RAR: RATE-ADAPTIVE ROLLOUT FOR TIME-CONSISTENT INTERACTIVE WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Video-based interactive world models aim to enable real-time, user-controllable environments. Yet most existing systems are trained and rolled out at a fixed temporal rate, limiting both runtime-rate adaptation and physical-horizon coverage—a limitation we term **temporal-rate coupling**. Specifically, when runtime generation throughput differs from the learned temporal rate, the generated world evolves unintentionally more slowly or quickly. Moreover, using a single temporal rate imposes an inherent trade-off between temporal granularity and physical horizon: higher rates capture finer motion but cover a shorter duration under a fixed context window. To address both limitations, we introduce **Rate-Adaptive Rollout** (**RAR**), a framework that enables interactive world models to roll out at controllable temporal rates. At its core, *Stride-RoPE* aligns latent frames sampled at different rates on a shared physical-time coordinate, enabling consistent multi-rate rollout, unseen-rate generalization, and online rate switching without discarding history. Meanwhile, RAR uses multi-stride video sampling to enable joint learning of fine-grained dynamics and long-horizon evolution. Across action- and camera-controlled environments, RAR achieves temporally consistent multi-rate rollout and online rate switching while preserving high-rate generation quality and extending physical horizons under matched training budgets. RAR also transfers to pretrained world models while preserving perceptual quality.

1 INTRODUCTION

Interactive world models are extending video generation from offline content synthesis toward real-time, user-controllable environments (Bruce et al., 2024; He et al., 2025; Savva et al., 2026; Hu et al., 2026). Unlike conventional video generation, these models must continuously receive user actions or camera trajectories and promptly synthesize subsequent visual observations consistent with the control inputs. The generated content then becomes part of the history that conditions future evolution. Recent systems have demonstrated high-quality, long-horizon, and even real-time interactive generation in game and open-world settings (Valevski et al., 2025; Wang et al., 2026; Chen et al., 2026; Gao et al., 2026), highlighting their potential as neural game engines and general-purpose environment simulators. As these systems move toward practical deployment, they must advance the generated world at the intended pace under varying runtime conditions, while also capturing both fine-grained dynamics and long-horizon evolution within a limited temporal context.

Nevertheless, most existing interactive world models are trained and rolled out at a single fixed temporal rate (He et al., 2025; Wang et al., 2026; Savva et al., 2026). We use the temporal rate f to denote the number of temporal steps represented per second of physical time[1]. In a conventional fixed-rate model, the training rate and target rollout rate are tied together, *i.e.*, $f_{\text{train}} = f_{\text{rollout}} = f$. We argue that this fixed-rate design gives rise to two closely related limitations in world modeling:

- **Rate–throughput mismatch.** With a runtime throughput of r generation updates per wall-clock second, the modeled world advances at a relative speed of $\rho = r/f_{\text{rollout}}$. For users, holding a movement command for the same wall-clock duration can therefore produce too little or too much displacement, making navigation and action timing inconsistent across runtime conditions.

[1]In our video-based setting, temporal steps correspond to video or latent frames, so f is instantiated as frames per second (FPS), and each temporal step represents a physical-time interval of $\Delta t = 1/f$ seconds.

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

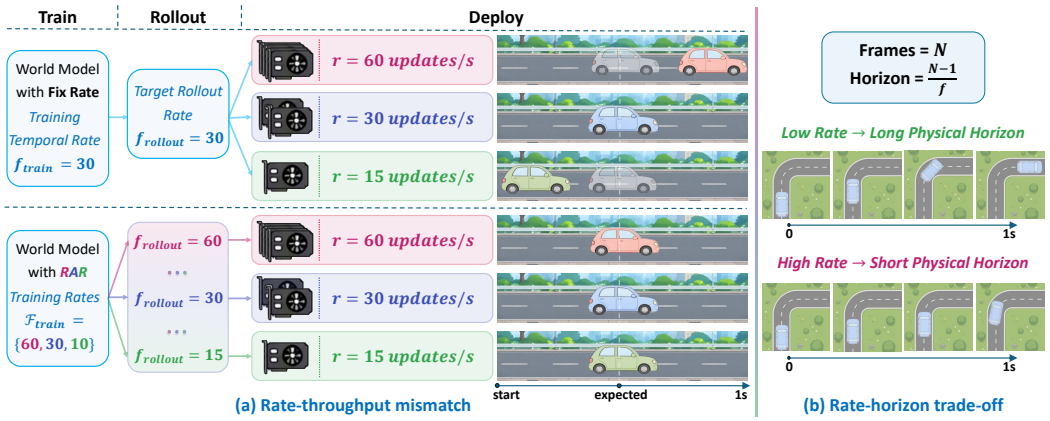

Figure 1: **(a)**: *Rate–throughput mismatch*. f_{train}, f_{rollout}, and r denote the training temporal rate, target rollout rate, and runtime generation throughput. After 1s of wall-clock time, colored cars mark generated positions, while gray cars mark the *expected* positions; their offsets indicate temporal mismatch. **(b)**: *Rate–horizon trade-off*. Under the same fixed budget of N frames, a higher rate provides finer temporal resolution but covers a shorter physical horizon, whereas a lower rate uses larger temporal steps to cover a longer horizon.

As shown in Fig. 1(a), the generated world lags behind or runs ahead of its expected state when $r < f_{\text{rollout}}$ or $r > f_{\text{rollout}}$, respectively. Although variable frame rates have been explored in offline video generation (Ho et al., 2022b; Blattmann et al., 2023; HaCohen et al., 2026), these methods do not address adapting the physical-time progression of continuous interactive rollouts to runtime generation throughput. Moreover, adjusting playback speed only changes display timing, not the physical-time interval represented by each generated transition; addressing this mismatch therefore requires controlling the temporal step size within the world model itself.

- **Rate–horizon trade-off.** With a fixed context budget of N temporal frames, the modeled physical horizon is $T = (N-1)/f$. As shown in Fig. 1(b), a higher f captures finer motion and control responses but covers a shorter duration, whereas a lower f extends the horizon with coarser temporal steps. This constrains the model's ability to learn both rapid control responses and longer-term environment evolution. Recovering a long horizon at a high temporal rate requires longer training sequences and substantially higher training cost per optimization update.

Although the two problems arise in different aspects of interactive world modeling, both stem from assigning a fixed physical-time increment to every model update. We refer to these two consequences collectively as **temporal-rate coupling**. Breaking this coupling requires learning controlled dynamics across temporal rates within a shared physical-time coordinate.

To this end, we introduce **R**ate-**A**daptive **R**ollout (**RAR**), which enables an interactive world model to be trained and rolled out across multiple temporal rates. RAR parameterizes the temporal rate by a stride s consistently during training and inference. To learn across different strides, the model must preserve consistent temporal relations between frames separated by the same physical-time interval. We therefore introduce *Stride-RoPE*, which modifies the original RoPE (Su et al., 2024), and maps latent frames sampled with different strides onto a shared physical-time coordinate for both visual observations and control conditioning.. During training, RAR synchronously samples video frames and control signals at multiple temporal strides, allowing the model to learn environment rollouts at varying temporal rates $\mathcal{F}_{\text{train}}$ by adjusting the stride $s \in \mathcal{S}_{\text{train}}$. This training paradigm also allows fine-rate samples to capture rapid local dynamics and control responses while coarse-rate samples cover longer physical horizons under the same sequence budget. At inference, the stride s directly controls the target rollout rate f_{rollout}, enabling the model to adapt its physical-time progression to runtime generation throughput. The stride can vary continuously within the learned range, including unseen intermediate rates, and can change online without discarding rollout history.

Across action- and camera-controlled interactive environments, RAR enables temporally consistent and controllable multi-rate rollout, including generalization to unseen rates and online rate

switching, while extending physical horizons without sacrificing high-rate generation quality under matched training budgets. RAR also transfers effectively to pretrained world models through finetuning, demonstrating its portability. Together, these results demonstrate that RAR effectively mitigates temporal-rate coupling by enabling controllable rollout rates to adapt world progression to runtime generation throughput and by extending physical horizons under a fixed sequence budget.

