FUTUREWORLD: A PLUG-AND-PLAY CAUSAL NEXT-STATE PREDICTOR FOR INTERACTIVE VIDEO WORLD EXPLORATION

Anonymous authors
Paper under double-blind review



Figure 1: Improving long-horizon consistency through explicit next-state prediction. (a) Reusing generated history can carry existing errors into subsequent generations, allowing scene distortions and semantic drift to accumulate across interactions (red dashed boxes). (b) Our plug-in predictor estimates the next latent state from the current generated latent and the next action. This future-state prior guides the autoregressive video DiT through feature modulation, helping preserve scene structure and semantic consistency as interactions proceed (green dashed boxes).

ABSTRACT

Interactive video world models aim to follow user actions while maintaining scene consistency over extended interactions. History-based methods reuse generated latents to preserve scene content, but three limitations of this reuse contribute to ***error recirculation***. **1)** The latent generated at each interaction is retained as history, potentially containing blurred details, geometric distortions, or semantic drift. **2)** Although retrieved historical content matches the current scene, it can still contain errors that are carried into the next chunk. **3)** New outputs then become history for later interactions, potentially reinforcing the same errors through repeated reuse. To limit this cycle, we aim to predict the future scene state from the current generated latent and the next action before generating the next interaction's video chunk. This future-state prior provides guidance for the next chunk, with the aim of reducing the errors carried forward from history. To this end, we propose a plug-and-play Causal Next-Chunk Predictor (C-NCP) with cross-model adaptability and prediction-guided modulation. We integrate C-NCP into four base generators and evaluate the augmented models across three benchmarks for interactive long-video generation and action-conditioned world modeling. Extensive experiments demonstrate reduced error accumulation and improved long-horizon consistency across diverse generation settings.

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

Interactive video world models (Li et al., 2025a; Sun et al., 2025; Team et al., 2026c; Wu et al., 2026a; Wang et al., 2026; Zhu et al., 2026; Zhao et al., 2026b; Team et al., 2026b;a; Chen et al., 2026) generate visual environments that evolve under user action control. Sustained interaction requires each new chunk to follow the next action while preserving the scene's structure and semantics. As interactions continue, local generation defects can affect subsequent chunks, making error propagation a central obstacle to long-horizon consistency.

To maintain scene consistency, history-based approaches (Sun et al., 2025; Wang et al., 2026; Li et al., 2025b; Zhu et al., 2025; Yu et al., 2025a; Zhao et al., 2026c; Yu et al., 2025b; Kim et al., 2026) repeatedly retrieve and reuse generated content. However, this reuse can lead to *error recirculation*. Defects from earlier generations enter new outputs, which are then added to history and reused in later interactions, as illustrated in fig. 1. Three limitations of history reuse contribute to this cycle. **First**, the latent generated at each interaction is retained as history, potentially containing visual defects such as blurred details, geometric distortions, or semantic drift. **Second**, although a retrieved history view matches the current scene, it can still contain errors. Reusing such a view can therefore propagate these errors to the next chunk. **Third**, the new output is then added to history and reused in later interactions, potentially reinforcing the same errors. To limit this cycle, the generator needs guidance for the next chunk that does not rely solely on reusing potentially flawed history.

Based on this analysis, our motivation is to predict the future state from the current generated latent and the next action, providing an explicit future-state prior before generating the video chunk for the next interaction. Supervision from real video transitions teaches the predictor how scenes change under action control. It uses these learned transitions to re-estimate the upcoming state, rather than only selecting content already present in history. The resulting future-state prior guides the next generation, with the aim of reducing errors carried forward from earlier outputs. To turn this prior into effective guidance, we need to meet the following two requirements. **(1) Adaptability.** To make future-state prediction usable across different interactive video generators, the predictor must adapt to their latent representations. Different models use VAE latents with different dimensions and distributions, so the predictor needs a way to accommodate these differences. **(2) Modulation.** The future-state prior must also help guide the generation of the next chunk. Direct concatenation places the predicted latent alongside the generator's main latent, but the generator must still learn how to use this extra input when producing the next chunk. Modulation instead uses the future-state prior to adjust the main latent directly, while keeping its tensor shape unchanged. These considerations lead to our central research question.

> 🔍 *How can we predict a future state from the current generated latent and the next action, and use it to guide subsequent generation and limit error propagation across interactions?*

To address this question, we propose the plug-and-play Causal Next-Chunk Predictor (C-NCP), which estimates the complete next latent chunk from the current interaction's generated latent and the next interaction's action control. We use Bayesian decision theory (Särkkä & Svensson, 2023) to formulate this prediction task as learning a future-state estimate that minimizes expected prediction loss given the current latent and next action. To enable adaptation, VAE-specific input adapters map each model's latents into a shared representation for the causal spatiotemporal predictor, while output heads map its predictions back to the corresponding model's latent space. To enable modulation, we use the future-state prior to adjust the generator's main latent through residual feature modulation based on adaptive instance normalization (Huang & Belongie, 2017). After each interaction, the newly generated latent becomes the input to the next prediction, so the future-state prior is updated as interaction continues. Our main contributions are summarized as follows.

- We introduce a future-state-guided correction paradigm for error recirculation in interactive video generation. Rather than only selecting content from generated history, this approach learns to re-estimate the state expected under the next action. The resulting prior guides each new chunk before it becomes part of history.
- We develop the plug-and-play Causal Next-Chunk Predictor with cross-model adaptability and prediction-guided modulation. VAE-specific adapters let the predictor work with different latent representations, while residual modulation uses its future-state prior to adjust the generator's main latent.

- We integrate our C-NCP into four base generators and evaluate the augmented models across three benchmarks for interactive long-video generation and action-conditioned world modeling. Extensive experiments demonstrate reduced error accumulation and improved long-horizon consistency across diverse generation settings.

2 Related Work

Interactive Video World Models. Interactive video world models (Li et al., 2025a; Sun et al., 2025; Team et al., 2026c; Wu et al., 2026a; Wang et al., 2026; Zhu et al., 2026; Zhao et al., 2026b; Team et al., 2026b;a; Chen et al., 2026) extend visual generation to environments that evolve under action control. Early token-based systems such as iVideoGPT (Wu et al., 2024) model observations, actions, and rewards within an autoregressive sequence. Matrix-Game 2.0 (He et al., 2025) and LingBot-World (Team et al., 2026c) develop real-time streaming generation across diverse environments, while SANA-WM (Zhu et al., 2026) combines hybrid linear attention and long-video refinement for efficient minute-scale generation. Recent work also addresses the mismatch between clean training contexts and generated histories. Matrix-Game 3.0 (Wang et al., 2026) reuses prediction errors to expose the model to imperfect conditioning, and DreamX-World (Team et al., 2026b) combines residual recycling with training on self-generated long-horizon contexts. These approaches improve the generator's ability to continue from its own outputs rather than assuming error-free history. However, remaining defects in generated history can still affect subsequent chunks. Our plug-and-play Causal Next-Chunk Predictor provides an explicit future-state prior before the next interaction's video chunk is generated. It learns from real video transitions, and VAE-specific adapters accommodate different latent representations so that the predictor can be integrated into different base generators.

History Modeling for Video Generation. History modeling preserves scene context through retrieval and compact memory. Spatial-memory world models (Wu et al., 2026b) reconstruct static 3D scenes for rendering along target trajectories, while WorldPlay (Sun et al., 2025) reconstructs relevant historical context. WorldKV (Yi et al., 2026) retrieves viewpoint-matched chunks and compresses redundant tokens; Matrix-Game 3.5 (Qian et al., 2026) uses geometry-aware patch memory; and ReWorld (Chen et al., 2026) combines a bounded KV cache with a pose-indexed landmark bank. Compression and adaptive selection further reduce context costs. FramePack (Zhang et al., 2026) uses frame-dependent compression and anti-drifting sampling, while TinyHistory (Zhang et al., 2025) learns compact history embeddings through frame-query pretraining and content-consistency learning. Infinite-World (Wu et al., 2026a) uses pose-free hierarchical memory, and CaR (Peng et al., 2026) combines context compression with attention-based implicit retrieval. OmniMem (Zhao et al., 2026a) sparsely retrieves KV blocks from historical caches. Future Forcing (Luo et al., 2026) uses future-query proxies to guide KV retention and merging. Retrieval and compression help preserve scene content, but the generated latents they reuse can still contain visual defects. When these defects enter new outputs and those outputs become history, repeated reuse can cause error recirculation. To address this problem, our predictor learns how scenes change under action control from real video transitions and predicts the next latent state from the current generated latent and the next action. The resulting future-state prior guides the generator's main latent through modulation, aiming to reduce the errors passed into subsequent chunks.

