substrate for high-capacity video prediction (Peebles & Xie, 2023; Lipman et al., 2023; Team Wan et al., 2025); autoregressive diffusion work studies how to continue generated history without excessive drift (Chen et al., 2024; Huang et al., 2025; Zheng et al., 2026). Policy evaluation poses a separate measurement question. WorldArena and TriWorldBench evaluate the quality of action-conditioned video predictions (Shang et al., 2026a;b; Liu et al., 2026; Xu et al., 2026), while real-world and simulation benchmarks evaluate robot policies through task success under specified evaluation protocols (Yakefu et al., 2025; Chen et al., 2025a; Jiang et al., 2026).

3 METHOD

3.1 OVERVIEW

We treat ProWorld as the environment in a repeated observe–act loop, with only the initial observation supplied by the real environment. As shown in Figure 2, at decision time t, the policy receives synchronized images $O_t = \{I_t^{(v)}\}_{v=1}^{V}$, proprioceptive state s_t, and a task instruction g. The policy returns an action chunk A_t, and ProWorld predicts the resulting camera observations and robot states over H steps:

$$A_t \sim \pi(\cdot \mid O_{\le t}, s_{\le t}, g), \qquad (\widehat{O}_{t+1:t+H}, \widehat{s}_{t+1:t+H}) \sim p_\theta(\cdot \mid O_{\le t}, s_{\le t}, A_t, g). \tag{1}$$

Here, $O_{\le t}$ and $s_{\le t}$ denote the available observation history; actions and observations are aligned by physical time even when their sampling rates differ. The predicted images and state become the policy's next input, and the loop repeats.

3.2 JOINT VIDEO–STATE PREDICTION

ProWorld jointly predicts visual observations and the resulting robot state using a shared transformer, a joint flow objective, and asymmetric attention. State queries read visual dynamics while both losses train the backbone, providing learned execution feedback for the next policy decision.

Representations and conditioning. A video VAE encodes each view into latent tokens. A lightweight projection embeds the normalized proprioceptive state into the transformer width, and a separate output head predicts its flow field. Text and metadata specify the task, embodiment, camera configuration, and sampling rate. Controls are grouped by their associated latent interval and projected into adaptive layer-normalization parameters. State tokens receive their own noise-level embedding; the control modulation is applied to the video tokens.

For end-effector trajectories, we use a velocity-style representation obtained from smoothed poses at the control rate. Let $\widetilde{r}_k$ contain the continuous pose coordinates and g_k the gripper state. The

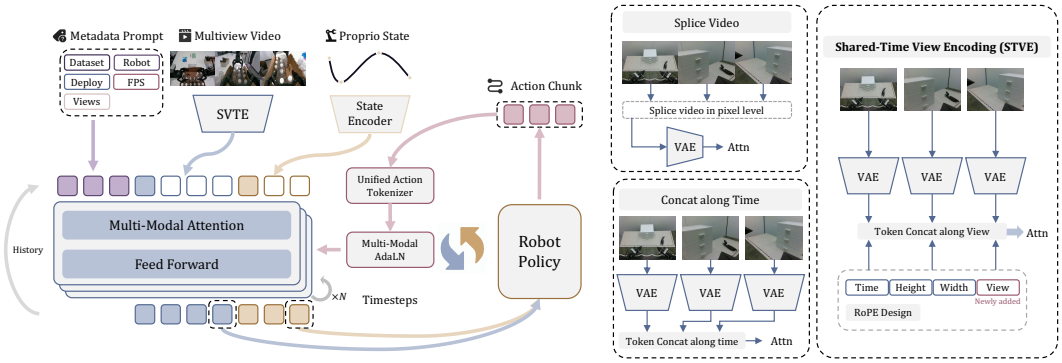


Figure 2: **ProWorld architecture and multi-view organization. Left:** Metadata, independently encoded camera streams, and proprioceptive state enter a shared transformer; action chunks condition visual dynamics through AdaLN, and predicted video and state feed back to the policy. **Right:** Comparison of three multi-view representations. Pixel tiling combines camera images before VAE encoding, while temporal concatenation independently encodes each view and concatenates the resulting tokens along the temporal sequence dimension. STVE instead combines independently encoded tokens along the view dimension and uses separate time, height, width, and view rotary coordinates, preserving shared time and per-view spatial structure.

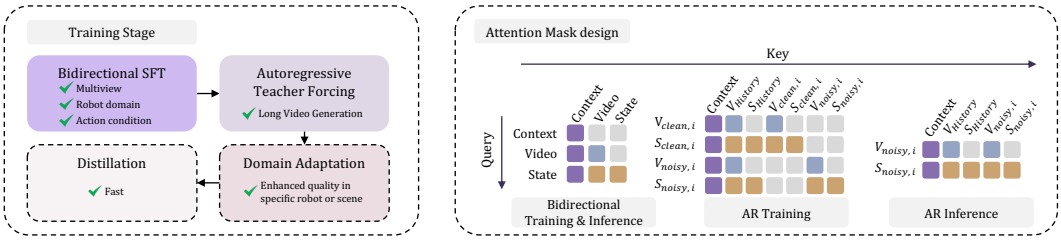


Figure 3: **Training stages and attention visibility.** The training recipe progresses from bidirectional video–state learning to autoregressive teacher forcing, domain adaptation, and distillation. Colored mask entries indicate allowed attention, with queries on rows and keys on columns. Autoregressive training separates clean history from current noisy targets; inference uses generated history.

control feature is

$$u_k = \left[\frac{\widetilde{r}_k - \widetilde{r}_{k-1}}{\Delta t},\ g_k \right], \qquad \bar{u}_k = \mathcal{N}_u(u_k), \qquad \bar{s}_k = \mathcal{N}_s(s_k), \tag{2}$$

where $\mathcal{N}_u$ and $\mathcal{N}_s$ are embodiment-specific normalizers. For joint-controlled deployments, the interface uses the policy's joint-command representation and predicts the corresponding measured joint state.

Joint flow matching. Let x^v denote clean video latents and x^s the normalized state sequence. For modality $m \in \{v, s\}$, we interpolate with Gaussian noise ϵ^m at noise level σ_m:

$$x^m_{\sigma_m} = (1 - \sigma_m)x^m + \sigma_m \epsilon^m, \qquad d^m = \epsilon^m - x^m. \tag{3}$$

The model predicts both flow fields under the same action-conditioned forward pass. We optimize

$$\mathcal{L} = \mathbb{E}\left[w_v(\sigma_v) \| f^v_\theta - d^v \|^2_{\text{mean}} + \lambda_s w_s(\sigma_s) \| f^s_\theta - d^s \|^2_{\text{mean}} \right], \tag{4}$$

with clean context excluded from the prediction loss. Here, λ_s controls the relative weight of state supervision. We use $\lambda_s = 2$ in all experiments with state outputs, except when explicitly varying this weight in the supplementary ablation. Sampling integrates from noise to data using the corresponding modality schedules (Lipman et al., 2023). Both losses update the shared transformer.

Attention structure. Video and state share a backbone but have different visibility. In bidirectional training, context and video tokens attend to each other; state queries read video and state tokens. Video queries do not read the noisy state stream. This makes state prediction a visual readout while retaining joint supervision of the backbone. For autoregressive training, each noisy chunk

reads clean preceding chunks and its own noisy tokens, but not the clean target for that chunk or any future chunk. The state stream also reads preceding states. Figure 3 summarizes these masks. During rollout, denoising reads a fixed history cache; the completed chunk refreshes the cache for the next interaction.

3.3 Shared-Time View Encoding

Figure 2 compares three ways to organize multi-view inputs. **Pixel tiling**, used by Cosmos 3 and GigaWorld-1 (NVIDIA et al., 2026; GigaWorld Team et al., 2026), combines camera images before VAE encoding, reusing the video pipeline but introducing artificial spatial boundaries. **Frame concatenation**, used in Cosmos-Predict2.5's multi-view implementation (NVIDIA et al., 2025) and ReCamMaster's source–target conditioning (Bai et al., 2025), independently encodes each camera stream and concatenates the resulting tokens along the temporal sequence dimension.

