FORGEWM: PROGRESSIVE CAUSAL TRAINING FOR FEW-STEP ACTION-CONDITIONED VIDEO WORLD MODELS

Anonymous authors
Paper under double-blind review



Figure 1: **ForgeWM rollouts.** Minecraft (top) under keyboard–mouse control and an FPS domain (bottom) under gamepad control, both produced by budget-specialized few-step students trained with the same four-stage framework.

ABSTRACT

Action-conditioned video world models require low-latency causal generation and reliable responses to game-native controls. Although causal distillation enables one- or few-step video synthesis, extending it to interactive world models remains challenging, as discrete keyboard states and continuous mouse motion must remain aligned with temporally compressed latent chunks during causal training and autoregressive rollout. We introduce *ForgeWM*, a progressive framework that transforms a bidirectional action-conditioned video generator into efficient few-step world models through domain adaptation, teacher-forced causal training, causal consistency distillation, and on-policy distribution matching with a bidirectional teacher. The budget-specialized students operate at steady-state denoising budgets of 1, 2, and 4 steps. ForgeWM further supports a dual-path deployment protocol combining latency-critical interaction with optional replay-time refinement, where the one-step student re-noises and refines its saved draft. On paired Minecraft trajectories, ForgeWM leads the evaluated systems in Imaging Quality, reference-aligned motion-profile agreement, action-sign accuracy, and mouse-control accuracy, while achieving the lowest reference LPIPS; the same four-stage recipe transfers to gamepad-controlled FPS gameplay. Replay-time refinement matches four-step reference quality while remaining roughly three times closer to the experienced trajectory than regeneration from noise. These results demonstrate ForgeWM's effectiveness for controllable few-step video generation.

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

Generative video models can simulate future visual states from observations and actions without an explicit graphics engine. Interactive deployment, however, requires causal generation, persistent action responsiveness, and few enough denoising steps to close the control loop. Recent systems demonstrate autoregressive generation with keyboard, mouse, or camera-pose control (He et al., 2025; Wang et al., 2026; Gao et al., 2026; DreamX Team et al., 2026). Yet aggressive sampling compression feeds visual, action, and cache errors into subsequent chunks, coupling fidelity, controllability, and multi-chunk stability. Figure 1 shows ForgeWM rollouts in two control domains.

The difficulty arises because causal generation changes both the input distribution and model state. At inference, the denoiser conditions on imperfect self-generated visual history rather than clean data, while action histories and key–value caches must remain synchronized with the generated latent chunks. Fewer denoising steps amplify errors in these states, which then propagate autoregressively. This coupling is especially pronounced for game-native controls: discrete keyboard states and continuous mouse motion arrive at video-frame rate and enter the denoiser through dedicated pathways rather than solely as a camera trajectory. A usable training framework must therefore preserve the control interface, action-to-latent alignment, and causal state-update protocol throughout adaptation and distillation.

Recent causal video methods (Huang et al., 2025; Zhao et al., 2026b; Zheng et al., 2026) combine teacher-forced causalization, few-step distillation, and on-policy self-rollout. ForgeWM studies how to preserve frame-rate discrete and continuous game controls through this conversion as clean context gives way to generated history.

To address this challenge, we introduce *ForgeWM*, a progressive four-stage framework that extends causal distillation to few-step action-conditioned video world models. It converts a bidirectional generator through domain adaptation, teacher-forced causal training, causal consistency distillation, and on-policy distribution matching. The two branches share the same initialization; during autoregressive self-rollout, the bidirectional teacher supervises the consistency-initialized student. A modular action interface preserves aligned keyboard-and-mouse conditioning throughout training. We train budget-specialized 1-, 2-, and 4-step students, providing quality–latency operating points.

We evaluate ForgeWM against interactive world-model baselines at its native 1-, 2-, and 4-step operating points, and isolate test-time solver scaling using a frozen one-step checkpoint. We further compare draft-preserving replay with from-noise generation, separating online interaction from optional replay-time refinement.

The contributions of this work are:

- We introduce a four-stage framework that converts a bidirectional action-conditioned generator into budget-specialized few-step causal world models.
- We preserve frame-aligned discrete and continuous game-native controls across latent compression, causal training, and autoregressive rollout.
- We provide 1-, 2-, and 4-step operating points and a same-checkpoint replay protocol that improves offline quality while retaining the experienced trajectory.

2 RELATED WORK

Interactive video world models. Early diffusion-based simulators such as DIAMOND and GameNGen model game observations conditioned on agent actions (Alonso et al., 2024; Valevski et al., 2025), while Genie learns action-controllable environments from unlabeled video (Bruce et al., 2024) and MineWorld targets real-time Minecraft interaction (Guo et al., 2025). Recent systems extend this setting with frame-level controls, long-horizon memory, and cross-game transfer (He et al., 2025; Wang et al., 2026; Tong et al., 2026). LingBot-World 2.0 combines MoBA causal pretraining with consistency and rollout-level distribution matching (Gao et al., 2026), while Cosmos 3 develops an omnimodal world-model backbone for Physical AI (NVIDIA et al., 2026).

