WoRL: Efficient Reinforcement Learning Framework for Video World Models with Next-Video Reward Modeling

Anonymous authors
Paper under double-blind review

Abstract

While video world models have achieved remarkable progress, reinforcement learning (RL) remains largely underexplored for their post-training. RL offers a promising way to improve these models through self-generated rollouts without real and long videos with interactive signal, yet scaling it in autoregressive video generation remains challenging due to expensive long-video optimization, the lack of history-aware reward models, and limited exposure to accumulated consistency failures during standard rollouts. We introduce **WoRL**, an end-to-end RL framework that addresses these challenges through three components. (1) **Efficient Train–Test-Matched RL.** We eliminate the intra-clip train–test mismatch in autoregressive DiffusionNFT and introduce Parallel Clean-Prefix Attention, context-aware policy scheduling, and distributed execution to scale full-attention RL to large models and long video contexts. (2) **Next-Video Reward Model (NVRM).** NVRM evaluates each generated clip conditioned on its preceding video history, enabling history-aware preference optimization over long horizons. We further train the model with a graph-based data construction pipeline that builds 312.4k video-continuation preference pairs without human expert annotation. (3) **Revisit-Centered Training.** We construct trajectories that revisit observed scenes to expose accumulated inconsistencies and introduce rewind regularization to improve learning under limited generation diversity. WoRL achieves a $9.2\times$ training speedup over the naive implementation. NVRM reaches 74.4% consistency agreement and 82.4% physical-plausibility agreement with blinded human judgments, outperforming HPSv3 and 3D Score on both criteria. Across action-conditioned LingBot-World-Fast-14B and text-conditioned Helios-Distilled-14B, WoRL consistently improves long-term consistency, controllability, and physical plausibility, yielding a 37.5-percentage-point human-preference margin. The qualitative results are best visualized as videos in the supplementary material.

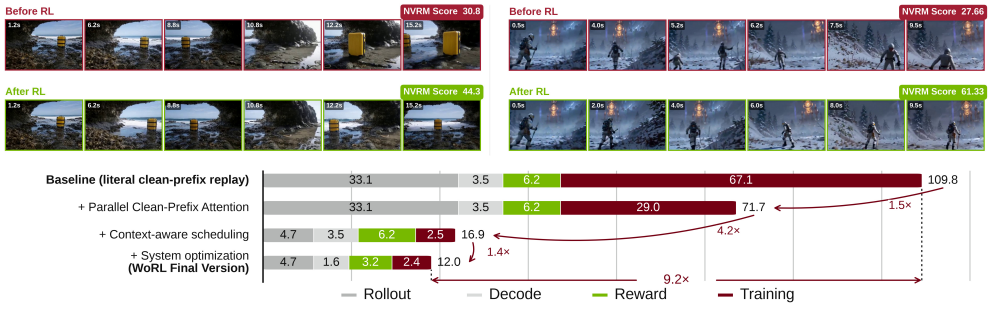


Figure 1: **WoRL is an efficient RL framework for improving the long-horizon quality of video world models. Top:** Matched rollouts before and after RL. With WoRL, the yellow suitcase (left) reappears consistently when the camera returns to the initial scene, while the main character (right) preserves its identity throughout the rollout instead of being replaced by a newly hallucinated character entering from the right. NVRM evaluates long-term consistency by scoring each generated continuation in the context of its preceding rollout history, assigning higher scores to continuations that remain consistent with that history. **Bottom:** Cumulative latency ablation on 16 B200 GPUs. Our optimizations reduce the update time from 109.8 to 12.0 minutes. Segment labels denote latency in minutes, while red arrows indicate stage-wise and overall speedups.

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.

1 INTRODUCTION

Autoregressive video world models create interactive pixel worlds by repeatedly extending their own predictions. Recent advances in long-video training and few-step distillation have enabled minute-scale rollouts (Huang et al., 2025a;b; Zhu et al., 2026b; Gao et al., 2026; Zhu et al., 2026a). As the generation horizon increases, however, the dominant failure mode shifts from local artifacts to accumulated long-term inconsistencies. A newly generated clip may appear realistic in isolation while contradicting entities, geometry, or physical states established much earlier; once introduced, these errors become part of the context and propagate through subsequent predictions. Supervised fine-tuning on expert videos is difficult to correct such self-induced errors because the model is not trained on its own imperfect rollout histories (Yin et al., 2025), while long, control-aligned videos with expert annotation are expensive to collect. Distribution-matching distillation over model-generated histories reduces this train–test mismatch (Huang et al., 2025b; Yang et al., 2025), but typically relies on a stronger teacher and primarily transfers the local distribution of short clips. Global inconsistencies can therefore persist even when every generated clip appears plausible.

Reinforcement learning (RL) provides a complementary post-training paradigm. It has improved image and bidirectional video diffusion models (Black et al., 2023; Liu et al., 2025b;a; Xue et al., 2025b) by directly optimizing properties such as visual text rendering (Liu et al., 2025a) and physical plausibility (Wang et al., 2026b), which are difficult to capture with standard denoising objectives alone. More importantly, RL learns from feedback on self-generated rollouts, exposing the model to inference-time states and failures without requiring long videos paired with interactive supervision, which are substantially harder to collect than passive video alone. This is particularly valuable for autoregressive world models, where each prediction conditions on the model's own generated history and inherits accumulated errors. However, scaling RL to long-horizon autoregressive video generation presents three unresolved challenges.

Scaling Train–Test-Matched RL for Video World Models. Online RL becomes prohibitively expensive as model size and rollout horizon increase. Recent methods adopt forward-process objectives such as DiffusionNFT (Zheng et al., 2025) on short clips with detached video history to avoid training computation over the long horizon (Wang et al., 2026c; Zhang et al., 2026). Clip-level DiffusionNFT keeps the rollout history clean but noising the within-clip context still introduces a train–test mismatch. Naively matching the rollout process requires block-wise evaluation and repeatedly recomputes the same growing prefix. This redundancy compounds across sampled candidates and policy evaluations during RL (Liu et al., 2025a; Xue et al., 2025b; Zheng et al., 2025), while existing approaches do not explicitly reuse the shared context across these stages. Scalable long-video RL requires both rollout-matched causal conditioning at the algorithm level and efficient management of the growing autoregressive context at the system level.

Reward Modeling for Long-Horizon Quality. Long-horizon failures are defined relative to the generated history rather than within an isolated clip. Existing visual reward models primarily assess local appearance, motion, prompt alignment, or physical plausibility (He et al., 2024; Ma et al., 2025; Liu et al., 2025b). Recent RL methods for autoregressive video world models inherit this limitation: WorldCompass and Astrolabe optimize rewards computed from short target clips without explicitly evaluating their long-term quality with the preceding rollout (Wang et al., 2026c; Zhang et al., 2026). A realistic continuation may receive a high reward even if it changes an established character, resets an object state, or contradicts earlier geometry. While large multi-modal models (Google DeepMind, 2026; OpenAI, 2026) have shown strong capability to understand long videos semantically, their judgments are significantly unstable as scalar rewards (Chen et al., 2024; Shi et al., 2024). Long-video RL therefore requires a deployable reward model that can assess long-term video quality.

Training Environments for Long-Term Improvement. RL can improve a long-horizon behavior only when the rollout distribution both exposes the corresponding failure and contains alternatives that the reward can distinguish. Generic forward-moving camera trajectories may never return to an established object or scene, leaving accumulated identity and geometry errors unobserved. Conversely, unconstrained trajectories that move far outside the base model's competence produce uniformly poor candidates and weak learning signals. An effective training environment must create delayed but observable dependencies across clips while remaining within a regime in which the model can produce meaningful positive and negative continuations. Constructing such environments

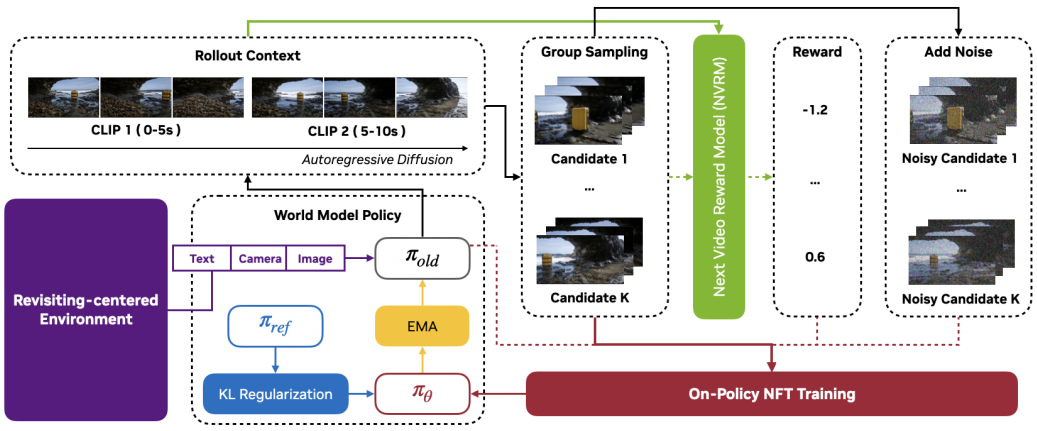


Figure 2: **Overview of WoRL including algorithm-system co-design, reward model and training environment.** The sampling policy π_{old} generates a history and branches K candidate next clips from the same state. NVRM scores each candidate **in the context of that history**. Selected clips are trained block by block: only the current block is noised, while its generated predecessors remain clean. KL regularization anchors the learned policy π_θ to π_{ref}, and an exponential moving average updates the sampler for the next iteration.

is therefore not merely data collection: it determines which long-term capabilities the RL loop can diagnose and improve.

To address these challenges, we introduce **WoRL**, an end-to-end RL framework for improving long-horizon autoregressive video world models. WoRL integrates three key components:

- **Efficient System for Train–Test-Matched RL.** We introduce Parallel Clean-Prefix Attention, which evaluates autoregressive target blocks in parallel while preserving the clean causal context used during rollout. At the system level, context-aware scheduling and group-local distributed execution reuse shared prefix computation across candidates and policy roles. Together, these designs enable efficient long-context RL while preserving train–test matching.
- **Next-Video Reward Model (NVRM).** We develop NVRM, an autoregressive reward model that evaluates each generated clip conditioned on its complete rollout history. To train NVRM without human annotation, we introduce a graph-based curation pipeline that consolidates noisy multimodal comparisons into consistent within-history preference labels. Trained on the resulting high-quality long-video preferences, NVRM provides scalable, context-aware rewards for long-term consistency and physical plausibility.
- **Revisit-Centered Training Environment.** We construct diverse object- and landmark-centered scenes with trajectories that leave and later revisit previously observed content, making accumulated changes in identity, geometry, and object state directly observable. Rewind regularization further supplies informative return candidates when successful sampled revisits are sparse, strengthening the learning signal for long-horizon behavior.

We evaluate WoRL across two complementary settings: the action-conditioned LingBot-World-Fast model (Gao et al., 2026) and the text-conditioned Helios model (Yuan et al., 2026). Under identical hardware settings, WoRL achieves a $9.2\times$ end-to-end training speedup over the baseline. NVRM attains 74.4% agreement with blinded human judgments on long-term consistency and 82.4% on physical plausibility, outperforming existing reward model by 16.3 and 35.3 percentage points, respectively. Across both models, WoRL consistently improves long-term consistency, camera controllability, and physical plausibility. In blinded human evaluation, WoRL is preferred in 53.1% of comparisons, compared with 15.6% for the baseline. These results demonstrate that WoRL makes policy optimization for long-term quality tractable in video world models.

2 Preliminaries

A video policy π_θ generates each clip $x^t = (x_1^t, \ldots, x_J^t)$ as J causal diffusion blocks, conditioned on history $h^t = (x^0, \ldots, x^{t-1})$ and inputs $c = (I, y, a)$: observation, instruction, and optional

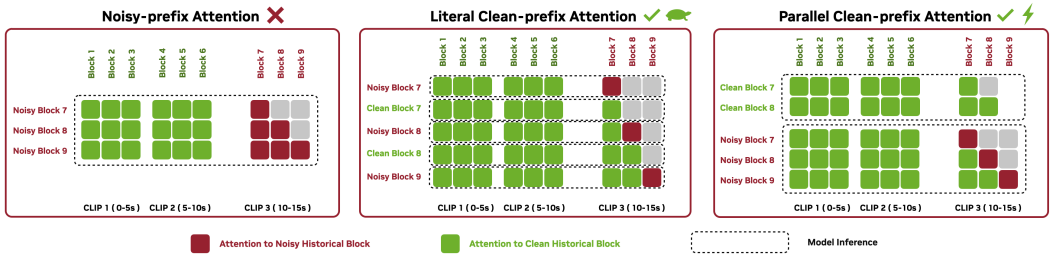


Figure 3: **Clean-prefix attention for train–test-matched RL. Left:** Noisy-prefix attention introduces a train–test mismatch because each target block can attend to preceding *noised* target blocks, whereas autoregressive rollout conditions on clean generated history. **Middle:** Literal clean-prefix attention restores clean predecessors separately for each noised target, preserving the test-time history at the cost of repeated inference. **Right:** Parallel Clean-Prefix Attention preserves clean history with one packed clean-store execution and one shared packed noisy-read execution.

control. Block $x^t_{j,0}$ is clean; $x^t_{j,\tau}$ is noised at diffusion time τ. DiffusionNFT (Zheng et al., 2025) re-noises rewarded candidates and uses a negative-aware forward-process objective to favor high-reward samples and suppress low-reward samples. The sampler π_{old} supplies candidates, π_θ receives gradients, and frozen π_{ref} regularizes updates; an EMA updates the sampler. Clip-wise training jointly noises the target blocks, whereas inference conditions on clean predecessors. Appendix B gives the full factorization and objective; related work appears in Appendix A.

3 Efficient System for Train–Test-Matched RL

Clip-wise training preserves history across clips but jointly noises blocks within the current clip. Consequently, a target block is optimized against corrupted predecessors, whereas rollout conditions on clean generated predecessors. This mismatch recurs at every generated clip and can compound over long autoregressive rollouts. We therefore modify policy evaluation to preserve the rollout conditioning structure during training, while leaving the reward, candidate selection, and DiffusionNFT objective unchanged.

