ALIGNING VIDEO WORLD MODELS WITH EXECUTABLE ROBOT ACTIONS VIA INVERSE DYNAMICS REWARDS

Anonymous authors
Paper under double-blind review

ABSTRACT

Video generative models offer a promising basis for robotic planning by predicting future visual rollouts from observations and task instructions. However, their training objectives typically lack explicit constraints on robot executability. Generated rollouts can exhibit embodiment-related artifacts, including arm deformation and temporal discontinuities, while the actions decoded from these rollouts may be unstable or violate robot motion limits. We term this mismatch between generated robot behavior and embodiment-specific execution requirements the *executability gap*. We introduce **Executable Video Alignment (EVA)**, a post-training framework that uses action-space feedback to improve both the visual integrity of generated robot motion and the feasibility of decoded actions. EVA trains an inverse dynamics model (IDM) on robot demonstrations and freezes it to decode actions from generated videos. We compute a reward from these actions that penalizes non-smooth motion and violations of velocity and acceleration limits, and use it to guide reinforcement-learning updates to the video generator. By feeding decoded-action constraints back into video generation, EVA links visual and action-level improvements through a shared alignment signal. Experiments on RoboTwin, LIBERO, and a real bimanual robot show that EVA reduces embodiment-related visual artifacts and improves task success when executing IDM-decoded actions. Further experiments show that video models fine-tuned with EVA generate more effective video–action pairs for training downstream VLAs.

1 INTRODUCTION

Developing generalist robot policies capable of executing diverse manipulation tasks remains a central pursuit in embodied AI. Vision-Language-Action (VLA) models (Black et al., 2025; Kim et al., 2024; Brohan et al., 2023; NVIDIA et al., 2025c) have made significant progress by mapping 2D visual observations and language instructions directly to low-level motor commands. However, scaling robust long-horizon behavior remains challenging when physical and temporal dynamics must be learned primarily from limited robot interaction data (Wang et al., 2025). In parallel, recent work has explored video generation models as world models for robotics (Chen et al., 2025a; Pai et al., 2025; Kim et al., 2026; Ye et al., 2026; Bi et al., 2025). Unlike static image-text pairs, videos provide rich spatiotemporal cues about state transitions and object interactions. This capability has catalyzed an emerging decoupled paradigm: a video world model first serves as a visual planner, generating a future visual trajectory conditioned on current observations and language instructions; subsequently, an inverse dynamics model (IDM) extracts the corresponding robot actions from the generated frames (Du et al., 2023; Bharadhwaj et al., 2024; Tan et al., 2025). By separating high-level spatiotemporal reasoning from low-level control, this formulation offers a highly promising route toward scalable robot learning grounded in internet-scale video data.

Despite this promise, we identify a critical limitation in this decoupled pipeline: the absence of explicit executability constraints. We define executability as the extent to which a generated video trajectory can be translated into motor commands that accomplish the intended task while respecting the robot's physical and kinematic constraints. Foundation video generative models (Brooks et al.,

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

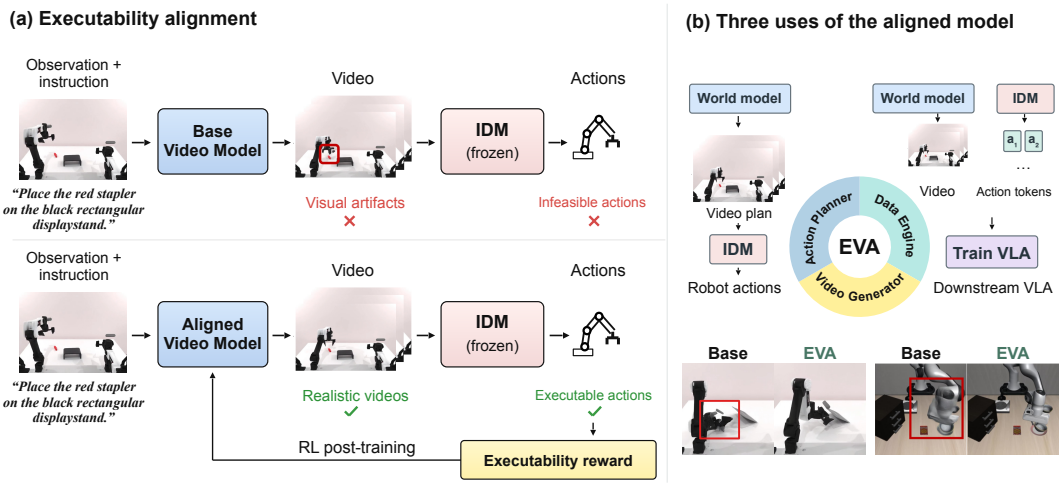


Figure 1: Overview of Executable Video Alignment (EVA). (a) EVA post-trains a video generator using rewards computed from actions decoded by a frozen inverse dynamics model (IDM), encouraging smoother, embodiment-consistent motion. (b) The aligned model supports video generation and video-based action planning, and can supply video–action pairs for downstream VLA training. The examples compare generated videos before and after alignment.

2024; Team Wan et al., 2025) are not explicitly trained to respect these constraints. Generated robot motion may exhibit arm deformation, self-intersections, or abrupt temporal discontinuities, which can translate into joint jumps or high-frequency jitter when decoded by an IDM. Even when such artifacts are not visually apparent, the decoded actions may still violate robot motion limits. We term this mismatch between generated robot behavior and embodiment-specific execution requirements the *executability gap*. Inference-time techniques such as rejection sampling can mitigate this gap, but require generating and evaluating multiple costly video candidates.

Our key idea is to use the IDM not only to decode actions for execution, but also to provide action-space feedback for post-training the video generator. We call this framework **Executable Video Alignment (EVA)**. We train the IDM on robot demonstrations and freeze it to decode actions from generated videos. From these actions, we construct a reward that (1) penalizes non-smooth trajectories through acceleration and jerk, and (2) penalizes violations of embodiment-specific velocity and acceleration limits. Aggregating these penalties over the sequence provides a trajectory-level reward that serves as a surrogate for executability. Inspired by the recent success of reinforcement learning (RL) in foundation-model alignment (Wu et al., 2025; Wang et al., 2026; Cai et al., 2025), we use this feedback to guide updates to the video generator, incorporating action-space constraints into its rollout distribution without executing each training rollout on a robot.

We evaluate our method on RoboTwin (Chen et al., 2025b), LIBERO (Liu et al., 2023), and a real-world robotic platform. We assess both the visual quality of generated rollouts and task success when executing their IDM-decoded actions. With the same IDM before and after alignment, the post-trained video model produces rollouts with fewer embodiment-related artifacts and improves downstream execution success. We further evaluate EVA as a data engine, showing that its generated video–action pairs improve downstream VLA performance when used as training data.

Our main contributions are summarized as follows:

- We identify and characterize the *executability gap* in video-based robotic planners: generated robot motion and its IDM-decoded actions may fail to respect the target robot's kinematic and embodiment constraints.
- We propose an IDM-based executability reward for aligning video world models. A frozen IDM translates generated videos into actions, whose smoothness and violations of velocity and acceleration limits provide feedback for RL post-training of the video generator.

- Experiments in simulation and on a real-robot platform show improved visual quality and execution success with the same IDM, as well as more effective synthetic training data for downstream VLAs.

2 Related Works

World Models for Policy Learning. World models connect future visual predictions with action generation for robot policies. One paradigm separates visual prediction from action decoding: UniPi and Vidar generate video plans and use an inverse dynamics model (IDM) to recover robot actions (Du et al., 2023; Feng et al., 2025), while VPP and mimic-video condition action generation on predictive video representations (Hu et al., 2025; Pai et al., 2025). Related approaches translate generated futures into control through dense correspondences or goal-conditioned policies (Ko et al., 2023; Black et al., 2023; Bharadhwaj et al., 2024). A second paradigm couples prediction and policy learning. VidMan transfers implicit video dynamics to action prediction, while Seer jointly learns future visual prediction and inverse dynamics (Wen et al., 2024; Tian et al., 2024). Unified video–action and world action models generate visual futures and actions through shared backbones or interacting experts (Zhu et al., 2025a; Cen et al., 2025; Bi et al., 2025; Kim et al., 2026; Ye et al., 2026). EVA follows the video-generation-and-IDM paradigm, extending the IDM from an action decoder to a source of feedback for video-model post-training.

Video Generation for Robotics. Beyond serving as policy components, video models provide synthetic experience for robot learning and evaluation. Video generators synthesize diverse scenes and interactions (Brooks et al., 2024; Team Wan et al., 2025; NVIDIA et al., 2025a). As data engines, DreamGen and GigaWorld generate manipulation videos paired with inferred actions to expand policy-training data (Jang et al., 2025; GigaWorld Team et al., 2025). As learned simulators, action-conditioned models predict the visual consequences of robot commands, supporting interaction modeling and policy evaluation (Zhu et al., 2025b; Guo et al., 2025; Liao et al., 2025; Gemini Robotics Team et al., 2025). Both uses require generated robot motion to be consistent and controllable. CLOVER regularizes video prediction with optical flow, while RoboMaster adds motion guidance and object appearance and shape conditioning (Bu et al., 2024; Fu et al., 2025). EVA complements these approaches by improving both the visual integrity and action executability of synthetic training pairs.