STVE instead assigns camera identity its own rotary coordinate (Figure 2, right). Each camera is encoded independently, with tokens indexed by (t, h, w, v): synchronized views share time t, (h, w) remain local to each camera, and v identifies the view. Disjoint rotary subspaces encode these coordinates:

$$\mathcal{R}(t, h, w, v) = \mathrm{diag}(R_t(t), R_h(h), R_w(w), R_v(v)), \qquad \widetilde{q}_i = \mathcal{R}_i q_i, \quad \widetilde{k}_i = \mathcal{R}_i k_i. \tag{5}$$

Joint attention thus distinguishes views without spatial seams or view-dependent time offsets, using the same coordinates for history and generated chunks.

The view coordinate fits within the existing attention-head width: MiniMax-H3 uses its unrotated tail, while Wan reserves a small portion of its spatial rotary channels, preserving the temporal allocation. STVE adds neither tokens nor a separate cross-view attention module. Table 5 compares the three structures in our model.

3.4 Failure Data and Closed-Loop Rollout

Successful demonstrations cover intended behavior extensively but offer fewer examples of policy mistakes. We collect failed trajectories in GenieSim (Yin et al., 2026), a high-fidelity simulator with realistic scenes and physical robot–object interaction, retaining synchronized multi-view observations, commands, and measured states. The observed aftermath of failed actions serves as the prediction target (Figure 4).

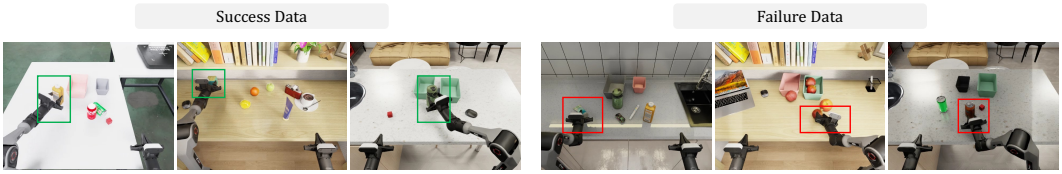


Figure 4: **Success and failure examples in the simulation data.** Successful interactions are shown on the left and failed interactions on the right. Green and red boxes highlight the corresponding robot–object interaction regions.

Under the same prediction objective, demonstrations supply broad manipulation behavior, robot-policy rollouts capture policy-induced execution, and simulated failures expand unsuccessful interaction coverage. Figure 6 compares generated and physical success rates.

Training follows the four stages in Figure 3. Bidirectional supervised fine-tuning learns action-conditioned video and state prediction. Chunk-wise teacher forcing extends the model to longer rollouts, following an established autoregressive training recipe (Zhao et al., 2026a). Domain adaptation specializes it to the target robot and observation setup. Following prior work on autoregressive distillation (Zhao et al., 2026b;a; Yuan et al., 2026b), we distill the model for few-step generation. At evaluation time, a frozen policy repeatedly receives generated images and predicted state in its native units, reacting to estimated execution rather than reusing its preceding command. The supplementary material provides action/state specifications and temporal alignment, failure-data collection details, and stage-specific training configurations.

4 EXPERIMENTS

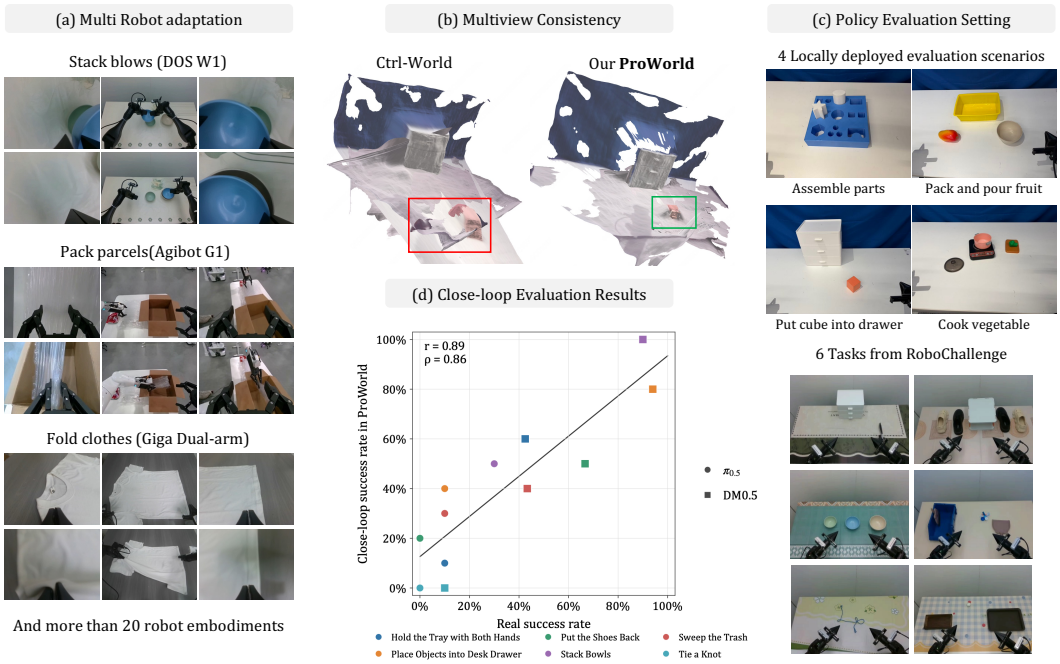


Figure 5: **ProWorld training data and closed-loop evaluation.** (a) Samples from our large-scale multi-embodiment training data. (b) After closed-loop rollout, ProWorld preserves cross-view geometry and reconstructs a consistent scene; Ctrl-World's reconstruction shows a blurred, deformed block. (c) Real-robot evaluation setups and tasks. (d) Generated success rates track physical outcomes on RoboChallenge.

4.1 EXPERIMENTAL SETUP

Training data and implementation. We pretrain on approximately 6,486 hours of heterogeneous manipulation data, combining large open datasets with a smaller private collection. The mixture includes InternData-A1 (264 h) (Tian et al., 2025), RoboCOIN (312 h) (Wu et al., 2025b), ABC-130K (2,420 h) (Allshire et al., 2026), InternData-M1 (156 h) (Chen et al., 2025b), AgiBotWorld-Beta (2,233 h) (AgiBot-World-Contributors et al., 2025), RealOmni-Open (199 h) (Genrobot, 2026), Galaxea Open-World (119 h) (Jiang et al., 2025), and RoboMIND (60 h) (Wu et al., 2025a), together with 71 hours of GenieSim data containing diverse failure trajectories and 652 hours of private data. We use MiniMax-H3 (MiniMax, 2026) and Wan (Team Wan et al., 2025) as video diffusion backbones, with details on their use and training settings provided in the supplementary material.

Evaluation settings. The local closed-loop study uses 3 policies on 4 manipulation tasks, with Ctrl-World and Cosmos3-Nano as alternative rollout environments. A second study evaluates 2 policies on 6 challenging RoboChallenge tasks (Yakefu et al., 2025), covering bimanual handling, placement, sweeping, stacking, and knot tying (Figure 5). For the representation ablations, we initialize models from a pretrained backbone and train them on AgiBotWorld-Alpha (AgiBot-World-Contributors et al., 2025). We evaluate action-conditioned video prediction on TriWorldBench (Liu et al., 2026), comprising 500 three-view trajectories (Table 2), and on WorldArena, comprising 1,000 single-view trajectories.