Parallel Clean-Prefix Attention. For candidate i in shared-history group g, block j should condition on clean history h_g and clean predecessors $x_{gi,<j,0}$. We noise only the target block:

$$x_{gi,j,\tau} = \alpha_\tau x_{gi,j,0} + \sigma_\tau \varepsilon, \qquad v_\theta(x_{gi,j,\tau}, \tau \mid h_g, x_{gi,<j,0}, c_g), \tag{1}$$

where $\varepsilon \sim \mathcal{N}(0, \mathbf{I})$. Sampling and reference policies use the same clean-prefix context without gradients. Standard causal masking does not satisfy this condition because earlier blocks in the current clip remain noised (Figure 3); literal replay does, but requires J forwards per policy.

PCPA realizes the same conditioning pattern with two attention executions per layer. The first computes and detaches each policy's KV cache for the shared history and clean prefix. The second packs independently noised target blocks and evaluates them in parallel. Each target attends to the shared history and earlier clean blocks, but not to other noisy targets. Gradients therefore pass only through the target computation, while prefix states are reused across targets. This preserves Equation 1 without repeatedly projecting the same prefix. Appendix G.5 gives the complete update.

Context-aware policy scheduling. The three policies π_{old}, π_θ, and π_{ref} have different weights. Each policy must therefore construct its own history KV state. During training, we evaluate all compatible candidates under one policy before switching to the next. The policy-major schedule amortizes cache construction across compatible candidates and avoids repeated KV-cache rebuilding and CPU offloading as Figure 4 shows. More details are in Appendix C.

Distributed execution. FSDP shards parameters, gradients, and optimizer states, while sequence parallelism partitions attention and KV states for full-attention rollout and training. Data-parallel replicas process disjoint groups across nodes. When memory permits, each group remains local through generation, decoding, reward computation, and selection. Decoding is performed in bounded microbatches and overlapped with CPU post-processing and asynchronous reward evaluation. Global synchronization is needed only for group-level normalization, selection, and op-

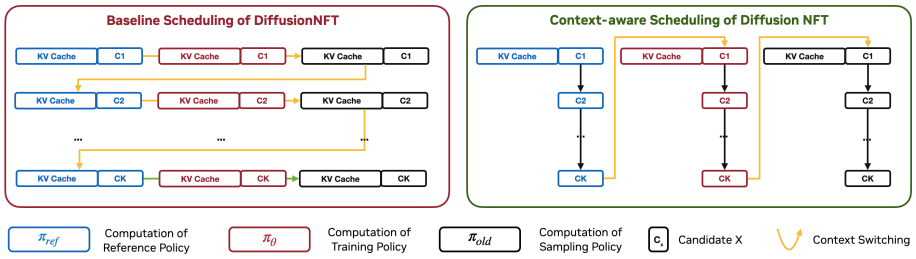


Figure 4: **Context-aware Scheduling. Left:** Candidate-major DiffusionNFT repeatedly switches among the reference, training, and sampling policies, rebuilding or offloading their policy-specific K/V caches. **Right:** Context-aware scheduling groups candidate computations under each active policy so that the corresponding cache remains reusable, reducing unnecessary context switches without sharing K/V across policies.



Figure 5: **Qualitative visualization of NVRM.** Each example shows the preceding video history (left) and two candidate continuations generated from the same history (right). NVRM is trained to assign a higher score to the continuation that better preserves the objects, characters, and scene state established by the history.

timizer updates, limiting cross-node communication to scalar statistics and optimizer reductions (Appendix G).

4 Next-Video Reward Modeling

A clip can look plausible in isolation yet contradict its history (Figure 5); for example, visually plausible lighting may remove red warning lights established earlier. A clip-only reward cannot capture such inconsistencies. We therefore score each continuation conditioned on its generated history.

History-conditioned utility. Let $r(x^{1:T}, c)$ denote the utility of a generated video. Under the optimality-weighted distribution used by DiffusionNFT (Zheng et al., 2025),

$$p^*(x^{1:T} \mid c) = \frac{1}{Z(c)} p_0(x^{1:T} \mid c) \exp\left(\frac{r(x^{1:T}, c)}{\beta}\right), \tag{2}$$

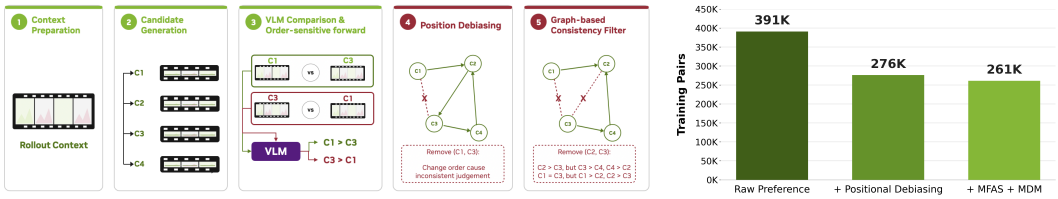
where p_0 is the reference distribution and $Z(c)$ normalizes it. Autoregressive factorization gives

$$\begin{aligned} r(x^{1:T}, c) &= \beta \sum_{t=1}^{T} \log \frac{p^*(x^t \mid x^{<t}, c)}{p_0(x^t \mid x^{<t}, c)} + \text{const}(c) \\ &= \sum_{t=1}^{T} r_t(x^t \mid x^{<t}, c) + \text{const}(c). \end{aligned} \tag{3}$$

The relevant reward is therefore the conditional utility of a continuation. NVRM models it as $q_\phi(x_0^{<t}, c, x_0^t)$. A history-independent score cannot penalize changes to previously established objects, identities, or scene state (Appendix D).

Pairwise learning. Following Bradley–Terry (Bradley & Terry, 1952), we train on preferred and dispreferred continuations sharing the same history and condition:

$$\mathcal{L}_{\text{NVRM}} = -\mathbb{E}_{(x_0^{<t}, c, x_{+,0}^t, x_{-,0}^t) \sim \mathcal{D}_{\text{pref}}} \log \sigma\big(q_\phi(x_0^{<t}, c, x_{+,0}^t) - q_\phi(x_0^{<t}, c, x_{-,0}^t)\big). \tag{4}$$



(a) Group-based curation pipeline. (b) Pair-retention statistics.

Figure 6: **Graph-based fake–fake preference curation. (a)** Candidates share a rollout context and are compared in both display orders. Position debiasing removes order conflicts, MFAS removes cycles, and MDM blocks unsupported paths. **(b)** Filtering reduces 391K raw pairs to 276K after position debiasing and 261K after MFAS and MDM.

The shared context isolates continuation quality rather than differences in history. Only within-history score differences are supervised; unrelated histories need not share an absolute scale. NVRM uses Qwen3-VL-4B-Instruct with LoRA and a scalar head. Appendix H.2 details the architecture and frame sampling.

Preference data and curation. Real trajectories provide only one observed future, not competing continuations from the same history. We therefore combine real–fake and fake–fake supervision. Real–fake pairs compare a ground-truth future with a generated continuation under the same real history, while fake–fake pairs compare branches from the model's own generated history and expose identity drift, inconsistent scene state, and implausible long-term consequences.

We generate multiple continuations from shared histories and compare them pairwise. Multimodal judges (Google DeepMind, 2026; OpenAI, 2026) evaluate both display orders; only comparisons with the same winner are retained. We then construct a preference graph: minimum feedback arc set removes cyclic contradictions, while minimum directed multicut prunes unsupported paths involving ambiguous pairs. Neither step adds transitive pseudo-labels. Position debiasing reduces 391K fake–fake comparisons to 276K, and graph filtering retains 260.5K. Adding 51.8K real–fake anchors yields 312.4K pairs, as shown in figure 6. Appendices D.2 and H.1 provide further details.

Reward integration. NVRM evaluates consistency and physical plausibility against the generated history. We combine it with HPSv3 (Ma et al., 2025) for visual quality and a control term r^{ctrl}: camera-control reward for action-conditioned models and dynamic score for text-conditioned models. These rewards have different numerical scales, so each component uses fixed pre-RL normalization $\hat{r}_m(r) = (r - \mu_m)/\sigma_m$. We keep μ_m and σ_m fixed throughout training to maintain the relative scale of the reward components (Appendix H.4):

$$R_{gi} = \hat{r}_{\text{HPS}}(r_{gi}^{\text{HPS}}) + \hat{r}_{\text{ctrl}}(r_{gi}^{\text{ctrl}}) + \hat{r}_{\text{NVRM}}(r_{gi}^{\text{NVRM}}), \tag{5}$$

5 Revisit-Centered Training Environment

Forward trajectories can hide drift by never returning to earlier content. We construct revisit trajectories that explicitly expose these failures (Figure 7). GPT-5.5 (OpenAI, 2026) generates diverse scenes, and a camera-planning agent creates anchor–departure–return trajectories in OpenScene: it establishes a recognizable object or landmark, moves away, and later returns to overlapping content. Automated validation and resampling retain valid paths. Most groups branch during the return after departure clips are committed; a smaller fraction uses ordinary short-horizon groups. NVRM evaluates each continuation against the generated history while allowing changes in viewpoint, illumination, occlusion, and scene state. Appendix G.6 gives the full construction.

Rewind regularization. Revisit trajectories expose accumulated drift, but distilled models can produce uniformly poor returns with little preference contrast. We increase candidate diversity by reversing the preceding committed clip and camera traversal and adding it as another candidate. This partially off-policy regularizer receives the same reward, ranking, and selection as generated samples, without privileged labels. Implausible reversals, such as reversed walking or falling, can simply receive low reward. More details are included in Appendix G.6.

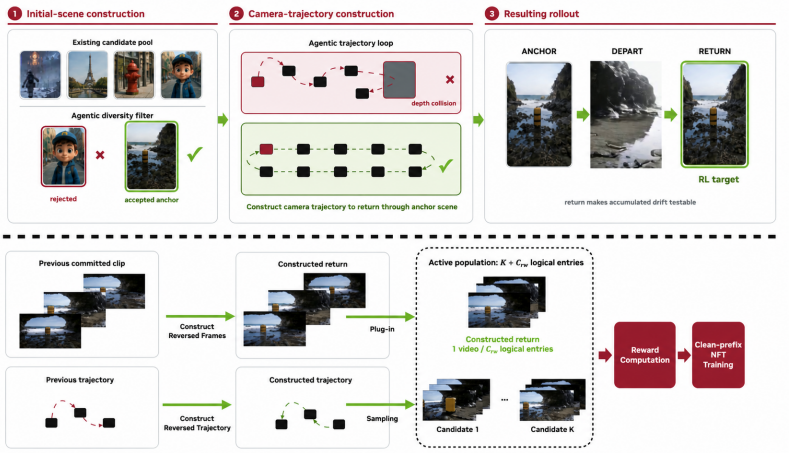


Figure 7: **Revisit-centered training. Top:** An anchor–departure–return trajectory brings earlier scene evidence back into view, turning accumulated drift into a scoreable next-clip error. **Bottom:** When successful sampled returns are sparse, rewind reverses the preceding committed clip and camera path to add candidate support. The constructed return and freshly sampled candidates are judged and selected by the same reward.

Table 1: **Main results on streaming action-to-video and long text-to-video generation. (a)** Action-to-video models on WBench-Navigation and RevBench. **(b)** Text-to-video models on WBench-Text and VBench-Long. Scores use a $[0, 100]$ scale and higher is better. Red arrows show the matched change from each base model to $+WoRL$. † marks a training-reward metric or its aggregate; in panel (a), $*$ marks a column containing VLM-backed submetrics.

(a) Streaming Action-to-Video Models

	WBench-Navigation						RevBench		
Method	Overall	Quality	Settings*	Interaction	Consistency	Physical*	Overall†	NVRM†	PI Camera†
HY-WorldPlay (Sun et al., 2025)	79.80	78.13	92.62	87.53	86.87	53.86	47.34	59.74	34.93
Matrix-Game 3.0 (Wang et al., 2026d)	73.99	75.43	92.16	83.54	74.52	44.31	45.32	53.70	36.94
SANA-WM-distilled (Zhu et al., 2026a)	76.24	77.33	91.83	83.00	80.46	48.60	42.45	44.02	40.89
LingBot-World-Fast (Gao et al., 2026)	77.45	79.34	92.15	79.16	84.95	51.64	43.85	43.48	44.22
+ WorldCompass (Wang et al., 2026c)	78.22	79.47	92.92	80.93	84.99	52.78	44.99	43.99	45.99
+ **WoRL**	**79.91** ↑2.46	**79.63** ↑0.29	**94.02** ↑1.87	**82.45** ↑3.29	**86.96** ↑2.01	**56.51** ↑4.87	**48.63** ↑4.78	**49.33** ↑5.85	**47.94** ↑3.72

(b) Streaming Text-to-Video Models

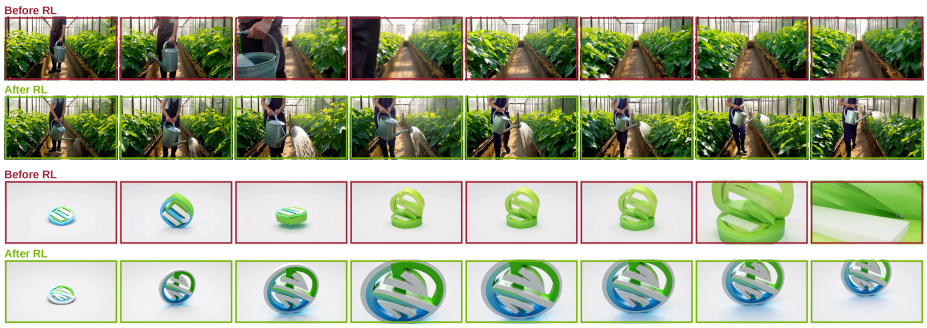
	WBench-Text						VBench-Long		
Method	Overall	Quality	Settings	Interaction	Consistency	Physical	VBench Score	NVRM†	Dynamic
LongCat-Video (Meituan LongCat Team et al., 2025)	70.57	79.38	96.49	29.60	87.68	59.70	81.70	58.39	66.70
HunyuanVideo-1.5 (Wu et al., 2025)	71.73	75.43	96.76	37.64	90.08	58.75	80.74	58.89	56.63
LTX-2.3 (HaCohen et al., 2026)	70.21	76.98	100.00	37.15	80.05	56.87	79.22	53.02	79.54
Helios-Distilled-14B (Yuan et al., 2026)	68.79	74.15	98.65	27.40	85.69	58.08	79.37	54.60	60.00
+ **WoRL**	**69.63** ↑0.84	**74.92** ↑0.77	**99.46** ↑0.81	**28.55** ↑1.15	**86.43** ↑0.74	**58.82** ↑0.74	**80.79** ↑1.42	**60.70** ↑6.10	**62.82** ↑2.82

6 Experiments

Experimental setup. We evaluate LingBot-World-Fast (Gao et al., 2026) and Helios-Distilled-14B (Yuan et al., 2026) before and after RL using matched prompts and settings, with three seeds per case. For LingBot, observations and controls are also matched. All scores are aggregated over matched prompt–seed pairs. WBench-Navigation (Ying et al., 2026) contains 158 held-out cases, while RevBench evaluates scene-state preservation on 150 out-of-training-distribution revisit trajectories. Helios is evaluated on WBench-Text and VBench-Long (Huang et al., 2024; 2025c).