Alignment and Post-Training in Generative Models. RL aligns generative models with human feedback (Ouyang et al., 2022) or automated rewards (DeepSeek-AI et al., 2025), commonly using PPO (Schulman et al., 2017) or GRPO (Shao et al., 2024). Flow-GRPO extends reward-based optimization to flow-matching models (Liu et al., 2025a). For video world models, Cosmos-Predict uses VLM-based feedback on text alignment, motion quality, and visual quality (NVIDIA et al., 2025b; Liu et al., 2025b), while ReWorld learns multi-dimensional rewards for physical realism, task completion, embodiment plausibility, and visual quality (Peng et al., 2026). RLIR rewards agreement between IDM-recovered actions and the actions conditioning video generation (**?**). EVA instead targets language-conditioned video planning without target action trajectories, deriving feedback from IDM-decoded actions to encourage temporal smoothness and embodiment-specific motion constraints.

3 Problem Setup

Given recent camera observations and a language instruction $c = (o_{t-h+1:t}, \ell)$, a video planner samples a future rollout $V \sim p_\theta(\cdot \mid c)$. An inverse dynamics model (IDM) f_ϕ maps temporal windows of V to a robot command sequence A. Our objective is to improve the video generator so that its rollouts exhibit fewer embodiment-related artifacts and support more successful task execution under the same action decoder.

Our backbone is a latent flow-matching video generator (Team Wan et al., 2025; Lipman et al., 2022). A video VAE maps a training video to z_1, and $z_0 \sim \mathcal{N}(0, \mathbf{I})$ supplies noise. With flow time $s \in [0, 1]$, the interpolation is $z_s = (1 - s)z_0 + sz_1$, and supervised training minimizes

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}_{z_0, z_1, s, c}\left[\|u_\theta(z_s, s, c) - (z_1 - z_0)\|_2^2\right]. \tag{1}$$

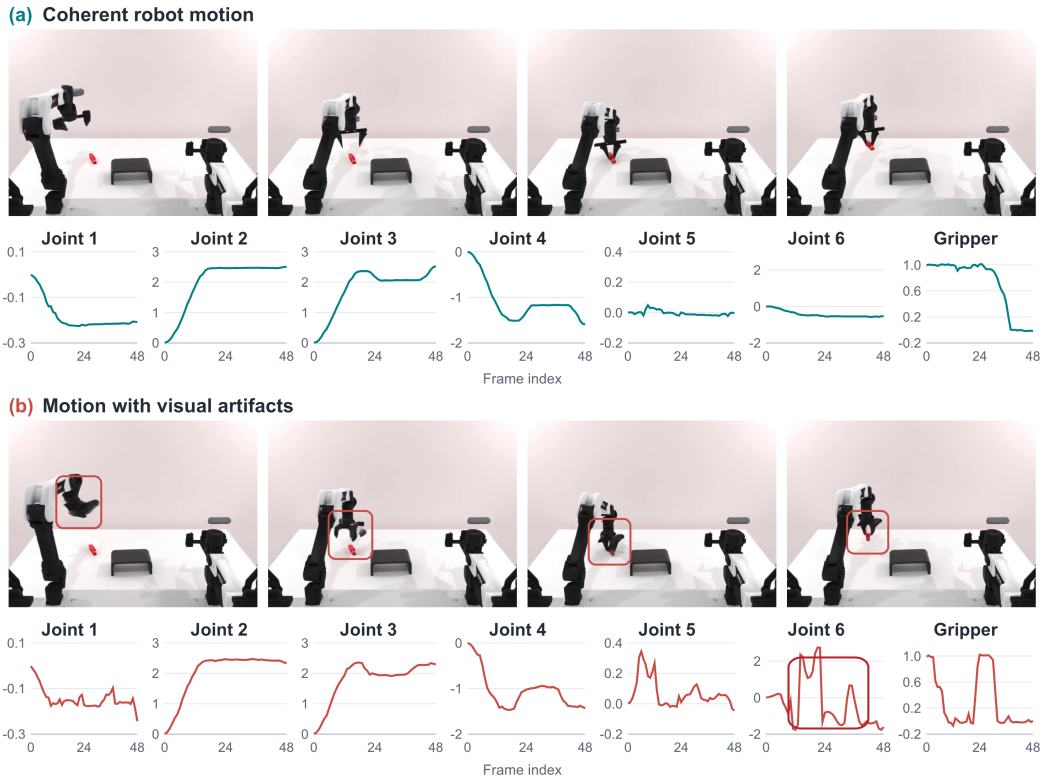


Figure 2: IDM-decoded actions from generated videos with and without visual artifacts. Plots show six left-arm joint-position commands and one gripper command, ordered from base to gripper. **Top:** coherent robot motion with smooth decoded commands. **Bottom:** visual artifacts (red boxes) accompanied by command fluctuations, especially in the distal joint and gripper (dimensions 6–7).

At inference time, ODE sampling transforms noise into a latent video, which the VAE decodes into frames.

4 Executable Video Alignment

4.1 Inverse Dynamics Model

The IDM infers robot control commands from a short temporal window of visual observations. Given frames $I_{t-k:t+k}$ centered at time t, the IDM predicts the joint-position command q_t. We train the IDM on robot trajectory data with supervised regression:

$$\mathcal{L}_{\text{IDM}} = \mathbb{E}\left[\sum_t \left\| f_\phi(I_{t-k:t+k}) - q_t^{\text{gt}} \right\|_2^2\right], \tag{2}$$

where k denotes the temporal context radius.

The IDM uses a convolutional backbone, spatial softmax, and an MLP action head (Levine et al., 2016). Architecture details and the spatial-representation ablation are provided in Appendix D. The trained IDM is frozen throughout video-model post-training.

4.2 IDM-Based Executability Reward

We assess executability through the smoothness and motion-limit violations of IDM-decoded actions, using them as a surrogate for downstream execution (Figure 2).

Given a generated video V, the frozen IDM predicts a sequence of joint-position commands $A = \{q_t\}_{t=1}^{T}$ at control interval Δt. We compute velocity $v_t = (q_{t+1} - q_t)/\Delta t$, acceleration $\alpha_t = (v_{t+1} - v_t)/\Delta t$, and jerk $j_t = (\alpha_{t+1} - \alpha_t)/\Delta t$. To penalize non-smooth motions, we apply a robust Huber penalty:

$$\mathrm{Huber}(x;\delta) = \begin{cases} \frac{1}{2}x^2, & |x| \le \delta, \\ \delta\left(|x| - \frac{1}{2}\delta\right), & |x| > \delta, \end{cases} \tag{3}$$

yielding acceleration and jerk penalties

$$\mathcal{P}_\alpha = \mathbb{E}_t[\mathrm{Huber}(\alpha_t;\delta_\alpha)], \qquad \mathcal{P}_j = \mathbb{E}_t[\mathrm{Huber}(j_t;\delta_j)]. \tag{4}$$

The Huber loss is applied componentwise and aggregated over command dimensions. We further penalize violations of the robot's velocity and acceleration bounds:

$$\mathcal{P}_{\mathrm{vel}} = \mathbb{E}_t \left\|\max(|v_t| - v_{\max}, 0)\right\|_2^2, \qquad \mathcal{P}_{\mathrm{acc}} = \mathbb{E}_t \left\|\max(|\alpha_t| - \alpha_{\max}, 0)\right\|_2^2. \tag{5}$$

The reward is the negative total penalty

$$R = -\left(\lambda_j \mathcal{P}_j + \lambda_\alpha \mathcal{P}_\alpha + \lambda_{v\text{-lim}} \mathcal{P}_{\mathrm{vel}} + \lambda_{a\text{-lim}} \mathcal{P}_{\mathrm{acc}}\right), \tag{6}$$

where the nonnegative weights balance the components. Maximizing this reward favors smoother decoded motions with fewer constraint violations. The feedback is computed from generated videos without executing each training rollout on a robot.

4.3 Post-Training and Deployment

For RL post-training with the IDM-based reward, we adopt Flow-GRPO (Liu et al., 2025a) for policy optimization. For each conditioning context, we sample G videos and compute rewards R_i with the frozen IDM. The group-relative advantage (Shao et al., 2024) is $\widehat{A}_i = (R_i - \mu_R)/(\sigma_R + \epsilon)$, where μ_R and σ_R denote the mean and standard deviation of the group rewards. The generator is then optimized using the clipped objective

$$\mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G} \min\Big(r_i(\theta)\widehat{A}_i, \mathrm{clip}(r_i(\theta), 1-\varepsilon, 1+\varepsilon)\widehat{A}_i\Big) - \beta D_{\mathrm{KL}}(\pi_\theta \,\|\, \pi_{\mathrm{ref}})\right]. \tag{7}$$

Here $r_i(\theta)$ is the current-to-behavior policy transition likelihood ratio, with the sampling-step index omitted; the expectation also averages over sampled transitions and the KL term regularizes deviation from the reference policy. Only generator LoRA (Hu et al., 2021) parameters are updated. The fine-tuned model follows the same sampling algorithm as the reference model.