3 Method

3.1 Overview

Reusing imperfect generated history can propagate existing defects into subsequent chunks, leading to error recirculation. We therefore learn a future-state prior from real video transitions to guide the next interaction's generation. As shown in fig. 2, the Causal Next-Chunk Predictor combines VAE-specific adapters for cross-model adaptability with a shared spatiotemporal trunk, learns through two-stage Next-Chunk Prediction Training, and uses prediction-guided Modulation to guide a frozen Autoregressive Video DiT during Autoregressive Inference. Prediction and generation follow

$$\widehat{\mathbf{z}}_{t+1}^{(d)} = P_{\phi}^{(d)}(\widetilde{\mathbf{z}}_{t}^{(d)}, \mathbf{u}_{t+1}), \qquad \widetilde{\mathbf{z}}_{t+1}^{(d)} = G_{\theta}^{(d)}\left(\mathcal{H}_t, \mathbf{u}_{t+1}; \widehat{\mathbf{z}}_{t+1}^{(d)}\right), \tag{1}$$

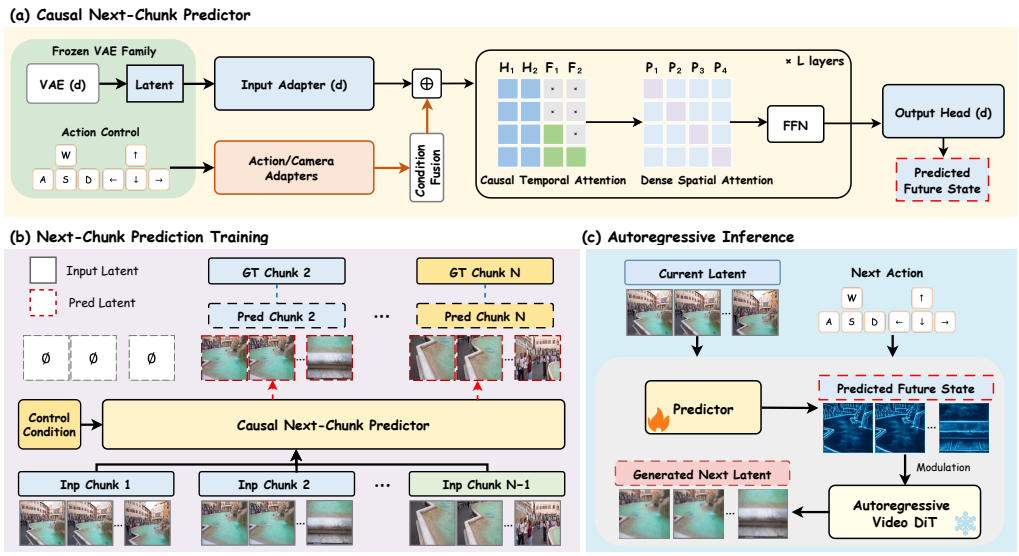

Figure 2: Overview of FutureWorld. Panel a presents the Causal Next-Chunk Predictor, which combines VAE-specific adapters, action conditioning, and a causal spatiotemporal backbone to predict the complete next latent chunk. Panel b illustrates next-chunk prediction training, where the predictor learns to forecast subsequent latent chunks from input chunks and control conditions under supervision from corresponding ground-truth chunks. Panel c shows autoregressive inference, where the predicted future state guides a frozen autoregressive video DiT through modulation to generate the next latent chunk.

where t indexes interaction chunks and d identifies the VAE domain. The latents $\widetilde{\mathbf{z}}_t^{(d)}$, $\widehat{\mathbf{z}}_{t+1}^{(d)}$, and $\widetilde{\mathbf{z}}_{t+1}^{(d)}$ are the current generated state, predicted next state, and completed next output, respectively. Each lies in $\mathbb{R}^{C_d \times T_d \times H_d \times W_d}$, with C_d channels, T_d temporal slices, height H_d, and width W_d. Here $\mathbf{u}_{t+1}$ is the next action, $\mathcal{H}_t$ is the generated history, and $P_\phi^{(d)}$ and $G_\theta^{(d)}$ are the predictor and generator with trainable parameters ϕ and frozen parameters θ. The semicolon denotes prediction-guided modulation. Completed outputs update history and feed subsequent predictions; real-video latents are denoted by $\mathbf{z}_t^{(d)}$.

3.2 Causal Next-Chunk Predictor

Errors in the current generated latent may propagate into subsequent generation. We therefore supervise the predictor with real future states to learn action-induced scene changes. To match sequential forecasting, we condition each future slice only on the current input, the next action, and preceding slices. The probability chain rule gives

$$p_*(\mathbf{Y} \mid C) = \prod_{\tau=1}^{T_d} p_*(\mathbf{Y}_\tau \mid \mathbf{Y}_{<\tau}, C), \tag{2}$$

where p_* is the data distribution, $C = (\mathbf{x}_t, \mathbf{u}_{t+1})$ is the conditioning information, and $\mathbf{x}_t$ is the predictor input, equal to $\widetilde{\mathbf{z}}_t$ at inference. The random tensor $\mathbf{Y}$ is the real next chunk, $\mathbf{Y}_\tau$ is its slice at time τ, and $\mathbf{Y}_{<\tau}$ denotes preceding slices. Rather than estimating a density, we learn a point estimate, motivated by Bayesian decision theory (Särkkä & Svensson, 2023)

$$a_\ell^*(C) \in \operatorname*{argmin}_{\mathbf{a}} \mathbb{E}_{p_*}[\ell(\mathbf{a}, \mathbf{Y}) \mid C], \tag{3}$$

where $\mathbf{a}$ is a candidate future latent, ℓ is its prediction loss, and $a_\ell^*(C)$ is an optimal estimate. The operator argmin selects minimizers and $\mathbb{E}_{p_*}[\cdot \mid C]$ denotes conditional expectation; expectations use p_* throughout. With squared error and finite second moments, the optimum is

$$m(C) = \mathbb{E}[\mathbf{Y} \mid C] \in \operatorname*{argmin}_{\mathbf{a}} \mathbb{E}\left[\|\mathbf{Y} - \mathbf{a}\|_F^2 \mid C\right], \tag{4}$$

where $m(C)$ is the conditional mean and $\|\cdot\|_F$ is the Frobenius norm. Its orthogonal-projection interpretation (Rényi, 2007) yields

$$\mathbb{E}\|\mathbf{Y} - f(C)\|_F^2 = \underbrace{\mathbb{E}\|\mathbf{Y} - m(C)\|_F^2}_{\text{conditional uncertainty}} + \underbrace{\mathbb{E}\|m(C) - f(C)\|_F^2}_{\text{excess prediction risk}}, \tag{5}$$

where $f(C)$ is any square-integrable predictor and expectations average over the joint distribution of C and $\mathbf{Y}$. This separates conditional uncertainty from reducible prediction error, motivating a learned future-state prior. The analysis concerns squared error, not the exact optimum of our normalized mixed loss or a guarantee of rollout error contraction; Appendix A.1 provides the derivation.