Training and evaluation. For each of 16 shared-history groups, training samples 24 candidates and retains the 4 highest- and 4 lowest-scoring candidates. Component ablations use 16 B200 GPUs, 8 prompts, 12 candidates per prompt, and 40 steps. We report the LingBot checkpoint after 51 optimizer commits (approximately 50 updates) and the Helios checkpoint after 240 commits. The efficiency workload is profiled separately from these training runs. Full protocols and configurations are provided in Appendices I and G.

Long-horizon quality. Table 1 summarizes the main results. On WBench-Navigation, LingBot improves Overall from 77.45 to 79.91 (+2.46), with the largest gains in Physical (+4.87), Interaction (+3.29), and Consistency (+2.01). On RevBench, Overall increases from 43.85 to 48.63, with gains



(a) Text-to-video: Helios-Distilled-14B



(b) Action-to-video: LingBot-World-Fast

Figure 8: **Qualitative results before and after RL.** Each row compares eight chronological frames before RL (red) and after RL (green) under matched conditions. **(a)** WoRL better preserves long-term appearance and structure: the gardener remains in view and the emblem retains its 3D shape under camera motion. **(b)** WoRL improves world-state consistency: the cauldron maintains a coherent perspective, and the skull and blackboard remain consistent after revisiting.

in both NVRM and PI Camera. Helios improves all five WBench-Text dimensions, raising Overall from 68.79 to 69.63, and improves VBench-Long from 79.37 to 80.79. The gains hold across both action- and text-conditioned generation, including consistency and physical plausibility.

Efficiency. Table 2 measures end-to-end update latency through the optimizer commit. Literal replay requires 109.8 minutes, of which 67.1 minutes are spent on training. PCPA reduces training time to 29.0 minutes; policy scheduling lowers total latency to 16.9 minutes; and the full system reaches 12.0 minutes, a $9.2\times$ speedup. In the final configuration, rollout, decoding, reward computation, and training take 4.7, 1.6, 3.2, and 2.4 minutes, respectively. The resulting latency makes train–test-matched RL feasible for long-context, full-attention video world models.

Component ablations. Clean-prefix conditioning improves RevBench NVRM from 43.78 to 45.61. In a separate matched comparison, adding rewind improves NVRM from 44.91 to 45.61 (Table 4), isolating the effect of additional candidate contrast from clean-prefix conditioning. Table 3 compares reward-only training under matched initialization and rollout budgets. NVRM reaches 78.55 WBench Overall, compared with 78.30 for 3D Score and 78.19 for HPSv3.

Qualitative analysis. Figure 5 shows how rollout history changes continuation preferences. A locally plausible continuation may still contradict previously established scene and character identity. NVRM instead favors continuations that remain consistent with the preceding history. Figure 8 shows the same effects over full rollouts. The examples test persistence under action and camera motion: the gardener continues its action, while revisited objects retain their structure and identity. Additional examples are provided in Appendix F.3.

Human-evaluation protocol. We evaluate 64 matched video pairs with identical prompts and seeds. For reward–human agreement, we exclude human ties, leaving 43 consistency and 34 physical-plausibility comparisons. Model identities are hidden, display order is randomized, and trajectories are synchronized. Ties are recorded explicitly. Reward agreement measures continuation ranking, while post-training preference measures overall generation quality.

Table 2: **RL training efficiency on 16 NVIDIA B200 GPUs.** Each row uses 8 prompts, 12 candidates per prompt, and a third-clip target. Optimizations are added cumulatively; entries are minutes per committed update. The final measurement protocol is detailed in Appendix I.6.

System variant	Total	Rollout	Decode	Reward	Training
Baseline (literal clean-prefix replay)	109.8	33.1	3.5	6.2	67.1
+ Parallel Clean-Prefix Attention	71.7	33.1	3.5	6.2	29.0
+ Context-aware scheduling	16.9	4.7	3.5	6.2	2.5
+ System optimization	**12.0**	4.7	1.6	3.2	2.4

Table 3: **Reward comparison.** WBench Overall after matched training with each reward.

	Reward model		
Metric	HPSv3	3D Score	NVRM
WBench Overall ↑	78.19	78.30	78.55

Table 4: **Component comparisons.** Ablation study on clean-prefix and rewind contrasts on LingBot-World-Fast.

	Conditioning		*Rewind regularization*	
Metric	Noisy-prefix	Clean-prefix	w/o rewind	w/ rewind
RevBench NVRM ↑	43.78	45.61	44.91	45.61

Table 5: **Reward–human preference alignment.** Observed agreement rates between reward model and human preferences for consistency and physical plausibility.

Reward model	Consistency	Physical Plausibility
HPSv3 (Ma et al., 2025)	67.4%	64.7%
3D Score (Zhao et al., 2026)	58.1%	47.1%
NVRM (ours)	74.4%	82.4%

Table 6: **Human preference before and after RL.** Observed outcome rates for consistency, camera control, physical plausibility, and overall preference; ties are retained.

Criterion	Before RL	Tie	After RL
Consistency	15.6%	37.5%	46.9%
Camera control	3.1%	65.6%	31.3%
Physical plausibility	9.4%	46.9%	43.8%
Overall	15.6%	31.3%	53.1%

Human evaluation. NVRM agrees with blinded judgments in 74.4% of consistency comparisons and 82.4% of physical-plausibility comparisons, exceeding 3D Score by 16.3 and 35.3 percentage points, respectively (Table 5). For overall generation preference, After RL is preferred in 53.1% of comparisons, Before RL in 15.6%, with 31.3% ties, corresponding to a 37.5-percentage-point margin (Table 6).

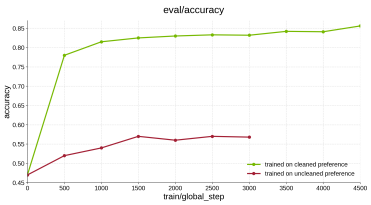


Figure 9: **NVRM preference cleaning.** Evaluation accuracy between the reward models trained on the complete cleaned and uncleaned data pipelines.

Preference curation. Figure 9 ablates bidirectional and graph-based filtering (Appendix F.2). At step 500, curation raises evaluation accuracy from approximately 0.52 to 0.78, and the gap remains through the shared training range up to step 3,000. Continued training on the curated set reaches approximately 0.86 at step 4,500. The reward model trained on curated data achieves higher evaluation accuracy significantly.

7 Conclusion

We presented WoRL, an efficient reinforcement-learning framework for post-training autoregressive video world models. At its core, WoRL aligns both policy optimization and reward evaluation with the autoregressive nature of video generation: each continuation is optimized under the generated history it is meant to continue. Parallel Clean-Prefix Attention and context-aware execution make train–test-matched optimization efficient, NVRM provides history-conditioned reward signals, and revisit-centered training explicitly exposes long-range consistency failures.

Across both action-conditioned and text-to-video models, WoRL consistently improves long-horizon generation quality, while the proposed system accelerates train–test-matched RL by $9.2\times$. Human evaluation further shows that NVRM better agrees with judgments of consistency and physical plausibility, and that the post-trained model is preferred overall. These results suggest that effective RL for video world models should optimize not only whether each generated clip is individually plausible, but whether it remains a coherent continuation of the world generated.

AI USE STATEMENT

We used AI tools to polish the language and improve the clarity of the manuscript. We used multimodal AI models to annotate pairwise preferences for NVRM, and GPT-5.5 to generate scene specifications for the revisit-centered training environment with Codex for camera-trajectory planning. The annotation and filtering procedures are described in Section 4 and Appendix H.1. The authors take responsibility for the paper's content, including AI-assisted writing and data annotation.

REFERENCES

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*, 2023.

Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952. doi: 10.1093/biomet/39.3-4.324.

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 https://openai.com/index/video-generation-models-as-world-simulators/.

Dongping Chen, Ruoxi Chen, Shilin Zhang, Yaochen Wang, Yinuo Liu, Huichi Zhou, Qihui Zhang, Yao Wan, Pan Zhou, and Lichao Sun. MLLM-as-a-judge: Assessing multimodal LLM-as-a-judge with vision-language benchmark. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 6562–6595. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/chen24h.html.

Yu Gao, Haoyuan Guo, Tuyen Hoang, Weilin Huang, Lu Jiang, Fangyuan Kong, Huixia Li, Jiashi Li, Liang Li, Xiaojie Li, et al. Seedance 1.0: Exploring the boundaries of video generation models. *arXiv preprint arXiv:2506.09113*, 2025.

Zelin Gao, Qiuyu Wang, Yanhong Zeng, Jiapeng Zhu, Ka Leong Cheng, Yixuan Li, Hanlin Wang, Yinghao Xu, Shuailei Ma, Yihang Chen, et al. Advancing open-source world models. *arXiv preprint arXiv:2601.20540*, 2026.

Google DeepMind. Veo: A text-to-video generation system. Technical Report, 2025. URL https://storage.googleapis.com/deepmind-media/veo/Veo-3-Tech-Report.pdf.

Google DeepMind. Gemini 3.5 Flash. Google AI for Developers, 2026. URL https://ai.google.dev/gemini-api/docs/whats-new-gemini-3.5.

Yoav HaCohen, Benny Brazowski, Nisan Chiprut, Yaki Bitterman, Andrew Kvochko, Avishai Berkowitz, Daniel Shalem, Daphna Lifschitz, Dudu Moshe, Eitan Porat, et al. Ltx-2: Efficient joint audio-visual foundation model. *arXiv preprint arXiv:2601.03233*, 2026.

Xuan He, Dongfu Jiang, Ge Zhang, Max Ku, Achint Soni, et al. VideoScore: Building automatic metrics to simulate fine-grained human feedback for video generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024.

Siqiao Huang, Jialong Wu, Qixing Zhou, Shangchen Miao, and Mingsheng Long. Vid2World: Crafting video diffusion models to interactive world models. *arXiv preprint arXiv:2505.14357*, 2025a.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self Forcing: Bridging the train-test gap in autoregressive video diffusion. *arXiv preprint arXiv:2506.08009*, 2025b.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. *arXiv preprint arXiv:2311.17982*, 2024.

Ziqi Huang, Fan Zhang, Xiaojie Xu, Yinan He, Jiashuo Yu, Ziyue Dong, Qianli Ma, Nattapol Chanpaisit, Chenyang Si, Yuming Jiang, Yaohui Wang, Xinyuan Chen, Ying-Cong Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench++: Comprehensive and versatile benchmark suite for video generative models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025c. doi: 10.1109/TPAMI.2025.3633890.

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. *arXiv preprint arXiv:2505.05470*, 2025a.

Jie Liu, Gongye Liu, Jiajun Liang, Ziyang Yuan, Xiaokun Liu, Mingwu Zheng, Xiele Wu, Qiulin Wang, Wenyu Qin, Menghan Xia, et al. Improving video generation with human feedback. *arXiv preprint arXiv:2501.13918*, 2025b.

Yuhang Ma, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. HPSv3: Towards wide-spectrum human preference score. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

Meituan LongCat Team, Xunliang Cai, Qilong Huang, Zhuoliang Kang, Hongyu Li, Shijun Liang, Liya Ma, Siyu Ren, Xiaoming Wei, Rixu Xie, and Tong Zhang. LongCat-Video technical report. *arXiv preprint arXiv:2510.22200*, 2025. URL https://arxiv.org/abs/2510.22200.

OpenAI. Introducing GPT-5.5. OpenAI Product Release, April 2026. URL https://openai.com/index/introducing-gpt-5-5/.

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.

Lin Shi, Chiyu Ma, Wenhua Liang, Weicheng Ma, and Soroush Vosoughi. Judging the judges: A systematic study of position bias in LLM-as-a-judge. *arXiv preprint arXiv:2406.07791*, 2024. URL https://arxiv.org/abs/2406.07791.

Wenqiang Sun, Haiyu Zhang, Haoyuan Wang, Junta Wu, Zehan Wang, Zhenwei Wang, Yunhong Wang, Jun Zhang, Tengfei Wang, and Chunchao Guo. WorldPlay: Towards long-term geometric consistency for real-time interactive world modeling. *arXiv preprint arXiv:2512.14614*, 2025. URL https://arxiv.org/abs/2512.14614.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. VGGT-Ω. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026a.

Weijie Wang, Xiaoxuan He, Youping Gu, Yifan Yang, Zeyu Zhang, Yefei He, Yanbo Ding, Xirui Hu, Donny Y. Chen, Zhiyuan He, Yuqing Yang, and Bohan Zhuang. World-R1: Reinforcing 3D constraints for text-to-video generation. *arXiv preprint arXiv:2604.24764*, 2026b.

Zehan Wang, Tengfei Wang, Haiyu Zhang, Xuhui Zuo, Junta Wu, Haoyuan Wang, Wenqiang Sun, Zhenwei Wang, Chenjie Cao, Hengshuang Zhao, Chunchao Guo, and Zhou Zhao. WorldCompass: Reinforcement learning for long-horizon world models. *arXiv preprint arXiv:2602.09022*, 2026c.

Zile Wang, Zexiang Liu, Jiaxing Li, Kaichen Huang, Baixin Xu, Fei Kang, Mengyin An, Peiyu Wang, Biao Jiang, Yichen Wei, Yidan Xietian, Jiangbo Pei, Liang Hu, Boyi Jiang, Hua Xue, Zidong Wang, Haofeng Sun, Wei Li, Wanli Ouyang, Xianglong He, Yang Liu, Yangguang Li, and Yahui Zhou. Matrix-Game 3.0: Real-time and streaming interactive world model with long-horizon memory. *arXiv preprint arXiv:2604.08995*, 2026d. URL https://arxiv.org/abs/2604.08995.