2 Related Work

Interactive World Models. World models learn predictive environment dynamics to anticipate future observations and support decision making (Ha & Schmidhuber, 2018). Recent video-based world models have progressed from action-conditioned future generation (Hu et al., 2023) toward continuously controllable environments that support interactive rollout through user actions or camera trajectories (Bruce et al., 2024; Sun et al., 2026; HunyuanWorld, 2025; Chen et al., 2026; Gao et al., 2026; Huang et al., 2026). Recent game- and open-world systems further scale this paradigm to more persistent closed-loop interaction, with stronger action–state coupling and improved long-horizon rollout consistency (Valevski et al., 2025; Po et al., 2026; He et al., 2025; Wang et al., 2026; Savva et al., 2026; Hu et al., 2026; DreamX et al., 2026). Despite these advances, existing systems generally associate each rollout update with a fixed temporal interval. Our work instead considers temporal rate itself as a controllable dimension of interactive rollout, rather than a fixed property.

Variable Frame-Rate Video Generation. Video generation models have explored varying temporal density through temporal resampling, super-resolution, and explicit FPS conditioning (Ho et al., 2022b;a; Chen et al., 2023; Xing et al., 2024; Blattmann et al., 2023). While FPS conditioning provides global rate control, it does not explicitly enforce shared temporal geometry across different rates. More recent methods incorporate frame rate into temporal positional coordinates to obtain physical-time-aware representations (HaCohen et al., 2024; 2026). However, these approaches primarily target finite offline videos with globally specified frame rates, rather than streaming interactive rollouts with time-consistent history across online rate changes.

Multi-Scale and Long-Horizon Video Modeling. Long-context methods extend the number of sequence positions available to a model for modeling longer-range dependencies through positional interpolation or RoPE rescaling (bloc97, 2023; Peng et al., 2024; Ding et al., 2024). In video modeling, temporal multi-scale and hierarchical approaches cover longer time spans by operating at multiple temporal resolutions or by generating sparse long-range structure before refining finer temporal details (Villar-Corrales et al., 2022; Du et al., 2023; Yin et al., 2023). These approaches extend temporal coverage by enlarging context or through hierarchical generation, whereas our setting varies the physical duration represented by a fixed-length rollout context.

3 Method

We introduce **Rate-Adaptive Rollout (RAR)**, a framework for multi-rate training and rollout of interactive world models. RAR is built around a unified temporal stride that parameterizes physical-time progression across training and inference. This stride defines rate-consistent temporal coordinates through *Stride-RoPE* (Sec. 3.2) and governs multi-stride training that balances rate and horizon, rate-controlled rollout and online rate switching (Sec. 3.3).

3.1 Problem Definition

We first formalize the problem. Let $\mathbf{x}_n = \mathbf{x}(t_n)$ denote the world state at physical time t_n, and let $\mathbf{a}_n$ denote the control applied during the transition from $\mathbf{x}_n$ to $\mathbf{x}_{n+1}$. Given the history of world states and controls, a causal interactive world model autoregressively predicts the next state as:

$$\widehat{\mathbf{x}}_{n+1} \sim p_\theta(\cdot \mid \mathbf{x}_{0:n}, \mathbf{a}_{0:n}) . \tag{1}$$

Each transition advances the modeled world by a physical-time interval $\Delta t_n = t_{n+1} - t_n$. Conventional models use a constant step size determined by the fixed training temporal rate, thereby tying each model update to a fixed amount of physical-time progression. We instead require the world model to support controllable temporal intervals Δt_n, both across training samples and within an ongoing rollout. Accordingly, we seek a variable-rate world model with two key properties. First,

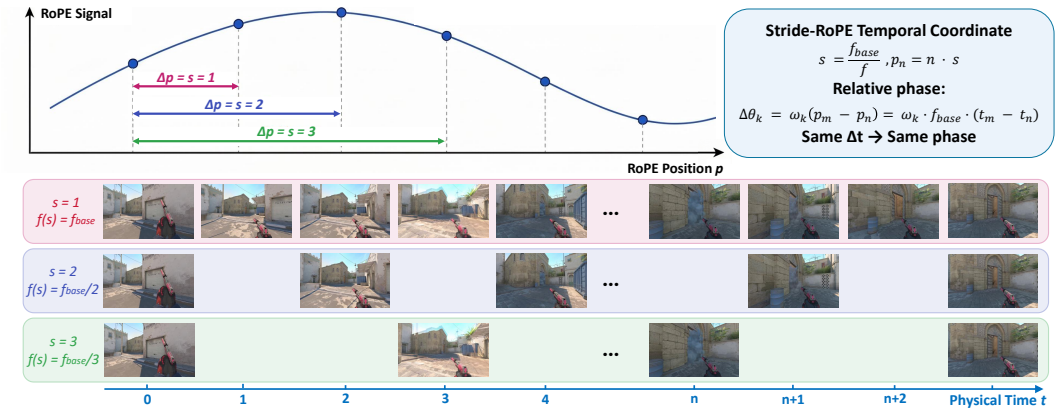

Figure 2: *Stride-RoPE*. Video sampled with temporal stride s uses RoPE positions $p_n = ns$, ensuring that equal physical-time intervals induce equal relative phases across sampling rates.

different temporal discretizations should preserve consistent temporal semantics at matched physical times, while changing the step size during rollout should not alter the physical-time relationships of previously generated states. Second, learning across multiple step sizes allows fine steps to capture fine-grained local dynamics and coarse steps to extend physical-time coverage under the same context budget. These requirements call for a shared physical-time representation across rates and a unified mechanism for learning and controlling temporal intervals.

3.2 STRIDE-ROPE

To support this physical-time consistency, we use a temporal positional representation in which equal physical-time intervals induce equal relative phases across sampling rates. We first examine the temporal geometry of RoPE, which encodes positions by rotating query and key subspaces. With frame-index-based temporal coordinates, the interaction between a query at index n and a key at index m in the k-th rotary subspace satisfies:

$$\langle R(n\omega_k)\mathbf{q}_{n,k}, R(m\omega_k)\mathbf{k}_{m,k}\rangle = \langle \mathbf{q}_{n,k}, R((m-n)\omega_k)\,\mathbf{k}_{m,k}\rangle\,, \tag{2}$$

where ω_k is the angular frequency of that subspace. The relative rotary phase is therefore determined by the frame-index difference $m - n$. For a sequence sampled at a constant temporal rate f, this difference corresponds to the physical-time interval:

$$t_m - t_n = \frac{m-n}{f}. \tag{3}$$

The relative phase can thus be written as $(m-n)\omega_k = f(t_m - t_n)\omega_k$. Consequently, the phase assigned to a given physical-time interval depends on the sampling rate: frame-index-based RoPE does not provide a shared physical-time reference across rates. Thus, the temporal positional encoding itself does not explicitly represent elapsed physical time.

To establish such a reference, we reparameterize the temporal RoPE coordinate in units of physical time. Given a base temporal rate f_{base}, we define the stride and the corresponding temporal coordinate for a sequence sampled at a constant rate f as:

$$s(f) = \frac{f_{\text{base}}}{f}, \qquad p_n = ns(f). \tag{4}$$

Instead of applying RoPE at the frame index n, *Stride-RoPE* applies it at the rescaled coordinate p_n. Taking $t_0 = 0$, we have $t_n = n/f$ and hence,

$$p_n = f_{\text{base}} t_n.$$

Thus, p_n measures physical time in units of the base-frame interval $1/f_{\text{base}}$. The relative rotary phase becomes:

$$(p_m - p_n)\omega_k = (m-n)\frac{f_{\text{base}}}{f}\omega_k = f_{\text{base}}(t_m - t_n)\omega_k. \tag{5}$$

For a fixed base rate, this phase depends on the physical-time interval rather than the sampling rate f. Equal physical-time intervals therefore induce equal relative rotary phases across rates, establishing a shared temporal positional geometry for multi-rate learning.

When $f = f_{\text{base}}$, we have $s = 1$ and $p_n = n$, so Stride-RoPE exactly recovers standard RoPE at the base rate and preserves its original positional geometry. The modification affects only the temporal coordinates of 3D RoPE: the rotary frequencies and spatial coordinates remain unchanged, and no additional learnable parameters are introduced. Wherever temporal RoPE is applied to visual observations and control conditioning, we use the same coordinate mapping to maintain a common physical-time reference for both.