To enable cross-model adaptability, we account for differences in VAE latent channel counts and distributions. Input Adapter (d) learns a $1 \times 1 \times 1$ channel mapping to a shared hidden space, preserving the temporal/spatial grid while accommodating these differences. Action/Camera Adapters and Condition Fusion produce

$$\mathbf{H}^{(0)} = [I_d(\mathbf{x}_t)\,;\, \mathbf{Q} + \Pi A_d(\mathbf{u}_{t+1})] + \mathbf{E}, \tag{6}$$

where $\mathbf{H}^{(0)}$ is the trunk input, I_d is Input Adapter (d), A_d encodes action control, and Π aligns control features to the temporal/spatial query grid. The queries $\mathbf{Q}$ encode a learned start token followed by preceding ground-truth slices during training or predicted slices during forecasting. Here $\mathbf{E}$ contains type/domain embeddings and $[\,;\,]$ concatenates temporal positions. Causal Temporal Attention uses

$$M_{ij} = 0 \text{ if } j \in \mathcal{I}_H \text{ or } (i = F_k,\ j = F_r,\ r \le k), \qquad M_{ij} = -\infty \text{ otherwise.} \tag{7}$$

where M_{ij} is the mask entry for query i and key j, $\mathcal{I}_H$ contains current-input positions, and F_k, F_r are future-query positions with $1 \le k, r \le T_d$. Zero permits attention and $-\infty$ blocks it. Shifted queries prevent access to their own or later targets. The trunk combines Causal Temporal Attention, Dense Spatial Attention, and feed-forward networks; Output Head (d) returns

$$\widehat{\mathbf{z}}_{t+1}^{(d)} = O_d\Big(\Big[T_\phi^{(L)}(\mathbf{H}^{(0)}; \mathbf{M})\Big]_F\Big), \tag{8}$$

where $T_\phi^{(L)}$ is the shared trunk of L blocks, $\mathbf{M} = (M_{ij})$, $[\cdot]_F$ selects future queries, and O_d maps features to native VAE channels. The output is a complete latent, not an input residual. Block details are in Appendix A.2.

3.3 Next-Chunk Prediction Training

Since C-NCP is trained from scratch, its early predictions are unreliable, and directly feeding them back across chunks can destabilize optimization. We therefore adopt a two-stage training schedule that first supports early learning with ground-truth-assisted inputs and then adapts the predictor to prediction-only inter-chunk feedback. Stage I adds aligned predicted and ground-truth chunks element-wise, retaining predicted content while providing ground-truth support. Stage II removes the direct ground-truth addition and continues optimizing the same predictor. Both stages retain within-chunk teacher forcing and start from a real initial chunk, with subsequent inter-chunk inputs constructed as follows

$$\begin{aligned} &\mathbf{x}_1^{(s)} = \mathbf{z}_1, \qquad \widehat{\mathbf{z}}_{n+1}^{(s)} = P_\phi^{\mathrm{TF}}(\mathbf{x}_n^{(s)}, \mathbf{u}_{n+1}; \mathrm{shift}(\mathbf{z}_{n+1})), \\ &\mathbf{x}_n^{(s)} = \begin{cases} \widehat{\mathbf{z}}_n^{(1)} + \mathbf{z}_n, & s = 1, \\ \widehat{\mathbf{z}}_n^{(2)}, & s = 2, \end{cases} \qquad n = 2, \dots, N-1, \end{aligned} \tag{9}$$

Here, $N \ge 3$ is the number of training chunks, n indexes chunks, and $s \in \{1, 2\}$ denotes the training stage. The tensors $\mathbf{x}_n^{(s)}$, $\widehat{\mathbf{z}}_n^{(s)}$, and $\mathbf{z}_n$ denote the input, predicted chunk, and real target, respectively, while $\mathbf{u}_{t+1}$ specifies the next action. For $n = 1, \dots, N-1$, the operator P_ϕ^{TF} performs within-chunk teacher-forced prediction, and shift prepends the learned start token and removes the final target slice. Thus, Stage II passes predicted chunks to subsequent prediction steps, while those chunks are still computed using ground-truth prefixes within each chunk. Both stages supervise predictions against real future chunks.

State matching alone does not explicitly supervise temporal changes. Both training stages combine normalized state losses with adjacent-slice difference supervision

$$(\Delta_T \mathbf{e}_n^{(s)})_\tau = (\widehat{\mathbf{z}}_{n,\tau}^{(s)} - \widehat{\mathbf{z}}_{n,\tau-1}^{(s)}) - (\mathbf{z}_{n,\tau} - \mathbf{z}_{n,\tau-1}), \quad \tau = 2, \dots, T_d, \tag{10}$$

where $\mathbf{e}_n^{(s)} = \widehat{\mathbf{z}}_n^{(s)} - \mathbf{z}_n$ is the prediction error at stage $s \in \{1, 2\}$, Δ_T takes temporal first differences, and the subscript τ selects a slice. This matches changes rather than suppressing motion. At each stage, we optimize

$$\mathcal{L}^{(s)}(\phi) = \frac{1}{N-1} \sum_{n=2}^{N} \left[\lambda_1 \left\langle \left| \frac{\mathbf{e}_n^{(s)}}{s_n} \right| \right\rangle + \lambda_2 \left\langle \left(\frac{\mathbf{e}_n^{(s)}}{s_n} \right)^2 \right\rangle + \lambda_{\text{temp}} \left\langle \left| \frac{\Delta_T \mathbf{e}_n^{(s)}}{r_n} \right| \right\rangle \right], \tag{11}$$

where $\mathcal{L}^{(s)}(\phi)$ is the loss at stage s and $\lambda_1, \lambda_2, \lambda_{\text{temp}} \geq 0$ weight absolute, squared, and temporal-difference errors. Brackets $\langle \cdot \rangle$ average element-wise quantities over all tensor entries; s_n and r_n are positive normalization factors computed from target-state and target-difference standard deviations, respectively. They depend only on real targets and are shared across stages; Appendix A.3 gives their definitions. Training updates only the predictor and its adapters.

3.4 Autoregressive Inference

At each interaction, C-NCP receives the completed generator output $\widetilde{\mathbf{z}}_t^{(d)}$ and the next action $\mathbf{u}_{t+1}$. It predicts the next latent chunk $\widehat{\mathbf{z}}_{t+1}^{(d)}$ autoregressively, starting from the learned start token and using preceding predicted slices for subsequent queries. No future ground-truth slices are used. The predicted chunk then provides a future-state prior for the frozen video generator. To incorporate this prior without enlarging the generator's main latent through concatenation, we use shape-preserving channel-wise modulation inspired by FiLM (Perez et al., 2018) and AdaIN (Huang & Belongie, 2017). Rather than directly matching reference statistics, we derive residual scale and shift adjustments from the predicted and current states.

$$\gamma_c = \frac{\sigma_c(\widehat{\mathbf{z}}_{t+1}^{(d)})}{\sigma_c(\widetilde{\mathbf{z}}_t^{(d)})} - 1, \qquad \beta_c = \mu_c(\widehat{\mathbf{z}}_{t+1}^{(d)}) - \mu_c(\widetilde{\mathbf{z}}_t^{(d)}), \tag{12}$$

where $c \in \{1, \ldots, C_d\}$ indexes channels, γ_c is the relative scale adjustment, and β_c is the mean shift. Channel means μ_c and population standard deviations σ_c use temporal/spatial entries, with standard deviations floored at $\epsilon > 0$. During synthesis, $\widetilde{\mathbf{z}}_{t+1}^{(d)}$ denotes the working main latent, updated by

$$\widetilde{\mathbf{z}}_{t+1,c}^{(d)} \leftarrow \widetilde{\mathbf{z}}_{t+1,c}^{(d)} + \alpha \left(\gamma_c \frac{\widetilde{\mathbf{z}}_{t+1,c}^{(d)} - \mu_c(\widetilde{\mathbf{z}}_{t+1}^{(d)})}{\sigma_c(\widetilde{\mathbf{z}}_{t+1}^{(d)})} + \beta_c \right), \tag{13}$$

Here, $\leftarrow$ denotes an in-place update, and $\alpha \geq 0$ controls the modulation strength. The channel-wise coefficients are broadcast over the temporal and spatial dimensions. After synthesis, the completed generator output $\widetilde{\mathbf{z}}_{t+1}^{(d)}$, rather than the predicted prior $\widehat{\mathbf{z}}_{t+1}^{(d)}$, updates the history and becomes the input to the next prediction. Both networks remain frozen during inference.

4 Experiments

4.1 Datasets and Metrics

We train the Causal Next-Chunk Predictor on Sekai (Li et al., 2026) and evaluate on Sekai-val, RealEstate10K-test (Zhou et al., 2018), and the official WBench (Ying et al., 2026) test set. All evaluation videos are 60 seconds long. We report six VBench (Huang et al., 2024) dimensions and their average on Sekai-val and RealEstate10K-test, and five WBench dimensions and their average on WBench. These metrics cover visual quality, temporal and scene consistency, and interactive behavior; metric names are defined in the table captions.