Evaluation protocol. A frozen policy starts from a real observation, then repeatedly predicts actions from generated images and state. We compare generated and physical success rates using Pearson and Spearman correlation, and paired video quality using PSNR, SSIM, and LPIPS (Zhang et al., 2018). For *video–state consistency error* (VSC), we train DreamGen's inverse-dynamics model (IDM) (Jang et al., 2025) on AgiBotWorld-Alpha and subtract its reference-video position error (approximately 9.1 mm) from the discrepancy between video-inferred and predicted trajecto-

Table 2: Action-conditioned three-view prediction on TriWorldBench. VLM consistency averages the three benchmark VLM scores. Higher is better.

Model	Score	Tri-view		Task	Physical/3D		Motion		Temporal	Visual
	TWB	Norm. PSNR	VLM consist.	Instr. following	Interaction quality	Perspective	State alignment	Trajectory accuracy	Subject consist.	Image quality
DreamDojo	51.72	59.78	79.53	49.72	**28.28**	59.39	60.76	16.61	77.37	24.45
Motus	42.35	60.13	72.82	52.28	25.00	44.20	59.31	13.77	77.37	16.08
Genie Envisioner	40.73	58.74	68.15	20.20	20.10	**87.91**	59.28	0.84	**91.40**	19.36
Ctrl-World	38.98	46.39	64.20	38.90	24.19	45.15	60.47	6.09	77.05	18.37
ProWorld	**53.68**	**62.55**	**84.07**	**62.32**	28.24	74.82	**62.01**	**30.39**	83.86	**26.99**

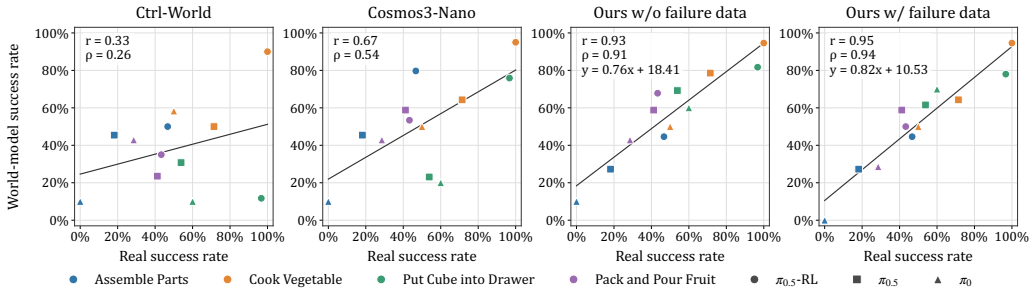


Figure 6: **Policy success in the real and generated worlds.** Colors identify four tasks; shapes identify three policies. Lines are linear fits in percentage units; panels report Pearson r and Spearman ρ. Failure-data training improves outcome agreement.

ries. VSC is reported in millimeters, alongside motion-direction cosine similarity (Dir.) and gripper accuracy (Grip.). Benchmark details appear in the supplementary material.

4.2 Closed-Loop Policy Evaluation

ProWorld's generated success rates closely track real outcomes across 12 local policy–task pairs (Figure 6), with Pearson $r = 0.95$ and Spearman $\rho = 0.94$. Both correlations exceed those of Ctrl-World ($0.33/0.26$) and Cosmos 3 Nano ($0.67/0.54$), indicating stronger agreement in success rates and outcome ordering.

Failure data improves Pearson/Spearman correlation from $0.93/0.91$ to $0.95/0.94$. The fitted relation changes from $y = 0.76x + 18.41$ to $y = 0.82x + 10.53$, reducing overestimation at low real success rates. This supports transferring failure supervision across embodiments.

RoboChallenge tasks often require over half a minute of stable closed-loop execution, with coherent observations throughout object manipulation. ProWorld supports these sustained interactions and preserves success-rate variation across policies and tasks (Figure 5d), extending evaluation beyond short demonstrations of a single successful motion.

4.3 Video Prediction and Multi-View Quality

On TriWorldBench (Table 2), ProWorld achieves the strongest overall score, with gains in cross-view agreement, task following, and motion. Head and wrist observations provide complementary constraints on robot motion and scene interaction.

On the single-view WorldArena benchmark, ProWorld performs especially well on trajectory accuracy, photometric consistency, and semantic alignment (Table 3). Together, these results indicate that motion, persistent scene appearance, and task content survive the generated transition. The supplementary material reports detailed results for both benchmarks and explains the exclusion of frame-rate-sensitive WorldArena motion-quality metrics and EWMScore.

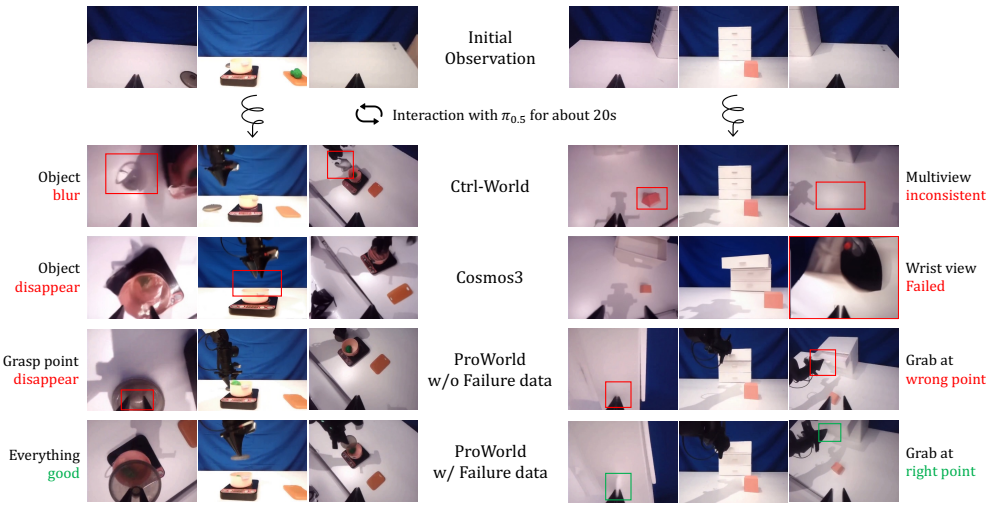


Figure 7: **Closed-loop visual comparison after roughly 20 seconds.** Ctrl-World, Cosmos 3 Nano, and ProWorld with and without failure data start from matched observations. Boxes highlight disappearing objects, misplaced grasps, and improved contact consistency with the full model.

Table 3: Single-view prediction on WorldArena Track 1. Higher is better; bold and underline mark the best and second-best results.

Model	Visual		Consistency		Physics		3D		Control	
	Image quality	JEPA similarity	Subject consist.	Photo. consist.	Interaction quality	Trajectory accuracy	Depth accuracy	Perspective	Instr. following	Semantic alignment
Ctrl-World	42.44	92.77	**83.56**	12.88	62.62	48.20	93.25	83.66	67.68	88.68
GigaWorld-1	51.18	96.77	80.97	<u>28.11</u>	75.10	54.27	**98.44**	95.60	82.14	89.42
ABot-PhysWorld	**61.03**	90.36	80.56	20.52	<u>81.96</u>	31.50	71.99	**98.94**	<u>92.10</u>	<u>89.58</u>
GE-Sim 2.0	46.73	<u>97.16</u>	<u>82.50</u>	7.05	73.60	<u>58.63</u>	96.49	93.64	82.70	89.05
Xiaomi-Robotics-U0	<u>53.94</u>	90.60	79.05	2.13	**87.30**	41.89	85.25	<u>98.84</u>	**93.86**	89.35
ProWorld	45.82	**98.39**	78.91	**35.34**	72.25	**62.19**	<u>98.36</u>	92.85	81.66	**90.41**

What changes during interaction? Figure 7 compares two tasks from matched initial scenes. Cooking requires retaining the pot and its graspable parts; drawer manipulation requires consistent gripper contact across wrist and workspace views. Baselines exhibit object disappearance, blurred structure, and incompatible views. ProWorld preserves these details, giving the policy stable visual context for its next decision.