Although training trajectories are sampled using a discrete set of strides, the coordinate $p_n = ns$ is continuous in s and well-defined for any real-valued $s > 0$. It therefore provides temporal coordinates for intermediate strides not explicitly observed during training. We next describe how RAR builds on Stride-RoPE to address temporal-rate coupling.

3.3 Rate-Adaptive Rollout Framework

RAR uses the stride variable to unify multi-rate training and rollout, with online rate switching achieved by varying the stride within an ongoing rollout. Fig. 3 summarizes our framework.

Multi-rate training. We choose a set of training strides $\mathcal{S}_{\text{train}}$ with respect to the fixed base temporal rate f_{base}, yielding the rate set $\mathcal{F}_{\text{train}} = \{f_{\text{base}}/s \mid s \in \mathcal{S}_{\text{train}}\}$. For a sampled stride $s \in \mathcal{S}_{\text{train}}$, we synchronously sample video and its associated control signals at the following physical times:

$$t_n^{(s)} = \frac{ns}{f_{\text{base}}}, \qquad \mathbf{x}_n^{(s)} = \mathbf{x}\left(t_n^{(s)}\right), \qquad \mathbf{a}_n^{(s)} = \mathbf{a}\left(t_n^{(s)}\right). \tag{6}$$

Here, $n = 0, \ldots, N-1$ indexes video frames and $n = 0, \ldots, N-2$ indexes transition controls. Every training sequence contains the same N video frames, with a uniform temporal interval $\Delta t(s) = s/f_{\text{base}}$, while its physical-time coverage is

$$T(s) = (N-1)\Delta t(s) = \frac{(N-1)s}{f_{\text{base}}}. \tag{7}$$

Fine-rate samples capture fine-grained local motion and control responses, whereas coarse-rate samples cover longer physical horizons under the same sequence budget. Joint training across these rates therefore enables learning at multiple temporal scales without increasing the training sequence length. This stride-controlled sampling strategy can be readily applied to different training stages, including bidirectional, causal training, and distillation.

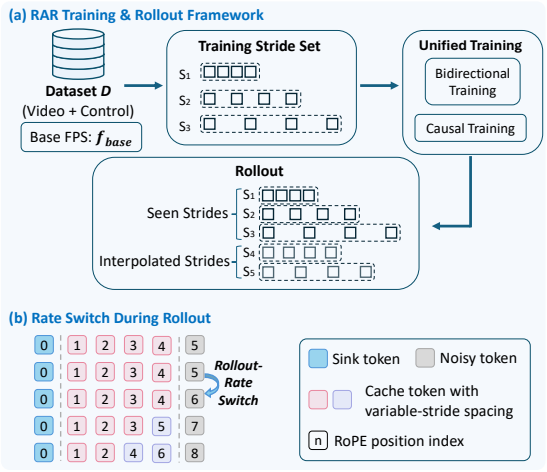

Figure 3: RAR pipeline. **(a)** Multi-stride training and rate-controlled rollout. **(b)** Rollout-rate switching with stride-aware temporal spacing.

Rate-controlled rollout. At inference, RAR allows the temporal interval to be selected at each autoregressive step. Let s_n denote the stride for the transition from $\mathbf{x}_n$ to $\mathbf{x}_{n+1}$, corresponding to the interval $\Delta t_n = s_n/f_{\text{base}}$. For a constant stride $s_n = s_{\text{rollout}}$ and runtime generation throughput r, the target rollout rate and relative physical-time progression are

$$f_{\text{rollout}} = \frac{f_{\text{base}}}{s_{\text{rollout}}}, \qquad \rho = \frac{r}{f_{\text{rollout}}} = \frac{r\, s_{\text{rollout}}}{f_{\text{base}}}. \tag{8}$$

Together with multi-rate training, the continuous stride parameterization of Stride-RoPE supports rollout at unseen intermediate rates within the training range. When r falls within this range, setting $s_{\text{rollout}} = f_{\text{base}}/r$ matches the target rollout rate to runtime throughput, yielding $\rho = 1$.



Figure 4: Qualitative multi-rate generation. Within each environment, rows share the same initial condition and underlying physical-time control trajectory, while columns are aligned by physical time. Frame indices are marked at the lower left of each frame. At matched physical times, RAR produces similar scene states across rollout rates.

More generally, the stride can vary from one transition to the next within the supported range, allowing the rollout rate to change online. We compute the temporal RoPE coordinate p_n of frame n by accumulating the selected strides:

$$\begin{aligned} p_0 &= 0, \qquad p_{n+1} = p_n + s_n, \\ p_\ell - p_i &= \sum_{j=i}^{\ell-1} s_j = f_{\text{base}}(t_\ell - t_i), \qquad i < \ell. \end{aligned} \tag{9}$$

For non-sink frame pairs, the relative rotary phase remains $(p_\ell - p_i)\omega_k = f_{\text{base}}(t_\ell - t_i)\omega_k$, extending the physical-time alignment in Eq. 5 to rollouts with time-varying strides.

We implement causal rollout with a KV cache following the *Unified Relative RoPE Recipe* (Bian et al., 2026). The cache stores pre-RoPE keys together with their corresponding values. Sink frames retain fixed RoPE positions following the adopted cache recipe. For the current frame ℓ and a cached frame i, temporal RoPE in the k-th rotary subspace is applied as:

$$\widetilde{\mathbf{q}}_{\ell,k} = R(p_\ell \omega_k)\mathbf{q}_{\ell,k}, \qquad \widetilde{\mathbf{k}}_{i,k} = R(p_i \omega_k)\mathbf{k}^{\text{raw}}_{i,k}. \tag{10}$$

At each attention step, the coordinates of active frames are determined from the accumulated stride history. These coordinates may be translated by a common offset within the active window without changing their relative rotary phases. Applying the resulting rotations directly to the cached raw keys allows the rollout rate to change without clearing the cache or recomputing historical features.

Table 1: Multi-rate generation under a fixed rollout budget. Drift is the ratio between last-quarter and first-quarter LPIPS. DOVER is the no-reference overall perceptual quality score in $[0, 1]$; higher is better. SR measures rate fidelity and has an ideal value of 1.

Environment	FPS	Type	Horizon (s)	LPIPS ↓	Drift	DOVER ↑	SR → 1
Doom	14	Train	15.1	0.4279	1.17	0.3205	0.923
	28	**Interp.**	7.6	0.4179	1.22	0.3269	1.003
	35	Train	6.1	0.3991	1.27	0.3153	0.999
	49	**Interp.**	4.3	0.4003	1.26	0.3156	0.986
	70	Train	3.0	0.3852	1.34	0.3049	1.082
CS2	16	Train	15.0	0.6489	1.52	0.3928	0.983
	24	**Interp.**	10.0	0.6200	1.60	0.4187	1.009
	32	Train	7.5	0.5980	1.61	0.4402	0.970
	48	**Interp.**	5.0	0.5944	1.67	0.4369	1.024
	64	Train	3.8	0.5823	1.69	0.4323	1.097
FPV	20	Train	12.0	0.5247	1.35	0.2827	1.031
	30	Train	8.0	0.5161	1.41	0.2766	1.040
	40	**Interp.**	6.0	0.5011	1.42	0.2733	0.960
	50	**Interp.**	4.8	0.5035	1.45	0.2843	1.007
	60	Train	4.0	0.4944	1.49	0.2956	0.917

4 Experiments

In this section, we first introduce the experimental setup and then evaluate RAR's rate control, training benefits, and portability. To address the rate–throughput mismatch, we examine two levels of rate control: **Q1:** Can RAR generate at specified rates, including unseen rates, while preserving visual quality and temporal consistency? **Q2:** Can it further switch rates online without interrupting the rollout? We then turn to the rate–horizon trade-off and ask **Q3:** Can multi-rate training alleviate this trade-off under matched training budgets? Finally, we assess portability: **Q4:** Can RAR enable multi-rate rollout in pretrained world models without training from scratch?

4.1 Experimental Setup

Environments. We evaluate RAR in three interactive environments: Doom (id Software, 1993), Counter-Strike 2 (CS2) (Valve Corporation, 2023), and FPV (Pfeiffer & Scaramuzza, 2021). Doom and CS2 are first-person interactive games controlled through discrete keyboard-and-mouse actions, whereas FPV, a first-person drone racing simulator, involves continuous camera-trajectory control.