4.2 Implementation Details

Each training sequence contains 1800 frames divided into 60-frame chunks. We train the predictor on eight A100 GPUs following the two-stage schedule in section 3.3, with 100k iterations per stage. The predictor and its adapters are optimized using AdamW with a learning rate of 10^{-4}. The normalized loss uses $\lambda_1 = 1$, $\lambda_2 = 1$, and $\lambda_{\text{temp}} = 0.25$. The VAEs and base generators remain frozen. Architecture and additional optimization settings are provided in Appendices A.2 and A.3. We use four base generators: Matrix-Game 3.0 (Wang et al., 2026), HY-World 1.5 (Tencent Hunyuan, 2025), LingBot-World-Fast (Team et al., 2026c), and SANA-WM (Zhu et al., 2026).

Table 1: Quantitative comparison on RealEstate10K-test using VBench. All videos are 60 seconds long. Higher is better for every metric. Bold rows denote base generators augmented with our predictor. T.F.: Temporal Flickering; M.S.: Motion Smoothness; S.C.: Subject Consistency; B.C.: Background Consistency; A.Q.: Aesthetic Quality; I.Q.: Imaging Quality; Avg.: Average.

Method	T.F. ↑	M.S. ↑	S.C. ↑	B.C. ↑	A.Q. ↑	I.Q. ↑	Avg. ↑
YUME 1.5	0.9127	0.9151	0.6928	0.9036	0.4128	0.7132	0.7584
FantasyWorld	0.9149	0.9128	0.6975	0.9020	0.4149	0.7104	0.7588
Infinite-World	0.9112	0.9170	0.6902	0.9067	0.4116	0.7160	0.7588
SANA-WM	0.9246	0.9764	0.7994	0.9034	0.5050	0.7401	0.8081
SANA-WM + Ours	**0.9366**	**0.9867**	**0.8437**	**0.9202**	**0.5192**	**0.7532**	**0.8266**
Matrix-game-3.0	0.9241	0.9748	0.7946	0.9042	0.5025	0.7740	0.8124
Matrix-game-3.0 + Ours	**0.9361**	**0.9884**	**0.8387**	**0.9210**	**0.5167**	**0.7877**	**0.8314**
HY-World 1.5	0.9238	0.9762	0.7965	0.9058	0.5040	0.7455	0.8086
HY-World 1.5 + Ours	**0.9358**	**0.9898**	**0.8407**	**0.9226**	**0.5182**	**0.7587**	**0.8276**
LingBot-World-Fast	0.9259	0.9783	0.8070	0.9002	0.5084	0.7009	0.8034
LingBot-World-Fast + Ours	**0.9380**	**0.9819**	**0.8518**	**0.9169**	**0.5227**	**0.7133**	**0.8208**

Table 2: Component ablation of the predictor with Matrix-game-3.0 as the base generator on RealEstate10K-test using VBench. All videos are 60 seconds long. Higher is better for every metric. The full method is highlighted in bold. Causal: causal prediction; Cond.: control conditioning; Temp.: temporal loss; Concat: direct concatenation; Mod.: modulation.

Variant	Causal	Cond.	Temp.	Concat	Mod.	T.F. ↑	M.S. ↑	S.C. ↑	B.C. ↑	A.Q. ↑	I.Q. ↑	Avg. ↑
Baseline	×	×	×	×	×	0.9241	0.9748	0.7946	0.9042	0.5025	0.7740	0.8124
Bidirectional Predictor	×	✓	✓	×	✓	0.9307	0.9823	0.8189	0.9134	0.5103	0.7815	0.8229
w/o Condition	✓	×	✓	×	✓	0.9295	0.9789	0.8144	0.9113	0.5103	0.7809	0.8209
w/o Temporal Loss	✓	✓	×	×	✓	0.9283	0.9789	0.8312	0.9180	0.5139	0.7850	0.8259
Direct Concat	✓	✓	✓	✓	×	0.9332	0.9850	0.8300	0.9180	0.5110	0.7848	0.8270
Full Ours	✓	✓	✓	×	✓	**0.9361**	**0.9884**	**0.8387**	**0.9210**	**0.5167**	**0.7877**	**0.8314**

4.3 Quantitative Comparison

We compare all four base generators with their predictor-augmented counterparts and include other interactive video world models as external references. Sekai-val and WBench results are provided in Appendices B and C. On RealEstate10K-test as shown in table 1, every predictor-augmented generator improved across all reported metrics. Subject and background consistency increased alongside temporal and visual-quality scores, with Matrix-Game 3.0 + Ours achieving the highest average. The improvements across all four bases indicate that the benefit is not confined to a single generator. Since the predictor was trained on Sekai, these results also support cross-dataset transfer of the learned transition prior. The gains are consistent with explicit future-state prediction providing a useful reference for maintaining scene continuity under action-driven viewpoint changes.

4.4 Qualitative Comparison

Figure 3 compares four base models with and without our predictor across the sampled 60-second rollouts. For Matrix-Game 3.0, our predictor better preserved the prominent painting and gallery layout, which changed in the baseline. For HY-World 1.5, it maintained the red car and garage structure, while the baseline drifted toward a different vehicle-filled setting. For LingBot-World-Fast, it retained vehicle appearance and road layout, whereas later baseline frames showed vehicle distortion and disrupted road content. For SANA-WM, it preserved clearer elephant head and ear contours than the baseline in later frames. These examples illustrate improved subject and scene consistency during continued interaction, in recognizable object details and the surrounding scene layout. Each forecast uses the current generated latent and the next action, so the reference follows the ongoing interaction. The reduced drift is consistent with this supervised future-state reference guiding each new chunk through modulation, helping refine generation beyond the reuse of potentially imperfect history. User-study results are provided in Appendix F.

4.5 Ablation Studies

Table 2 compares the full model against five variants: the baseline without a predictor, bidirectional prediction, removal of action conditioning, removal of temporal loss, and direct concatenation in-

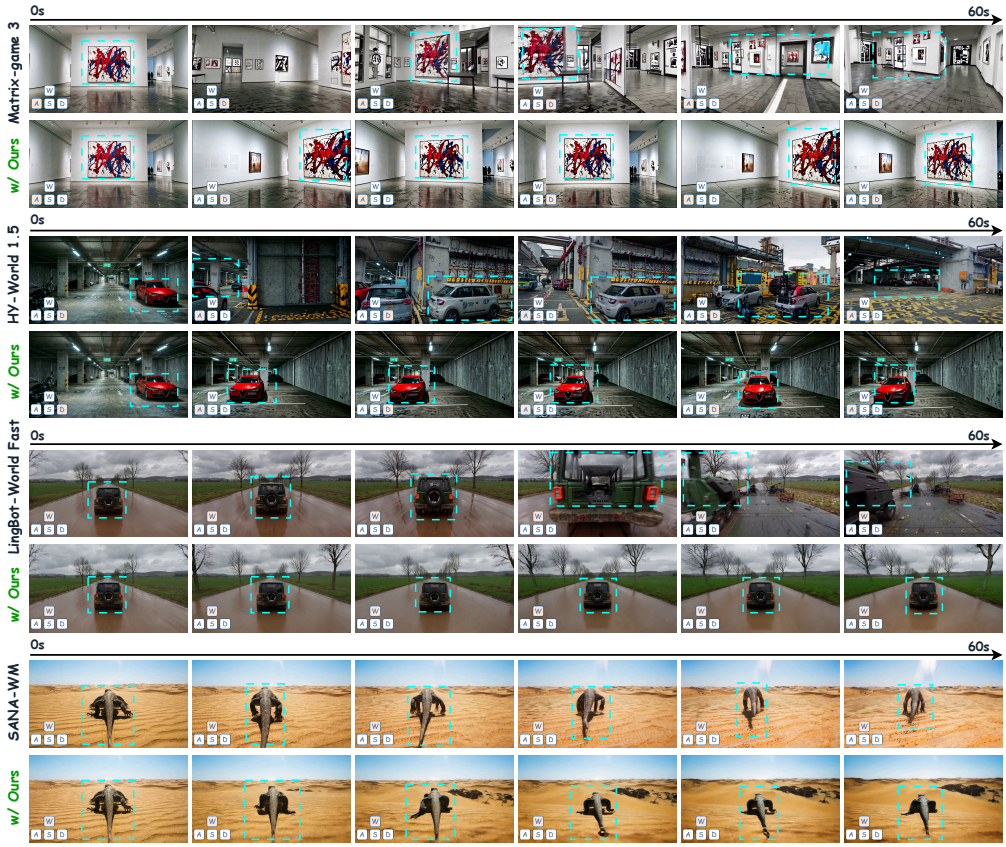

Figure 3: Qualitative comparisons over 60-second interactive rollouts with four base generators. Each pair shows the original generator above and its predictor-augmented counterpart below, with six frames ordered from left to right. Action controls appear in the lower-left corners, and cyan dashed boxes highlight subjects and scene regions for comparison. The examples show improved preservation of subject appearance and scene layout with our predictor.

stead of modulation. The full model achieved the highest score on every metric, while all predictor-equipped variants exceeded the baseline, supporting the value of an explicit future-state reference.