A complementary family parameterizes interaction through camera geometry. PRoPE encodes relative projective transformations in attention (Li et al., 2025), minWM combines camera-controllable adaptation with causal distillation (Zhao et al., 2026a), and WorldPlay uses dual action representa-

tions with reconstituted context memory (Sun et al., 2025). DreamX-World 1.0 extends this line with E-PRoPE, causal forcing, long-rollout distribution matching, and geometry-retrieved scene memory (DreamX Team et al., 2026). Relative to these camera-, prompt-, or omnimodal systems, ForgeWM centers frame-aligned keyboard-and-mouse controls across budget-specialized causal students and draft-preserving replay.

Few-step autoregressive video distillation. One- and few-step generation builds on consistency models, which learn direct noise-to-data mappings (Song et al., 2023), and distribution matching distillation, which aligns a fast student's distribution with that of a diffusion teacher (Yin et al., 2024). Recent video methods adapt these ideas to causal autoregressive generation. Diffusion Forcing assigns independent noise levels to sequence tokens (Chen et al., 2024), while Self-Forcing reduces exposure bias through autoregressive self-rollout (Huang et al., 2025) and AAPT provides an adversarial student-forcing alternative for one-NFE generation (Lin et al., 2025).

Causal Forcing initializes the student along a causal teacher trajectory before asymmetric distribution matching (Zhu et al., 2026b). Causal Forcing++ replaces offline ODE pairs with online causal consistency distillation (Zhao et al., 2026b), and Causal-rCM unifies teacher-forcing consistency learning with Self-Forcing distribution matching (Zheng et al., 2026). ForgeWM extends this progression to game-native world modeling through a training-to-deployment framework that produces separate 1-, 2-, and 4-step causal checkpoints while preserving the alignment of discrete and continuous controls.

Most fixed-budget models can be evaluated with additional solver steps, but quality need not improve outside their trained regime. AnyFlow instead learns flow-map transitions for flexible test-time budgets (Gu et al., 2026); ForgeWM trains budget-specialized students and separately evaluates a frozen one-step checkpoint under off-budget schedules.

Replay-time and video refinement. Adding noise to an existing sample and denoising it with a generative prior is the basis of SDEdit (Meng et al., 2022). Recent video systems develop more specialized correction mechanisms. AutoRefiner learns pathwise noise refinement and a reflective cache for autoregressive video diffusion (Yu et al., 2025b); Pathwise Test-Time Correction uses the initial frame to calibrate intermediate stochastic states without retraining (Xiang et al., 2026); and SANA-WM employs a trained long-video refiner in a two-stage minute-scale pipeline (Zhu et al., 2026a). ForgeWM reconditions a saved rollout with the deployed low-step student at a larger schedule, requiring no dedicated refiner or second checkpoint. We evaluate reference quality and draft retention separately.

3 METHOD

ForgeWM transforms a bidirectional action-conditioned video generator into causal world models specialized for different sampling budgets. Figure 2 summarizes its game-native action interface, four-stage training graph, and dual-path deployment. Two branches start from the same base generator: a bidirectional domain teacher and a causal student. Their outputs meet during on-policy distribution matching over the student's autoregressive self-rollouts. We next describe the shared generator and action interface, followed by the training and deployment procedures.

3.1 PROBLEM SETUP AND BASE GENERATOR

Let $x_{0:L}$ be a video, x_0 its initial observation, and $a_{0:L} = (k_{0:L}, m_{0:L})$ the corresponding control stream, where k is a discrete keyboard state and $m = (\Delta u, \Delta v)$ is continuous mouse motion. A fixed VAE maps the video to a clean latent sequence z. Following the Matrix-Game 2.0 image-to-video design (He et al., 2025), the generator receives the encoded first frame through both channel-wise latent conditioning and a visual-context branch.

We adapt the base generator using the flow-matching objective (Lipman et al., 2023). Using the noise level $\sigma \in [0, 1]$ as the path coordinate and Gaussian noise ϵ, we construct

$$z_\sigma = (1 - \sigma)z + \sigma\epsilon, \qquad v^*(z_\sigma, \sigma) = \epsilon - z, \tag{1}$$