Training and evaluation. We follow Solaris's three-stage training pipeline (Savva et al., 2026), initializing Doom and CS2 models from Matrix-Game 2.0 (He et al., 2025) and FPV from SkyReels (Chen et al., 2025). For each environment, we train one model on its temporal-rate set $\mathcal{F}_{\text{train}}$. The training details are provided in Appd. A. We evaluate each model at both training rates and unseen intermediate rates within the training range. Cross-rate comparisons use matched initial conditions, random seeds, and the same underlying physical-time control trajectories. Within each environment, we fix the number of generated frames across rate settings.

Metrics. We evaluate perceptual generation quality using the reference-based LPIPS (Zhang et al., 2018) and the no-reference DOVER overall score (Wu et al., 2023). To evaluate temporal semantics, we introduce the speed ratio SR as a rate-fidelity metric:

$$\mathrm{SR}(f) = \frac{v_{\text{pred}}(f)/\bar{v}_{\text{pred}}}{v_{\text{GT}}(f)/\bar{v}_{\text{GT}}}. \tag{11}$$

Here, $v(f)$ denotes motion speed estimated from RAFT (Teed & Deng, 2020) optical flow using the physical-time interval between the compared frames, and $\bar{v}$ is its geometric mean across the rates within the training range. A value of $\mathrm{SR} = 1$ indicates that the generated and ground-truth videos exhibit the same cross-rate temporal scaling. Values above or below 1 indicate over- or under-scaling of motion at that rate compared with the ground-truth cross-rate trend. For SR evaluation, we filter out cases where optical flow estimation fails and remove spatial regions with excessively small or large motion magnitudes to reduce the influence of unreliable flow estimates. The complete per-case definition, aggregation, and filtering procedure are provided in Appd. B.

4.2 Rate-Controlled Rollout

4.2.1 Multi-Rate Generation (Q1)

Settings. We compare rollouts at training and unseen intermediate rates, holding the target rate constant within each rollout. Qualitative comparisons use frames at matched physical times, while quantitative evaluations fix the number of generated frames within each environment.

Results. As shown in Fig. 4, RAR produces similar scene evolution and action progress at matched physical times across rollout rates. This consistency also holds at unseen intermediate rates, supporting interpolation with the same model without retraining.

Tab. 1 reports quantitative results under a fixed rollout budget. With the same number of generated frames, lowering the rollout rate from the highest to the lowest evaluated rate extends the physical horizon by factors of 5, 4, and 3 on Doom, CS2, and FPV, respectively. These extensions are accompanied by modest LPIPS increases of 6.1%–11.4% relative to the respective highest-rate rollouts, while DOVER scores remain broadly comparable across rates within each environment. Unseen intermediate rates also achieve LPIPS and DOVER scores comparable to neighboring training rates. RAR also preserves consistent temporal scaling across rates. The mean $|\mathrm{SR} - 1|$ is 3.5% on Doom, 3.5% on CS2, and 4.0% on FPV. Across all six unseen intermediate rates, SR ranges from 0.960 to 1.024, with a maximum deviation of 4.0% from the ideal value. Their deviations are comparable to those of the training rates, indicating no systematic loss of rate fidelity at unseen rates.

Together, these results show that a single RAR model supports temporally consistent rollout across multiple rates and interpolates to unseen intermediate rates within the training range without rate-specific retraining. Comparisons with alternative rate-control methods and results beyond the training range are provided in Appd. C and Appd. D, respectively.

4.2.2 Dynamic Rollout-Rate Switching (Q2)

Settings. Beyond fixed-rate generation, we evaluate whether RAR can follow changing rollout rates within a continuous rollout without clearing the KV cache or restarting generation. We use two stride schedules: a smoothly varying curve and an abrupt schedule comprising six constant-stride segments. For each rollout, we rescale the optical-flow measurements to the target stride scale using a global factor before fitting the response.

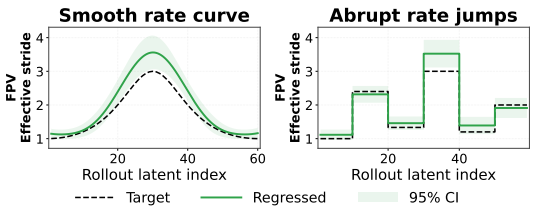

Figure 5: Dynamic rollout-rate switching results on FPV model. Dashed curves denote target strides; solid curves show the mean calibrated and regressed stride estimates over retained cases, with shading indicating 95% confidence intervals.

Results. Fig. 5 shows the mean fitted response on FPV and its 95% confidence interval over the retained cases. The reported Pearson correlations with the target schedules are 0.998 for the smooth schedule and 0.970 for the abrupt schedule. These results support RAR's ability to track changes in the target stride during a continuous rollout. Evaluation details and additional results on Doom and CS2 are provided in Appd. E.

4.3 Rate–Horizon Trade-off (Q3)

Settings. We conduct two comparisons at the bidirectional training stage on Doom. (1) Matched updates: We fix the training window (213 frames), model architecture, and number of updates, comparing RAR with a single-rate model trained only at 70 FPS. (2) Matched training time: We compare the 213-frame RAR model with fixed-rate models trained using 425- or 849-frame windows at 70 FPS under approximately matched wall-clock budgets. For each matched-time comparison, both models are evaluated at 70 FPS using the same rollout length of 425 or 849 frames.

Results. (1) Under matched updates, RAR substantially improves generation quality at the lower rollout rates while retaining nearly identical performance at 70 FPS, as shown in Tab. 2 (top). This indicates that exposing the model to longer physical horizons through coarse-rate samples does not

Table 2: Rate–horizon comparisons on Doom: LPIPS under matched updates (top) and approximately matched wall-clock time (bottom). The baseline uses Stride-RoPE at inference.

Protocol	Evaluation	Baseline	RAR
10k updates	14 FPS	0.4376	**0.3834**
	35 FPS	0.3863	**0.3413**
	70 FPS	**0.3113**	0.3135
Matched time	425 frames	0.3500	**0.3383**
	849 frames	0.3745	**0.3552**

Table 3: Multi-rate adaptation of SANA-WM. Baseline model directly applies Stride-RoPE at inference without additional finetuning. † marks the baseline model's native output rate.

		SR → 1		LPIPS ↓	
FPS	Type	Base	RAR	Base	RAR
10	Train	0.598	**0.641**	0.5702	0.4956
16†	**Interp.**	**1.203**	1.238	0.5684	0.5009
20	Train	0.983	**0.989**	0.5707	0.4979
24	**Interp.**	**1.173**	**1.173**	0.5673	0.5022
30	Train	1.234	**1.127**	0.5678	0.4999

come at the expense of its fine-rate generation capability. (2) For the matched-time comparison, the 425- and 849-frame training configurations take $1.56\times$ and $5.25\times$ as long per update as the 213-frame configuration, respectively. At the approximately matched checkpoints reported in Tab. 2 (bottom), RAR achieves lower LPIPS than the corresponding longer-window models.

These results show that RAR alleviates the rate–horizon trade-off from two complementary perspectives: multi-rate training preserves fine-rate generation quality while exposing the model to longer physical-time contexts, and achieves long-horizon generation more effectively under a fixed training budget than directly lengthening high-rate training sequences. Appd. F provides additional experimental details and long-rollout results, further examining the benefits of multi-rate training across checkpoints under comparable training budgets.

4.4 Portability to Pretrained World Models (Q4)

Settings. We apply a two-stage adaptation to pretrained SANA-WM (Zhu et al., 2026), using a subset of its training data (Li et al., 2025). We fine-tune a bidirectional teacher on data sampled at 10, 20, and 30 FPS for 3k steps, followed by 3k steps of multi-rate Self-Forcing (Huang et al., 2025) to obtain a streaming student. We compare the student with the released SANA-WM Streaming model. The baseline applies Stride-RoPE at inference with $f_{\text{base}} = 16$, whereas the adapted model uses $f_{\text{base}} = 30$ for both training and rollout. Both models are evaluated under the same sampling protocol for SR. We also report LPIPS as an auxiliary reference for perceptual quality.