Effectiveness of Causal Prediction and Action Conditioning. Replacing bidirectional prediction with causal prediction improved temporal stability, subject consistency, and visual quality. The causal mask restricts future queries to available context and preceding slices, aligning prediction with the temporal direction of generation. Removing action conditioning yielded the lowest average and subject-consistency scores among predictor-equipped variants. Action control specifies which transition should follow the current latent, helping the predictor anticipate the intended scene change rather than infer it from appearance alone.

Effectiveness of Temporal Supervision and Feature Fusion. Adding temporal supervision improved flickering, motion smoothness, and scene consistency. Supervising differences between adjacent latent slices encourages coherent transitions rather than accurate individual states alone. Direct concatenation also improved over the baseline, but modulation outperformed it on every metric. Through channel-wise scale and shift, modulation lets the future-state prior refine the generator's existing main latent without replacing it. This comparison supports modulation as an effective way to incorporate the predicted reference during synthesis. Training-schedule ablations are reported in Appendix D.

Effectiveness of Prediction-Guided Modulation. The upper panel of fig. 4 shows recognizable vehicle contours and street structure in the predicted state, providing a visual reference for the subsequent modulation. Without modulation, the main latent exhibits fragmented vehicle boundaries and indistinct building details; modulation produces more continuous vehicle outlines, facades, and

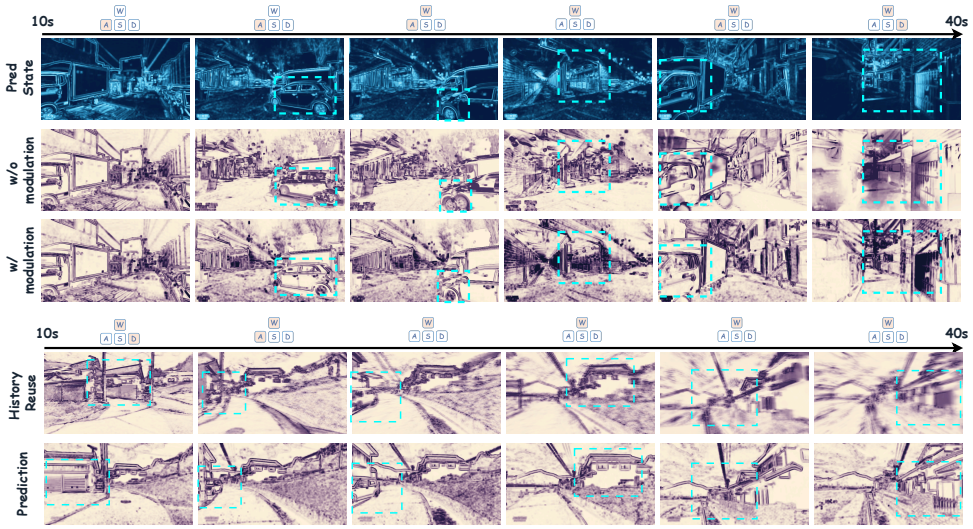

Figure 4: The upper panel compares the generator's main latent before and after modulation, with the predicted future state shown in blue. The lower panel compares main latents obtained with historical retrieval and predictor-guided modulation.

street boundaries. These observations are consistent with the scale-and-shift update in eq. (13) using the future-state reference to refine the generator's existing scene structure during synthesis.

History Reuse vs. Prediction. In the lower panel of fig. 4, main latents obtained with historical retrieval exhibit scene drift and blurring, particularly in later snapshots as interaction continues. Predictor-guided modulation preserves more stable scene structure and clearer building and road contours. A forecast learned from real transitions offers an additional action-conditioned reference when the reused history contains defects. These observations illustrate its benefit for refining generation on top of history reuse, where existing defects can otherwise persist in subsequent outputs.

4.6 Inference-Time Analysis

Table 3 quantifies the additional inference time introduced by the predictor. Across four interactions, the total generation time increased from 26.707 s to 29.174 s, while the average time per interaction increased from 6.677 s to 7.294 s. This corresponds to an additional 0.617 s per interaction, or 9.24% relative to the baseline. The predictor therefore provides explicit future-state guidance with a modest increase in generation time.

Table 3: Inference-time comparison of Matrix-game-3.0 with and without the predictor when generating four 720P video chunks on an A800 GPU. Avg. denotes the time per interaction.

Metric	Baseline	***w/* Predictor**
Total (s)	26.707	29.174
Avg. (s/interaction)	6.677	7.294
Extra (s/interaction)	–	0.617
Overhead	–	9.24%

5 Conclusion

We presented the Causal Next-Chunk Predictor, a plug-and-play module that guides interactive video generation through explicit future-state prediction. Trained through direct supervision from real video transitions, it estimates the next latent chunk from the current interaction's generated latent and the next action control. Feature modulation incorporates this learned reference into the generator's main latent to refine each new chunk before it becomes part of the history. Experiments with four base generators on Sekai-val, RealEstate10K-test, and WBench show consistent improvements in visual quality and long-horizon consistency over the corresponding baselines. These findings support future-state prediction as an effective guide for more coherent generation in existing world models during sustained interaction.

AI USE STATEMENT

In this work, we used generative AI tools to improve the clarity, grammar, and presentation of the manuscript. All AI-assisted content was carefully reviewed and verified by the authors. All scientific ideas, methodological designs, experimental results, and conclusions were developed and validated by the authors. We take full responsibility for the final content of this work.

REFERENCES

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

Xianglong He, Chunli Peng, Zexiang Liu, Boyang Wang, Yifan Zhang, Qi Cui, Fei Kang, Biao Jiang, Mengyin An, Yangyang Ren, et al. Matrix-game 2.0: An open-source real-time and streaming interactive world model. *arXiv preprint arXiv:2508.13009*, 2025.

Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *2017 IEEE international conference on computer vision (ICCV)*, pp. 1510–1519. IEEE, 2017.

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. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21807–21818. IEEE, 2024.

Youngrae Kim, Qixin Hu, C-C Jay Kuo, and Peter Beerel. Memrope: Training-free infinite video generation via evolving memory tokens. In *European Conference on Computer Vision*, pp. 244–261. Springer, 2026.

Guangyuan Li, Bo Li, Jinwei Chen, Xiaobin Hu, Lei Zhao, and Peng-Tao Jiang. Magicworld: Towards long-horizon stability for interactive video world exploration. *arXiv preprint arXiv:2511.18886*, 2025a.

Jiaqi Li, Junshu Tang, Zhiyong Xu, Longhuang Wu, Yuan Zhou, Shuai Shao, Tianbao Yu, Zhiguo Cao, and Qinglin Lu. Hunyuan-gamecraft: High-dynamic interactive game video generation with hybrid history condition. *arXiv preprint arXiv:2506.17201*, 2(3):6, 2025b.

Zhen Li, Chuanhao Li, Xiaofeng Mao, Shaoheng Lin, Ming Li, Shitian Zhao, Zhaopan Xu, Xinyue Li, Yukang Feng, Jianwen Sun, et al. Sekai: A video dataset towards world exploration. *Advances in Neural Information Processing Systems*, 38, 2026.

Jiayi Luo, Qiyan Liu, Tengyang Wang, JunHao Liu, Jiayu Chen, Cong Wang, Hanxin Zhu, Chen Gao, Xiaobin Hu, Qingyun Sun, et al. Future forcing: Future-aware training-free kv cache policy for autoregressive video generation. *arXiv preprint arXiv:2605.30083*, 2026.