Bing Wu, Chang Zou, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Jack Peng, Jianbing Wu, Jiangfeng Xiong, Jie Jiang, et al. HunyuanVideo 1.5 technical report. *arXiv preprint arXiv:2511.18870*, 2025. URL https://arxiv.org/abs/2511.18870.

Shuchen Xue, Chongjian Ge, Shilong Zhang, Yichen Li, and Zhi-Ming Ma. Advantage weighted matching: Aligning RL with pretraining in diffusion models. *arXiv preprint arXiv:2509.25050*, 2025a.

Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, and Ping Luo. DanceGRPO: Unleashing GRPO on visual generation. *arXiv preprint arXiv:2505.07818*, 2025b.

Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, Yingcong Chen, Yao Lu, et al. LongLive: Real-time interactive long video generation. *arXiv preprint arXiv:2509.22622*, 2025.

Tianwei Yin, Qiang Zhang, Richard Zhang, William T. Freeman, Frédo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22963–22974, June 2025.

Kaining Ying, Hengrui Hu, Siyu Ren, Jiamu Li, Fengjiao Chen, Ziwen Wang, Xuezhi Cao, Xunliang Cai, and Henghui Ding. WBench: A comprehensive multi-turn benchmark for interactive video world model evaluation. *arXiv preprint arXiv:2605.25874*, 2026.

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*, 2026. URL https://arxiv.org/abs/2603.04379.

Songchun Zhang, Zeyue Xue, Siming Fu, Jie Huang, Xianghao Kong, Y Ma, Haoyang Huang, Nan Duan, and Anyi Rao. Astrolabe: Steering forward-process reinforcement learning for distilled autoregressive video models. *arXiv preprint arXiv:2603.17051*, 2026. URL https://arxiv.org/abs/2603.17051.

Yuke Zhao, Wangbo Zhao, Weijie Wang, Zeyu Zhang, Dakai An, Akide Liu, Yinghao Yu, Jiasheng Tang, Fan Wang, Wei Wang, and Bohan Zhuang. WorldOlympiad: Can your world model survive a triathlon? *arXiv preprint arXiv:2606.11129*, 2026.

Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, Kai Jiang, Hang Su, Stefano Ermon, Jun Zhu, and Ming-Yu Liu. DiffusionNFT: Online diffusion reinforcement with forward process. *arXiv preprint arXiv:2509.16117*, 2025.

Haoyi Zhu, Haozhe Liu, Yuyang Zhao, Tian Ye, Junsong Chen, Jincheng Yu, Tong He, Song Han, and Enze Xie. SANA-WM: Efficient minute-scale world modeling with hybrid linear diffusion transformer. *arXiv preprint arXiv:2605.15178*, 2026a.

Hongzhou Zhu, Min Zhao, Guande He, Hang Su, Chongxuan Li, and Jun Zhu. Causal Forcing: Autoregressive diffusion distillation done right for high-quality real-time interactive video generation. *arXiv preprint arXiv:2602.02214*, 2026b.

A Extended Related Work

Video world models. Full-sequence diffusion systems such as Sora (Brooks et al., 2024), Seedance (Gao et al., 2025), and Veo (Google DeepMind, 2025) set the quality frontier for video generation, but they rely on bidirectional diffusion and are not designed for causal prediction from generated history. CausVid (Yin et al., 2025), Self-Forcing (Huang et al., 2025b), and LongLive (Yang et al., 2025) convert this capability into few-step autoregressive generation and reduce the train–test gap over model-generated histories. LingBot-World-Fast (Gao et al., 2026), HY-WorldPlay (Sun et al., 2025), and SANA-WM (Zhu et al., 2026a) further combine long-horizon generation with action control, explicit memory, or efficient long-context architectures. However, supervision remains a bottleneck. Even with model-generated context, denoising and distillation still mainly match short-clip distributions from bidirectional diffusion models (Huang et al., 2025b; Gao et al., 2026). Long, action-aligned videos that preserve state across revisits are also scarce and expensive to collect. Moreover, maximizing likelihood in pixel space does not directly optimize the long-horizon behavior users prefer, such as consistency and physical plausibility. RL offers an alternative way to directly incentivize these capabilities.

Reinforcement learning for video generation. GRPO (Shao et al., 2024) enabled practical group-relative online RL for LLMs and inspired reverse-process extensions such as Flow-GRPO (Liu et al., 2025a) and DanceGRPO (Xue et al., 2025b) for diffusion and flow models. AWM (Xue et al., 2025a) and DiffusionNFT (Zheng et al., 2025) instead optimize rewarded endpoints through the forward process, avoiding reverse-trajectory storage and solver dependence. WorldCompass (Wang et al., 2026c) and Astrolabe (Zhang et al., 2026) are the closest applications to autoregressive video world models. WorldCompass uses HPSv3 (Ma et al., 2025) and action-following rewards without conditioning on the generated history, while Astrolabe applies DiffusionNFT with HPSv3 (Ma et al., 2025) and VideoAlign (Liu et al., 2025b) feedback within a local window. Both retain an autoregressive state mismatch: jointly noised target windows expose later causal blocks to noisy predecessors, whereas sampling reaches them through clean generated prefixes. WoRL instead uses clean-prefix conditioning to preserve the clean generated predecessor content of ordinary rollouts, Parallel Clean-Prefix Attention to execute that contract efficiently, and a history-conditioned reward.

Reward models for video generation. HPSv3 (Ma et al., 2025) and VideoAlign (Liu et al., 2025b) provide strong clip-local signals for visual quality and prompt alignment, while camera-based action scores measure immediate control fidelity (Wang et al., 2026c). Because these rewards do not condition on the generated history, a clip may score well while changing an entity, undoing an event, or violating geometry established earlier. World-R1 (Wang et al., 2026b) moves beyond these local 2D signals by using a reconstruction-based 3D Score to post-train a bidirectional video diffusion model. The score combines scene reconstruction, novel-view rendering, and camera-trajectory alignment. Its reliability depends on faithfully reconstructing the generated video: imperfect samples can yield inaccurate reconstructions, while independently moving or deformable objects violate the rigid-scene assumptions on which the score relies. Reconstructing and rendering every rollout also adds substantial reward latency, limiting scalability to long videos and dynamic worlds. Autoregressive long-video generation therefore calls for a different signal: each next clip must be evaluated against the world established by its preceding generated history. State-of-the-art multimodal models such as Gemini 3.5 Flash (Google DeepMind, 2026) and GPT-5.5 (OpenAI, 2026) already exhibit strong video-understanding capabilities, yet their judgments can be unstable and order-sensitive (Shi et al., 2024), making them unsuitable as direct reward signals. We therefore curate offline multimodal judgments into preference supervision and train NVRM, a dedicated reward model that evaluates each candidate next clip in the context of its preceding generated history.

B Full Preliminaries

Autoregressive Video Diffusion Model. An autoregressive video diffusion model extends a generated video sequence one block at a time (Huang et al., 2025b; Yang et al., 2025; Gao et al., 2026), with each block generated through a conditional diffusion process. We define a *block* (e.g., a 0.7-second video segment) as the basic diffusion unit. A *clip*, by contrast, comprises multiple consecutive blocks (e.g., a 5-second segment) and is the shortest temporal unit over which short-term video quality can be reliably evaluated.

At clip step t, let $h^t = (x^0, \ldots, x^{t-1})$ denote the previously generated clips, and let $c = (I, y, a)$ denote the conditioning inputs: the initial observation I, instruction y, and optional control signal a, such as an action or camera trajectory. Let π_θ denote the autoregressive video diffusion policy parameterized by θ. The next clip, $x^t = (x_1^t, \ldots, x_J^t)$, consists of J causally generated latent blocks, and its conditional distribution factorizes as

$$\pi_\theta(x^t \mid h^t, c) = \prod_{j=1}^{J} \pi_\theta(x_j^t \mid h^t, x_{<j}^t, c). \tag{6}$$

We denote the clean state of block j by $x_{j,0}^t \equiv x_j^t$ and its noisy state at diffusion timestep τ by $x_{j,\tau}^t$. When denoising block j, the preceding clips h^t and earlier blocks $x_{<j}^t$ have already been generated and remain fixed in their clean states.

DiffusionNFT (Zheng et al., 2025) optimizes diffusion policies directly through the forward process. For each condition c, the sampling policy π_{old} generates K candidates $\{x_0^{(k)}\}_{k=1}^K$ with raw rewards $\{R_k\}_{k=1}^K$ and optimality probabilities $\{r_k\}_{k=1}^K$:

$$r_k = \frac{1}{2} + \frac{1}{2}\,\text{clip}\left(\frac{R_k - \bar{R}}{Z_c}, -1, 1\right), \qquad \bar{R} = \frac{1}{K}\sum_{k=1}^{K} R_k,$$

where $Z_c > 0$ is a normalization scale. Rather than optimizing the likelihood of the sampling trajectory directly, DiffusionNFT re-noises each clean candidate through the forward process:

$$x_\tau^{(k)} = \alpha_\tau x_0^{(k)} + \sigma_\tau \epsilon, \qquad v_\tau^{(k)} = \dot{\alpha}_\tau x_0^{(k)} + \dot{\sigma}_\tau \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I).$$

The training policy π_θ is optimized using the negative-aware objective

$$\begin{aligned}\mathcal{L}_{\text{NFT}} = \mathbb{E}_{k,\tau,\epsilon}\Big[& r_k \left\| v_\theta^+(x_\tau^{(k)}, c, \tau) - v_\tau^{(k)} \right\|_2^2 + (1 - r_k) \left\| v_\theta^-(x_\tau^{(k)}, c, \tau) - v_\tau^{(k)} \right\|_2^2 \Big] \\ & + \lambda_{\text{ref}} \mathcal{L}_{\text{KL}}(\pi_\theta, \pi_{\text{ref}}),\end{aligned} \tag{7}$$

where $v_\theta^+ = (1-\beta)v_{\text{old}} + \beta v_\theta$ and $v_\theta^- = (1+\beta)v_{\text{old}} - \beta v_\theta$ are the implicit positive and negative velocity predictors. During each update, π_{old} remains fixed, π_θ receives gradients, and the frozen pretrained policy π_{ref} regularizes the update. The sampling policy is subsequently updated using an exponential moving average of π_θ. Intuitively, this objective steers the learned policy toward the distribution of high-reward samples and away from that of low-reward samples. Appendix G.5 provides the exact forward-process targets and reference loss.

Clip-wise DiffusionNFT for AR Diffusion is adopted by existing work applying RL to video world models (Wang et al., 2026c; Zhang et al., 2026). Given a shared clean inter-clip history h^t, the sampling policy generates a group of candidate continuations $\{x^{t,(k)}\}_{k=1}^K$, each comprising J causal blocks. The entire clip of multiple blocks is jointly noised at diffusion timestep τ:

$$x_\tau^{t,(k)} = \alpha_\tau x^{t,(k)} + \sigma_\tau \epsilon, \qquad v_\tau^{t,(k)} = \dot{\alpha}_\tau x^{t,(k)} + \dot{\sigma}_\tau \epsilon.$$

Intuitively, this extends DiffusionNFT by conditioning both velocity predictors v_θ and v_{old} on the frozen history, thereby shifting the optimization target from standalone video generation to history-conditioned continuation. However, jointly noising all blocks in the current clip introduces an intra-clip train–test mismatch, even though the inter-clip history h^t remains clean and fixed. Specifically, the velocity predictor for block j is conditioned on noisy earlier blocks during clip-wise training, whereas these blocks are fully denoised during autoregressive inference:

$$\underbrace{v_\theta\big(x_{j,\tau}^t, \tau \mid h^t, x_{<j,\tau}^t, c\big)}_{\text{clip-wise DiffusionNFT training}} \qquad \text{vs.} \qquad \underbrace{v_\theta\big(x_{j,\tau}^t, \tau \mid h^t, x_{<j,0}^t, c\big)}_{\text{autoregressive inference}}.$$

Clip-wise DiffusionNFT therefore preserves clean conditioning across clips but not the clean-prefix conditioning within each clip, causing it to optimize a conditional distribution different from inference.

C Further Details on Train–Test-Matched RL

Section 2 shows that clip-wise DiffusionNFT keeps the inter-clip history clean but jointly noises all blocks within the current clip. Consequently, during training, the prediction for block j is conditioned on a noisy intra-clip prefix $x^t_{<j,\tau}$, whereas autoregressive rollout conditions on the clean prefix $x^t_{<j,0}$. We therefore seek to match both inter-clip and intra-clip contexts between training and rollout when optimizing DiffusionNFT for autoregressive diffusion models.

Parallel Clean-Prefix Attention. For candidate i selected from shared-history group g, train–test matching requires each target block j to be optimized under the same causal context used during generation: the generated history h_g and the preceding clean blocks $x_{gi,<j,0}$. Rather than jointly noising all blocks within a clip and predicting their velocities in a single forward pass, a literal train–test-matched implementation performs a separate policy forward pass for each target block, keeping its preceding context clean while noising only block j. For $\varepsilon \sim \mathcal{N}(0, \mathbf{I})$ and diffusion time τ, it evaluates

$$x_{gi,j,\tau} = \alpha_\tau x_{gi,j,0} + \sigma_\tau \varepsilon, \qquad v_\theta(x_{gi,j,\tau}, \tau \mid h_g, x_{gi,<j,0}, c_g)\,. \tag{8}$$

The sampling and reference policies are evaluated under the same context without gradients. This modification affects only policy evaluation; candidate rewards, advantages, top/bottom selection, and the DiffusionNFT objective remain unchanged and are detailed in Appendix G.5. Importantly, ordinary causal masking is insufficient: although it masks future blocks, block j can still attend to corrupted earlier blocks when the entire clip is jointly noised, as illustrated by the noisy-prefix attention in Figure 3. Equation 8 instead preserves the clean causal context encountered during rollout. However, this literal implementation requires J separate forward passes per policy for a J-block clip, repeatedly recomputing projections over the same history and an increasingly long clean prefix.

To eliminate this redundancy, we propose Parallel Clean-Prefix Attention (PCPA), which parallelizes these target-wise evaluations within a packed execution. For each policy, the first pass of each attention layer computes and detaches the KV cache for the shared history and clean prefix only once. A subsequent attention pass packs all independently noised target blocks and evaluates them in parallel. Specifically, each noisy target block j attends only to the cached clean context from previous clips h_g and clean prefix within current clip $x_{gi,<j,0}$. Consequently, each target preserves the clean-prefix conditioning in Equation 8, while the shared prefix projections are computed only once.