Results. As shown in Tab. 3, LPIPS does not increase at any evaluated rate, providing a complementary quality check after adaptation. For rate fidelity, except at the base model's native rate of 16 FPS, RAR yields SR values that are closer to 1 or effectively unchanged: it improves the results at 10, 20, and 30 FPS and remains comparable at 24 FPS. These results suggest that multi-rate adaptation can further improve stride-based rate control in the pretrained model. The baseline applies Stride-RoPE without finetuning, while rates near its native 16 FPS, such as 20 FPS, require only modest temporal rescaling. These factors may partly explain why the baseline already achieves reasonable rate fidelity at some non-native rates. The residual under-scaling at 10 FPS may reflect the difficulty of adapting to larger temporal intervals under a limited training budget, with optical-flow measurement uncertainty potentially contributing to the observed deviations. Overall, RAR adaptation improves rate fidelity at multiple rollout rates while preserving perceptual quality, supporting its portability.

5 Conclusion

We introduce Rate-Adaptive Rollout (RAR) to address temporal-rate coupling in interactive world models. This coupling encompasses the mismatch between runtime throughput and the target rollout rate, as well as the trade-off between temporal resolution and physical horizon under a fixed budget. RAR combines physical-time-aligned Stride-RoPE with synchronized multi-stride sampling of videos and controls to learn controlled dynamics across temporal scales and support controllable rollout. RAR demonstrates consistent generation across training and unseen rates, online rate switching, and an improved quality–horizon trade-off. These capabilities allow RAR to adapt the rollout rate to runtime throughput while preserving consistent physical-time semantics, offering a practical path toward flexible, real-time interactive world models across hardware conditions.

AI USE STATEMENT

Generative AI tools were used to assist with language editing and improving the readability of the manuscript, to help draft preliminary versions of figures and visualizations, and to provide limited assistance with code development and debugging. All AI-assisted figures and code were reviewed, modified, and verified by the authors. Generative AI tools were not used to formulate the main research ideas, design the methodology or experiments, interpret experimental results, or draw scientific conclusions. The authors take full responsibility for the final content of this work.

REFERENCES

Yuxuan Bian, Zeyue Xue, Songchun Zhang, Shiyi Zhang, Weiyang Jin, Yaowei Li, Junhao Zhuang, Haoran Li, Jie Huang, Haoyang Huang, Nan Duan, and Qiang Xu. Echo-infinity: Learnable evolving memory for real-time infinite video generation. *arXiv preprint arXiv:2606.04527*, 2026.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

bloc97. NTK-Aware Scaled RoPE allows LLaMA models to have extended (8k+) context size without any fine-tuning and minimal perplexity degradation., 2023. URL https://www.reddit.com/r/LocalLLaMA/comments/14lz7j5/ntkaware_scaled_rope_allows_llama_models_to_have/.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first international conference on machine learning*, 2024.

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

Guibin Chen, Dixuan Lin, Jiangping Yang, Chunze Lin, Junchen Zhu, Mingyuan Fan, Hao Zhang, Sheng Chen, Zheng Chen, Chengcheng Ma, Weiming Xiong, Wei Wang, Nuo Pang, Kang Kang, Zhiheng Xu, Yuzhe Jin, Yupeng Liang, Yubing Song, Peng Zhao, Boyuan Xu, Di Qiu, Debang Li, Zhengcong Fei, Yang Li, and Yahui Zhou. Skyreels-v2: Infinite-length film generative model, 2025. URL https://arxiv.org/abs/2504.13074.

Haoxin Chen, Menghan Xia, Yingqing He, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, et al. Videocrafter1: Open diffusion models for high-quality video generation. *arXiv preprint arXiv:2310.19512*, 2023.

Zhifei Chen, Luozhou Wang, Guibao Shen, Dongyu Yan, Shuai Yang, Tianshuo Xu, Yihua Du, Wei Wang, Tianyi Gui, Lianghua Huang, and Yingcong Chen. Reworld: An interactive world model with long-horizon memory. *arXiv preprint arXiv:2608.23565*, 2026.

Yiran Ding, Li Lyna Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. Longrope: Extending llm context window beyond 2 million tokens. *arXiv preprint arXiv:2402.13753*, 2024.

Team DreamX, Yancheng Bai, Rui Chen, Xiangxiang Chu, Rujing Dang, Hao Dou, Bingjie Gao, Qiwen Gu, Siyu Hong, Jiachen Lei, et al. Dreamx-world 1.0: A general-purpose interactive world model. *arXiv preprint arXiv:2606.16993*, 2026.

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *Advances in neural information processing systems*, 36:9156–9172, 2023.

Zelin Gao, Qiuyu Wang, Jiapeng Zhu, Jingye Chen, Zichen Liu, Qingyan Bai, Jiahao Wang, Yufeng Yuan, Hanlin Wang, Yichong Lu, Ka Leong Cheng, Haojie Zhang, Jian Gao, Tianrui Feng, Yuzheng Liu, Yao Yao, Yinghao Xu, Xing Zhu, Yujun Shen, and Hao Ouyang. Infinite worlds with versatile interactions. *arXiv preprint arXiv:2607.07534*, 2026.

Zeyuan Gong and Weiyu Ma. Cs2 demo to dataset: Hltv and player-pov demo to synchronized frames + tick parquet, 2026. URL https://github.com/vast7777/cs2-demo-to-dataset. Software.

David Ha and J"urgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, et al. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024.

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.

Xianglong He, Chunli Peng, Zexiang Liu, Boyang Wang, Yifan Zhang, Qi Cui, Fei Kang, Biao Jiang, Mengyin An, Yangyang Ren, Baixin Xu, Hao-Xiang Guo, Kaixiong Gong, Cyrus Wu, Wei Li, Xuchen Song, Yang Liu, Eric Li, and Yahui Zhou. Matrix-game 2.0: An open-source, real-time, and streaming interactive world model. *arXiv preprint arXiv:2508.13009*, 2025.

Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022a.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in neural information processing systems*, 35:8633–8646, 2022b.

Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. Gaia-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023.

Anthony Hu, Václav Volhejn, Adrien Ramanana Rahary, Chris Mulder, Aditya Makkar, Alyx Liao, Amélie Royer, Manu Orsini, Adam Jelley, Eloi Alonso, Florian Laurent, Fredrik Norén, James Swingos, Jan Hünermann, Kent Rollins, Lucas Hosseini, Matthieu Le Cauchois, Maxim Peter, Pim de Witte, Tim Brown, Vincent Micheli, Moritz Böhle, Gabriel de Marmiesse, Viktoriia Sharmanska, Lucia Specia, Michael Black, and Patrick Pérez. Multiplayer interactive world models with representation autoencoders, 2026. URL https://arxiv.org/abs/2607.05352.

Junchao Huang, Guian Fang, Shengju Qian, Xianghao Kong, Zhuoran Zhao, Wei Huang, Yihua Du, Zixin Zhang, Justin Cui, Yuchao Gu, et al. Solarwm: Open data and scalable training for long-horizon video world models. *arXiv preprint arXiv:2609.02886*, 2026.

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*, 2025.

Team HunyuanWorld. Hy-world 1.5: A systematic framework for interactive world modeling with real-time latency and geometric consistency. *arXiv preprint*, 2025.

id Software. DOOM. Video game, 1993.

Zhen Li, Chuanhao Li, Xiaofeng Mao, Shaoheng Lin, Ming Li, Shitian Zhao, Zhaopan Xu, Xinyue Li, Yukang Feng, Jianwen Sun, Zizhen Li, Fanrui Zhang, Jiaxin Ai, Zhixiang Wang, Yuwei Wu, Tong He, Jiangmiao Pang, Yu Qiao, Yunde Jia, and Kaipeng Zhang. Sekai: A video dataset towards world exploration. *arXiv preprint arXiv:2506.15675*, 2025.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. Yarn: Efficient context window extension of large language models. In *International Conference on Learning Representations*, volume 2024, pp. 31932–31951, 2024.

Aleksei Petrenko, Zhehui Huang, Tushar Kumar, Gaurav S. Sukhatme, and Vladlen Koltun. Sample factory: Egocentric 3d control from pixels at 100000 FPS with asynchronous reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7652–7662. PMLR, 2020. URL http://proceedings.mlr.press/v119/petrenko20a.html.