Zhan Peng, Jie Ma, Huiqiang Sun, Chong Gao, Zhijie Xue, Zhiyu Pan, Zhiguo Cao, Jun Liang, and Jing Li. Compression and retrieval: Implicit memory retrieval for video world models. *arXiv preprint arXiv:2606.23105*, 2026.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, 2018.

Runjia Qian, Zile Wang, Jihai Zhang, Kai Zou, Wei Yu, Jiaxing Li, Zexiang Liu, Yaokun Li, Fei Kang, Kaichen Huang, et al. Matrix-game 3.5: Enhancing real-time streaming interactive world models with patch memory. *arXiv preprint arXiv:2608.29910*, 2026.

Alfréd Rényi. *Probability theory*. Courier Corporation, 2007.

Simo Särkkä and Lennart Svensson. *Bayesian filtering and smoothing*, volume 17. Cambridge university press, 2023.

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.

AlayaWorld Team, Kaipeng Zhang, Chuanhao Li, Yifan Zhan, Yongtao Ge, Yuanyang Yin, Jiaming Tan, Kang He, Liaoyuan Fan, Ruicong Liu, et al. Alayaworld: Long-horizon and playable video world generation. *arXiv preprint arXiv:2607.06291*, 2026a.

DreamX Team, 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*, 2026b.

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

Tencent Hunyuan. HY-World 1.5: A systematic framework for interactive world modeling with real-time latency and geometric consistency. Technical report, 2025. URL https://3d-models.hunyuan.tencent.com/world/world1_5/HYWorld_1.5_Tech_Report.pdf.

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.

Jialong Wu, Shaofeng Yin, Ningya Feng, Xu He, Dong Li, Jianye Hao, and Mingsheng Long. ivideogpt: Interactive videogpts are scalable world models. *Advances in Neural Information Processing Systems*, 37:68082–68119, 2024.

Ruiqi Wu, Xuanhua He, Meng Cheng, Tianyu Yang, Yong Zhang, Zhuoliang Kang, Xunliang Cai, Xiaoming Wei, Chunle Guo, Chongyi Li, and Ming-Ming Cheng. Infinite-World: Scaling interactive world models to 1000-frame horizons via pose-free hierarchical memory. In *International Conference on Machine Learning*, 2026a. URL https://arxiv.org/abs/2602.02393.

Tong Wu, Shuai Yang, Ryan Po, Yinghao Xu, Ziwei Liu, Dahua Lin, and Gordon Wetzstein. Video world models with long-term spatial memory. *Advances in Neural Information Processing Systems*, 38:49371–49393, 2026b.

Jung Yi, Minjae Kim, Paul Hyunbin Cho, Wooseok Jang, Sangdoo Yun, and Seungryong Kim. Worldkv: Efficient world memory with world retrieval and compression. *arXiv preprint arXiv:2605.22718*, 2026.

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.

Jiwen Yu, Jianhong Bai, Yiran Qin, Quande Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Xihui Liu. Context as memory: Scene-consistent interactive long video generation with memory retrieval. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pp. 1–11, 2025a.

Yifei Yu, Xiaoshan Wu, Xinting Hu, Tao Hu, Yangtian Sun, Xiaoyang Lyu, Bo Wang, Lin Ma, Yuewen Ma, Zhongrui Wang, et al. Videossm: Autoregressive long video generation with hybrid state-space memory. *arXiv preprint arXiv:2512.04519*, 2025b.

Lvmin Zhang, Shengqu Cai, Muyang Li, Chong Zeng, Beijia Lu, Anyi Rao, Song Han, Gordon Wetzstein, and Maneesh Agrawala. Tinyhistory: Lightweight video history embeddings via two-stage context learning. *arXiv preprint arXiv:2512.23851*, 2025.

Lvmin Zhang, Shengqu Cai, Muyang Li, Gordon Wetzstein, and Maneesh Agrawala. Frame context packing and drift prevention in next-frame-prediction video diffusion models. *Advances in Neural Information Processing Systems*, 38:30546–30566, 2026.

Lin Zhao, Yushu Wu, Yifan Gong, Yanzhi Wang, and Pu Zhao. Omnimem: Scalable and adaptive memory retrieval for long video generation. *arXiv preprint arXiv:2605.30519*, 2026a.

Min Zhao, Hongzhou Zhu, Bokai Yan, Zihan Zhou, Yimin Chen, Wenqiang Sun, Kaiwen Zheng, Guande He, Xiao Yang, Chongxuan Li, et al. minwm: A full-stack open-source framework for real-time interactive video world models. *arXiv preprint arXiv:2605.30263*, 2026b.

Zengqun Zhao, Yanzuo Lu, Ziquan Liu, Jifei Song, Jiankang Deng, and Ioannis Patras. Relax forcing: Relaxed kv-memory for consistent long video generation. *arXiv preprint arXiv:2603.21366*, 2026c.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 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.

Tianrui Zhu, Shiyi Zhang, Zhirui Sun, Jingqi Tian, and Yansong Tang. Memorize-and-generate: Towards long-term consistency in real-time video generation. *arXiv preprint arXiv:2512.18741*, 2025.

SUMMARY OF THE APPENDIX

This supplementary material provides additional quantitative results, analyses, and method details for *FutureWorld: A Plug-and-Play Causal Next-State Predictor for Interactive Video World Exploration*. The appendix is organized as follows.

- §A gives the decision-theoretic derivation, predictor architecture, and normalized training objective.
- §B reports VBench results on Sekai-val and analyzes the gains across four base generators.
- §C reports results on the official WBench test set and analyzes interactive generation quality and consistency.
- §D compares one-stage and two-stage training strategies for the predictor.
- §E presents additional qualitative comparisons in desert and coastal scenes.
- §F presents a user study evaluating perceptual preferences across four base generators.

A ADDITIONAL METHOD DETAILS

A.1 CONDITIONAL-MEAN INTERPRETATION

The squared-error interpretation in eqs. (4) and (5) separates uncertainty in the future state from error in its estimate. For finite-second-moment $\mathbf{Y}$ and a square-integrable predictor $f(C)$, write $\mathbf{Y} - f(C) = (\mathbf{Y} - m(C)) + (m(C) - f(C))$. The cross term vanishes by conditional expectation

$$\begin{aligned}&\mathbb{E}\langle \mathbf{Y} - m(C), m(C) - f(C)\rangle_F \\ &\quad = \mathbb{E}\langle \mathbb{E}[\mathbf{Y} - m(C) \mid C], m(C) - f(C)\rangle_F = 0,\end{aligned}$$

where $\langle \cdot, \cdot \rangle_F$ is the Frobenius inner product over latent entries and $m(C) = \mathbb{E}[\mathbf{Y} \mid C]$. Expanding the squared norm then gives eq. (5). The first term measures uncertainty remaining under the available context; the second measures deviation from its conditional mean. This motivates learning a future-state reference from real transitions, but does not establish error contraction on generated inputs. The normalized mixed objective in eq. (11), which also includes absolute-error and temporal terms, need not share the conditional-mean optimum. Finite-capacity training and differences between training and generated contexts further limit this idealized interpretation.

A.2 PREDICTOR ARCHITECTURE

The predictor supports three VAE latent domains with 16, 48, and 128 channels through domain-specific $1 \times 1 \times 1$ input adapters and output heads, together with action-control adapters. The VAE-specific input maps in eq. (6) are trained jointly with the predictor to accommodate channel-count, scale, and channel-statistic differences before the shared trunk. They preserve each domain's temporal/spatial grid rather than requiring identical grids across VAEs. The future-query stream starts with a learned token and encodes preceding ground-truth slices during training or preceding predictions during forecasting. Together with the mask in eq. (7), this shift prevents access to the current target or later targets.

Each block applies Causal Temporal Attention and a feed-forward network (FFN), followed by Dense Spatial Attention and an FFN over each slice's full spatial grid. These sublayers use pre-normalization, residual connections, and positional embeddings. The trunk has four blocks, hidden width 768, 12 attention heads, an FFN expansion ratio of four, and no dropout. Domain-specific output heads map the future-query features in eq. (8) back to native VAE channels. The model predicts the full future latent directly; additive feedback in eq. (9) modifies inputs, not outputs.