At inference time, the aligned video model follows standard video-planning procedures. Longer videos are generated by conditioning each new chunk on preceding frames. For closed-loop control, the IDM decodes each chunk into robot actions; after execution, the model replans from the latest camera observation.

5 Experiments

We evaluate whether action-space feedback improves three aspects of robotic video generation: visual rollout quality, task success when executing IDM-decoded actions, and the utility of generated video–action pairs for downstream VLA training.

5.1 Experimental Settings

Models and training. We initialize the Wan2.1-14B video backbone (Team Wan et al., 2025) from the pretrained Large Video Planner (Chen et al., 2025a) and perform embodiment-specific supervised fine-tuning with diffusion forcing (Chen et al., 2024), yielding the base model **Base**. Starting from this checkpoint, we use EVA to perform RL post-training with the IDM-based reward, yielding the fine-tuned model **EVA**. The IDM is trained on robot demonstrations and remains frozen throughout post-training. Training configurations are provided in Appendix C.

Benchmarks and data. On RoboTwin 2.0 (Chen et al., 2025b), we train the video model on 1,050 demonstrations across 21 tasks and evaluate Base and EVA on these tasks under the clean setting

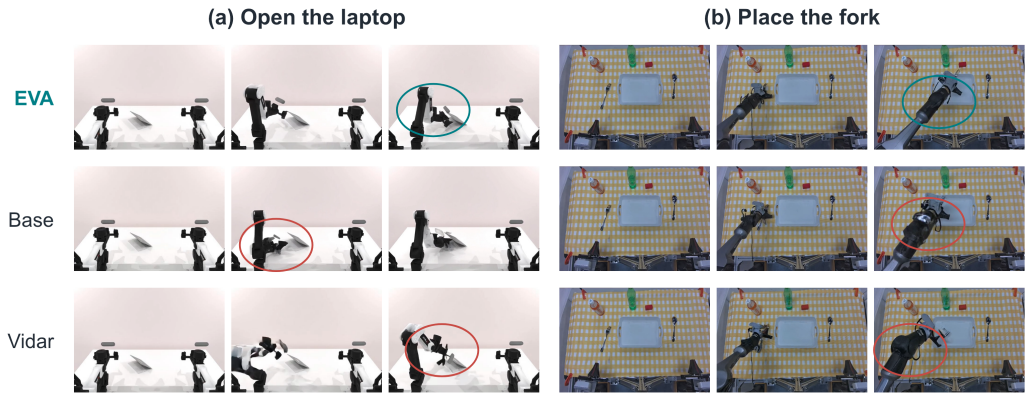


Figure 3: Qualitative video comparison on two tasks. Rows show EVA, Base, and Vidar; red circles highlight arm artifacts in Base and Vidar, while EVA shows more coherent robot geometry.

Table 1: Visual planning quality across 210 prompts, including Vidar (Feng et al., 2025). Entries are human-rated pass rates (%); Perfect requires all criteria and visually depicted task completion.

Method	Kinematic	Interaction	Instruction	Perfect
Vidar (Wan2.2)	67.6	66.7	87.6	62.9
Base	70.5	83.3	**90.5**	68.1
EVA	**91.4**	**86.2**	89.5	**83.8**

(`demo_clean`), with 50 trials per task. On LIBERO (Liu et al., 2023), we train both the video model and IDM on demonstrations from the Spatial, Goal, and Object suites and report execution success separately for each suite. For real-robot experiments, we use the AgileX CobotMagic platform and collect 250 training demonstrations covering five tasks. We evaluate on these five seen tasks and five additional out-of-distribution (OOD) tasks, with 20 trials per task.

Evaluation and baselines. Execution follows the closed-loop protocol in Section 4.3; success requires task completion without human intervention or safety interruption. Our main comparison is Base versus EVA. We additionally compare visual quality with Vidar (Wan2.2-5B) (Feng et al., 2025) and real-robot execution with ACT, π_0, Vidar, and GE-Act (Figure 5). Full RoboTwin baseline comparisons and per-task results are provided in Appendix E.

5.2 Visual Rollout Quality

We use structured human evaluation to assess whether action-space feedback reduces embodiment-related artifacts in generated videos. The evaluation contains 210 held-out observation–instruction prompts, with ten prompts per RoboTwin task. Generated videos are anonymized and shuffled, then rated for *kinematic plausibility* (arm visual integrity and temporal continuity), *interaction plausibility* (contact and object motion), and *instruction following*. A rollout is rated *perfect* if it satisfies all three criteria and visually depicts task completion. This rubric evaluates the generated plan; actual execution success is assessed separately below.

Kinematic plausibility improves most, with a 20.9% gain over Base (Table 1), while interaction plausibility increases modestly and instruction following remains similar. This pattern is consistent with the reward's focus on decoded motion: action-space feedback can discourage visually inconsistent robot motion without directly supervising task semantics or object interactions. The improvement in visual quality suggests that the frozen IDM provides a useful connection between action constraints and video generation. Qualitative examples (Fig. 3) show the reduction in arm deformation.

5.3 Direct Action Execution

Simulation. EVA improves mean success by 6.48% on RoboTwin and by 6.5–7.5 points across the three LIBERO suites (Table 2). On RoboTwin, success improves on 18 of 21 tasks (Appendix E).

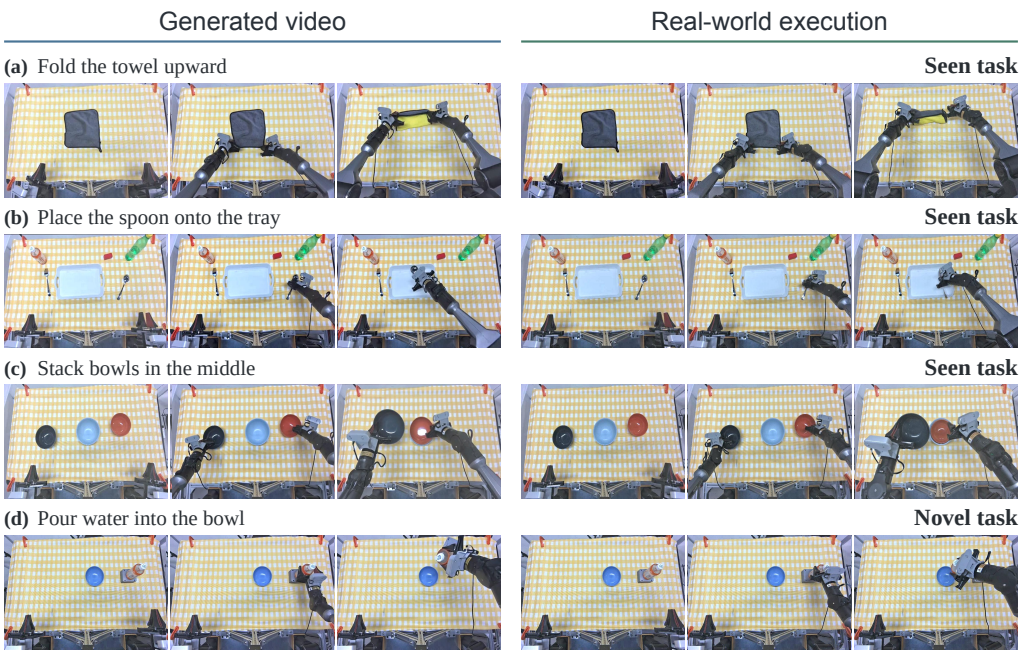


Figure 4: Real-robot deployment. Left: Generated video plans ; Right: corresponding robot executions.

Together with the visual evaluation, these gains suggest that more coherent generated robot motion can translate into more reliable downstream execution: reducing motion discontinuities in visual plans may help the IDM decode smoother action sequences that are easier to execute. A similar pattern is observed across all three LIBERO suites, where EVA consistently improves execution success over Base.

Table 2: Direct execution success. RoboTwin reports successes/50 trials for three representative tasks and mean success (%) over all 21 tasks under the clean setting. LIBERO reports success rates (%) over 200 trials per suite. Δ: EVA minus Base, in counts for individual RoboTwin tasks and percentage points otherwise.

Method	RoboTwin 2.0					LIBERO		
	Place Mouse	Place Obj. Stand	Turn Switch	...	Mean (%) 21 tasks	Spatial	Goal	Object
Base	10/50	28/50	23/50	...	45.81	46.5	49.5	63.5
EVA	**15/50**	**33/50**	**30/50**	...	**52.29**	**53.5**	**57.0**	**70.0**
Δ	+5	+5	+7	...	+6.48	+7.0	+7.5	+6.5

Real-robot deployment. The video-planning pipeline achieves higher overall success than the evaluated VLA baseline π_0, with a particularly clear advantage on OOD tasks (Figure 5). Applying EVA post-training increases execution success on both seen and OOD tasks. Smoother commands that remain within motion limits may make generated plans easier to execute on the physical robot, reducing failures caused by jitter and abrupt joint motions that the robot cannot reliably follow. The gains on OOD tasks suggest that these motion-level benefits can extend beyond the training tasks. Figure 4 pairs generated plans with their physical executions.