Christian Pfeiffer and Davide Scaramuzza. Human-piloted drone racing: Visual processing and control. *IEEE Robotics and Automation Letters*, 6(2):3467–3474, 2021.

Ryan Po, David Junhao Zhang, Amir Hertz, Gordon Wetzstein, Neal Wadhwa, and Nataniel Ruiz. Multigen: Level-design for editable multiplayer worlds in diffusion game engines, 2026. URL https://arxiv.org/abs/2603.06679.

Georgy Savva, Oscar Michel, Daohan Lu, Suppakit Waiwitlikhit, Timothy Meehan, Dhairya Mishra, Srivats Poddar, Jack Lu, and Saining Xie. Solaris: Building a multiplayer video world model in minecraft. *arXiv preprint arXiv:2602.22208*, 2026.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

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, 2026. URL https://arxiv.org/abs/2512.14614.

Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pp. 402–419. Springer, 2020.

Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. In *International Conference on Learning Representations*, volume 2025, pp. 73754–73776, 2025.

Valve Corporation. Counter-Strike 2. Video game, 2023. URL https://www.counter-strike.net/cs2.

Angel Villar-Corrales, Ani Karapetyan, Andreas Boltres, and Sven Behnke. Mspred: Video prediction at multiple spatio-temporal scales with hierarchical recurrent networks. *arXiv preprint arXiv:2203.09303*, 2022.

Zile Wang, Zexiang Liu, Jiaxing Li, Kaichen Huang, Baixin Xu, Fei Kang, Mengyin An, Peiyu Wang, Biao Jiang, Yichen Wei, et al. Matrix-game 3.0: Real-time and streaming interactive world model with long-horizon memory. *arXiv preprint arXiv:2604.08995*, 2026.

Haoning Wu, Erli Zhang, Liang Liao, Chaofeng Chen, Jingwen Hou, Annan Wang, Wenxiu Sun, Qiong Yan, and Weisi Lin. Exploring video quality assessment on user generated contents from aesthetic and technical perspectives. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20087–20097. IEEE, 2023.

Marek Wydmuch, Michał Kempka, and Wojciech Jaśkowski. ViZDoom Competitions: Playing Doom from Pixels. *IEEE Transactions on Games*, 11(3):248–259, 2019. doi: 10.1109/TG.2018.2877047. The 2022 IEEE Transactions on Games Outstanding Paper Award.

Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Wangbo Yu, Hanyuan Liu, Gongye Liu, Xintao Wang, Ying Shan, and Tien-Tsin Wong. Dynamicrafter: Animating open-domain images with video diffusion priors. In *European Conference on Computer Vision*, pp. 399–417. Springer, 2024.

Shengming Yin, Chenfei Wu, Huan Yang, Jianfeng Wang, Xiaodong Wang, Minheng Ni, Zhengyuan Yang, Linjie Li, Shuguang Liu, Fan Yang, et al. Nuwa-xl: Diffusion over diffusion for extremely long video generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1309–1320, 2023.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 586–595. IEEE, 2018.

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*, 2026.

Appendix

A Experimental Details

We evaluate RAR on Doom, CS2, and FPV. We generate the Doom data using ViZDoom (Wydmuch et al., 2019) and Sample Factory (Petrenko et al., 2020), and construct the CS2 data from game demos (Gong & Ma, 2026); both datasets include keyboard-and-mouse controls. For FPV, we use the public Human-Piloted Drone Racing dataset and its camera trajectories (Pfeiffer & Scaramuzza, 2021). We initialize the Doom and CS2 models from Matrix-Game 2.0 (He et al., 2025), and initialize the FPV model from SkyReels (Chen et al., 2025). After adding action control modules, the Doom and CS2 models contain approximately 1.8B parameters each, while the FPV model contains approximately 1.3B parameters. Tab. 4 summarizes the main settings for each environment. All training rates within an environment share the same model parameters and fixed sequence length, and therefore correspond to different physical horizons. We have also provided the details of the training we conducted in the SANA-WM fine-tuning process (Sec. 4.4).

Table 4: Model and training settings. Bid., DF, and SF denote bidirectional training, Diffusion Forcing, and Self-Forcing, respectively. F is the sequence length in video frames. Bid. and DF entries report steps / learning rate; SF entries report steps / student learning rate / fake-model learning rate.

Env.	Res.	Params.	f_{base}	$\mathcal{F}_{\text{train}}$	F	Bid.	DF	SF
Doom	360p	1.8B	70	$\{14, 35, 70\}$	213	$10k/10^{-4}$	$12k/10^{-4}$	$5k/10^{-7}$, 3×10^{-8}
CS2	360p	1.8B	64	$\{16, 32, 64\}$	241	$10k/10^{-4}$	$10k/10^{-4}$	$2k/10^{-7}$, 3×10^{-8}
FPV	480p	1.3B	60	$\{20, 30, 60\}$	241	$15k/10^{-4}$	$11k/10^{-4}$	$5k/10^{-6}$, 3×10^{-7}
SANA-WM	720p	1.6B	30	$\{10, 20, 30\}$	961	$3k/10^{-5}$	—	$3k/10^{-5}$, 10^{-6}

Following Solaris (Savva et al., 2026), we adopt a similar three-stage training pipeline consisting of bidirectional training, Diffusion Forcing (Chen et al., 2024), and Self-Forcing (Huang et al., 2025). The environment-specific training steps and learning rates are listed in Tab. 4.

B Rate-Fidelity Evaluation

Here, we provide the complete definition of Eq. 11. Let $m_{k,d}(f, c)$ denote the optical-flow magnitude between frames separated by k steps for video $d \in \{\text{pred}, \text{GT}\}$ in case c at temporal rate f. Since this interval corresponds to k/f seconds, we define the physical motion speed for that case and rate as

$$v_d(f, c) = \frac{m_{k,d}(f, c)}{k/f}. \tag{12}$$

To focus on cross-rate scaling rather than the overall motion magnitude of each case, we normalize speeds separately for generated and ground-truth videos. For each case, the reference speed is the geometric mean over $\mathcal{F}_{\text{in}}$, the set of evaluated rates within the training range, including both training and interpolation rates:

$$\bar{v}_d(c) = \exp\left(\frac{1}{|\mathcal{F}_{\text{in}}|}\sum_{f'\in\mathcal{F}_{\text{in}}} \log v_d(f', c)\right). \tag{13}$$

The ratio $v_d(f, c)/\bar{v}_d(c)$ measures the motion speed at rate f relative to the reference speed of the same case, removing variations in motion magnitude across scenes. We take its median across cases for robust aggregation and divide it by the same statistic on the GT side to reduce systematic biases in optical-flow measurement. This yields the rate fidelity

$$\text{SR}(f) = \frac{\text{median}_c\left[v_{\text{pred}}(f, c)/\bar{v}_{\text{pred}}(c)\right]}{\text{median}_c\left[v_{\text{GT}}(f, c)/\bar{v}_{\text{GT}}(c)\right]}, \tag{14}$$

where a value close to one indicates agreement with the ground-truth cross-rate scaling, rather than absolute speed accuracy. For the main experiments, we estimate optical flow over short generated

clips using approximately matched physical-time intervals and apply two-stage filtering to ensure measurement reliability. We first exclude low-quality samples at the case level according to quality metrics, and then remove spatial regions with excessively small or large motion magnitudes, retaining regions with reliable motion signals for rate estimation.

C Comparison with Alternative Rate-Control Methods

We compare RAR with two alternative rate-control methods using the same backbone, three-stage training pipeline, optimization-step budgets, and other training settings, except for the training-rate sampling and rate-conditioning mechanisms described below. **Single** is trained only at the base FPS of each environment (70 FPS for Doom, 64 FPS for CS2, and 60 FPS for FPV) and directly applies Stride-RoPE at inference to control the rollout rate. **Embed** uses the same multi-rate training data as RAR but retains frame-index-based temporal RoPE and introduces an explicit FPS embedding. Inspired by FPS conditioning in Stable Video Diffusion (Blattmann et al., 2023) and VideoCrafter (Chen et al., 2023), the Embed implementation first normalizes the target FPS by the base FPS, encodes the resulting continuous scalar using Fourier embeddings, and adds the FPS embedding to the diffusion timestep embedding. The multi-rate training sets are same with RAR.