Context-aware policy scheduling. Train–test-matched RL further requires each policy to evaluate candidates under its own historical state. In particular, the history KV caches for π_{old}, π_θ, and π_{ref} must be computed using their corresponding policy weights. Approximating these states by sharing them across adapters introduces cross-policy mismatch, which can destabilize training and ultimately limit the scalability of RL. This requirement becomes particularly expensive for long videos and large full-attention models, where the growing history KV cache can exceed GPU memory and trigger frequent cache CPU offloading and repeated prefilling. A naive execution order therefore incurs substantial overhead from repeatedly switching and reconstructing policy-specific history states, as illustrated in Figure 4.

We introduce a context-aware policy schedule to minimize these history-state transitions. During rollout, WoRL computes the committed history once and branches the K candidate clips from its final state. During training, rather than processing each candidate sequentially through π_{old}, π_θ, and π_{ref}, we adopt a policy-major schedule: each policy constructs its history state once and evaluates all compatible candidates before switching roles. This ordering maximizes reuse of the resident history KV cache and minimizes cache swapping and repeated prefilling.

System optimization for distributed training. To scale RL for large video world models with full attention over long videos, we combine Fully Sharded Data Parallel (FSDP) and sequence parallelism (SP) to distribute model rollout and training across multiple GPUs and nodes. This enables reinforcement learning for large models over substantially longer video contexts without exceeding per-GPU memory limits. Beyond distributed model execution, we optimize the rollout and reward pipeline to reduce non-model overhead. Whenever memory permits, each shared-history group remains local throughout candidate generation, decoding, reward evaluation, and selection, avoiding communication of large video or history tensors. Candidate decoding is performed in bounded microbatches and overlapped with CPU-side post-processing and asynchronous reward evaluation,

while only lightweight reward statistics are synchronized when necessary. Further implementation details of distributed training are provided in Appendix G.

In summary, we build an efficient system that makes train–test-matched RL practical for large video world models with long contexts. Parallel Clean-Prefix Attention removes redundant intra-clip computation while preserving rollout-matched conditioning; context-aware policy scheduling minimizes reconstruction and movement of policy-specific history states; and distributed system optimizations scale full-attention rollout and training while reducing decoding, reward, and communication overhead. Together, these designs substantially improve the efficiency and scalability of train–test-matched RL for video world models.

D Reward Derivation and Preference Data

A key challenge in rewarding autoregressive video generation is that the quality of a continuation cannot be determined from the current clip alone, whereas existing RL frameworks assume otherwise (Wang et al., 2026c; Zhang et al., 2026; Wang et al., 2026b; Ma et al., 2025). Consider the two candidate next clips in the first case of Figure 5. Without observing their preceding history, the first candidate may appear preferable due to its more visually appealing lighting and shading. However, if the preceding video establishes a scene dominated by red warning lights, the second candidate becomes the better continuation because it preserves the established environment. Thus, the preference between two candidate clips can change once the generated history is taken into account. A clip-wise reward $r_{\text{clip}}(x^t, c)$ cannot capture such conditional preference.

D.1 Autoregressive Reward Formulation

Our goal is to improve the quality of an entire autoregressively generated video rather than that of each clip in isolation. Let $r(x^{1:T}, c)$ denote the sequence-level utility of a generated video $x^{1:T}$ under condition c. Following the optimality formulation of DiffusionNFT (Zheng et al., 2025), we define the optimality-weighted distribution

$$p^*(x^{1:T} \mid c) = \frac{1}{Z(c)} p_0(x^{1:T} \mid c) \exp\left(\frac{r(x^{1:T}, c)}{\beta}\right), \tag{9}$$

where p_0 is the reference generation distribution and $Z(c)$ is the normalization constant. Since both p^* and p_0 admit autoregressive factorization, the sequence-level reward can be decomposed as

$$\begin{aligned} r(x^{1:T}, c) &= \beta \sum_{t=1}^{T} \log \frac{p^*(x^t \mid x^{<t}, c)}{p_0(x^t \mid x^{<t}, c)} + \text{const}(c) \\ &= \sum_{t=1}^{T} r_t(x^t \mid x^{<t}, c) + \text{const}(c), \end{aligned} \tag{10}$$

where $r_t(x^t \mid x^{<t}, c)$ denotes the conditional utility of the next clip given its generated history. Thus, optimizing long-video quality naturally corresponds to optimizing a sequence of history-conditioned continuation decisions.

Existing video reward models instead typically score each clip independently, effectively approximating the history-conditioned utility $r_t(x^t \mid x^{<t}, c)$ with a clip-wise reward $r_{\text{clip}}(x^t, c)$. This assumes that the preference over a clip is independent of the history, which is poorly suited to long-horizon video generation: a candidate that appears preferable in isolation may contradict objects, identities, or scene states established earlier in the rollout. Consequently, improving standalone clip-wise reward does not necessarily improve the quality of the complete generated video.

Motivated by Equation 10, we introduce the *Next-Video Reward Model* (NVRM), which directly models the conditional utility of a candidate continuation:

$$q_\phi(x_0^{<t}, c, x_0^t) \approx r_t(x_0^t \mid x_0^{<t}, c). \tag{11}$$

Following the Bradley–Terry formulation (Bradley & Terry, 1952), we train NVRM from pairwise preferences between candidate continuations under the same history to learn the reward. Given two

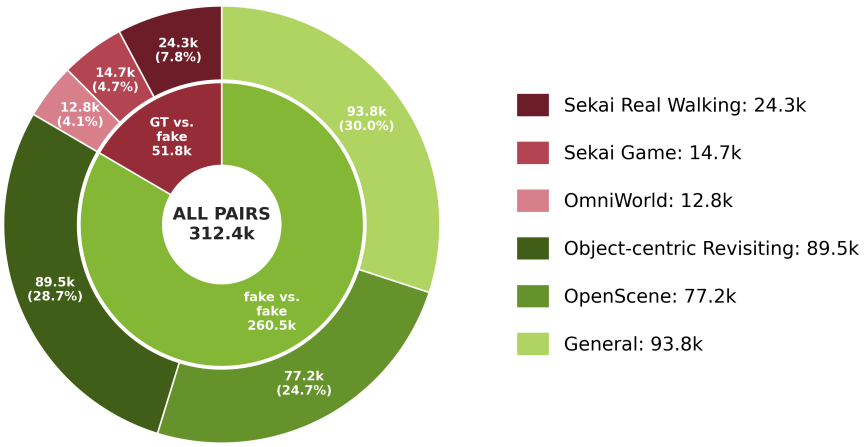


Figure 10: **NVRM training-data composition.** Ground-truth–fake pairs supply a directional real-video preference anchor, while fake–fake pairs cover model-generated histories across general, revisiting, and interactive domains.

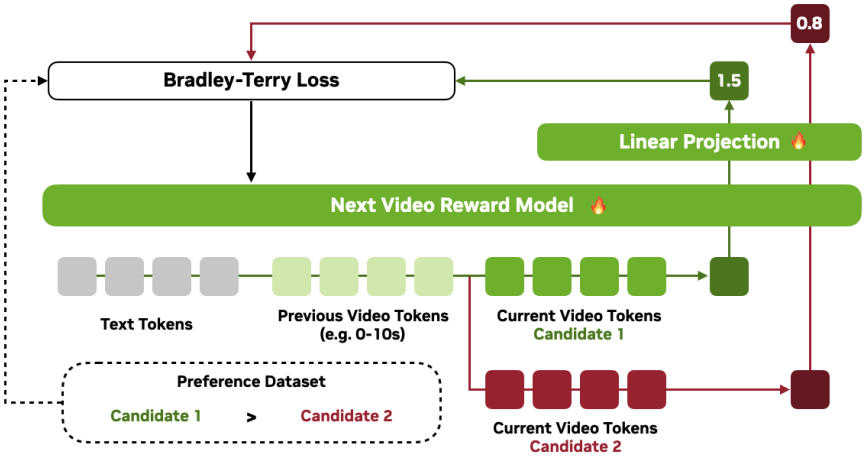


Figure 11: **Pairwise training of NVRM.** Two candidates share the same instruction and generated-history tokens, are scored independently by a shared scalar model, and are coupled only through the Bradley–Terry loss.

candidate continuations $x^t_{u,0}$ and $x^t_{v,0}$, we model

$$\Pr_{\phi}[x^t_{u,0} \succ x^t_{v,0} \mid x_0^{<t}, c] = \sigma\big(q_\phi(x_0^{<t}, c, x^t_{u,0}) - q_\phi(x_0^{<t}, c, x^t_{v,0})\big)\,, \tag{12}$$

and optimize the corresponding preference objective

$$\mathcal{L}_{\text{NVRM}} = -\mathbb{E}_{(x_0^{<t}, c, x^t_{+,0}, x^t_{-,0}) \sim \mathcal{D}_{\text{pref}}} \log \sigma\big(q_\phi(x_0^{<t}, c, x^t_{+,0}) - q_\phi(x_0^{<t}, c, x^t_{-,0})\big)\,. \tag{13}$$

This formulation imposes a key requirement on the training data: candidate continuations must be compared under the same preceding history and generation condition, such that the preference reflects how well each candidate continues the established world rather than differences in context. Moreover, training NVRM at scale requires diverse and reliable preference supervision that captures long-term consistency failures without expensive human expert annotation. These requirements motivate our history-matched preference collection and graph-based curation pipeline described next.

D.2 Preference Data for Video Continuation

Training NVRM requires preference supervision between alternative continuations under the same preceding history. Such data is rarely available in natural video corpora: a real trajectory provides

only one observed future for a given history, rather than multiple alternative continuations that can be directly compared. We therefore construct preference data by branching model-generated continuations from shared video histories. As summarized in Figure 10, our dataset combines real–fake pairs, which provide reliable real-video anchors, with fake–fake pairs, which expose a broader range of model-generated continuation failures.

Real–fake preference. We first pair a ground-truth continuation with a model-generated continuation conditioned on the same real-video history. Because the two candidates share the same preceding context, the ground-truth continuation provides a natural positive anchor for realistic temporal evolution. This gives relatively reliable supervision without requiring pairwise human annotation. Real–fake data alone, however, is insufficient for learning long-horizon consistency. Its histories are constrained by ground-truth trajectories, which rarely deviate from or revisit previously observed scenes in ways that expose accumulated model errors. Moreover, sufficiently long and interaction-rich real videos are limited in scale. Consequently, real–fake pairs provide strong anchors but cover only a restricted distribution.

Fake–fake preference. To better cover the model's own rollout distribution, we further generate multiple candidate continuations from the same model-generated history and compare them against one another. Because all candidates branch from an identical preceding world and generation condition, their preference directly reflects which continuation better preserves the established video context. These comparisons expose finer-grained failures that may not appear in real–fake data, including identity drift, scene-state inconsistency, and implausible consequences of previous events.

Constructing fake–fake preference data at scale introduces two additional challenges. First, exhaustive human expert annotation is prohibitively expensive, motivating the use of multimodal models (Google DeepMind, 2026; OpenAI, 2026) as automatic judges. Their pairwise decisions, however, can be sensitive to presentation order and inconsistent across comparisons. Second, autoregressive video world models, particularly distilled models, often exhibit limited diversity under the same history and control trajectory. Candidate continuations can therefore be highly similar, making their preference margins small and further amplifying judgment noise. Directly using all raw VLM comparisons would introduce substantial noisy data.

Graph-based preference curation. Rather than generating an isolated candidate pair for each preference example, we sample a group of continuations from the same history and construct preferences jointly within the group. This group-based design provides two advantages. First, an individual pairwise comparison can be ambiguous or noisy, whereas multiple comparisons within the same candidate group provide additional self-consistency constraints that can be used to identify unreliable judgments. Second, sampling multiple continuations from the same state increases the chance of observing candidates with substantially different continuation quality, making it easier to discover informative failure–success pairs even when the underlying model has limited generation diversity.

Based on this group structure, we develop the fully automatic curation pipeline illustrated in Figure 6. For each shared-history group, we first generate multiple candidate continuations and compare candidate pairs using a multimodal judge (Google DeepMind, 2026; OpenAI, 2026). Each pair is evaluated in both display orders, and we retain a preference only when the predicted winner remains unchanged after swapping the candidates, filtering position-sensitive judgments (Shi et al., 2024). The surviving decisions form the group's directed preference graph.

The graph structure allows us to further exploit consistency across multiple pairwise judgments. We apply a minimum feedback arc set (MFAS) to remove cyclic contradictions and a minimum directed multicut (MDM) to prune unsupported preference paths involving ambiguous candidate pairs. Importantly, these operations are used only to filter noisy observed comparisons; we neither infer a complete ranking over the group nor introduce transitive pseudo-labels. As shown in Figure 6, position debiasing reduces 391K raw fake–fake comparisons to 276K stable preferences, while graph-based consistency filtering retains 260.5K pairs. Together with 51.8K real–fake anchors, our final corpus contains 312.4K preference pairs without human expert annotation. Detailed formulations of MFAS and MDM are provided in Appendix H.1.

D.3 Integration with Short-Term Rewards

NVRM focuses on history-dependent properties such as long-term consistency and physical plausibility, while existing reward models remain effective for evaluating short-term visual quality and controllability. We therefore combine NVRM with complementary clip-wise rewards during RL. Specifically, we use HPSv3 (Ma et al., 2025) for visual quality, together with a **camera-control reward** for action-conditioned video world models or a **motion score** for text-conditioned video generation. Because these reward models operate on different numerical scales, we normalize each component using fixed statistics estimated before RL. For reward component m, we define

$$\hat{r}_m(r) = \frac{r - \mu_m}{\sigma_m}, \tag{14}$$

where μ_m and σ_m are preset and kept frozen throughout training. The final reward for candidate i in group g is

$$R_{gi} = \hat{r}_{\mathrm{HPS}}(r_{gi}^{\mathrm{HPS}}) + \hat{r}_{\mathrm{ctrl}}(r_{gi}^{\mathrm{ctrl}}) + \hat{r}_{\mathrm{NVRM}}(r_{gi}^{\mathrm{NVRM}}), \tag{15}$$

where r^{ctrl} denotes the camera-control reward for action-conditioned models and the motion score for text-conditioned models. This composition combines short-term generation quality with history-aware continuation quality while keeping the relative scale of each reward component fixed during RL. Further details are provided in Appendix H.4.