5.4 Synthetic Data for VLA Training

Following WorldArena's data-engine evaluation framework (Shang et al., 2026), we fine-tune $\pi_{0.5}$ (Physical Intelligence et al., 2025) on Base- or EVA-generated videos paired with IDM-decoded actions for Click Bell, Place A2B Right, and Place Container Plate. Both conditions use the same

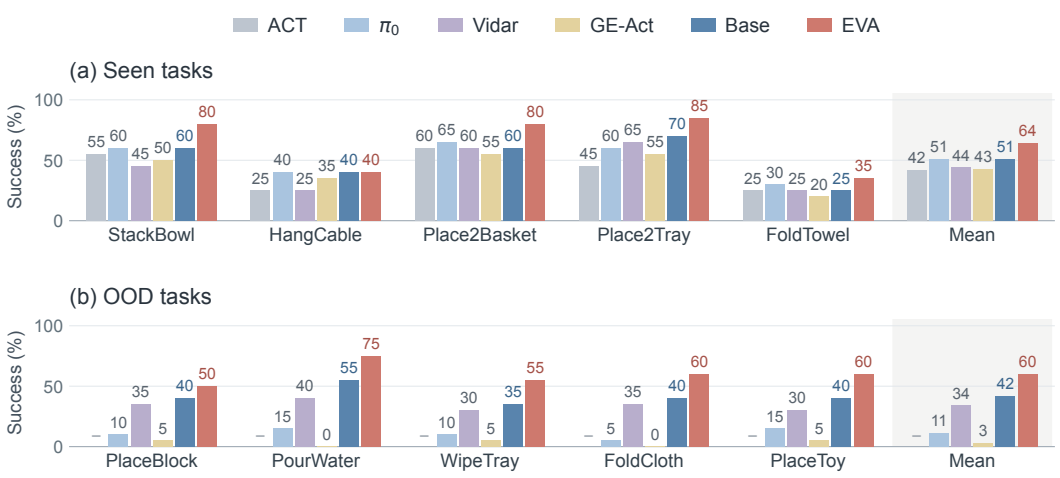


Figure 5: Real-robot success (%) on seen and OOD tasks with 20 trials each: ACT (Zhao et al., 2023), π_0 (Black et al., 2025), Vidar (Feng et al., 2025), and GE-Act (Liao et al., 2025). Shaded groups show means; dashes denote unreported ACT OOD results.

data quantity and policy-training settings. Each policy is evaluated over 100 trials per task (Table 3); implementation details are provided in Appendix C.

Table 3: Downstream $\pi_{0.5}$ success after fine-tuning on Base- or EVA-generated data for three RoboTwin tasks. Entries are success rates (%) over 100 trials per task.

Data source	Click Bell	Place A2B Right	Place Container Plate	Mean
Base-generated data	92	18	53	54.3
EVA-generated data	**98**	**26**	**67**	**63.7**

With matched data quantity and policy-training settings, EVA-generated data improves $\pi_{0.5}$ success on all three tasks, increasing mean success from 54.3% to 63.7% (Table 3). These results provide initial evidence that executability alignment can improve the usefulness of world-model-generated data for downstream robot learning. Appendix H further shows that EVA can generate plausible rollouts from synthetic initial scenes, although the downstream training utility of this fully synthetic setting remains to be evaluated.

5.5 Ablations and Analysis

Auxiliary training is evaluated on all 21 RoboTwin tasks; reward ablation and inference-time controls use separate three-task evaluations.

Reward diagnostics. Artifact-free rollouts and successful executions generally receive higher rewards (Appendix Figure S3), suggesting that decoded actions expose motion inconsistencies relevant to both generation and execution. However, nearly static motion incurs little smoothness penalty, and incorrect interactions can still yield smooth, in-limit commands (Appendix Figure S5). The reward therefore guides how the robot moves without directly assessing whether it achieves the intended interaction.

Reward components. Smoothness penalties yield higher execution success than motion limits alone, despite similar visual perfect rates (Table 4b). Commands can fluctuate within admissible bounds, so limit penalties alone may leave jitter that is difficult to execute. Conversely, smooth motion can still exceed robot limits. The full reward performs best, supporting the complementary roles of temporal smoothness and motion-limit constraints.

Training and inference-time strategies. We examine how the proposed action-space feedback can be used during training and inference (Table 4a). IDM-aux SFT augments the supervised video objective with the negative of the same IDM-based reward used by EVA. Starting from the same Base

checkpoint and using the same robot data with comparable optimization budgets, both methods improve execution success, supporting action-space feedback as a useful alignment signal across optimization objectives. EVA directly optimizes this reward on model-generated rollouts without paired target videos during post-training. It thus applies feedback to the same free-running generation process encountered at inference, while achieving modestly higher success than auxiliary SFT.

We also compare inference-time interventions. Local smoothing around motion-limit violations or abrupt command changes does not improve execution success, suggesting that such corrections cannot reliably repair motion errors in generated rollouts and may distort task-relevant movements. Reward-based rejection sampling improves Base by selecting the best of four candidates, demonstrating the reward's utility for inference-time selection. However, it requires multiple video generations and IDM evaluations per planning round. EVA incorporates the same feedback during post-training and achieves slightly higher success with a single candidate. Together, these results support IDM-derived action-space feedback across training and inference, with RL integrating this signal into free-running video generation and avoiding the additional sampling cost of inference-time selection.

Table 4: Ablations and alternative strategies on RoboTwin. (a) Auxiliary training uses 21 tasks; inference-time controls use a separate three-task evaluation. Base + Smooth smooths decoded actions; G_{test} denotes candidates per planning round. (b) Reward ablation uses another three-task evaluation. (c) EVA applied to Vidar (Feng et al., 2025) on OpenLaptop. All rates are percentages.

(a) Training and inference

Method	G_{test}	Success (%)
Auxiliary training (21 tasks)		
Base	1	45.81
IDM-aux SFT	1	50.29
EVA	1	**52.29**
Inference-time controls (3 tasks)		
Base	1	53.0
Base + Smooth	1	51.7
Base + RS-4	4	63.3
EVA	1	**65.0**

(b) Reward components

Reward setting	Video perfect (%)	Execution (%)
Base	63.3	46.67
Acc. + jerk only	76.7	63.33
Limits only	75.0	56.67
Full reward	**81.7**	**68.33**

(c) EVA applied to Vidar

Task	Vidar	Vidar + EVA
OpenLaptop	50.0	**65.0**

Transfer across video models and action decoders. Applying EVA to Vidar improves success on OpenLaptop (Table 4c), providing initial evidence of applicability to another video model. EVA also improves success by 6.2 percentage points with an independently trained ViT-based IDM never used for reward computation, supporting transfer beyond the reward-producing decoder (Appendix D).

6 Conclusion

We presented **Executable Video Alignment (EVA)**, a framework for addressing the executability gap in robotic video generation through action-space feedback. A frozen IDM decodes the motion implied by generated rollouts, and penalties on non-smooth motion and velocity- and acceleration-limit violations guide reinforcement-learning updates to the video generator. Experiments on RoboTwin, LIBERO, and a real bimanual robot show reduced embodiment-related visual artifacts and improved task success when executing IDM-decoded actions. EVA-generated video–action pairs also improve downstream VLA performance when used as training data. These results support action-space feedback as a practical way to improve robotic video models for both direct control and synthetic data generation.

Limitations and Future Work. Our reward focuses on kinematic feasibility and smoothness, and does not explicitly model contact dynamics such as forces, friction, or torques, which are critical for precision contact-rich manipulation. Diffusion-based video generation is also computationally expensive in our current setup, limiting applicability to high-frequency reactive control. Future work will explore richer dynamics-aware reward signals and faster sampling (e.g., distillation or fewer-step samplers) to enable more responsive closed-loop deployment.

AI USE STATEMENT

AI tools assisted with literature research for Related Work and language polishing.

REFERENCES

Homanga Bharadhwaj, Debidatta Dwibedi, Abhinav Gupta, Shubham Tulsiani, Carl Doersch, Ted Xiao, Dhruv Shah, Fei Xia, Dorsa Sadigh, and Sean Kirmani. Gen2Act: Human video generation in novel scenarios enables generalizable robot manipulation. *arXiv preprint arXiv:2409.16283*, 2024. URL http://arxiv.org/abs/2409.16283.

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

Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pretrained image-editing diffusion models. *arXiv preprint arXiv:2310.10639*, 2023.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. In *Proceedings of Robotics: Science and Systems*, June 2025. doi: 10.15607/RSS.2025.XXI.010.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023. URL http://arxiv.org/abs/2307.15818.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. OpenAI technical report, 2024. URL http://openai.com/index/video-generation-models-as-world-simulators/.

Qingwen Bu, Jia Zeng, Li Chen, Yanchao Yang, Guyue Zhou, Junchi Yan, Ping Luo, Heming Cui, Yi Ma, and Hongyang Li. Closed-loop visuomotor control with generative expectation for robotic manipulation, October 2024.

Yuanhao Cai, Kunpeng Li, Menglin Jia, Jialiang Wang, Junzhe Sun, Feng Liang, Weifeng Chen, Felix Juefei-Xu, Chu Wang, Ali Thabet, et al. PhyGDPO: Physics-aware groupwise direct preference optimization for physically consistent text-to-video generation. *arXiv preprint arXiv:2512.24551*, 2025. URL http://arxiv.org/abs/2512.24551.

Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, Deli Zhao, and Hao Chen. WorldVLA: Towards autoregressive action world model. *arXiv preprint arXiv:2506.21539*, 2025. URL http://arxiv.org/abs/2506.21539.

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, pp.

24081–24125. Curran Associates, Inc., 2024. doi: 10.52202/079017-0759. URL http://proceedings.neurips.cc/paper_files/paper/2024/file/2aee1c4159e48407d68fe16ae8e6e49e-Paper-Conference.pdf.

Boyuan Chen, Tianyuan Zhang, Haoran Geng, Caiyi Zhang, Peihao Li, Kiwhan Song, William T. Freeman, Jitendra Malik, Pieter Abbeel, Russ Tedrake, Vincent Sitzmann, and Yilun Du. Large video planner enables generalizable robot control. *arXiv preprint arXiv:2512.15840*, 2025a. URL http://arxiv.org/abs/2512.15840.

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*, 2025b. URL http://arxiv.org/abs/2506.18088.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin CM Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems*, July 2023. doi: 10.15607/RSS.2023.XIX.026.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. URL http://arxiv.org/abs/2501.12948.

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 9156–9172. Curran Associates, Inc., 2023. doi: 10.52202/075280-0403. URL http://proceedings.neurips.cc/paper_files/paper/2023/file/1d5b9233ad716a43be5c0d3023cb82d0-Paper-Conference.pdf.

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

Xiao Fu, Xintao Wang, Xian Liu, Jianhong Bai, Runsen Xu, Pengfei Wan, Di Zhang, and Dahua Lin. Learning video generation for robotic manipulation with collaborative trajectory control, July 2025.

Gemini Robotics Team, Krzysztof Choromanski, Coline Devin, Yilun Du, Debidatta Dwibedi, Ruiqi Gao, Abhishek Jindal, Thomas Kipf, Sean Kirmani, Isabel Leal, Fangchen Liu, Anirudha Majumdar, Andrew Marmon, Carolina Parada, Yulia Rubanova, Dhruv Shah, Vikas Sindhwani, Jie Tan, Fei Xia, Ted Xiao, Sherry Yang, Wenhao Yu, and Allan Zhou. Evaluating Gemini Robotics policies in a Veo world simulator. *arXiv preprint arXiv:2512.10675*, 2025. URL http://arxiv.org/abs/2512.10675.

GigaWorld Team, Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Haoyun Li, Jiagang Zhu, Kerui Li, Mengyuan Xu, Qiuping Deng, Siting Wang, Wenkang Qin, Xinze Chen, Xiaofeng Wang, Yankai Wang, Yu Cao, Yifan Chang, Yuan Xu, Yun Ye, Yang Wang, Yukun Zhou, Zhengyuan Zhang, Zhehao Dong, and Zheng Zhu. GigaWorld-0: World models as data engine to empower embodied AI. *arXiv preprint arXiv:2511.19861*, 2025. URL http://arxiv.org/abs/2511.19861.

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. URL http://arxiv.org/abs/2510.10125.

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

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. URL http://arxiv.org/abs/2106.09685.

Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 24328–24346. PMLR, 13–19 Jul 2025. URL http://proceedings.mlr.press/v267/hu25g.html.

Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, Yen-Chen Lin, Loic Magne, Ajay Mandlekar, Avnish Narayan, You Liang Tan, Guanzhi Wang, Jing Wang, Qi Wang, Yinzhen Xu, Xiaohui Zeng, Kaiyuan Zheng, Ruijie Zheng, Ming-Yu Liu, Luke Zettlemoyer, Dieter Fox, Jan Kautz, Scott Reed, Yuke Zhu, and Linxi Fan. DreamGen: Unlocking generalization in robot learning through video world models. *arXiv preprint arXiv:2505.12705*, 2025. URL http://arxiv.org/abs/2505.12705.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. URL http://arxiv.org/abs/2406.09246.

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

Po-Chen Ko, Jiayuan Mao, Yilun Du, Shao-Hua Sun, and Joshua B Tenenbaum. Learning to act from actionless videos through dense correspondences. *arXiv preprint arXiv:2310.08576*, 2023.

Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(39):1–40, 2016.

Yue Liao, Pengfei Zhou, Siyuan Huang, Donglin Yang, Shengcong Chen, Yuxin Jiang, Yue Hu, Jingbin Cai, Si Liu, Jianlan Luo, Liliang Chen, Shuicheng Yan, Maoqing Yao, and Guanghui Ren. Genie Envisioner: A unified world foundation platform for robotic manipulation. *arXiv preprint arXiv:2508.05635*, 2025. URL http://arxiv.org/abs/2508.05635.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. URL http://arxiv.org/abs/2210.02747.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 44776–44791. Curran Associates, Inc., 2023. doi: 10.52202/075280-1939. URL http://proceedings.neurips.cc/paper_files/paper/2023/file/8c3c666820ea055a77726d66fc7d447f-Paper-Datasets_and_Benchmarks.pdf.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-GRPO: Training flow matching models via online RL. In *Advances in Neural Information Processing Systems*, volume 38, pp. 40783–40818. Curran Associates, Inc., 2025a. doi: 10.52202/085713-1362. URL http://proceedings.neurips.cc/paper_files/paper/2025/file/3a10c46572628d58cb44fb705f25cbbf-Paper-Conference.pdf.

Jie Liu, Gongye Liu, Jiajun Liang, Ziyang Yuan, Xiaokun Liu, Mingwu Zheng, Xiele Wu, Qiulin Wang, Menghan Xia, Xintao Wang, Xiaohong Liu, Fei Yang, Pengfei Wan, Di Zhang, Kun Gai, Yujiu Yang, and Wanli Ouyang. Improving video generation with human feedback. In *Advances in Neural Information Processing Systems*, volume 38, pp. 82155–82192. Curran Associates, Inc., 2025b. doi: 10.52202/

085713-2750. URL http://proceedings.neurips.cc/paper_files/paper/2025/file/76227feb18ea0ee40bd15cf02c33e18e-Paper-Conference.pdf.

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 arXiv:2410.07864*, 2024. URL http://arxiv.org/abs/2410.07864.

NVIDIA, Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, et al. Cosmos world foundation model platform for physical AI. *arXiv preprint arXiv:2501.03575*, 2025a. URL http://arxiv.org/abs/2501.03575.

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

NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025c. URL http://arxiv.org/abs/2503.14734.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744. Curran Associates, Inc., 2022. doi: 10.52202/068431-2011. URL http://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf.

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. URL http://arxiv.org/abs/2512.15692.

Baorui Peng, Wenyao Zhang, Liang Xu, Zekun Qi, Jiazhao Zhang, Hongsi Liu, Wenjun Zeng, and Xin Jin. Reworld: Multi-dimensional reward modeling for embodied world models, January 2026.

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

vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025. URL http://arxiv.org/abs/2504.16054.

Naina Raisinghani. Introducing Nano Banana Pro. Google, The Keyword, November 2025. URL http://blog.google/innovation-and-ai/products/nano-banana-pro/.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL http://arxiv.org/abs/1707.06347.

Yu Shang, Zhuohang Li, Yiding Ma, Weikang Su, Xin Jin, Ziyou Wang, Lei Jin, Xin Zhang, Yinzhou Tang, Haisheng Su, et al. WorldArena: A unified benchmark for evaluating perception and functional utility of embodied world models. *arXiv preprint arXiv:2602.08971*, 2026. URL http://arxiv.org/abs/2602.08971.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. URL http://arxiv.org/abs/2402.03300.

Hengkai Tan, Yao Feng, Xinyi Mao, Shuhe Huang, Guodong Liu, Zhongkai Hao, Hang Su, and Jun Zhu. AnyPos: Automated task-agnostic actions for bimanual manipulation. *arXiv preprint arXiv:2507.12768*, 2025. URL http://arxiv.org/abs/2507.12768.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. URL http://arxiv.org/abs/2503.20314.

Yang Tian, Sizhe Yang, Jia Zeng, Ping Wang, Dahua Lin, Hao Dong, and Jiangmiao Pang. Predictive inverse dynamics models are scalable learners for robotic manipulation, December 2024.

Jie Wang, Matthew Leonard, Kostas Daniilidis, Dinesh Jayaraman, and Edward S. Hu. Evaluating π_0 in the wild: Strengths, problems, and the future of generalist robot policies. GRASP Lab, University of Pennsylvania, 2025. URL http://penn-pal-lab.github.io/Pi0-Experiment-in-the-Wild/.

Jin Wang, Jianxiang Lu, Guangzheng Xu, Comi Chen, Haoyu Yang, Linqing Wang, Peng Chen, Mingtao Chen, Zhichao Hu, Longhuang Wu, et al. TAGRPO: Boosting GRPO on image-to-video generation with direct trajectory alignment. *arXiv preprint arXiv:2601.05729*, 2026. URL http://arxiv.org/abs/2601.05729.

Youpeng Wen, Junfan Lin, Yi Zhu, Jianhua Han, Hang Xu, Shen Zhao, and Xiaodan Liang. Vidman: Exploiting implicit dynamics from video diffusion model for effective robot manipulation, November 2024.