A.3 NORMALIZED TRAINING OBJECTIVE

Both training stages use the normalized objective in eq. (11). The target-dependent factors are

$$s_n = \max\big(\mathrm{Std}(\mathbf{z}_n), 10^{-3}\big), \qquad r_n = \max\big(\mathrm{Std}(\Delta_T \mathbf{z}_n), 10^{-3}\big),$$

where s_n normalizes state errors, r_n normalizes temporal-difference errors, and Std computes the population standard deviation over all entries of the corresponding target tensor. The floor prevents division by very small values. Absolute values, divisions, and squares in eq. (11) act element-wise before averaging. When $T_d = 1$, no adjacent temporal slices exist and the temporal term is omitted.

Stage I adds aligned predicted and ground-truth chunks element-wise to support early optimization while retaining predicted content in the input. The sum is not averaged, and the supervision target remains the complete real future chunk. Stage II continues optimizing the same predictor after removing the direct ground-truth addition, which can change both the content and the scale of the input. Both stages retain within-chunk teacher forcing and supervision from real future chunks. The predicted chunks used as inter-chunk feedback during training are therefore computed using ground-truth prefixes. At inference, C-NCP instead uses predicted prefixes within each chunk, and the completed video-generator output becomes the input for the next interaction. The two-stage schedule thus adapts the predictor to prediction-only inter-chunk feedback without making the training and inference distributions identical.

We optimize the predictor and its adapters with AdamW using a learning rate of 10^{-4}, weight decay of 10^{-4}, bfloat16 precision, gradient-norm clipping at 1.0, and a batch size of one per accelerator. The normalized absolute-error, squared-error, and temporal-difference loss weights are $\lambda_1 = 1$, $\lambda_2 = 1$, and $\lambda_{\mathrm{temp}} = 0.25$, respectively. Each stage runs for 100k iterations, with the VAEs and base video generators frozen throughout.

B Additional Results on Sekai-val

As shown in table A1, adding our predictor improved all six VBench dimensions and the average score for every base generator. Higher subject and background consistency accompanied improvements in temporal stability, motion smoothness, and visual quality, with LingBot-World-Fast + Ours achieving the highest average among the compared methods. These results indicate that the benefit of future-state guidance extends across different generators. A plausible explanation is that the predictor estimates how the current scene should evolve under the next action, providing a supervised visual reference that helps maintain scene content while guiding the upcoming transition.

Table A1: Quantitative comparison on Sekai-val using VBench. All videos are 60 seconds long. Higher is better for every metric. Bold rows denote base generators augmented with our predictor. T.F.: Temporal Flickering; M.S.: Motion Smoothness; S.C.: Subject Consistency; B.C.: Background Consistency; A.Q.: Aesthetic Quality; I.Q.: Imaging Quality; Avg.: Average.

Method	T.F. ↑	M.S. ↑	S.C. ↑	B.C. ↑	A.Q. ↑	I.Q. ↑	Avg. ↑
YUME 1.5	0.9002	0.9321	0.7115	0.8921	0.4270	0.6877	0.7584
FantasyWorld	0.9024	0.9298	0.7162	0.8905	0.4291	0.6849	0.7588
Infinite-World	0.8987	0.9340	0.7089	0.8952	0.4258	0.6905	0.7588
SANA-WM	0.9413	0.9713	0.7364	0.9119	0.4522	0.7237	0.7895
SANA-WM + Ours	**0.9534**	**0.9850**	**0.7804**	**0.9288**	**0.4743**	**0.7371**	**0.8098**
Matrix-game-3.0	0.9360	0.9708	0.7600	0.9104	0.4535	0.7555	0.7977
Matrix-game-3.0 + Ours	**0.9482**	**0.9843**	**0.8022**	**0.9273**	**0.4663**	**0.7689**	**0.8162**
HY-World 1.5	0.9319	0.9718	0.7209	0.9148	0.4303	0.7014	0.7785
HY-World 1.5 + Ours	**0.9440**	**0.9853**	**0.7609**	**0.9318**	**0.4424**	**0.7138**	**0.7964**
LingBot-World-Fast	0.9560	0.9713	0.7284	0.9104	0.4729	0.7142	0.7922
LingBot-World-Fast + Ours	**0.9679**	**0.9854**	**0.7781**	**0.9273**	**0.5141**	**0.7286**	**0.8169**

C Additional Results on WBench

On the official WBench test set, our predictor improved Quality, Setting, Interaction, Consistency, and Physical scores for all four generators, as shown in table A2. Consistency showed the largest gain within each pair, while HY-World 1.5 + Ours achieved the highest average among the compared methods. The joint improvements in interaction and consistency suggest that future-state guidance helps preserve scene content while responding to action controls. This trend is consistent with

repeatedly predicting the next state from the current generated latent and the upcoming action, then using the forecast to refine generation before its output becomes the context for another interaction.

Table A2: Quantitative comparison on the official WBench test set using 60-second videos. Higher is better for every metric. Bold rows denote base generators augmented with our predictor. Qual.: Quality; Set.: Setting; Inter.: Interaction; Cons.: Consistency; Phys.: Physical; Avg.: Average.

Method	Qual. ↑	Set. ↑	Inter. ↑	Cons. ↑	Phys. ↑	Avg. ↑
YUME 1.5	77.6	72.4	72.0	80.1	65.2	73.5
FantasyWorld	72.4	71.3	72.1	86.4	66.8	73.8
Infinite-World	77.0	69.3	75.9	80.0	62.1	72.9
DreamX-World	77.5	80.8	78.4	74.9	63.3	75.0
SANA-WM	79.3	76.1	82.1	80.7	61.9	76.0
SANA-WM + Ours	**80.3**	**77.1**	**83.7**	**85.2**	**63.1**	**77.9**
Matrix-game-3.0	75.5	63.6	83.5	74.5	59.3	71.3
Matrix-game-3.0 + Ours	**76.5**	**64.5**	**85.1**	**78.6**	**60.4**	**73.0**
HY-World 1.5	78.1	72.2	87.5	86.9	66.3	78.2
HY-World 1.5 + Ours	**79.1**	**73.2**	**89.1**	**91.7**	**67.5**	**80.1**
LingBot-World-Fast	79.4	77.9	79.4	84.9	65.7	77.5
LingBot-World-Fast + Ours	**80.4**	**79.0**	**80.9**	**89.6**	**66.9**	**79.4**

D Training Strategy Ablation

Table A3: Training-strategy ablation with Matrix-game-3.0 on RealEstate10K-test using VBench. All videos are 60 seconds long. Stage inputs specify inter-chunk feedback; Pred+GT denotes element-wise addition. Higher is better for every metric, and the full method is highlighted in bold. T.F.: Temporal Flickering; M.S.: Motion Smoothness; S.C.: Subject Consistency; B.C.: Background Consistency; A.Q.: Aesthetic Quality; I.Q.: Imaging Quality; Avg.: Average.

Variant	Stage I Input	Stage II Input	T.F. ↑	M.S. ↑	S.C. ↑	B.C. ↑	A.Q. ↑	I.Q. ↑	Avg. ↑
One-stage Pred	Pred	–	0.9290	0.9785	0.8110	0.9110	0.5050	0.7831	0.8196
One-stage GT	GT	–	0.9315	0.9825	0.8205	0.9145	0.5090	0.7840	0.8237
One-stage Pred+GT	Pred+GT	–	0.9330	0.9842	0.8268	0.9168	0.5118	0.7846	0.8262
Two-stage GT→Pred	GT	Pred	0.9345	0.9860	0.8320	0.9185	0.5135	0.7860	0.8284
Full Ours	Pred+GT	Pred	**0.9361**	**0.9884**	**0.8387**	**0.9210**	**0.5167**	**0.7877**	**0.8314**

Table A3 compares five inter-chunk input schedules; within-chunk teacher forcing is retained. One-stage Pred achieved the lowest scores, consistent with errors in early predictions making subsequent inputs less reliable. One-stage GT performed better, but dependence on real latent inputs leaves a mismatch with prediction-based feedback. One-stage Pred+GT improved further by exposing the predictor to its own outputs alongside ground-truth information, although it never adapts to prediction-only inputs. Two-stage GT→Pred outperformed all one-stage variants, supporting a transition from ground-truth-supported learning to prediction-dependent training. Full Ours achieved the highest score on every metric. Its first stage combines predicted and real chunks, providing ground-truth support while exposing the model to its own prediction errors; the second removes inter-chunk ground-truth addition to adapt the predictor to its own feedback. These results support combining a ground-truth-supported warm-up with prediction-only adaptation for more reliable next-state prediction.