E Further Details on Revisit and Rewind

Long-horizon consistency failures can only be corrected if they are explicitly exposed during rollout. In a standard forward trajectory, a world model may gradually alter an object, character, or scene geometry while each individual continuation remains locally plausible. If the trajectory never revisits previously observed content, such accumulated drift may remain hidden from the training signal and therefore cannot be directly corrected. Simply extending the rollout provides longer history, but does not necessarily create the observations needed to reveal long-term inconsistency.

To address this, we train the model in a revisit-centered training environment. Instead of relying on arbitrary long trajectories, we explicitly construct anchor–departure–return rollouts: the model first observes an identifiable scene state, moves sufficiently far for that state to leave the current view, and later returns to overlapping content. The departure creates a genuinely long-range dependency, while the return acts as a consistency checkpoint where accumulated drift becomes directly observable. Importantly, revisit does not require pixel-level reconstruction of the original view; viewpoint, illumination, occlusion, and legitimate state changes may differ. The environment determines *where* long-term consistency becomes observable, while NVRM determines *which* returned continuation is preferable.

Revisit trajectory construction. We construct diverse anchor–departure–return trajectories from OpenScene environments. Each instance contains an identifiable object, character, or landmark together with a camera trajectory that first establishes the anchor, moves sufficiently far away, and subsequently returns to an overlapping view. Scene and trajectory proposals are automatically checked to ensure that the anchor is recognizable before departure, the intervening motion produces a meaningful viewpoint change, the return recovers overlapping scene content, and the camera path remains geometrically valid. Invalid trajectories are resampled. These constraints create training states in which long-range consistency is deliberately delayed and later made observable, rather than using the checks themselves as reward signals.

During RL, most revisit groups branch candidates at the return state, after the preceding departure clips have been committed into the generated history. All candidates therefore share the same long-range context and differ only in how they reconstruct or continue the revisited world. We retain a smaller fraction of ordinary short-horizon groups to preserve general generation quality. The exact trajectory composition and sampling ratios are provided in Appendix G.6.

Rewind regularization. Revisit makes long-term failures observable, but informative preference contrast may still be scarce. This is particularly problematic for distilled autoregressive video models, whose continuations from the same history often exhibit limited diversity. Early in RL, a group may contain only poor returns, providing little positive signal for group-relative optimization.

We introduce *rewind regularization* to augment such groups with an additional history-consistent candidate. Given the preceding committed clip, we construct a rewind continuation by reversing its temporal evolution together with the corresponding camera traversal, yielding a candidate that approximately returns toward the previously observed state. Although this constructed continuation is not sampled from the current policy and therefore introduces a partially off-policy candidate into the group, we use it only as a regularizer to increase the chance of providing an informative positive contrast when ordinary samples lack sufficient diversity.

Importantly, the rewind candidate is not treated as ground truth or assigned a privileged positive label. It participates in the same reward computation, group-relative ranking, and selection as generated candidates. Since temporal reversal can be unnatural for cases such as walking or falling objects, poor rewind candidates receive low rewards and provide little positive signal. Rewind therefore introduces structured off-policy diversity while letting the reward determine whether each constructed candidate provides useful supervision.

Figure 7 summarizes the resulting training environment. Revisit trajectories expose accumulated long-term errors, NVRM evaluates the resulting history-conditioned continuations, and rewind regularization improves the availability of informative candidate contrasts under limited generation diversity. Implementation details are provided in Appendix G.6.

F Additional Experimental Details

We evaluate WoRL from four complementary perspectives: (1) whether WoRL improves long-horizon generation quality across different types of models; (2) whether the proposed system makes train–test-matched RL computationally practical; (3) how the main optimization and training components contribute to downstream performance; and (4) whether NVRM provides preference signals that align with human judgments. Detailed evaluation protocols and statistical settings are provided in Appendix I.

F.1 Experimental Setup

Models. We evaluate WoRL on two substantially different autoregressive video models. LingBot-World-Fast (Gao et al., 2026) is an action-conditioned video world model, while Helios-Distilled-14B (Yuan et al., 2026) is a long-horizon text-to-video model. For all before–after comparisons, the pretrained and post-trained models use identical evaluation prompts, generation settings, and random seeds; LingBot additionally shares the same initial observations and action controls.

Evaluation benchmarks. We evaluate action-conditioned generation on WBench-Navigation (Ying et al., 2026), which contains 158 held-out cases, and text-conditioned generation on WBench-Text. Both benchmarks report Quality, Settings, Interaction, Consistency, Physical, and Overall scores. For Helios, we additionally report VBench-Long (Huang et al., 2024; 2025c) as a complementary long-video benchmark. RevBench contains 150 anchor–departure–return out-of-training-distribution trajectories and specifically evaluates behavior under scene revisiting. Each benchmark case is evaluated with three generation seeds. Beyond automatic metrics, we conduct blinded human studies for preference between generations before and after RL.

Training and efficiency setup. In terms of DiffusionNFT settings, we sample 24 candidates for each of 16 shared-history groups and retain the 4 highest- and 4 lowest-scoring candidates for optimization. Unless stated otherwise, training uses the aggregated reward defined in Equation 5. For component ablations, we use a fixed 16-GPU B200 workload with 8 prompts and 12 candidates per prompt for 40 steps; efficiency is measured over one complete optimizer update. This controlled setting isolates the computational effect of each system optimization from differences in training configuration. Complete training and system configurations are provided in Appendix G.

F.2 Preference-Curation Analysis

Preference curation. We further evaluate the effect of the preference-data curation pipeline on NVRM training. Figure 9 compares models trained with and without bidirectional filtering and graph-based reconciliation. At the shared 500-step checkpoint, evaluation accuracy reaches approximately 0.78 with curation compared with 0.52 without it, and the gap remains substantial

throughout their common training range up to step 3,000. The curated run further reaches approximately 0.86 at step 4,500. Together with the data-retention statistics in Figure 6, these results indicate that group-level preference curation substantially improves the learnability of automatically collected pairwise supervision.

F.3 Additional Qualitative Analysis

Qualitative analysis. Long-horizon consistency is inherently relational: a clip may be visually plausible in isolation while contradicting states established much earlier in the trajectory. Figure 5 isolates this effect by holding the preceding history fixed and comparing alternative continuations. Higher NVRM scores consistently correspond to candidates that better preserve established scene geometry, foreground objects, and character identity.

Figure 8 illustrates these improvements across complete autoregressive rollouts. In the two text-conditioned cases, the text-conditioned examples show that WoRL better preserves long-term appearance and structure: the gardener remains within the scene instead of drifting out of view, the rotating emblem maintains a stable 3D structure rather than morphing over time. The action-conditioned examples show similar improvements in world-state consistency: the cauldron preserves a coherent perspective during camera movement, the skull and the writing on the blackboard remain consistent after revisiting the scene.

These examples also clarify that long-term consistency is distinct from static reconstruction. The post-trained model continues actions, object motion, and controlled camera movement while better preserving identity, topology, and spatial state. This behavior is consistent with the objective of WoRL: maintaining constraints established by the generated history without forcing later frames to reproduce earlier observations pixel by pixel.

G Implementation Details

G.1 Training Checkpoints

The LingBot $+WoRL$ row in Table 1 uses the checkpoint at global step 51, after 51 optimizer commits; this corresponds to about 50 updates. The Helios result uses the checkpoint at global step 240, after 240 optimizer commits.

G.2 Training Profiles

Table 7 places the two evaluated model profiles side by side. The shared symbols follow the main text: G is the number of shared-history groups, K is the number of sampled candidates per group, and B is the number retained for training. The controlled efficiency workload is separate and is specified in Appendix I.6.

G.3 Policy Update and Candidate Selection

The training, sampling, and reference policies are separate policy roles over the same frozen backbone. For the LingBot profile, the trainable state is a LoRA adapter; the sampling and reference roles have their own adapter weights and therefore their own history states. After committed optimizer step s, the sampling policy is updated by

$$\theta_{\text{old}} \leftarrow d_s\theta_{\text{old}} + (1 - d_s)\theta, \qquad d_s = \min(0.001s, 0.5). \tag{16}$$

For group g with temporal target t_g, let $h_g = (x_g^0, \ldots, x_g^{t_g-1})$ be the shared generated history and $c_g = (I_g, y_g, a_g)$ the condition defined in Section 2. Let $a_{g,\text{eff}}$ equal a_g in an ordinary group and the resolved reversed camera control in an active rewind group, and write $c_{g,\text{eff}} = (I_g, y_g, a_{g,\text{eff}})$. The sampling policy draws

$$\mathcal{B}_g = \{x_{gi,0}^{t_g}\}_{i=1}^{K}, \qquad x_{gi,0}^{t_g} \sim \pi_{\text{old}}^{(s)}(\cdot \mid h_g, c_{g,\text{eff}}). \tag{17}$$

Table 7: Training and generation configurations for the two evaluated backbones. The controlled efficiency workload is reported separately in Appendix I.6.

(a) LingBot-World-Fast

Setting	Value
Backbone	LingBot-World-Fast
Output clip	464×832, 16 FPS, 81 frames (5 s)
Latent geometry	7 blocks $\times$ 3 frames
Conditioning	Initial image, text, frame-wise camera trajectory
Chain / target	5 clips; 20% ordinary clip-0 targets and 80% terminal return-state targets
Groups / candidates	$G = 16$; $K = 24$ per group
Selection	$B = 8$ (top 4 and bottom 4)
Rewind	Attempt probability 0.2
Reward	Fixed-z sum of HPSv3, NVRM, and PI Camera
PI input	VGGT-Ω-1B-512; 24 prefix and 8 target frames
Policy adapters	LoRA rank/alpha $= 128/128$
NFT	$\beta = 1$, $A_{\max} = 5$, $\lambda_{\text{ref}} = 10^{-4}$
Numerics	BF16 latent input; FP32 prediction output
Optimizer	AdamW, lr 10^{-5}, $\boldsymbol{\beta} = (0.9, 0.999)$, $\epsilon = 10^{-8}$
Gradient clipping	Global max norm 1
Parallelism	FSDP full-shard with SP$=$ 4, DP$=$ 4 on 16 B200 GPUs
Training steps	51

(b) Helios-Distilled-14B

Setting	Value
Backbone	Helios-Distilled-14B
Output rollout	384×640, 16 FPS, 396 frames (24.75 s)
AR partition	12 blocks $\times$ 33 frames
Conditioning	Text
Chain / target	12-block autoregressive rollout with generated-prefix conditioning
Groups / candidates	$G = 16$; $K = 24$ per group
Selection	$B = 8$ (top 4 and bottom 4)
Rewind	N/A
Reward	Fixed-normalized sum of HPSv3, NVRM, and motion score
PI input	N/A
Policy adapters	LoRA rank/alpha $= 128/256$
NFT	$\beta = 1$, $A_{\max} = 5$, $\lambda_{\text{ref}} = 10^{-4}$
Numerics	BF16 latent input; FP32 prediction output
Optimizer	AdamW, lr 10^{-5}, $\boldsymbol{\beta} = (0.9, 0.999)$, $\epsilon = 10^{-8}$
Gradient clipping	Global max norm 1
Parallelism	DDP$=$ 32 on 32 H100 GPUs; no FSDP or SP
Training steps	240

An active rewind group appends C_{rw} constructed identities. For population $\mathcal{P}_g$, $N_g = |\mathcal{P}_g|$, and rewards R_{gi}, advantages are

$$A_{gi} = \frac{R_{gi} - \mu_g}{s_R + \epsilon}, \qquad \mu_g = \frac{1}{N_g} \sum_{i=1}^{N_g} R_{gi}, \quad s_R = \text{Std}_{g,i}(R_{gi}), \tag{18}$$

where s_R is the update-wide pre-selection population standard deviation. This realizes the normalization scale $Z_c = (s_R + \epsilon) A_{\max}$ in the DiffusionNFT notation of Section 2. Each group retains its top and bottom $B/2$ candidates, with deterministic tie-breaking.

G.4 Clean-Prefix Conditioning, Parallel Attention, and Scheduling

Clean-prefix conditioning defines the target-wise computation independently of its implementation. For selected candidate $x^{t_g}_{gi,0} = (x^{t_g}_{gi,1,0}, \ldots, x^{t_g}_{gi,J,0})$ and target block j, the history h_g and clean current-clip predecessors $x^{t_g}_{gi,<j,0}$ are stopped context, whereas only $x^{t_g}_{gi,j,\tau_j}$ is noised. Literal execution therefore evaluates each policy role $r \in \{\theta, \text{old}, \text{ref}\}$ on

$$v_r\left(x^{t_g}_{gi,j,\tau_j}, \tau_j \mid h_g, x^{t_g}_{gi,<j,0}, c_{g,\text{eff}}\right) \quad \text{in a separate call for every } j. \tag{19}$$

Parallel Clean-Prefix Attention realizes these J calls with two packed transformer executions per selected candidate and policy role, as illustrated in Figure 3. A detached clean-store execution processes $[h_g, x^{t_g}_{gi,1,0}, \ldots, x^{t_g}_{gi,J-1,0}]$ once. A noisy-read execution packs $[x^{t_g}_{gi,1,\tau_1}, \ldots, x^{t_g}_{gi,J,\tau_J}]$, while a block-indexed mask routes each noisy query only to its own clean causal prefix. At layer ℓ and role r, this gives

$$\begin{aligned} K^{\ell}_{r,gij} &= \left[\text{sg}\, K^{\ell}_r(h_g), \text{sg}\, K^{\ell}_r(x^{t_g}_{gi,<j,0}), K^{\ell}_r(x^{t_g}_{gi,j,\tau_j})\right], \\ V^{\ell}_{r,gij} &= \left[\text{sg}\, V^{\ell}_r(h_g), \text{sg}\, V^{\ell}_r(x^{t_g}_{gi,<j,0}), V^{\ell}_r(x^{t_g}_{gi,j,\tau_j})\right], \\ z^{\ell}_{r,gij} &= \text{Attn}\left(Q^{\ell}_r(x^{t_g}_{gi,j,\tau_j}), K^{\ell}_{r,gij}, V^{\ell}_{r,gij}\right). \end{aligned} \tag{20}$$

The mask excludes future clean blocks and every other target's noisy block. Only the noisy-read prediction of π_θ builds an autograd graph; its clean-store state is detached, and both π_{old} and π_{ref} are evaluated under `no_grad`. Packing therefore changes execution, but not the clean causal condition in Equation 1.