Jialong Wu, Shaofeng Yin, Ningya Feng, and Mingsheng Long. RLVR-World: Training world models with reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 125312–125350. Curran Associates, Inc., 2025. doi: 10.52202/085713-4177. URL http://proceedings.neurips.cc/paper_files/paper/2025/file/b63a24a1832bd14fa945c71f535c0095-Paper-Conference.pdf.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Malik, Kyungmin Lee, William Liang, Nadun Ranawaka, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, Johan Bjorck, Jing Wang, Gwanghyun Kim, Dantong Niu, Ruijie Zheng, Yuqi Xie,

Jimmy Wu, Qi Wang, Ryan Julian, Danfei Xu, Yilun Du, Yevgen Chebotar, Scott Reed, Jan Kautz, Yuke Zhu, Linxi "Jim" Fan, and Joel Jang. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026. URL http://arxiv.org/abs/2602.15922.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Proceedings of Robotics: Science and Systems*, July 2023. doi: 10.15607/RSS.2023.XIX.016.

Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets, May 2025a.

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 (ICCV)*, pp. 9834–9844, October 2025b.

A Failure Modes and Additional Qualitative Results

We examine failures in rollouts generated by unaligned video models during real-robot evaluation. Visual inconsistencies can propagate through the IDM into unstable commands, while incorrect interactions or goals can prevent task completion even when the motion appears plausible.

Figure S1 illustrates representative examples. We group the dominant failure patterns into three categories. **Implausible kinematics** refers to violations of rigid-body consistency, including morphological deformation of the robot arm, ambiguous joint articulation, and abrupt temporal discontinuities across frames. **Wrong contact** denotes physically inconsistent object interactions, such as penetration or missing contact events. **Incorrect goal** occurs when the generated rollout fails to make meaningful progress toward the instruction-conditioned objective.

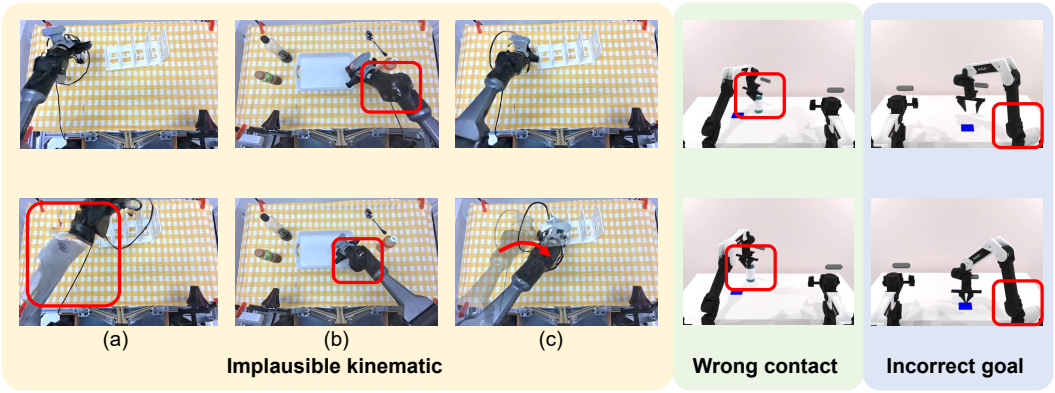


Figure S1: Common failure modes observed in unaligned video world models during real-world execution. **Implausible kinematics:** violations of rigid-body consistency, including (a) morphological deformation, (b) ambiguous joint articulation, and (c) temporal discontinuity. **Wrong contact:** physically inconsistent object interaction. **Incorrect goal:** failure to make progress toward the instruction-conditioned objective.

LIBERO visual comparison. Figure S2 compares Base and EVA rollouts on two LIBERO tasks. The soup-can example follows a single planning round for each model, while the drawer example illustrates Base artifacts and later EVA interaction stages. Red boxes identify visible arm deformation; the caption specifies the frame selection for each panel.

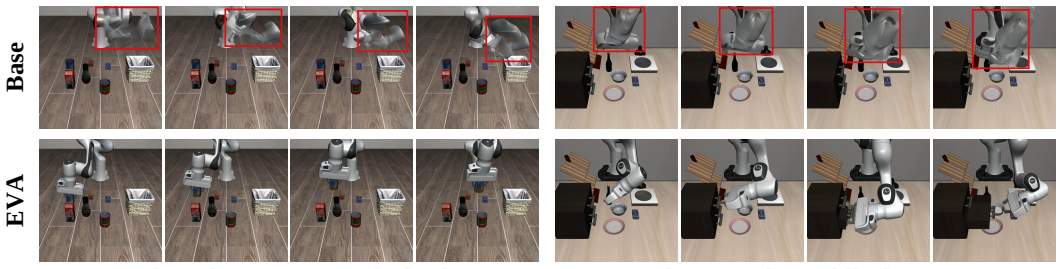


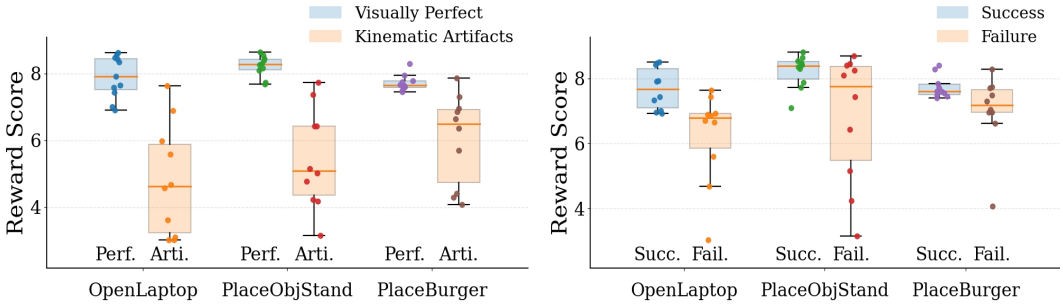
Figure S2: Generated video comparison on LIBERO. Base is shown in the top row and EVA in the bottom row. The left four columns show placing the soup can in the basket (LIBERO-Object), and the right four show opening the middle drawer (LIBERO-Goal). Panel (a) samples frames 12, 24, 36, and 48 in chronological order from the third generated chunk of each model. Panel (b) shows representative Base artifacts from an earlier chunk and EVA interaction stages from subsequent planning rounds, rather than temporally matched frames. Red boxes highlight visible arm deformation in Base rollouts.

B Reward Diagnostics

We examine whether the IDM-based reward distinguishes rollouts with different visual quality and execution outcomes. Both analyses use three representative tasks, with ten rollouts per group for each task.

Visual quality. We compare visually plausible rollouts with rollouts exhibiting embodiment-related kinematic artifacts. Artifact-free videos generally receive higher scores in Figure S3(a), suggesting that the reward is sensitive to inconsistencies in generated robot motion.

Execution outcomes. We decode generated videos into actions and execute them in simulation, then compare ten successful and ten failed rollouts per task. Successful executions generally receive higher scores in Figure S3(b). Together, these observations support the reward as a useful surrogate within the sampled cases; they do not establish that a high reward guarantees task completion.



(a) Reward vs. visual artifacts

(b) Reward vs. execution success

Figure S3: Relationship between reward scores and rollout quality. (a) Reward scores grouped by whether the generated rollout contains visible embodiment-related artifacts. (b) Reward scores grouped by downstream execution outcomes in simulation. Results are shown for three representative tasks, with 10 rollouts per group for each task. Each point represents one rollout, and box plots summarize the reward distributions.

C Implementation and Training Details

Supervised fine-tuning. We first fine-tune the video generation model using supervised learning on 8 NVIDIA A800 GPUs with Fully Sharded Data Parallel (FSDP) and mixed-precision training in bf16. Each training sample consists of a 49-frame video clip. For the RoboTwin simulation dataset, the model is trained for approximately 4,500 optimization steps at an input resolution of 640×480. For the real-world dataset collected on the physical robot platform, the model is trained for approximately 2,580 optimization steps at a resolution of 832×480. The per-device batch size is set to 1 with gradient accumulation of 4 steps, giving a total batch size of 32. We use the AdamW optimizer with learning rate 8×10^{-6}, $\beta_1 = 0.9$, $\beta_2 = 0.95$, and weight decay of 0.05. The learning rate follows a constant schedule with 100 warm-up steps. We additionally enable diffusion forcing with random history conditioning during training to improve temporal consistency.

GRPO post-training. After supervised fine-tuning, we further align the model using GRPO on eight NVIDIA A800 GPUs with a total batch size of 32. We optimize the policy with AdamW using a learning rate of 2×10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.95$, and weight decay of 0.05. For each prompt, 8 candidate videos are sampled for policy optimization. The clipping range is set to 0.001, with advantage clipping at 5.0 and a maximum gradient norm of 1.0. We also apply KL regularization with coefficient $\beta = 0.004$ during optimization. Only the LoRA parameters (rank 32) are updated, while the backbone remains frozen. In the simulation setting, GRPO is trained for 136 optimization steps with 4 inner policy epochs per iteration, taking approximately 6 days. In the real-world setting, we use 2 inner policy epochs and train for 46 optimization steps, taking approximately 3 days. The corresponding training reward curves are shown in Figure S4.