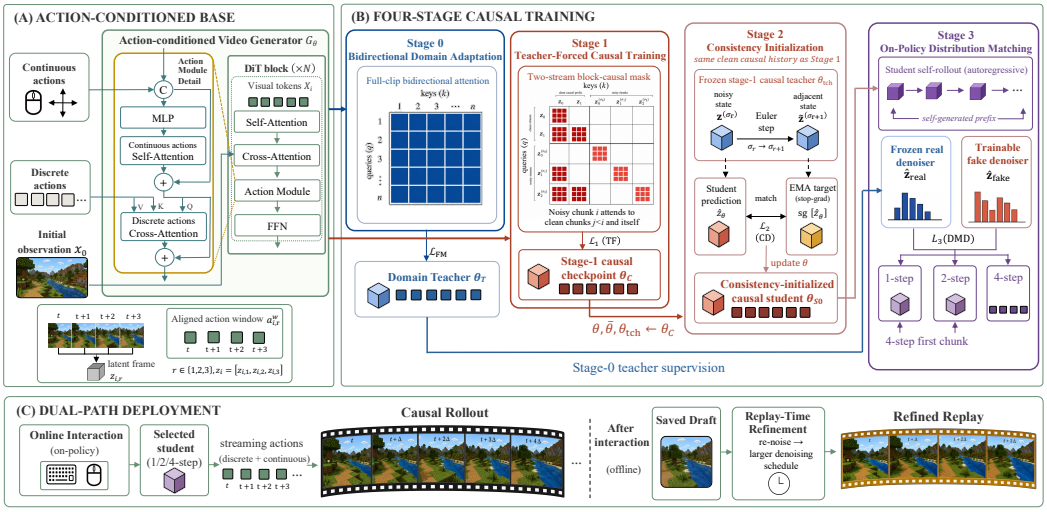

Figure 2: **Overview of ForgeWM.** (A) Frame-aligned keyboard and mouse controls condition latent chunks. (B) A shared base yields a bidirectional teacher and budget-specialized causal students through four-stage training. (C) Deployment separates low-latency interaction from optional Replay-Time Refinement (Figure 6).

and train the velocity predictor v_θ using

$$\mathcal{L}_{\text{FM}} = \mathrm{E}_{z,\epsilon,\sigma}\left[w(\sigma)\,\|v_\theta(z_\sigma,\sigma;c) - v^*(z_\sigma,\sigma)\|_2^2\right], \tag{2}$$

where c contains the initial-frame and action conditions. The induced clean prediction is

$$\hat{z}_\theta(z_\sigma,\sigma) = z_\sigma - \sigma\, v_\theta(z_\sigma,\sigma)\,, \tag{3}$$

which is the quantity the consistency and distribution-matching stages compare. The first stage uses bidirectional temporal attention over each training clip; later stages change the temporal execution and training distribution while retaining this conditioning interface.

3.2 Game-Native Action Conditioning

Following Matrix-Game 2.0 (He et al., 2025), the *ActionModule* injects frame-rate discrete actions and continuous controls through separate pathways, without reducing them to camera poses or pose-based encodings such as PRoPE (Li et al., 2025). Discrete inputs serve as cross-attention keys and values, while windowed continuous inputs are fused with visual features and processed by temporal attention.

The VAE compresses four video frames into one latent frame, and each three-latent causal chunk therefore spans twelve video frames. We group actions by latent interval and retain the preceding controls required by each token's temporal window. During rollout, the model maintains a visual key–value cache together with separate keyboard and mouse caches at action-enabled blocks, so each new chunk receives its complete aligned action window. The same alignment and cache-update protocol is used throughout causal training, distillation, and inference.

3.3 Four-Stage Causal Training

ForgeWM progressively changes the temporal execution pattern, sampling objective, and conditioning-history distribution across four stages. Figure 2(B) visualizes their computational dependencies; Table A1 in Appendix A summarizes each stage's initialization, context, and objective, together with the corresponding optimization settings and stage-wise ablations.

Stage 0: bidirectional adaptation. We first adapt the action-conditioned image-to-video generator to the target game domain using $\mathcal{L}_{\text{FM}}$. Full-clip bidirectional attention learns the visual and control prior. We retain this checkpoint as the frozen real denoiser used during Stage 3.

Stage 1: teacher-forced causal training. In parallel to Stage 0, we initialize a second branch from the same base generator and replace full temporal attention with block-wise causal attention. Frames inside a latent chunk attend bidirectionally, while each chunk can attend only to earlier chunks. Training concatenates the clean and noisy token streams and applies a block mask under which noisy chunk i attends to clean chunks $j < i$ and to itself, so the history is exact rather than model-generated. A single noise level is drawn per chunk, giving the teacher-forced objective

$$\mathcal{L}_1 = \mathrm{E}\left[w(\sigma_i) \left\| v_\theta\left(z_i^{(\sigma_i)}, \sigma_i; c_i, z_{<i} \right) - (\epsilon_i - z_i) \right\|_2^2 \right], \tag{4}$$

where $z_{<i}$ are the clean preceding chunks, c_i is the initial-frame and action condition for chunk i, and w is the same flow-matching weighting as Stage 0. This stage learns the causal execution pattern without yet exposing the model to errors from its own rollout.