A block cursor exposes the valid range of detached clean-support K/V. A reusable history snapshot is keyed by generated history, condition, policy role, policy-weight version, position/block layout, and model mode. Thus π_θ, π_{old}, and π_{ref} never share a policy-dependent K/V cache. The LingBot profile retains at most three CPU-resident entries, while clean or noisy target state remains candidate-private. For one group with L history blocks, J target blocks, B selected candidates, and $M = 3$ policy roles, the prefix-projection counts are

$$(C_{\text{literal}}, C_{\text{parallel}}, C_{\text{sched}}) = \left(MB\left[JL + \frac{J(J-1)}{2}\right],\ MB[L+J-1],\ ML + MB(J-1) \right). \tag{21}$$

The middle term packs target blocks within each candidate–policy evaluation; the final term additionally reuses the compatible group history while one policy remains active.

Training follows the policy-major order in Section 3: for a fixed group, role r is activated once, its compatible history snapshot is restored or constructed once, and all B selected candidates are evaluated before the next role is loaded. Only committed history is reused; the current-clip clean store remains candidate-specific. This reordering preserves the selected candidate–block evaluations because π_{old} and π_{ref} are frozen and only π_θ accumulates gradients. Rollout uses only π_{old} to commit history and branch K candidates. At each optimizer boundary, derived caches are invalidated before role versions advance and are never checkpointed or reused across weight versions.

G.5 DiffusionNFT Objective and Blockwise WoRL Update

We now state the complete blockwise objective using exactly the forward-process notation of Section 2. For selected candidate i in group g, the context for target block j is

$$\tilde{c}_{gij} = \big(\text{sg}(h_g), \text{sg}(x^{t_g}_{gi,<j,0}), c_{g,\text{eff}}\big), \tag{22}$$

where sg denotes stop-gradient. Sample $\tau_j \sim p(\tau)$ and $\varepsilon_{gij} \sim \mathcal{N}(0, \mathbf{I})$ independently for each block, and noise only the target:

$$\begin{aligned} x^{t_g}_{gi,j,\tau_j} &= \alpha_{\tau_j} x^{t_g}_{gi,j,0} + \sigma_{\tau_j} \varepsilon_{gij}, \\ v^\star_{gi,j,\tau_j} &= \dot{\alpha}_{\tau_j} x^{t_g}_{gi,j,0} + \dot{\sigma}_{\tau_j} \varepsilon_{gij}. \end{aligned} \tag{23}$$

All three roles evaluate the identical tuple $(x^{t_g}_{gi,j,\tau_j}, \tau_j, \tilde{c}_{gij})$ and produce $v_{\theta,gij}$, $v_{\text{old},gij}$, and $v_{\text{ref},gij}$. The latter two predictions are detached. DiffusionNFT maps the clip-level advantage to the optimality probability used for every block of that clip:

$$\bar{A}_{gi} = \text{clip}(A_{gi}, -A_{\max}, A_{\max}), \qquad r_{gi} = \frac{1}{2} + \frac{\bar{A}_{gi}}{2A_{\max}}. \tag{24}$$

$$v^+_{gij} = (1-\beta) v_{\text{old},gij} + \beta v_{\theta,gij}, \qquad v^-_{gij} = (1+\beta) v_{\text{old},gij} - \beta v_{\theta,gij}. \tag{25}$$

The blockwise policy and reference terms are

$$\ell^{gij}_{\text{policy}} = r_{gi} \| v^+_{gij} - v^\star_{gi,j,\tau_j} \|_2^2 + (1 - r_{gi}) \| v^-_{gij} - v^\star_{gi,j,\tau_j} \|_2^2, \tag{26}$$

$$\ell^{gij}_{\text{ref}} = \| v_{\theta,gij} - v_{\text{ref},gij} \|_2^2, \tag{27}$$

$$\mathcal{L}_{\text{WoRL}} = \mathbb{E}_{(g,i)\in\mathcal{S}} \frac{1}{J} \sum_{j=1}^{J} \left(\ell^{gij}_{\text{policy}} + \lambda_{\text{ref}} \ell^{gij}_{\text{ref}} \right), \tag{28}$$

where $\mathcal{S}$ is the selected top/bottom candidate set and every squared norm is averaged over non-batch elements. Only $v_{\theta,gij}$ carries gradients. In the forward-process implementation, the regularizer denoted $\mathcal{L}_{\text{KL}}$ in Equation 7 is instantiated by the velocity-space reference MSE ℓ^{gij}_{ref}. For the rectified-flow parameterization $\alpha_\tau = 1 - \sigma_\tau$ and $v^\star = \varepsilon - x_0$, a predicted velocity also gives $\hat{x}_0(v) = x_\tau - \sigma_\tau v$. WoRL changes only the conditioning of each blockwise policy evaluation to the clean autoregressive context in Equation 22; reward, selection, and the DiffusionNFT objective remain unchanged.

G.6 Long-Horizon Targets and Rewind Population

The reference chain contains five clips indexed $0, \dots, 4$. It samples an ordinary short-horizon group by targeting clip 0 with probability 0.2, and a revisit group by targeting the terminal return state, clip 4, with probability 0.8. Thus the reported mixture is exactly 20% ordinary groups and 80% revisit groups. The chosen target, camera path, sampling-policy version, and random seeds are fixed in the chain plan before candidate generation. OpenScene paths are obtained by the propose–check–resample process in Section 5.

Rewind activation separates cheap eligibility from post-draw input availability. Cheap eligibility requires only a noninitial target and explicit rewind support from the current model profile,

$$
\begin{aligned}
e_g &= \mathbf{1}[t_g > 0]\,\mathbf{1}[\mathrm{Cap}_{\mathrm{rw}}(g)], \\
\mathrm{state}_g &= \begin{cases} \textsc{inactive}, & e_g = 0 \text{ or } (e_g = 1, z_g = 0), \\ \textsc{active}, & e_g = z_g = v_g = 1, \\ \textsc{abort}, & e_g = z_g = 1,\ v_g = 0, \end{cases} \qquad b_g = \mathbf{1}[\mathrm{state}_g = \textsc{active}].
\end{aligned} \tag{29}
$$

For $e_g = 1$, the implementation makes a seed-fixed draw $z_g \sim \mathrm{Bernoulli}(p_{\mathrm{rw}})$ before candidate generation and independently of rewards and sampled outcomes; no draw is consumed for $e_g = 0$. Only after a positive draw does it inspect the complete, compatible latent, committed-RGB, and camera inputs, summarized by v_g. An unavailable positive attempt fails closed and aborts before target sampling rather than silently reverting to the inactive population. An ineligible or negative-draw group uses its original target control and contains only ordinary sampling-policy candidates.

In an active group, only the camera component of the target condition is replaced. Let RevCam denote the model-native control sequence that traverses the recorded preceding camera path in reverse temporal direction. We construct

$$
\begin{aligned}
\Gamma^{t_g}_{g,\mathrm{rw}} &= \mathrm{RevCam}(\Gamma^{t_g-1}_{g}), \\
\widetilde{a}^{t_g}_{g} &= \mathrm{ReplaceCam}(a^{t_g}_{g}, \Gamma^{t_g}_{g,\mathrm{rw}}).
\end{aligned} \tag{30}
$$

For LingBot, RevCam fills the target window's camera-to-world pose samples and intrinsics with the preceding window's samples in reverse temporal order. Target timestamps are retained, as are the trajectory coordinate convention and intrinsics units.

The generated history, initial observation, instruction, and all non-camera conditions remain fixed. All K ordinary candidates are sampled under $\widetilde{a}^{t_g}_{g}$ in an active group; only then is the preceding committed clip reversed and encoded once,

$$
x^{t_g}_{g,\mathrm{rw},0} = \mathrm{Enc}\big(\mathrm{rev}(\mathcal{V}^{t_g-1}_{g})\big)\,. \tag{31}
$$

In an inactive group, candidates are sampled under $a^{t_g}_{g}$; no constructed media are encoded.

The single encoded return is expanded to $C_{\mathrm{rw}} = 4$ provenance-tagged logical identities:

$$
\begin{aligned}
\mathcal{W}_g &= \begin{cases} \{(x^{t_g}_{g,\mathrm{rw},0}, j) : j = 1, \dots, C_{\mathrm{rw}}\}, & b_g = 1, \\ \varnothing, & b_g = 0, \end{cases} \\
\mathcal{P}_g &= \mathcal{B}_g \uplus \mathcal{W}_g, \qquad |\mathcal{P}_g| = K + b_g C_{\mathrm{rw}}.
\end{aligned} \tag{32}
$$

The logical identities reuse one media tensor but participate separately in group centering, update-wide scaling, deterministic ranking, and top/bottom selection; coincident identities may therefore occupy multiple selected slots. Sampled and constructed entries share the HPSv3–NVRM–PI reward, advantage computation, selection rule, and complete clean-prefix training loss in Eq. 28. A constructed entry receives no privileged label or auxiliary loss, and if selected its encoded blocks provide the clean target and predecessor blocks. Its provenance is retained because it was not sampled from π_{old}.

G.7 Distributed Execution

For G groups, K candidates, J_{hist} history blocks, J target blocks, and S denoising evaluations per generated block, rollout costs $GS(J_{\mathrm{hist}} + KJ)$ sampler evaluations; training evaluates $GBJM$

Algorithm 1 Rewind augmentation for group g

Require: plan $\mathcal{C}_g$; stopped history h_g; $\pi_{\text{old}}^{(s)}$; K; C_{rw}
set $b_g \leftarrow 0$ and $a_{g,\text{eff}} \leftarrow a_g$
if $e_g = 1$ and the seed-fixed Bernoulli draw activates rewind **then**
  if the required compatible inputs are unavailable **then**
    abort before candidate sampling
  end if
  set $b_g \leftarrow 1$ and replace the target camera control using Equation 30
end if
sample $\mathcal{B}_g$ of K candidate clips from $\pi_{\text{old}}^{(s)}$ under $a_{g,\text{eff}}$
if $b_g = 1$ **then**
  encode the reversed preceding clip once and form $\mathcal{W}_g$ with C_{rw} logical identities
else
  $\mathcal{W}_g \leftarrow \varnothing$
end if
return $\mathcal{P}_g \leftarrow \mathcal{B}_g \uplus \mathcal{W}_g$

policy-block predictions for $M = 3$ roles. SP shards attention and K/V, FSDP shards parameters, gradients, and optimizer state within each SP replica, and DP reduces corresponding parameter shards across replicas that own disjoint groups.

Whenever capacity permits, a complete group remains on one node through rollout, decoding, reward evaluation, and selection. Videos are decoded in bounded microbatches while completed media undergo CPU-side staging and asynchronous reward evaluation. Group normalization and deterministic selection wait for all rewards. Only keyed scalar statistics and optimizer reductions cross nodes, rather than video, history, or K/V tensors. The loss is normalized by the actual global optimized-candidate count, and the optimizer commits at a common rank boundary. Appendix I.6 gives the concrete microbatches, timing boundary, and phase totals for the system summarized in Section 3.

H Reward-Model and Control Details

H.1 NVRM Preference Construction

NVRM training combines ground-truth–generated anchors with generated–generated comparisons under a shared history. We use the main-text notation $h_g = (x_g^0, \ldots, x_g^{t_g-1})$ for generated history and $c_g = (I_g, y_g, a_g)$ for the generation condition.

For a real–fake anchor, the observed continuation and a sampled continuation share the same history and condition:

$$\big(h_g, c_g, x_{g,+,0}^{t_g}, x_{g,-,0}^{t_g}\big), \qquad x_{g,+,0}^{t_g} = x_{g,\text{real},0}^{t_g}, \quad x_{g,-,0}^{t_g} \sim \pi_{\text{col}}(\cdot \mid h_g, c_g). \tag{33}$$

Here π_{col} denotes the checkpoint used for preference-data collection; it is not the online RL sampling policy π_{old}.

For a generated–generated group, sample

$$x_{gk,0}^{t_g} \sim \pi_{\text{col},g}(\cdot \mid h_g, c_g), \quad k \in V_g = \{1, \ldots, K_{\text{pref}}\}, \qquad K_{\text{pref}} = 12. \tag{34}$$

The 12 candidates yield the unordered comparison set

$$P_g = \{\{u, v\} : u, v \in V_g,\ u < v\}, \qquad |P_g| = \binom{K_{\text{pref}}}{2} = 66. \tag{35}$$

Each pair is annotated by a multimodal GPT judge using the shared-context template in Figure 12. The complete generated history and generation prompt are common evidence for both candidates, and the judge is instructed to evaluate only the two continuations. The rubric prioritizes identity consistency, scene consistency, and physical plausibility over superficial visual appeal. Content-specific sub-dimensions and paired scores elicit comparative evidence, while the retained annotation is the categorical overall winner.

Each pair is judged in both A/B orders, giving $2|P_g| = 132$ calls per group, and dataset splits are assigned by complete group. Let $J_g^{AB}(u, v)$ and $J_g^{BA}(u, v)$ denote the winner identity after mapping

System. You are a rigorous video-continuation quality judge. Compare two candidate continuations conditioned on the same complete video context. Prioritize identity consistency, scene consistency, and physical plausibility. Return only valid JSON matching the requested schema.

Task and inputs. Video A and Video B are generated from the *same* context and generation prompt. Judge only the candidate continuations; use the complete context as evidence for established identities, scene state, spatial layout, lighting, motion state, and physical constraints.

Generation prompt: `[shared prompt]`
Complete shared context: `[context video]`
Video A: `[candidate continuation A]`
Video B: `[candidate continuation B]`

Evaluation priorities.

1. **ID consistency:** preserve the identity, appearance, attributes, count, and temporal persistence of subjects, objects, characters, and distinctive anchors.
2. **Scene consistency:** preserve environment, geometry, spatial layout, landmarks, lighting, materials, and continuity across the context–candidate boundary.
3. **Physical plausibility:** preserve causal motion, object permanence, contact, support, gravity, inertia, scale, camera motion, and parallax; penalize warping, flicker, discontinuities, and impossible state changes.

Decision rules. For each category, define content-specific sub-dimensions and cite concrete temporal evidence. For every sub-dimension, require `score_a + score_b = 10`; `5/5` means that the evidence does not distinguish the candidates. Prefer the less inconsistent continuation even when both contain defects, and use `Tie` only when they are genuinely indistinguishable overall.