Without failure data, errors concentrate around contact: handles, grasp points, and gripper–object positions. Failure examples teach the model how scenes evolve when contact is absent or incomplete. In Figure 8, an empty lift leaves the banana behind, and a one-sided grasp produces a one-sided fold. These distinctions lead to different closed-loop task outcomes.

4.4 Ablations

Further ablations of action conditioning, visual context, noise schedules, and training resolution are provided in the supplementary material.

Action representation and joint state learning. In Table 4, velocity-style control improves visual prediction over absolute-state conditioning by directly relating the commanded change to visible motion. Adding state supervision improves the shared visual representation and supplies proprioceptive feedback. Motion-based conditioning with asymmetric attention gives the strongest combination: state queries read the evolving video, while state loss trains the backbone to retain robot-motion information.

Compared with full attention between noisy streams, directed readout lowers VSC and improves motion direction with little change in gripper accuracy, improving continuous motion and visual agreement.

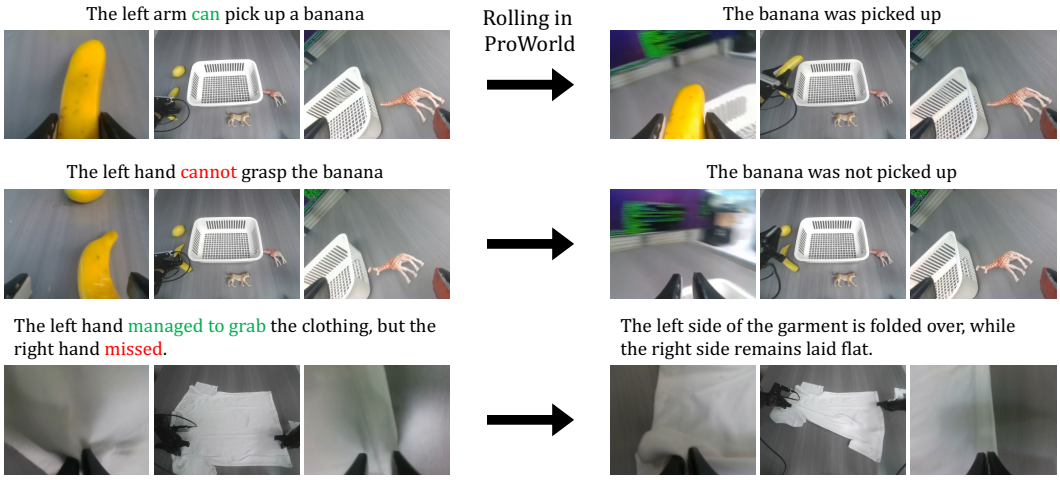


Figure 8: **Generated three-view successes and failures.** Rows: successful and missed banana pickups, then an incomplete cloth fold. Right: ProWorld continuations after the actions at left.

Table 4: Control, prediction target, and attention ablation on AgiBotWorld-Alpha.

		Video quality			State quality		
Control	Targets	PSNR↑	SSIM↑	LPIPS↓	VSC↓	Dir.↑	Grip.↑
Absolute state	Video	21.21	0.798	0.1342	—	—	—
Velocity	Video	22.18	0.819	0.1167	—	—	—
Absolute state	Video + state	21.45	0.806	0.1130	20.5	0.380	**0.976**
Velocity	Video + state (full attn.)	22.47	0.821	**0.1053**	20.3	0.408	0.974
Velocity	Video + state (ours)	**22.60**	**0.829**	0.1069	**14.2**	**0.464**	0.974

Multi-view encoding. Table 5 compares the three structures in Figure 2. Continuous-time concatenation improves all video metrics over pixel tiling. Replacing its view-dependent temporal offsets with STVE further raises PSNR from 21.56 to 22.60 and SSIM from 0.809 to 0.829, and lowers LPIPS from 0.1266 to 0.1069. State results are mixed: STVE improves directional agreement and matches gripper accuracy, but time concatenation has lower VSC. Thus, STVE benefits visual prediction rather than uniformly improving all outputs.

Table 5: Multi-view encoding ablation. State metrics follow Table 4.

	Video quality			State quality		
Encoding	PSNR↑	SSIM↑	LPIPS↓	VSC↓	Dir.↑	Grip.↑
Pixel tiling	17.95	0.619	0.2843	23.2	0.318	0.969
Time concatenation	21.56	0.809	0.1266	**13.3**	0.416	**0.974**
STVE (ours)	**22.60**	**0.829**	**0.1069**	14.2	**0.464**	**0.974**

5 Conclusion and Limitations

ProWorld evaluates robot policies through closed-loop interaction, combining joint visual–proprioceptive feedback, STVE for camera identity and synchronization, and simulated failures to reduce interaction hallucinations and success-rate inflation. Strong correlation with real policy success and competitive video prediction support complete observation feedback and faithful failure outcomes as foundations for policy evaluation.

Object deformation and drift over exceptionally long rollouts remain limitations. Improving geometry persistence and rollout stability could reduce accumulated evaluation errors.

AI USE STATEMENT

Generative AI tools were used to organize, draft parts of, translate, and edit the manuscript; assist with writing code for data generation and collection in simulation; inspect implementation details; and provide feedback on experimental presentation and interpretation of results. AI assistance in constructing the synthetic dataset concerned the simulation data-collection code; the data themselves were collected through simulation. The authors are responsible for verifying all AI-assisted work and for the final text, citations, code, datasets, methods, and reported results.

ETHICS STATEMENT

This work studies action-conditioned video generation to support robotics research and policy evaluation. The use of third-party code, models, and datasets is subject to their respective licenses and usage terms. Generated videos and evaluation results may not fully reflect real-world robot behavior and should not be taken as guarantees of safety or reliability in physical deployment.

REPRODUCIBILITY STATEMENT

We provide implementation details and core implementation code with the supplementary material to facilitate inspection of the proposed method. The complete training and inference pipeline is being prepared for public release and requires further testing and documentation. We will open-source the full codebase and publicly release the failure dataset upon acceptance.

REFERENCES

AgiBot-World-Contributors, Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, et al. AgiBot World Colosseo: A Large-scale Manipulation Platform for Scalable and Intelligent Embodied Systems. *arXiv preprint arXiv:2503.06669*, 2025.

Arthur Allshire, Himanshu Gaurav Singh, Ritvik Singh, Adam Rashid, Hongsuk Choi, et al. Scalable Behavior Cloning with Open Data, Training, and Evaluation. *arXiv preprint arXiv:2606.27375*, 2026.

Jianhong Bai, Menghan Xia, Xiao Fu, Xintao Wang, Lianrui Mu, Jinwen Cao, Zuozhu Liu, Haoji Hu, Xiang Bai, Pengfei Wan, and Di Zhang. ReCamMaster: Camera-Controlled Generative Rendering from A Single Video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14834–14844, 2025.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, et al. Motus: A Unified Latent Action World Model. *arXiv preprint arXiv:2512.13030*, 2025.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion Forcing: Next-token Prediction Meets Full-Sequence Diffusion. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-0759.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, 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*, 2025a.

Xinyi Chen, Yilun Chen, Yanwei Fu, Ning Gao, Jiaya Jia, et al. InternVLA-M1: A Spatially Guided Vision-Language-Action Framework for Generalist Robot Policy. *arXiv preprint arXiv:2510.13778*, 2025b.

Dexmal. DW0.5: A Closed-Loop World Model for VLA. Official technical article, 2026.

Shenyuan Gao, William Liang, Kaiyuan Zheng, Ayaan Malik, Seonghyeon Ye, Sihyun Yu, et al. DreamDojo: A Generalist Robot World Model from Large-Scale Human Videos. *arXiv preprint arXiv:2602.06949*, 2026.