For RoboTwin and the real-world setting, we use $(\lambda_j, \lambda_\alpha, \lambda_{v\text{-lim}}, \lambda_{a\text{-lim}}) = (0.5, 0.25, 2.0, 1.0)$ and $\Delta t = 1/16\,\mathrm{s}$. For LIBERO, whose OSC controller operates at 20 Hz, we use $\Delta t = 1/20\,\mathrm{s}$. The

Huber thresholds are $\delta_\alpha = 0.5\alpha_{\max}$ and $\delta_j = 0.5\alpha_{\max}/\Delta t$. Rewards are computed from unstandardized actions, with gripper dimensions weighted at 0.2 relative to arm joints. The reward is the negative weighted sum of the four penalties defined in the main text.



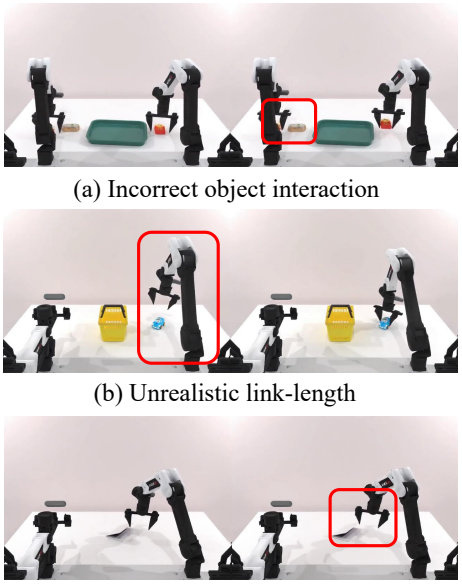
(a) Simulation setting (b) Real-world setting

Figure S4: Training reward curves in simulation and real-robot experiments. Rewards are normalized to $[0, 10]$ for visualization. The blue curve shows the normalized reward, while the orange curve shows its exponential moving average (EMA). In both settings, the reward exhibits a generally increasing trend during alignment, showing the evolution of the optimization objective.

LIBERO implementation details. The IDM predicts a 7-D command $q_t = (p_t, r_t, g_t)$, comprising absolute end-effector position $p_t \in \mathbb{R}^3$, absolute rotation-vector orientation $r_t \in \mathbb{R}^3$, and gripper command g_t. At 20 Hz, each target pose is converted into an incremental operational-space control (OSC) command relative to the observed pose, using the shortest relative rotation. Rewards are computed before OSC conversion from component-wise first-, second-, and third-order differences of q_t with $\Delta t = 1/20\,\mathrm{s}$, following Sec. 4.2. The soft coordinate-wise thresholds are

$$v_{\max} = [0.75, 0.75, 0.75, 4, 4, 4, 30],$$
$$a_{\max} = [10, 10, 10, 50, 50, 50, 450].$$

For position, rotation-vector, and gripper channels, first-order units are $\mathrm{m/s}$, $\mathrm{rad/s}$, and s^{-1}; second-order units are $\mathrm{m/s^2}$, $\mathrm{rad/s^2}$, and s^{-2}. Rotational differences serve as coordinate-space smoothness proxies. The gripper weight is 0.2 relative to pose channels. Each suite is evaluated over 200 trials.



(a) Incorrect object interaction

(b) Unrealistic link-length

(c) Static behavior

Figure S5: High-reward failure cases during GRPO post-training.

Inference settings. Long-horizon video generation uses 49-frame chunks and conditions the next chunk on the last four frames of the preceding chunk. In closed-loop control, the IDM translates each generated chunk into an action chunk, which is executed before the next chunk is generated from the resulting camera observation. These two modes share the same generator and IDM but differ in whether the generated chunk is only concatenated into a video or executed on the robot.

Reward hacking. The IDM-based reward is designed as a surrogate for motion-level executability rather than task completion. During the early and intermediate stages of post-training, increasing reward is generally accompanied by improvements in robot-motion integrity and downstream execution. With prolonged optimization, however, the proxy can be over-optimized: we observe high-reward failures including incorrect contacts, unrealistic robot geometry, and nearly static rollouts that do not complete the task (Figure S5).

To mitigate this behavior, we use KL regularization to constrain deviation from the reference video model and monitor a held-out validation set that is excluded from both policy optimization and final test evaluation. During training, we qualitatively assess two validation criteria: **robot integrity**, which captures embodiment consistency and temporal continuity, and **task progress**, which assesses whether generated rollouts make meaningful progress toward the instructed goal. We stop optimization once improvements on these validation criteria have largely saturated, before continued reward maximization leads to persistent visible degeneration. The checkpoint at this stopping point is used for evaluation; no downstream test success is used for stopping or model selection.

Synthetic data and VLA training. For each data source in Table 3, Base and EVA, we generate 50 trajectories per task for Click Bell, Place A2B Right, and Place Container Plate, yielding 150 trajectories per source. Generation starts from the head-camera images of the corresponding demonstration episodes, and each generated video matches its source episode's frame count. The frozen IDM decodes the generated videos into actions. For each data source, we initialize a separate $\pi_{0.5}$ policy from the same official pretrained checkpoint and jointly fine-tune it on all three tasks. Both conditions use 10,000 training steps, a batch size of 64, an action horizon of 40, and eight GPUs with FSDP. The policy uses only head-camera images, with wrist-camera inputs disabled. Evaluation uses the RoboTwin clean setting with the same head-camera configuration as the source data, with 100 trials per task.

D IDM Architecture and Ablation

Architecture. The IDM follows a standard visuomotor design (Levine et al., 2016): an ImageNet-pretrained ResNet-50 (He et al., 2016) extracts spatial features, a spatial softmax layer converts each channel into a 2D coordinate, and an MLP maps these coordinates to actions. To accommodate temporally stacked RGB frames, we expand the first convolution along the input-channel dimension, initializing its weights by repeating the pretrained filters and dividing by the number of input frames. The MLP has two hidden layers of widths 1,024 and 512. Let $F \in \mathbb{R}^{C\times H\times W}$ denote the feature map obtained from temporally stacked input frames. For feature channel m, spatial softmax is defined as

$$p_{ij}^{m} = \frac{\exp(F_{ij}^{m})}{\sum_{i',j'} \exp(F_{i'j'}^{m})}, \qquad (x_m, y_m) = \sum_{i,j} p_{ij}^{m}(i,j). \tag{8}$$

The coordinates $\{(x_m, y_m)\}_{m=1}^{C}$ are concatenated and fed into an MLP to predict q_t.

Spatial-representation ablation. We evaluate the IDM's spatial representation by replacing spatial softmax with global average pooling. Test accuracy is the fraction of predicted action vectors for which all 14 dimensions satisfy the error thresholds simultaneously: absolute error below 0.05 rad for each arm joint and below 0.5 for each $0/1$-encoded gripper command. We also evaluate task success by executing the decoded actions in simulation.

As shown in Table S1, replacing spatial softmax reduces prediction accuracy from 0.9864 to 0.7738 and execution success from 89.52% to 84.29%. This comparison supports retaining explicit spatial features in the action decoder.

Table S1: Ablation study on the IDM architecture. Test accuracy requires all 14 action dimensions to have absolute errors below 0.05 rad for arm joints and 0.5 for $0/1$ gripper commands simultaneously. Execution success is measured in simulation.

Inverse Dynamics Model	Test Accuracy	Test Success Rate
Spatial softmax	0.9864	89.52%
Global average pooling	0.7738	84.29%

Cross-architecture IDM evaluation. Since the same IDM provides rewards and decodes actions in our main experiments, we test whether EVA's gains persist with a decoder that did not guide alignment. The evaluation IDM uses a ViT-S/16 backbone, spatial softmax, and an MLP action head to predict q_t. It is independently trained on the same robot demonstrations with the same regression

objective. This IDM is used exclusively for evaluation, with no involvement in EVA post-training or reward computation. Decoding Base and EVA videos with this same ViT-IDM yields success rates of 43.5% and 49.7%, respectively (Table S2). The 6.2-point gain closely matches the original IDM's 6.5-point gain, suggesting that EVA's improvements to generated motion transfer to a decoder with a different visual backbone, beyond the one supplying the reward.

Table S2: Cross-architecture IDM evaluation. Entries report execution success rates (%); Δ denotes the improvement in percentage points. ResNet IDM-A is used for reward computation, while ViT IDM-B is used only for evaluation.

Video planner	ResNet IDM-A	ViT IDM-B
Base	45.8	43.5
EVA	52.3	49.7
Δ	+6.5	**+6.2**

E ROBOTWIN DIRECT-EXECUTION RESULTS

Table S3 compares performance on 21 RoboTwin tasks under `demo_clean`. The baselines use single-task policies, while Base and EVA use a single multi-task video model. Relative to Base, EVA improves success on 18 tasks and matches it on two; Stamp Seal decreases from 13/50 to 8/50.

Table S3: RoboTwin clean-setting results on 21 tasks, comparing ACT (Zhao et al., 2023), DP (Chi et al., 2023), RDT (Liu et al., 2024), and π_0 (Black et al., 2025). Baseline entries are rounded /50 equivalents of published success rates; Base and EVA report observed successes out of 50 trials. Baseline averages are computed from the rounded entries.