Table 5: Comparison of rate-control methods. Entries report SR, whose ideal value is 1. † marks rates unseen by both multi-rate models (Embed and RAR); Single is trained only at the base FPS of each environment. MAE is the equal-weight mean of $|\mathrm{SR} - 1|$ over the five displayed rates, expressed as a percentage. Bold indicates the SR closest to 1 in each row or the lowest MAE.

(a) Doom

FPS	Single	Embed	RAR
14	0.106	0.921	**0.923**
28†	0.851	0.917	**1.003**
35	1.110	0.936	**0.999**
49†	2.483	1.395	**0.986**
70	3.818	**0.925**	1.082
MAE (%)	109.08	13.92	**3.54**

(b) CS2

FPS	Single	Embed	RAR
16	0.412	0.900	**0.983**
24†	0.720	0.954	**1.009**
32	1.169	**0.974**	0.970
48†	1.678	1.181	**1.024**
64	1.897	**1.009**	1.097
MAE (%)	52.24	7.24	**3.54**

(c) FPV

FPS	Single	Embed	RAR
20	1.151	1.051	**1.031**
30	0.833	0.950	**1.040**
40†	1.644	**1.009**	0.960
50†	2.134	0.844	**1.007**
60	2.268	1.204	**0.917**
MAE (%)	67.28	9.40	**4.02**

Tab. 5 shows that RAR enables more consistent multi-rate generation than Single, highlighting the benefit of multi-rate training over inference-only temporal rescaling. RAR also achieves lower overall SR MAE than Embed in all three environments, with a larger aggregate advantage at unseen rates. Averaged over the nine seen and six unseen environment–rate pairs, Embed and RAR obtain seen-rate MAEs of 7.31% and 5.09%, respectively, but unseen-rate MAEs of 14.50% and 1.62%. This gap indicates that Embed does not interpolate as reliably to unseen FPS values despite its continuous FPS embedding, whereas RAR better preserves cross-rate temporal consistency.

D Extrapolation Beyond the Training Range

Tab. 6 reports the rate fidelity of rollout rates outside the training range. For comparison, the table also lists the interpolation rates and the intermediate training rate of each environment. All SR values at extrapolated rates are computed per case adapting Eq. 14 for extrapolation: the GT normalization denominator is the geometric mean of the in-scope GT speeds for the same case, and the extrapolated rate is not included in this reference. The per-case ratios are formed first and then aggregated by their median. At the low-rate side, the rollout stride exceeds the largest stride observed during training: 7 FPS on Doom, 8 FPS on CS2, and 10 FPS on FPV correspond to strides of 10, 8, and 6, respectively. Their SR values are 0.739, 0.636, and 0.634, all clearly below one, indicating under-scaling of motion relative to the ground-truth cross-rate trend at these rates. At the high-rate side, Doom at 84 FPS and CS2 at 72 FPS yield SR values of 1.191 and 1.273, respectively, indicating overshoot, whereas FPV at 80 FPS yields 0.864 and therefore undershoots. The direction of the out-of-range bias thus depends on the environment and extrapolation regime rather than following a universal pattern. These results lie outside the scope of our claims and are reported only to characterize the boundary of the method.

Table 6: Rate fidelity at out-of-range rollout rates. Out-of-range rates are listed together with in-range interpolation and training rates for comparison. Out-of-range rates lie outside the scope of our claims.

Environment	FPS	Type	$\mathrm{SR} \to 1$
Doom	7	*Extrap.*	0.739
	28	**Interp.**	1.003
	35	Train	0.999
	49	**Interp.**	0.986
	84	*Extrap.*	1.191
CS2	8	*Extrap.*	0.636
	24	**Interp.**	1.009
	32	Train	0.970
	48	**Interp.**	1.024
	72	*Extrap.*	1.273
FPV	10	*Extrap.*	0.634
	20	Train	1.031
	40	**Interp.**	0.960
	50	**Interp.**	1.007
	80	*Extrap.*	0.864

E Dynamic Rollout-Rate Switching

Because absolute motion scales vary across scenes, we apply a single global multiplicative calibration to each rollout, aligning the geometric mean of the measured motion with that of the target stride. This calibration only matches the vertical scale and preserves the relative temporal variation of the curve. The evaluation therefore measures how well the model tracks a dynamic schedule rather than its absolute stride calibration. For the smooth schedule, we first group adjacent intervals into bins and then fit each rollout using a second-order periodic harmonic regression. For the abrupt schedule, we perform piecewise-constant regression over its six constant-stride segments.

To reduce interference from low-quality generations and unreliable motion measurements, we apply a two-stage case filtering procedure. First, for each schedule, we retain the half of the cases with the lowest mean LPIPS between the GT and generated rollouts. We then assess measurement reliability using the tracking nRMSE obtained from the GT videos under the same protocol and retain cases with lower measurement errors. These criteria evaluate generation quality and measurement reliability, respectively, without selecting cases according to the generated rollouts' final tracking scores. The solid curves and shaded regions show the mean regressed response over the retained cases and its 95% confidence interval, respectively.

As shown in Fig. 6, both Doom and CS2 recover the overall variation of the target stride under the smooth and abrupt schedules. On Doom, the smooth and abrupt schedules achieve Pearson correlations of 0.941 and 0.938, with nRMSE values of 0.219 and 0.185, respectively. The corresponding results on CS2 are 0.899 and 0.963 for Pearson correlation and 0.248 and 0.157 for nRMSE. In particular, under the abrupt schedule, both environments correctly recover the alternating increases and decreases across all six segments, showing that RAR responds to discrete rollout-rate switches without interrupting the rollout.

Compared with FPV in the main paper, the smooth responses on Doom and CS2 are flatter and have wider confidence intervals. One possible reason is that observable motion in these game environments primarily arises from intermittent turns, firing events, or short movements, rather than the persistent motion provided by the continuous camera trajectories in FPV. Consequently, local optical flow reflects both the target stride and content-dependent motion, making gradual changes in the smooth schedule harder to isolate. The GT-side diagnostic supports this interpretation: under the same measurement protocol, the GT Pearson correlations for the smooth schedule are only 0.841 on Doom and 0.896 on CS2, indicating that the optical-flow measurement cannot fully recover the target curve even without generation errors. In contrast, the abrupt schedule aggregates motion over

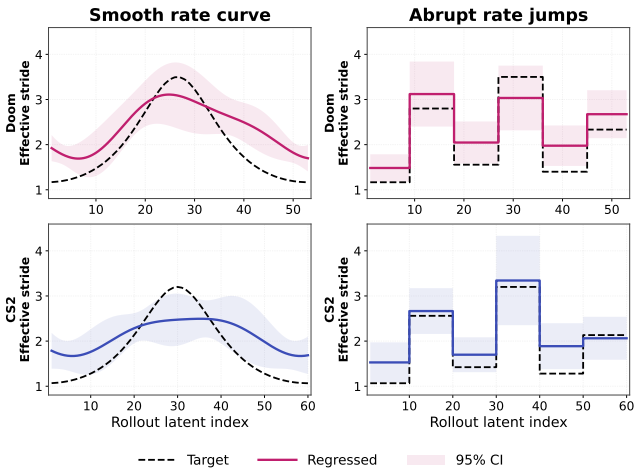

Figure 6: Additional dynamic rollout-rate switching results on Doom and CS2. The evaluation protocol and plot conventions follow Fig. 5.

longer constant-stride segments, increasing the GT Pearson correlations to 0.931 and 0.996, respectively, while also yielding lower nRMSE for the generated rollouts. The residual discrepancies in the figure should therefore be attributed partly to measurement uncertainty caused by intermittent motion, rather than being interpreted entirely as errors in rollout-rate control.

The qualitative examples in Fig. 7 show no noticeable degradation in visual quality during smooth or abrupt rollout-rate changes on Doom, CS2, and FPV.