Genrobot. RealOmni-Open Dataset. Official dataset release, 2026.

GigaWorld Team, Angyuan Ma, Boyuan Wang, Bohan Li, Chaojun Ni, Guo Li, et al. GigaWorld-1: A Roadmap to Build World Models for Robot Policy Evaluation. *arXiv preprint arXiv:2607.02642*, 2026.

Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-World: A Controllable Generative World Model for Robot Manipulation. *arXiv preprint arXiv:2510.10125*, 2025.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self Forcing: Bridging the Train-Test Gap in Autoregressive Video Diffusion. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-5576.

Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, et al. DreamGen: Unlocking Generalization in Robot Learning through Video World Models. *arXiv preprint arXiv:2505.12705*, 2025.

Byeongguk Jeon, Seonghyeon Ye, JaeHyeok Doo, Sungdong Kim, Minjoon Seo, Hyungmok Son, and Kimin Lee. RoboWorld: Fast and Reliable Neural Simulators for Generalist Robot Policy Evaluation. *arXiv preprint arXiv:2607.01060*, 2026.

Tao Jiang, Tianyuan Yuan, Yicheng Liu, Chenhao Lu, Jianning Cui, et al. Galaxea Open-World Dataset and G0 Dual-System VLA Model. *arXiv preprint arXiv:2509.00576*, 2025.

Tianchong Jiang, Xiangshan Tan, Samuel Wheeler, Luzhe Sun, Tewodros W. Ayalew, and Matthew Walter. What Are We Actually Benchmarking in Robot Manipulation? *arXiv preprint arXiv:2606.04233*, 2026.

Xinghang Li, Jun Guo, Qiwei Li, Long Qian, Hang Lai, et al. Xiaomi-Robotics-U0: Unified Embodied Synthesis with World Foundation Model. *arXiv preprint arXiv:2607.11643*, 2026.

Yue Liao, Pengfei Zhou, Siyuan Huang, Donglin Yang, Shengcong Chen, Yuxin Jiang, et al. Genie Envisioner: A Unified World Foundation Platform for Robotic Manipulation. *arXiv preprint arXiv:2508.05635*, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow Matching for Generative Modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

Xuanyi Liu, Haofeng Wang, Ruiqi Li, Danni Yu, Rui Wan, Ruixu Zhang, Siyu Tao, Xue Yang, Shaofeng Zhang, Zicheng Zhang, Jiaqi Zhang, and Siwei Ma. TriWorldBench: A tri-view consistency perspective on embodied world models. *arXiv preprint arXiv:2609.26314*, 2026.

MiniMax. MiniMax H3: An open model breaking the boundaries between tasks and modalities. Technical announcement, 2026.

NVIDIA, Arslan Ali, Junjie Bai, Maciej Bala, Yogesh Balaji, Aaron Blakeman, et al. World Simulation with Video Foundation Models for Physical AI. *arXiv preprint arXiv:2511.00062*, 2025.

NVIDIA et al. Cosmos 3: Omnimodal World Models for Physical AI. *arXiv preprint arXiv:2606.02800*, 2026.

William Peebles and Saining Xie. Scalable Diffusion Models with Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

Boxiang Qiu, Liliang Chen, Yue Liao, Nan Wang, Lintao Wang, Jiayi Luo, et al. GE-Sim 2.0: A Roadmap Towards Comprehensive Closed-loop Video World Simulators for Robotic Manipulation. *arXiv preprint arXiv:2605.27491*, 2026.

Younggyo Seo, Junsu Kim, Stephen James, Kimin Lee, Jinwoo Shin, and Pieter Abbeel. Multi-View Masked World Models for Visual Robotic Manipulation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 30613–30632, 2023.

Yu Shang, Zhuohang Li, Yiding Ma, Weikang Su, Xin Jin, Ziyou Wang, et al. WorldArena: A Unified Benchmark for Evaluating Perception and Functional Utility of Embodied World Models. *arXiv preprint arXiv:2602.08971*, 2026a.

Yu Shang, Yinzhou Tang, Yiding Ma, Zhuohang Li, Lei Jin, Weikang Su, et al. WorldArena 2.0: Extending Embodied World Model Benchmarking on Modality, Functionality and Platform. *arXiv preprint arXiv:2605.17912*, 2026b.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, et al. Wan: Open and Advanced Large-Scale Video Generative Models. *arXiv preprint arXiv:2503.20314*, 2025.

Yang Tian, Yuyin Yang, Yiman Xie, Zetao Cai, Xu Shi, et al. InternData-A1: Pioneering High-Fidelity Synthetic Data for Pre-training Generalist Policy. *arXiv preprint arXiv:2511.16651*, 2025.

Wei-Cheng Tseng, Jinwei Gu, Qinsheng Zhang, Hanzi Mao, Ming-Yu Liu, Florian Shkurti, and Lin Yen-Chen. Scalable Policy Evaluation with Video World Models. *arXiv preprint arXiv:2511.11520*, 2025.

Yixuan Wang, Rhythm Syed, Fangyu Wu, Mengchao Zhang, Aykut Onol, Jose Barreiros, Hooshang Nayyeri, Tony Dear, Huan Zhang, and Yunzhu Li. Interactive World Simulator for Robot Policy Training and Evaluation. *arXiv preprint arXiv:2603.08546*, 2026.

Kun Wu, Chengkai Hou, Jiaming Liu, Zhengping Che, Xiaozhu Ju, et al. RoboMIND: Benchmark on Multi-embodiment Intelligence Normative Data for Robot Manipulation. In *Proceedings of Robotics: Science and Systems*, 2025a. doi: 10.15607/RSS.2025.XXI.152.

Shihan Wu, Xuecheng Liu, Shaoxuan Xie, Pengwei Wang, Xinghang Li, et al. RoboCOIN: An Open-Sourced Bimanual Robotic Data COllection for INtegrated Manipulation. *arXiv preprint arXiv:2511.17441*, 2025b.

Chendong Xiang, Jiajun Liu, Jintao Zhang, Xiao Yang, Zhengwei Fang, et al. Geometry-Aware Rotary Position Embedding for Consistent Video World Model. *arXiv preprint arXiv:2602.07854*, 2026.

Xiaojie Xu, Zhengyuan Lin, Kang He, Yukang Feng, Xiaofeng Mao, Yuanyang Yin, Kaipeng Zhang, and Yongtao Ge. WorldMark: A Unified Benchmark Suite for Interactive Video World Models. *arXiv preprint arXiv:2604.21686*, 2026.

Adina Yakefu, Bin Xie, Chongyang Xu, Enwen Zhang, Erjin Zhou, Fan Jia, et al. RoboChallenge: Large-scale Real-robot Evaluation of Embodied Policies. *arXiv preprint arXiv:2510.17950*, 2025.

Sherry Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Leslie Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning Interactive Real-World Simulators. In *The Twelfth International Conference on Learning Representations*, 2024.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, et al. World Action Models are Zero-shot Policies. *arXiv preprint arXiv:2602.15922*, 2026.

Chenghao Yin, Da Huang, Di Yang, Jichao Wang, Nanshu Zhao, Chen Xu, et al. Genie Sim 3.0 : A High-Fidelity Comprehensive Simulation Platform for Humanoid Robot. *arXiv preprint arXiv:2601.02078*, 2026.

Ge Yuan, Qiyuan Qiao, Jing Zhang, and Dong Xu. AdaWorldPolicy: World-Model-Driven Diffusion Policy with Online Adaptive Learning for Robotic Manipulation. *arXiv preprint arXiv:2602.20057*, 2026a.

Shenghai Yuan, Yuanyang Yin, Zongjian Li, Xinwei Huang, Xiao Yang, and Li Yuan. Helios: Real real-time long video generation model. *arXiv preprint arXiv:2603.04379*, 2026b.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018.