E Additional Qualitative Results

Figure 5 provides two additional examples of sustained interactive generation. In the desert scene, the baseline gradually changed the camel's body shape and appearance and introduced irregular sky textures, whereas the predictor-augmented sequence retained a more stable camel silhouette and scene appearance. In the coastal scene, later baseline frames remained around a similar section of the trail, with less distinct foreground vegetation and path details. With our predictor, the sequence progressed along the rocky path toward an open viewpoint while preserving recognizable

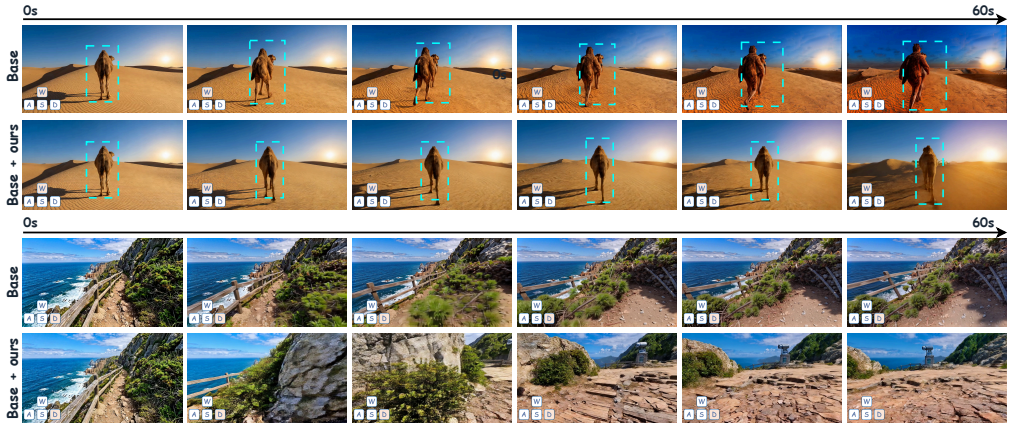

Figure 5: Additional qualitative comparisons over 60-second interactive rollouts in desert and coastal scenes. Each pair shows the base generator above and its predictor-augmented counterpart below, with six snapshots ordered from left to right. Action controls appear in the lower-left corners, and cyan dashed boxes highlight the camel in the desert example.

rock surfaces and path boundaries. These examples illustrate the value of future-state guidance for maintaining subject appearance while allowing the scene to evolve during interaction. By estimating the upcoming state from the current generated latent and the next action, the predictor provides a future-state prior for each new chunk. Modulation uses this prior to refine generation, helping limit the continuation of visual defects without simply preserving the current view.

F User Study

We conducted a paired user study with 100 participants, each evaluating all 20 video pairs, with five pairs per base generator. The study covered five criteria. *Scene consistency* assessed whether object appearance, position, and scene layout remained consistent through turns, return paths, and revisits. *Action adherence* assessed whether the generated motion followed the displayed controls, including forward and backward movement, lateral movement, and turning. *Temporal consistency* assessed flickering, abrupt changes, and continuity across action switches and chunk boundaries. *Visual quality* assessed clarity, texture, structural integrity, and visible distortions or artifacts. *Overall preference* captured which result participants preferred after considering these aspects together.

As shown in fig. 6, participants preferred predictor-augmented videos on all five criteria for all four generators. This preference was stronger for scene consistency, action adherence, and temporal consistency, while visual quality showed a more modest advantage. The pattern is consistent with the predictor's role in anticipating scene transitions. By estimating the next state from the current generated latent and the next action, the predictor supplies a visual reference for how the scene should evolve. Modulation uses this reference to refine each new chunk before it becomes part of the history. The preference pattern thus supports future-state guidance for preserving coherent, action-responsive scenes during sustained interaction.

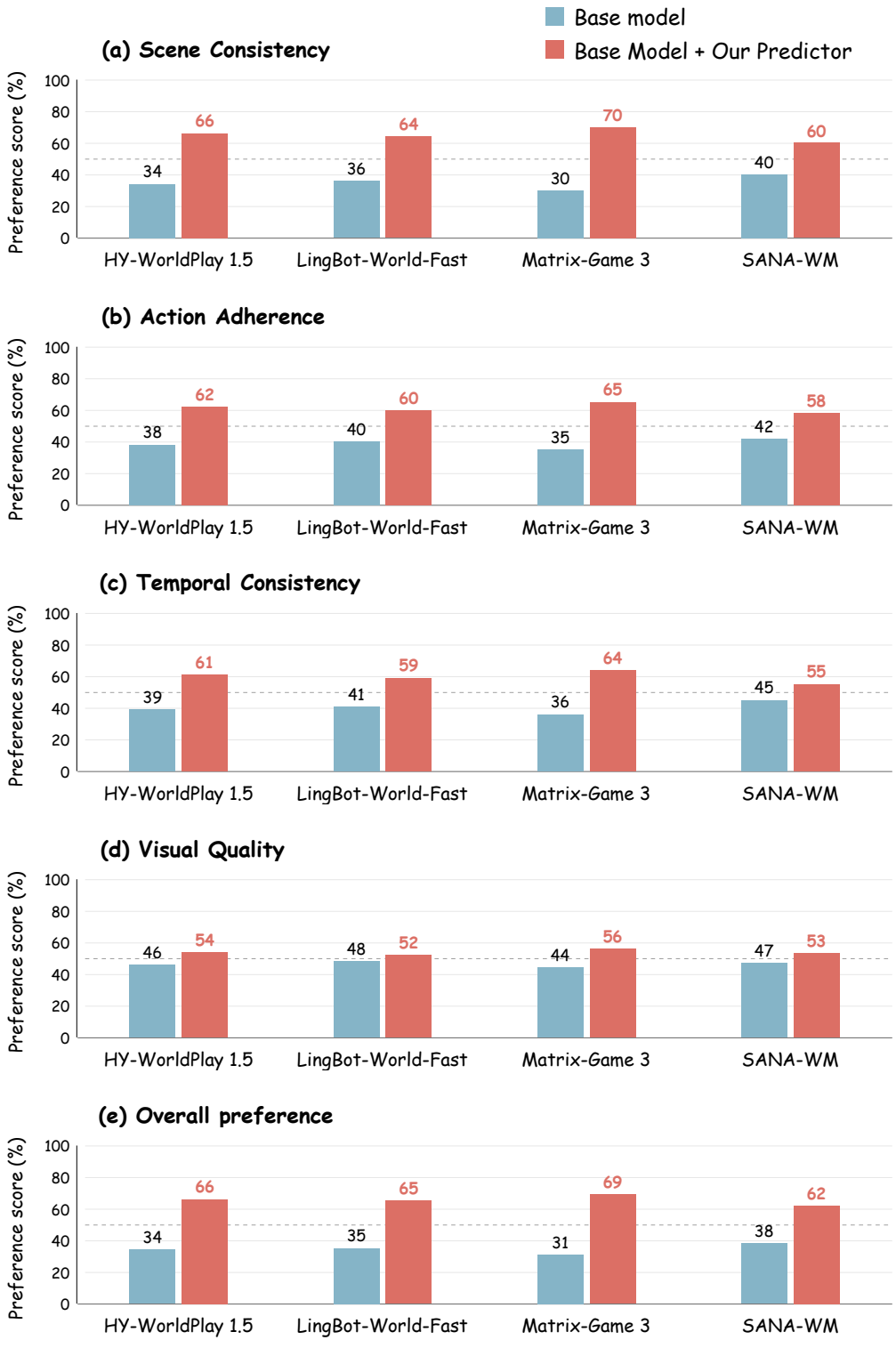

Figure 6: User-study preferences across four base generators on scene consistency, action adherence, temporal consistency, visual quality, and overall preference. Blue and red bars show the preference percentages for the base model and its predictor-augmented counterpart, respectively. The dashed line marks equal preference at 50%.