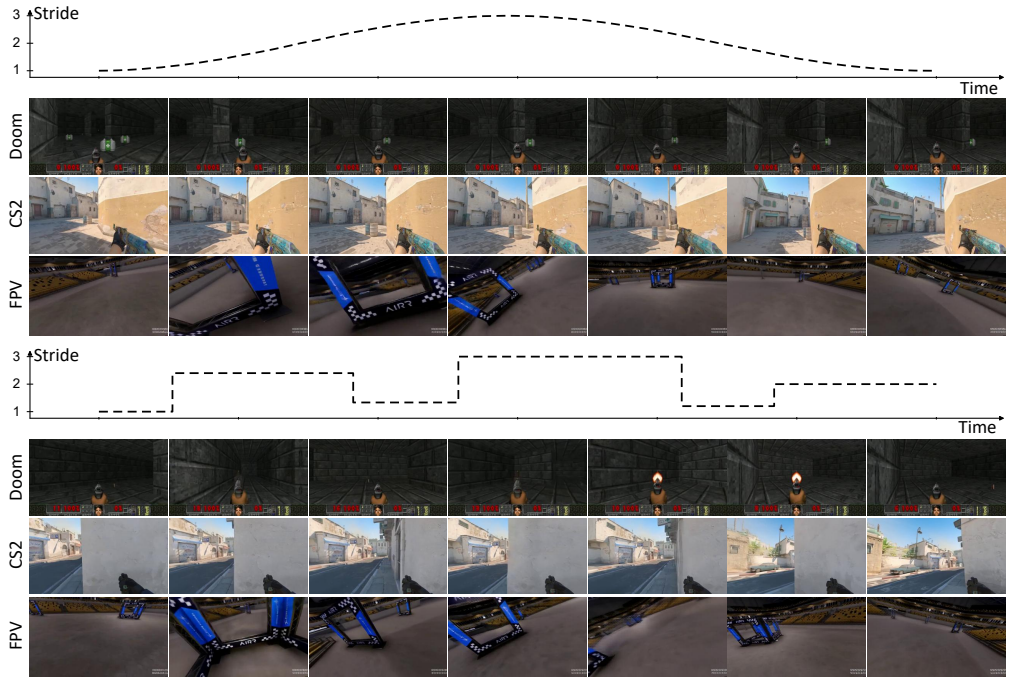

Figure 7: Qualitative dynamic rollout-rate switching on Doom, CS2, and FPV. The upper and lower groups show generated frames under smooth and abrupt schedules, respectively.

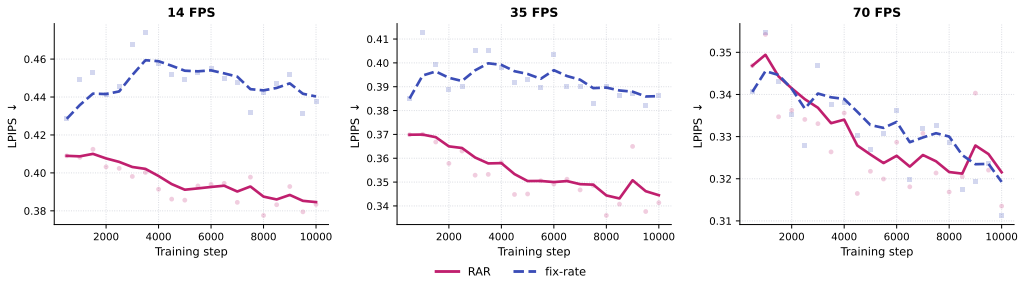

Figure 8: LPIPS under a fixed training-update budget on Doom. RAR is jointly trained at 14, 35, and 70 FPS, whereas the fixed-rate baseline is trained only at 70 FPS. Light markers denote individual checkpoints, and curves show the corresponding trends.

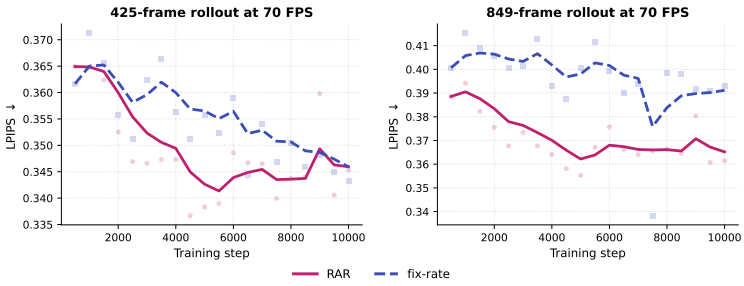

Figure 9: Long-rollout LPIPS under a fixed training-update budget on Doom. Both models are trained with 213-frame windows and evaluated at 70 FPS using 425- and 849-frame rollouts.

F Rate–Horizon Trade-off: Additional Results

Fixed training steps. We first compare RAR with the fixed-rate baseline under the same number of training updates. Both start from the same bidirectional initialization and use the same model architecture and 213-frame training window; RAR is jointly trained with data at 14, 35, and 70 FPS, whereas the fixed-rate baseline uses only 70-FPS data. Fig. 8 plots LPIPS over training at the three evaluation rates. The curves exhibit a clear training trend. At 14 and 35 FPS, the LPIPS of RAR decreases steadily, while the fixed-rate baseline remains at a substantially higher level after its initial fluctuations and shows no corresponding learning trend at the lower rates. At 70 FPS, both curves decrease similarly and gradually converge later in training. This indicates that RAR progressively acquires lower-rate generation capabilities during training, rather than benefiting from fluctuations at a particular checkpoint, while preserving its generation capability at the highest training rate. For reference, after 10k updates, the LPIPS values of RAR at 14 and 35 FPS are 0.3834 and 0.3413, compared with 0.4376 and 0.3863 for the fixed-rate baseline. At 70 FPS, the two models obtain comparable values of 0.3135 and 0.3113, respectively.

We further evaluate both 213-frame-trained models on longer rollouts by jointly denoising the full sequence, without changing their training windows. Fig. 9 compares their LPIPS when generating 425 and 849 frames at 70 FPS. For 425-frame rollouts, RAR improves more rapidly during the first half of training and reaches lower LPIPS earlier; with further training, the two curves gradually converge, reaching 0.3453 and 0.3432 after 10k updates. On the more challenging 849-frame rollouts, RAR achieves lower LPIPS from the early stages of training and maintains a consistent advantage thereafter, reaching 0.3614 after 10k updates compared with 0.3929 for the fixed-rate baseline. The transition from faster convergence at 425 frames to a persistent gap at 849 frames shows that the benefit of multi-rate training becomes more pronounced as the evaluation rollout length increases. These results indicate that the different physical horizons induced by the training rates improve long-horizon generation stability, allowing RAR to extend to rollouts substantially longer than its training window without increasing the training sequence length.

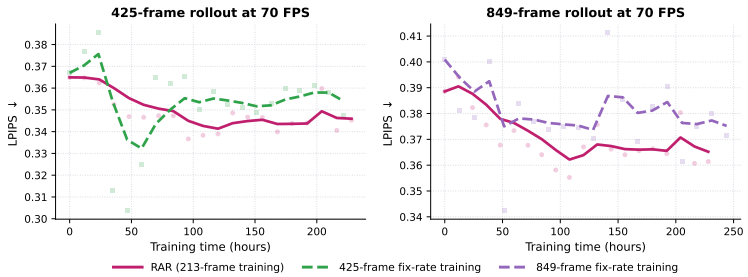

Figure 10: LPIPS under approximately matched wall-clock training time on Doom. RAR is trained with 213-frame multi-rate sequences and compared with fixed-rate models trained directly using 425- or 849-frame sequences at 70 FPS. Each model is evaluated at its indicated rollout length.

Matched training time. Finally, under approximately matched wall-clock training time, we compare RAR trained with a 213-frame window against fixed-rate models trained directly with 425- and 849-frame windows. Each model is evaluated at 70 FPS using the corresponding long-rollout length. As shown in Fig. 10, on the 425-frame evaluation, the LPIPS of RAR decreases steadily with training time and reaches a level comparable to or lower than that of 425-frame fixed-rate training during the middle and later stages, whereas the latter exhibits larger fluctuations without establishing a persistent advantage. On the 849-frame evaluation, RAR reaches lower LPIPS earlier and maintains a clear gap from the 849-frame fixed-rate model over most of the training period. This trend shows that the efficiency advantage of RAR persists throughout training under the same wall-clock budget, rather than appearing only at a particular final checkpoint. Compared with simply increasing the training sequence length, RAR covers multiple physical horizons by jointly training on different rates using shorter sequences and a lower cost per update, yielding comparable or better long-horizon generation quality within the same training time.