Min Zhao, Hongzhou Zhu, Bokai Yan, Zihan Zhou, Yimin Chen, Honglie Wang, Wenqiang Sun, Zhengwei Fang, Zizheng Xun, Zihao Li, Kaiwen Zheng, Guande He, Xiao Yang, Chongxuan Li, Fan Bao, and Jun Zhu. minWM: A full-stack open-source framework for real-time interactive video world models. *arXiv preprint arXiv:2605.30263*, 2026a.

Min Zhao, Hongzhou Zhu, Kaiwen Zheng, Zihan Zhou, Bokai Yan, Xinyuan Li, Xiao Yang, Chongxuan Li, and Jun Zhu. Causal Forcing++: Scalable few-step autoregressive diffusion distillation for real-time interactive video generation. *arXiv preprint arXiv:2605.15141*, 2026b.

Kaiwen Zheng, Guande He, Min Zhao, Jintao Zhang, Huayu Chen, Jianfei Chen, et al. CausalrCM: A Unified Teacher-Forcing and Self-Forcing Open Recipe for Autoregressive Diffusion Distillation in Streaming Video Generation and Interactive World Models. *arXiv preprint arXiv:2606.25473*, 2026.

Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. IRASim: A Fine-Grained World Model for Robot Manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9834–9844, 2025.

Supplementary Material

Overview and Video Supplement

Organization. This supplementary material comprises four parts. Appendix A describes training, the action/state interface and temporal alignment, metadata prompts, failure data collection, and the closed-loop evaluation protocol. Appendix B presents additional ablation studies and the trajectory-quality evaluation protocol. Appendix C reports detailed WorldArena and TriWorldBench results. Appendix D provides qualitative examples of cross-view recovery.

Data, code, and model release. We include the core implementation code with the supplementary material to facilitate inspection of the proposed method. The complete training and inference pipeline is being prepared for public release and requires further testing and documentation. We will open-source the full codebase and publicly release our failure dataset and model weights upon acceptance.

Video supplement. We strongly encourage reviewers to watch the supplementary videos. The video supplement includes open-loop rollouts, closed-loop policy rollouts, and success and failure examples. Please view the full sequences to assess temporal coherence, multi-view consistency, and the progression of policy–world-model interactions, which are difficult to convey through static figures alone. The videos are available at https://proworld.pages.dev/.

A Training and Interface Details

A.1 Training Details

Backbones and initial fine-tuning. We instantiate ProWorld with two pretrained video backbones: Wan2.1-14B and MiniMax-H3. We perform full-parameter fine-tuning on Wan2.1-14B and low-rank adaptation (LoRA) on MiniMax-H3, using learning rates of 2×10^{-5} and 10^{-4}, respectively. Both models are initially fine-tuned on our proposed dataset for 36,000 steps with a batch size of 64.

Wan-based training and experiments. Following this initial fine-tuning, we continue training the Wan-based model using the standard Helios Stage 1 procedure with an autoregressive attention mask. The ablation studies in the main paper and the RoboChallenge experiments use this Wan-based model.

MiniMax-H3-based training and experiments. For MiniMax-H3, we follow the standard minWM pipeline for long-video generation training and distillation after the initial fine-tuning. The additional ablation studies in this supplementary material use the MiniMax-H3-based model.

A.2 Action/State Definitions and Temporal Alignment

Definitions and layouts. The control feature u_k conditions prediction, whereas s_k is the absolute proprioceptive target. End-effector data use 32-dimensional vectors: each arm has Cartesian position (3), a 6D rotation representation (6), and gripper opening (1), occupying entries 0–9 and 10–19; unused entries are zero. Continuous pose coordinates are median-filtered and Savitzky–Golay smoothed before differencing at the control rate. Translation features have units of length per second; rotation features are 6D-coordinate rates, not angular velocities. Grippers retain raw absolute openings. State targets remain the original absolute poses and openings. These trajectory-derived controls are motion proxies. For joint data with command logs, absolute commanded and measured positions remain separate. The RoboTwin adapter, for example, places each arm's six joints and gripper in entries 0–6 and 16–22, with zeros elsewhere.

Normalization. $\mathcal{N}_u$ and $\mathcal{N}_s$ follow the dataset/embodiment schema. Scalar coordinates use 1st/99th-percentile affine scaling; absolute 6D rotations retain their bounded representation. Velocity features use separate statistics, with continuous coordinates clipped to $[-5, 5]$ and grippers to

$[-1, 1]$ after normalization. Joint data use joint-layout statistics. State predictions are mapped back to native units before policy feedback; clipping is not invertible.

Temporal alignment. Synchronized views share source-frame indices. For clip start b, source rate f_s, video rate f_v, and control rate f_u, the binary loader samples

$$i_j^v = b + \mathrm{round}(jf_s/f_v), \qquad i_k^u = b + \mathrm{round}(kf_s/f_u). \tag{6}$$

The initial image uses $j = 0$; target controls start at $k = 1$. Each state target is read at its control's source index. At equal video/control rates, (u_k, s_k) aligns with target image O_k. For $k > 1$, the derived control describes the interval ending at that sample. Its first sampled continuous-coordinate velocity is zero; the gripper retains its absolute value. Across rates, alignment follows physical time rather than equal array positions. History uses the same sampling grid, with control indices before the episode start clamped to its initial sample.

A.3 Metadata Prompt Format

Fields and serialization. The metadata prompt describes the data source, embodiment, and observation setup in a compact JSON object. Its fields have a fixed order: `dataset` identifies the dataset group or individual data subset; `robot` names the embodiment; `fps` gives the control sampling rate in Hz; `deploy` specifies the domain (e.g., `real` or `simulation`); and `views` lists camera names in the same order as the input video streams. Thus, `fps` corresponds to f_u in Appendix A.2, and the camera list is reordered whenever the views are shuffled.

Example and optional task text. For AgiBotWorld-Alpha at 10 Hz, group-level metadata take the following form (line breaks are added for readability):

```
{"dataset":"agibot/agibot_alpha","robot":"agibot_alpha",
 "fps":10,"deploy":"real",
 "views":["head_color","hand_left_color","hand_right_color"]}
```

The loader serializes this object without extra whitespace. Group-level naming permits samples with the same metadata to share a prompt; subset-level naming retains the individual dataset identifier while removing a trailing train/evaluation split name. If task conditioning is enabled and a nonempty instruction is available, a final `task` string is appended to the object. Otherwise, that field is omitted. Numeric actions and proprioceptive targets are supplied through the separate interface in Appendix A.2, rather than inserted into this JSON object.

A.4 Failure Data Collection

Perturbed physical replay. We generate missed-grasp examples from successful GenieSim trajectories. Each replay restores the recorded scene and initial robot–object poses, then tracks the recorded joint trajectory with a Cartesian offset applied to the active arm. A damped least-squares Jacobian update maps the offset to joint corrections, ramped in before gripper closure and maintained during lifting. The target object remains a dynamic rigid body, allowing contact and gravity to determine the outcome; non-target objects remain kinematic.

Failure selection and recording. We label a perturbed replay as a missed grasp only when the target's maximum vertical displacement from its first replay frame is below a threshold adjusted during collection and bounded by 5 cm. Replays reaching this threshold are excluded from the failure set. Each replay saves synchronized camera videos, the perturbed joint reference trajectory, and per-frame target and end-effector poses. A manifest links each variant to its successful source and records the requested offset, achieved offset at closure, and lift-based label.

A.5 Closed-Loop Evaluation Configuration

Adaptation and policy prerequisites. Before closed-loop evaluation, we fine-tune ProWorld for the target embodiment and tasks using trajectories collected on the corresponding real-robot platform. GenieSim failure trajectories constitute 10% of the fine-tuning mixture. Evaluation additionally requires locally deployable robot-policy weights and real-robot success rates for the same policies and tasks.