Required JSON. Return `categories`, each containing the category name, sub-dimension evidence and paired scores, a category-level reason, and `cat_winner`; then return an overall evidence-based `reason` and `winner` in {`Video A`, `Video B`, `Tie`}.

Figure 12: GPT pairwise-annotation template for generated–generated NVRM data. The two display orders contain the same shared context and candidate pair; only the A/B positions are exchanged.

the two display orders back to candidate indices. We retain a directed edge $u \to v$ only when both calls select u over v:

$$\begin{aligned} E_g &= \{(u,v) : \{u,v\} \in P_g,\ J_g^{AB}(u,v) = J_g^{BA}(u,v) = u\}, \\ U_g &= P_g \setminus \{\{u,v\} : (u,v) \in E_g \text{ or } (v,u) \in E_g\}. \end{aligned} \tag{36}$$

Thus U_g contains ties, invalid judgments, and order conflicts. The directed graph $G_g = (V_g, E_g, w_g)$ contains only bidirectionally stable observed preferences, and we use unit edge weights, $w_g(e) = 1$.

Minimum feedback arc set. Among all bijective orders π, we recover a maximum-weight consistent order

$$\pi_g^* \in \arg \max_{\pi : V_g \to \{1,\dots,K_{\text{pref}}\}} \sum_{(u,v) \in E_g} w_g(u,v)\, \mathbf{1}[\pi(u) < \pi(v)] \tag{37}$$

We remove the backward edges to obtain the acyclic set

$$E_g^\dagger = \{(u,v) \in E_g : \pi_g^*(u) < \pi_g^*(v)\}, \qquad G_g^\dagger = (V_g, E_g^\dagger, w_g). \tag{38}$$

The 12-node ordering problem is solved exactly by subset dynamic programming.

Minimum directed multicut. To prevent an uncertain pair from being reintroduced through a directed path, we solve

$$\begin{aligned} D_g^* \in \arg \min_{D \subseteq E_g^\dagger} & \sum_{e \in D} w_g(e) \\ \text{s.t.} \quad & \text{reach}_{G_g^\dagger \setminus D}(u,v) = 0, \\ & \text{reach}_{G_g^\dagger \setminus D}(v,u) = 0, \quad \forall \{u,v\} \in U_g. \end{aligned} \tag{39}$$

We export only surviving observed edges

$$E_g^* = E_g^\dagger \setminus D_g^*; \tag{40}$$

without completing the order or adding transitive pseudo-labels. The retained training pairs are therefore observed preferences accepted in both display orders.

Figure 6 counts unordered candidate pairs: the two A/B calls are not separate training examples. Of approximately 391K raw fake–fake pairs, strict order agreement retains 276K and MFAS–MDM filtering retains 260.5K. Together with 51.8K real–fake anchors, the independently rounded stage totals give the reported 312.4K-pair corpus. Complete groups are split before training, so no candidate or paired comparison crosses splits. Template scores and rationales are retained as auxiliary annotations; graph construction uses only the categorical winner with unit weight.

H.2 NVRM Architecture and Inputs

NVRM initializes from Qwen3-VL-4B-Instruct and trains LoRA adapters with rank 64, scale 128, and zero dropout on the query, key, value, output, gate, up, and down projections in all 36 language layers. The vision encoder and multimodal merger remain trainable, while the language backbone is updated only through LoRA. A bias-free 1×2560 scalar head is trained on the final non-padding state and the resulting model is evaluated in BF16.

Training configuration. We train NVRM for two epochs on eight GPUs with AdamW, learning rate 2×10^{-6}, and effective batch size 128 (8 per GPU with two accumulation steps). Training uses 5% linear warmup followed by a constant rate, BF16, gradient checkpointing, an 8,192-token limit, ZeRO stage 1, and global gradient clipping at norm 1; evaluation and checkpointing run every 100 steps.

For target clip $t > 0$, the input contains the condition text, one midpoint frame from each earlier history clip $0, \ldots, t-2$, and 1-FPS samples from the immediately preceding clip and candidate continuation. For an 81-frame, 16-FPS LingBot clip, the 1-FPS indices are $\{0, 16, 32, 48, 64, 80\}$. Frames are resized to height 360 while preserving aspect ratio. The two members of a preference pair always reuse the identical sampled history frames, so the only visual difference presented to the scorer is the candidate continuation.

H.3 Conditional Utility and Identifiability

NVRM trains on preferred and dispreferred continuations under a matched history and condition. We sample matched tuples and abbreviate their two scores as

$$(h^t, c, x^t_{+,0}, x^t_{-,0}) \sim \mathcal{D}_{\text{pref}}, \qquad q_\pm = q_\phi(h^t, c, x^t_{\pm,0}). \tag{41}$$

Its Bradley–Terry objective is

$$\mathcal{L}_{\text{NVRM}} = -\mathbb{E}_{\mathcal{D}_{\text{pref}}} \log \sigma(q_+ - q_-). \tag{42}$$

Pairwise supervision identifies score differences within a history, not an absolute utility shared by unrelated scenes. In particular, any history-dependent offset leaves every supervised comparison unchanged:

$$\begin{aligned} q'_\phi(h^t, c, x^t_0) &= q_\phi(h^t, c, x^t_0) + b(h^t, c), \\ q'_\phi(h^t, c, x^t_{a,0}) - q'_\phi(h^t, c, x^t_{b,0}) &= q_\phi(h^t, c, x^t_{a,0}) - q_\phi(h^t, c, x^t_{b,0}). \end{aligned} \tag{43}$$

WoRL uses NVRM only to compare candidates from the same history and centers their composite rewards within that group. Fixed pre-RL calibration supplies a stable numerical scale for combining reward components; it does not impose a globally identifiable NVRM utility.

H.4 Online Reward Composition

For every active reward component m, fixed pre-RL statistics define the normalization used in Equation 5:

$$\hat{r}_m(s) = \frac{s - \mu_m}{\sigma_m}. \tag{44}$$

For LingBot, $r^{\text{ctrl}} = r^{\text{PI}}$ is the perspective-invariant camera reward; for Helios, $r^{\text{ctrl}} = r^{\text{motion}}$ is the text-to-video motion score. The candidate reward is therefore

$$\begin{aligned} R_{gi} = \hat{r}_{\text{HPS}}(r^{\text{HPS}}_{gi}) + \hat{r}_{\text{ctrl}}(r^{\text{ctrl}}_{gi}) \\ + \mathbf{1}[t_g > 0]\hat{r}_{\text{NVRM}}(r^{\text{NVRM}}_{gi}), \end{aligned} \tag{45}$$

Equation 5 states this composition for continuation targets with $t_g > 0$. For clip 0, NVRM is omitted because no preceding video history exists. LingBot uses

$$\begin{aligned}(\mu_{\mathrm{HPS}}, \sigma_{\mathrm{HPS}}) &= (3.56, 2.04),\\ (\mu_{\mathrm{NVRM}}, \sigma_{\mathrm{NVRM}}) &= (-2.3933, 4.6782),\\ (\mu_{\mathrm{PI}}, \sigma_{\mathrm{PI}}) &= (0.4435, 0.1739).\end{aligned} \tag{46}$$

These constants remain fixed throughout RL; Equation 18 subsequently centers the combined reward within each shared-history group and uses the update-wide population standard deviation. Helios uses a separately estimated set of fixed calibration statistics.

H.5 Perspective-Invariant Camera Reward

The Perspective-Invariant Camera (PI Camera) reward uses VGGT-Ω-1B-512 (Wang et al., 2026a): eight candidate frames for clip 0, or 24 prefix and eight candidate frames for a continuation. Only the candidate suffix $\mathcal{C}$ is scored.

After re-referencing predicted and target translations p_t, q_t to their first pose, let D stack $D_t = R_t - R_0$ and let $D^\dagger$ be its pseudoinverse. Scale and fixed-pivot alignment are

$$\tilde{p} = \mathrm{vec}(p) - DD^\dagger \mathrm{vec}(p), \qquad \tilde{q} = \mathrm{vec}(q) - DD^\dagger \mathrm{vec}(q), \tag{47}$$

$$s = \frac{\tilde{p}^\top \tilde{q}}{\tilde{p}^\top \tilde{p}}, \qquad \Delta = D^\dagger \mathrm{vec}(sp - q), \tag{48}$$

$$p_t^{\mathrm{deorbit}} = sp_t - D_t\Delta. \tag{49}$$

With $s = 1$ for a degenerate or nonpositive solution and $\mathrm{Path}(r_{\mathcal{C}})$ the adjacent-translation path length,

$$e_{\mathrm{trans}} = \frac{\sqrt{|\mathcal{C}|^{-1}\sum_{t\in\mathcal{C}} \|p_t^{\mathrm{deorbit}} - q_t\|_2^2}}{\max\{\mathrm{Path}(q_{\mathcal{C}}), \mathrm{Path}(p^*_{\mathcal{C}}), \mathrm{Path}((D\Delta)_{\mathcal{C}}), 10^{-6}\}}, \tag{50}$$

where $p^* = sp$ for the initial clip and $p^* = p^{\mathrm{deorbit}}$ for a continuation.

For rotations re-referenced to the first candidate frame,

$$\theta(R_a, R_b) = \frac{180}{\pi}\arccos\left(\mathrm{clip}_{[-1,1]}\frac{\mathrm{tr}(R_a^\top R_b) - 1}{2}\right), \qquad e_{\mathrm{rot}} = \sqrt{|\mathcal{C}|^{-1}\sum_{t\in\mathcal{C}} \theta(\hat{R}_t^{\mathcal{C}}, R_t^{\mathcal{C}})^2}. \tag{51}$$

The final reward uses $\tau_{\mathrm{trans}} = 0.4$ and $\tau_{\mathrm{rot}} = 40°$:

$$r^{\mathrm{PI}} = \mathrm{clip}_{[0,1]}\left[\exp\left(-\frac{1}{2}\left(\frac{\max(e_{\mathrm{trans}}, 0)}{\tau_{\mathrm{trans}}} + \frac{\max(e_{\mathrm{rot}}, 0)}{\tau_{\mathrm{rot}}}\right)\right)\right]. \tag{52}$$

Pose estimates must be finite to receive a PI score.

I Evaluation Protocol

Before–after comparisons match prompts or cases, generation settings, and evaluation seeds; action-conditioned comparisons additionally match initial observations and controls. Evaluation cases are held out from post-training and model selection.

I.1 WBench Protocols

For action-conditioned models, we use the released 158-case WBench-Navigation split (Ying et al., 2026) with three rollouts per case, totaling 474 outputs. WBench-Text is used for Helios-Distilled-14B with three matched generation seeds per prompt. We retain the released prompts, preprocessing, frame sampling, parsing, and aggregation.

The asterisks in Table 1(a) identify action columns that contain VLM-backed submetrics. Specifically, Gemini 3.5 Flash (`gcp/google/gemini-3.5-flash`) (Google DeepMind, 2026) evaluates `scene_adherence` and `subject_adherence` within Settings and `causal_fidelity` within Physical, using medium reasoning effort, at most 5,090 output tokens, and temperature 0.1. The other action dimensions use non-VLM WBench evaluators.

I.2 VBench-Long Protocol

For Helios, VBench-Long (Huang et al., 2024; 2025c) is evaluated on the same before–after prompts, generation duration, and three-seed pairing used by the matched text-to-video comparison. We retain the released VBench-Long temporal sampling and aggregation and report its dataset-level score separately from WBench-Text.

I.3 RevBench Protocol

RevBench contains 150 fixed revisiting cases with three seeds per case. Each case establishes an anchor, leaves or occludes it, and later returns. The scene–trajectory combinations are disjoint from post-training data and model selection, and differ from the OpenScene revisit paths used to construct the RL environment. Models share initial observations, prompts, controls, and candidate seeds. The two displayed case-level metrics are **NVRM**, $100[\mathrm{clip}_{[-20,20]}(q_\phi) + 20]/40$, and **PI Camera**, 100 times Eq. 52. RevBench Overall is the arithmetic mean of the unrounded dataset-level NVRM and PI Camera aggregates, followed by final display rounding.

I.4 Reward-Model and Data-Cleaning Comparisons

Table 3 compares reward configurations with matched initialization and rollout/optimization budgets. Table 4 reports two matched component comparisons under the same evaluation protocol.

I.5 Human Evaluation

We collect 64 matched video pairs for human evaluation. Each pair is identity-blinded, randomly ordered, and shown as synchronized trajectories. The reward-model study reports agreement on consistency and physical plausibility, while the downstream study reports before-RL, tie, and after-RL preferences for consistency, camera control, physical plausibility, and overall quality.

I.6 Efficiency Measurement

We measure one complete optimizer update on 16 NVIDIA B200 GPUs in four four-GPU nodes, with SP$=4$, DP$=4$, 8 prompts, and 12 candidates per prompt. The update targets clip 2 of a three-clip, five-seconds-per-clip chain with 14 transformer history blocks and rollout/decode/training microbatches $2/2/1$; this profiling workload differs from the $G = 16$, $K = 24$ training profile in Table 7a.

Timing spans candidate execution through the optimizer commit. Synchronized rank-maximum timers measure candidate generation (380.7 seconds), reward scoring and exchange (193.2 seconds), selection through commit (145.0 seconds), and the full update (718.9 seconds). Candidate generation consists of 284.0 seconds of rollout and 96.7 seconds of decode/media staging. The complete update therefore takes 12.0 minutes, a $9.2\times$ speedup over the 109.8-minute literal baseline. Earlier rows preserve same-scale phase deltas, and the final bundle primarily changes Decode and Reward.

J Limitations

System speedups depend on workload, parallelism, memory, and interconnect. Parallel Clean-Prefix Attention removes redundant prefix projection but still attends to the available history. The NFT update and rewind augmentation optimize learned rewards and can exploit reward misspecification.

NVRM may miss short-lived events or become unreliable under ambiguous occlusion, and automated cleaning cannot correct systematic errors shared by the multimodal judge. Rewind is less applicable to irreversible events or scenes with independently moving objects. The camera score is also limited by failures of the underlying geometry estimator.

K Broader Impact and Responsible Use

Reward-driven world models can be useful for simulation and embodied-agent training, but they may also reinforce unsafe behavior or exploit weaknesses in the reward signal. Practical deployment therefore requires careful data review, human evaluation, monitoring, and red-team testing, and generated outputs should not be used directly to control safety-critical systems.