Task	ACT	DP	RDT	π_0	Base	EVA
Click Alarmclock	16/50	31/50	31/50	32/50	48/50	50/50
Click Bell	29/50	27/50	40/50	22/50	42/50	48/50
Handover Block	21/50	5/50	23/50	23/50	0/50	0/50
Handover Mic	43/50	27/50	45/50	49/50	1/50	8/50
Move Pillbottle Pad	0/50	1/50	4/50	11/50	8/50	16/50
Move Stapler Pad	0/50	1/50	1/50	0/50	0/50	1/50
Open Laptop	28/50	25/50	30/50	43/50	15/50	25/50
Place A2B Right	0/50	7/50	1/50	14/50	26/50	28/50
Place Bread Basket	3/50	7/50	5/50	9/50	36/50	38/50
Place Burger Fries	25/50	36/50	25/50	40/50	30/50	32/50
Place Cans Plasticbox	8/50	20/50	3/50	17/50	20/50	22/50
Place Container Plate	36/50	21/50	39/50	44/50	34/50	40/50
Place Dual Shoes	5/50	4/50	2/50	8/50	2/50	3/50
Place Mouse Pad	0/50	0/50	1/50	4/50	10/50	15/50
Place Object Basket	8/50	8/50	17/50	8/50	6/50	6/50
Place Object Stand	1/50	11/50	8/50	18/50	28/50	33/50
Press Stapler	16/50	3/50	21/50	31/50	44/50	49/50
Shake Bottle	37/50	33/50	37/50	49/50	46/50	47/50
Shake Bottle Horizontally	32/50	30/50	42/50	50/50	49/50	50/50
Stamp Seal	1/50	1/50	1/50	2/50	13/50	8/50
Turn Switch	3/50	18/50	18/50	14/50	23/50	30/50
Success Rate	29.71%	30.10%	37.52%	46.48%	45.81%	**52.29%**

F Real-World Experimental Setup

F.1 Robot Platform Setup

Our real-world experiments are conducted on an AgileX CobotMagic platform, as shown in Figure S6 and Table S4. We use its dual-arm tabletop configuration, where each task may involve either single-arm or dual-arm manipulation depending on the task requirement.



Figure S6: Real-world tabletop manipulation platform used in our experiments, built with AgileX PiPER robotic arms in a fixed multi-arm configuration.

Table S4: Key manipulator and gripper specifications of the AgileX PiPER arms used in our real-world experiments.

Component	Specification
Manipulator type	AgileX PiPER
Arm DoF	6 DoF per arm
Reach	626.75 mm
Repeatability	± 0.1 mm
Rated payload	1.5 kg per arm
Joint motion range	J1: $\pm 154^\circ$; J2: 0°–195°; J3: -175°–0°; J4: -106°–106°; J5: -75°–75°; J6: $\pm 100^\circ$
Joint max speed	J1: 180°/s; J2: 195°/s; J3: 180°/s; J4/J5/J6: 225°/s
Gripper type	Two-finger gripper
Gripper opening range	0–70 mm
Gripper accuracy	± 0.5 mm
Rated clamping force	40 N
Max clamping force	50 N

F.2 Task Descriptions

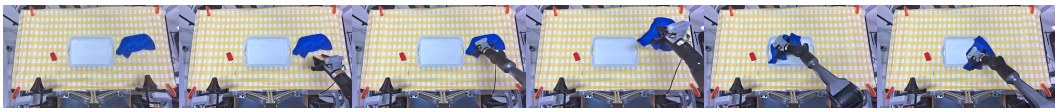
We summarize the real-world evaluation tasks in Table S5. The first five tasks are seen during training; the remaining five evaluate out-of-distribution (OOD) manipulation. Each task is evaluated in 20 trials.

Table S5: Real-world evaluation tasks used in our experiments.

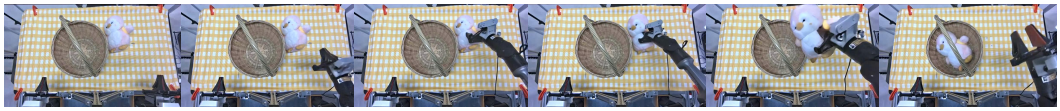
Task Name	Arm Type	Task Description
StackBowl	Dual-Arm	The robot uses both arms to grasp two bowls from the two sides and stack them onto the target bowl placed at the center of the table.
HangCable	Single-Arm	The robot uses a single arm to grasp a black cable and hang it onto the designated rack.
Place2Basket	Single-Arm	The robot uses a single arm to pick up an object from the tabletop and place it into a nearby basket.
Place2Tray	Single-Arm	The robot uses a single arm to grasp an object from the table and place it onto the tray.
FoldTowel	Dual-Arm	The robot uses both arms to grasp the two bottom corners of a towel and fold it upward.
PlaceBlock (OOD)	Single-Arm	The robot uses a single arm to pick up a block and place it into the bowl with the specified color and position.
PourWater (OOD)	Single-Arm	The robot uses a single arm to grasp a bottle and pour water into a bowl on the table.
WipeTray (OOD)	Single-Arm	The robot uses a single arm to grasp a towel and wipe the white tray on the tabletop.
FoldCloth (OOD)	Dual-Arm	The robot uses both arms to grasp the two bottom corners of a piece of clothing and fold it upward.
PlaceToy (OOD)	Single-Arm	The robot uses a single arm to grasp a soft plush toy and place it into a basket.

G Additional Out-of-Distribution Video Generation

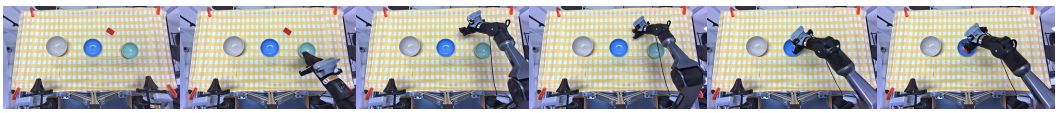
Figure S7 illustrates EVA rollouts on OOD task instructions. Each sequence starts from a conditioning image and shows the generated progression. These examples assess qualitative video generation rather than measured robot execution success.



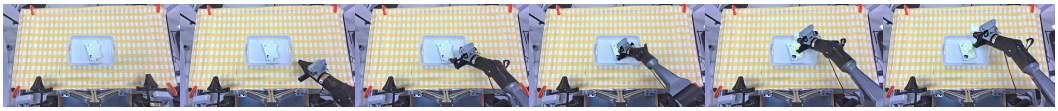
Grasp the blue towel and wipe the white tray.



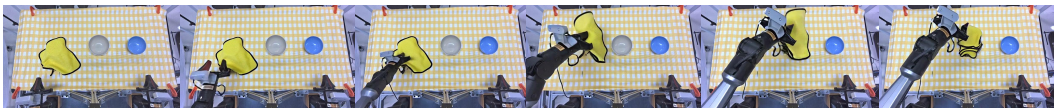
Grasp the plush toy and place it into the basket.



Pick up the red block with the right arm and place it into the middle blue bowl.



Pull out a tissue using the right arm.



Pick up the yellow towel and drop it into the middle bowl.



Fold the clothing upwards from the bottom edge using both arms.

Figure S7: EVA video generation on out-of-distribution tasks. Each row shows a synthesized video sequence of six evenly sampled frames, where the first frame serves as the input conditioning image and the instructions below correspond to the task prompts.

H SCALING EMBODIED DATA VIA ZERO-SHOT GENERATION

We also explore a fully synthetic input pipeline for generating manipulation videos. A text-to-image generator (e.g., Nano Banana Pro (Raisinghani, 2025)) produces initial scene images, which condition EVA to generate subsequent robot motion. Figure S8 shows representative sequences.

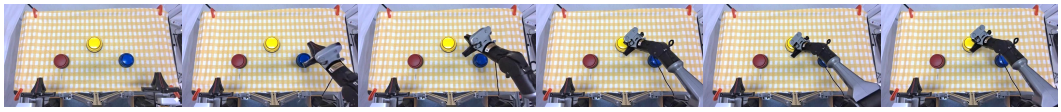
This pipeline offers a potential source of diverse visual trajectories without collecting a teleoperated demonstration for every scene. The examples illustrate qualitative generation from synthetic observations; downstream data utility requires policy-training evaluation, as considered in the WorldArena experiment.



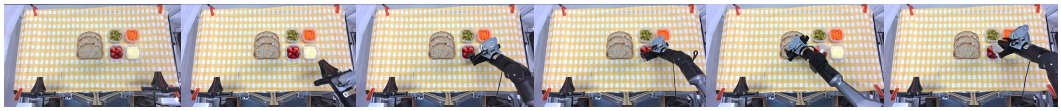
Grasp the cup and pour coffee into the basket.



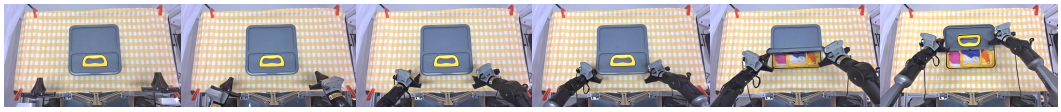
Use the left arm to grasp the apple and place it into the basket.



Press the yellow button with the left arm.



Spread strawberries on the bread.



Lift the box lid with both arms.



Place the shoe into the shoebox with the left arm.

Figure S8: EVA rollouts conditioned on synthetic initial scenes. Each row shows a generated video sequence whose first frame is supplied by the text-to-image generator.