Closed-loop protocol. Evaluation recordings provide only the initial frames for world-model rollouts; subsequent recorded video and actions are never supplied to either ProWorld or the policy. After initialization, the policy acts on observations generated by ProWorld, which predicts the next multi-view observations and proprioceptive state for continued policy feedback. Thus, all subsequent actions are chosen online by the policy. Below, one interaction round denotes one policy query followed by the corresponding world-model transition.

RoboChallenge. We use the Table30v2 dataset for adaptation and evaluate $\pi_{0.5}$ and DM0.5, selected for their publicly released fine-tuned policy weights and nontrivial success rates reported on the official test platform. We use these official rates as the real-robot reference and collect 120 evaluation trajectories from the RoboChallenge website across the two policies and six tasks. These recordings supply initial frames under the protocol above. Most tasks require more than 80 interaction rounds to complete; success in the world-model rollouts is judged manually.

Cobot. For our local Cobot evaluation, we collect 200 real-robot trajectories across three policies and four tasks. These recordings serve only to measure real-robot policy success rates and provide initial frames for world-model evaluation. Cobot tasks require approximately 25 policy–world-model interaction rounds. A locally deployed vision–language model (VLM) scores the generated rollouts to determine task success.

B Additional Ablation Studies

We examine action conditioning, the state-loss weight, visual context, noise schedules, and training resolution. The action-conditioning, visual-context, noise-schedule, and training-resolution ablations use MiniMax-H3 on AgiBotWorld-Alpha, with 108 evaluation clips per setting. Slight differences in the starting configurations make absolute scores across ablations not directly comparable.

B.1 Trajectory-Quality Evaluation Protocol

Inputs and alignment. The AgiBotWorld-Alpha ablations assess video–state agreement and state prediction separately. For video–state agreement, a fixed inverse dynamics model (IDM) reads only the head-camera video at 10 FPS, resized to 224×224. We remove the initial conditioning frame to align video frames with saved state predictions. The IDM takes pairs of frames 16 steps apart and predicts 16 motion steps; overlapping predictions are averaged and denormalized. Cumulative motion is anchored at the first state of the trajectory being compared, while gripper outputs remain absolute.

Video–state consistency (VSC). Let $E(V, S)$ denote the mean Euclidean distance, in millimeters, between the IDM-reconstructed end-effector positions from video V and the positions in state sequence S. We average over time and both arms within each clip, then equally over clips. With generated video $\hat{V}$, predicted states $\hat{S}$, and recorded references (V, S), we report

$$\mathrm{VSC} = E(\hat{V}, \hat{S}) - E(V, S). \tag{7}$$

Thus, VSC measures excess video–state disagreement relative to the IDM's reference-video error; lower is better. This reference error is an empirical calibration, not a guaranteed lower bound. VSC covers the head view only and does not measure task success.

State prediction (Dir. and Grip.). Dir. is the motion-direction cosine similarity between predicted and ground-truth end-effector trajectories. Grip. is the accuracy of predicted gripper open/closed states against ground truth. Both are higher-is-better measures computed from predicted versus reference states, independently of the IDM. State metrics are omitted for models without state supervision.

B.2 Action Conditioning and State Supervision

Direct AdaLN provides a strong interface between actions and visual dynamics (Table 6). Replacing it with velocity tokens reduces PSNR by 3.48 dB and increases LPIPS from 0.1698 to 0.2786.

Action guidance recovers part of this gap, but direct modulation remains substantially stronger. State supervision also benefits visual prediction: removing it costs 1.36 dB and degrades both structural and perceptual quality.

Table 6: **Action conditioning and state supervision.** Direct AdaLN improves visual prediction over token-based conditioning.

Variant	PSNR↑	SSIM↑	LPIPS↓
Direct AdaLN	**19.03**	**0.747**	**0.1698**
w/o state supervision	17.67	0.720	0.1996
Velocity tokens	15.55	0.672	0.2786
Velocity tokens with action CFG	16.23	0.699	0.2422

B.3 State-Loss Weight

The state-loss weight controls how strongly proprioceptive supervision contributes to joint video–state learning. Figure 9 varies λ_s, the coefficient of the state term in the main paper's joint flow-matching objective, over $\{0, 0.05, 0.1, 0.5, 1, 2, 5, 15, 100\}$. The figure uses λ as shorthand for λ_s; $\lambda_s = 0$ removes state supervision, and state-related metrics are omitted for this setting.

A moderate state-loss weight provides a favorable balance between visual quality and video–state agreement. Among the tested values, $\lambda_s = 2$ yields the lowest LPIPS and VSC and the highest gripper accuracy, while PSNR and SSIM are slightly higher at $\lambda_s = 1$. Motion-direction similarity does not peak at $\lambda_s = 2$, so this choice does not optimize every metric individually. Larger weights, particularly 15 and 100, degrade visual quality and increase VSC relative to 2. We therefore adopt $\lambda_s = 2$ for all experiments with state outputs, except for this explicit weight sweep.

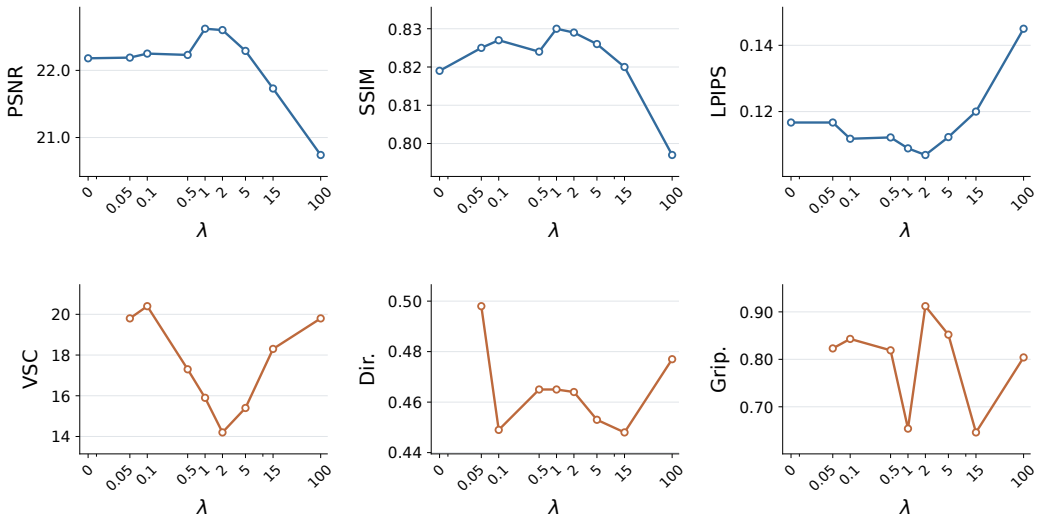


Figure 9: **Ablation of the state-loss weight.** The horizontal-axis label λ denotes λ_s in the main paper. Higher PSNR (dB), SSIM, motion-direction cosine similarity (Dir.), and gripper accuracy (Grip.) are better; lower LPIPS and video–state consistency error (VSC, mm) are better. State-related metrics are shown only for nonzero weights. The default setting is $\lambda_s = 2$.

B.4 Visual Context and Noise Scheduling

Initial scene descriptions and noise scheduling improve different aspects of prediction (Table 7). Descriptions derived from synchronized initial views reduce video–state consistency error from 9.71 to 7.93 mm, with little change in visual quality. Increasing the shared video/state flow shift to 12 gives the strongest visual scores, improving PSNR by 0.43 dB and reducing LPIPS to 0.1472. The lower shift of 5 preserves better motion direction and video–state agreement, revealing a trade-off between visual fidelity and motion accuracy.

Table 7: **Visual context and training flow shift.** Video and state use the same shift. VSC is measured in mm.

Variant	PSNR↑	SSIM↑	LPIPS↓	VSC↓	Dir.↑	Grip.↑
Default	19.88	0.757	0.1536	9.71	**0.443**	**0.996**
Initial multi-view descriptions	19.73	0.756	0.1553	**7.93**	0.437	**0.996**
Shared flow shift 5	20.02	0.758	0.1523	8.24	0.425	**0.996**
Shared flow shift 12	**20.31**	**0.762**	**0.1472**	8.64	0.407	0.995

B.5 Coordinating Video and State Denoising

Separate noise schedules let video and state prediction use different flow shifts while sharing the same denoising progress. Table 8 compares a shared shift of 12 with separate video/state shifts of 12/5, using either aligned or independently sampled progress. Aligned progress gives the lowest VSC and highest motion-direction similarity, while maintaining the visual quality of the shared schedule. Independent sampling raises VSC from 7.83 to 8.91 mm. Coordinating denoising progress therefore matters even when the two modalities use different noise levels.

Table 8: **Video and state noise schedules.** Shift pairs denote video/state shifts; aligned progress shares a sampled denoising index across the two schedules.

Variant	PSNR↑	SSIM↑	LPIPS↓	VSC↓	Dir.↑	Grip.↑
12/12, shared schedule	**20.54**	**0.765**	**0.1438**	8.38	0.422	**0.996**
12/5, aligned progress	20.51	0.764	0.1445	**7.83**	**0.440**	0.995
12/5, independent progress	20.38	0.763	0.1458	8.91	0.430	0.995

B.6 Mixed-Resolution Training

Mixed-resolution training substantially improves 480p prediction (Table 9). Compared with training exclusively at 480p, it raises PSNR by 2.11 dB, lowers VSC by 5.28 mm, and improves motion-direction similarity from 0.247 to 0.403. At 288p, however, the dedicated model retains better visual quality and motion accuracy. Resolution mixing thus offers a practical route to higher-resolution generation, with a trade-off in lower-resolution performance.

Table 9: **Mixed-resolution training.** Results are grouped by output resolution; the mixed model is evaluated at both resolutions.

Variant / output	PSNR↑	SSIM↑	LPIPS↓	VSC↓	Dir.↑	Grip.↑
480p training / 480p	16.09	0.723	0.2598	16.11	0.247	0.994
Mixed training / 480p	**18.20**	**0.757**	**0.2036**	**10.82**	**0.403**	**0.995**
288p training / 288p	**19.88**	**0.757**	**0.1536**	**9.71**	**0.443**	**0.996**
Mixed training / 288p	18.14	0.723	0.1906	13.59	0.397	**0.996**

C Detailed Benchmark Results

Tables 10 and 11 provide the complete results for the retained WorldArena metrics and all 19 Tri-WorldBench submetrics, respectively. Scores are reported on a 0–100 scale; higher is better. Bold and underlined values indicate the best and second-best results among the compared models.

Frame-rate sensitivity in WorldArena. WorldArena's motion-quality metrics are strongly affected by frame rate: lowering the evaluation frame rate can substantially increase these scores without a corresponding improvement in generated video quality. We therefore omit motion quality and the aggregate EWMScore from our reported WorldArena tables, including the main-paper comparison, rather than select a lower frame rate solely to improve benchmark scores.

Table 10: WorldArena Track 1 results on the retained metrics.

	Visual quality			Content consistency		
Model	IQ	Aes	JEPA	Subj	Bg	Photo
Ctrl-World	42.44	37.05	92.77	**83.56**	90.30	12.88
GigaWorld-1	51.18	41.17	96.77	80.97	86.43	28.11
ABot-PhysWorld	**61.03**	**41.83**	90.36	80.56	89.02	20.52
GE-Sim 2.0	46.73	39.07	97.16	82.50	**90.80**	7.05
Xiaomi-Robotics-U0	53.94	40.79	90.60	79.05	86.44	2.13
ProWorld	45.82	39.91	**98.39**	78.91	85.70	**35.34**

	Physics adherence		3D accuracy		Controllability	
Model	Inter	Traj	Depth	Persp	Instr	Sem
Ctrl-World	62.62	48.20	93.25	83.66	67.68	88.68
GigaWorld-1	75.10	54.27	**98.44**	95.60	82.14	89.42
ABot-PhysWorld	81.96	31.50	71.99	**98.94**	92.10	89.58
GE-Sim 2.0	73.60	58.63	96.49	93.64	82.70	89.05
Xiaomi-Robotics-U0	**87.30**	41.89	85.25	98.84	**93.86**	89.35
ProWorld	72.25	**62.19**	98.36	92.85	81.66	**90.41**

Abbreviations: IQ, image quality; Aes, aesthetic quality; JEPA, JEPA similarity; Subj, subject consistency; Bg, background consistency; Photo, photometric consistency; Inter, interaction quality; Traj, trajectory accuracy; Depth, depth accuracy; Persp, perspectivity; Instr, instruction following; Sem, semantic alignment.

Table 11: TriWorldBench results: TWB-Score and all 19 submetrics.

	Score	Tri-view consistency						Task alignment		
Model	TWB	N-PSNR	SSIM	VLM-I	VLM-II	VLM-III	VQA	Instr	Sem	JEPA
DreamDojo	51.72	59.78	71.58	77.14	73.26	88.19	47.82	49.72	88.28	30.73
Motus	42.35	60.13	76.31	63.68	67.56	87.22	45.33	52.28	86.54	10.24
Genie Envisioner	40.73	58.74	74.44	69.40	57.13	77.93	36.72	20.20	79.30	0.02
Ctrl-World	38.98	46.39	64.62	46.44	63.69	82.46	40.90	38.90	81.94	10.32
ProWorld	**53.68**	**62.55**	**79.06**	**78.15**	**80.73**	**93.32**	**54.05**	**62.32**	**88.75**	**42.08**

	Physical/3D		Motion quality			Temporal consistency			Visual quality	
Model	Inter	Persp	State	Flow	Traj	Subj	Bg	Photo	IQ	Aes
DreamDojo	**28.28**	59.39	60.76	6.52	16.61	77.37	**89.29**	**15.86**	24.45	17.58
Motus	25.00	44.20	59.31	0.80	13.77	77.37	1.00	1.31	16.08	16.44
Genie Envisioner	20.10	**87.91**	59.28	0.40	0.84	**91.40**	1.07	4.08	19.36	15.57
Ctrl-World	24.19	45.15	60.47	**21.29**	6.09	77.05	49.02	12.44	18.37	14.92
ProWorld	28.24	74.82	**62.01**	17.82	**30.39**	83.86	25.45	7.16	**26.99**	**22.09**

N-PSNR, normalized PSNR; VLM-I/II/III, the three VLM consistency protocols; VQA, VQA consistency; Instr, instruction following; Sem, semantic alignment; Inter, interaction quality; Persp, perspective; State, state alignment; Traj, trajectory accuracy; Subj, subject consistency; Bg, background consistency; Photo, photometric smoothness; IQ, image quality; Aes, aesthetic quality.

D Additional Qualitative Observation

Cross-view recovery. We observe that ProWorld can use the other two camera views to recover content in a target view. Figure 10 shows qualitative examples alongside the corresponding ground-truth frames. The recovered views capture the broad scene layout and object appearance, although fine details can differ. We report this as an additional qualitative observation of the model's cross-view prediction capability.

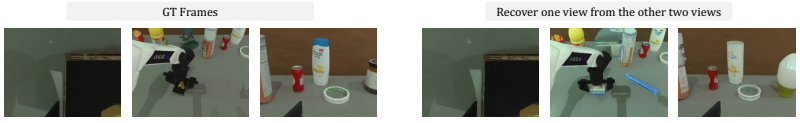


Figure 10: **Cross-view recovery.** Ground-truth frames (left) and target views recovered using the other two camera views (right).