Stage 2: causal consistency initialization. Starting from the Stage 1 causal checkpoint, we perform online causal consistency distillation, building on consistency distillation (Song et al., 2023) and its causal video adaptations (Zhao et al., 2026b). Generator, its exponential-moving-average copy, and a frozen teacher are all initialized from that checkpoint. We use a discrete grid of $N = 48$ noise levels $\sigma_0 > \cdots > \sigma_{N-1}$ and sample one adjacent pair (σ_i, σ_{i+1}) per step. The frozen teacher takes one classifier-free-guided Euler step from σ_i towards σ_{i+1},

$$\tilde{z}^{(\sigma_{i+1})} = z^{(\sigma_i)} + (\sigma_{i+1} - \sigma_i) \left[v_{\mathrm{tch}}^{\emptyset} + \omega \left(v_{\mathrm{tch}}^{c} - v_{\mathrm{tch}}^{\emptyset} \right) \right], \tag{5}$$

with guidance weight ω and v_{tch}^{c}, $v_{\mathrm{tch}}^{\emptyset}$ the conditional and unconditional teacher velocities. Because $\sigma_{i+1} < \sigma_i$, the step moves towards the data end of the path. The student's original-level clean prediction is then matched to the EMA stop-gradient at the teacher-advanced point:

$$\mathcal{L}_2 = \mathrm{E}\left[\left\| \hat{z}_\theta\left(z^{(\sigma_i)}, \sigma_i \right) - \mathrm{sg}\left[\hat{z}_{\bar{\theta}}\left(\tilde{z}^{(\sigma_{i+1})}, \sigma_{i+1} \right) \right] \right\|_2^2 \right], \tag{6}$$

where $\bar{\theta}$ is the EMA parameter vector and $\mathrm{sg}[\cdot]$ denotes stop-gradient. All three networks are conditioned on clean causal history at this stage, which is what lets the teacher's Euler step be evaluated in one forward pass instead of an autoregressive unroll. This local consistency objective initializes few-step sampling without an offline trajectory dataset.

Stage 3: on-policy distribution matching. Finally, the student performs autoregressive self-rollout, so each new causal chunk is conditioned on its own previously generated history. The student is initialized from Stage 2, while the domain-adapted Stage 0 checkpoint provides the real-distribution supervision. Let $\hat{z}$ be the clean latent produced by a K-step self-rollout and let $\hat{z}^{(\sigma)}$ be a re-noised copy of it. Following distribution matching distillation (Yin et al., 2024) and its autoregressive video adaptations in Self-Forcing and Causal Forcing (Huang et al., 2025; Zhu et al., 2026b), we form the gradient direction from the disagreement between a frozen real denoiser $\hat{z}_{\mathrm{real}}$ and a trainable fake denoiser $\hat{z}_{\mathrm{fake}}$, which provide score-equivalent predictions under the denoiser parameterization of Eq. (3),

$$g = \frac{\hat{z}_{\mathrm{fake}}\left(\hat{z}^{(\sigma)}, \sigma \right) - \hat{z}_{\mathrm{real}}\left(\hat{z}^{(\sigma)}, \sigma \right)}{\mathrm{mean}\left(\left| \hat{z} - \hat{z}_{\mathrm{real}}\left(\hat{z}^{(\sigma)}, \sigma \right) \right| \right)}, \tag{7}$$

where the denominator averages the absolute deviation over all elements of each sample and acts as an adaptive normalizer. We apply this direction through the generator surrogate

$$\mathcal{L}_3 = \mathrm{E}\left[\frac{1}{2} \left\| \hat{z} - \mathrm{sg}[\hat{z} - g] \right\|_2^2 \right], \tag{8}$$

whose gradient with respect to $\hat{z}$ is exactly g; the fake denoiser is updated concurrently with its own flow-matching loss on the student's samples. Training on the induced rollout distribution reduces the clean-history mismatch left by teacher forcing. We train separate budget-specialized students for $K \in \{1, 2, 4\}$. Following the first-frame enhancement strategy of Yang et al. (2026), the 1- and 2-step students use a fixed four-step schedule for the first generated latent chunk and the matched K-step schedule thereafter; the 4-step student uses four denoising evaluations throughout. Thus, "1-step" and "2-step" denote the steady-state denoising budget rather than the total number of generator calls for the complete rollout.

3.4 BUDGET-SPECIALIZED INTERACTION

For every generated chunk, the model follows its configured denoising schedule and appends the clean prediction to the causal history. The primary deployment uses the student matched to each target budget. We distinguish this native evaluation from an off-budget analysis that freezes the one-step student and changes only its test-time solver schedule. Because the low-step models retain a four-evaluation first chunk, "one-step" and "two-step" refer to the steady-state budget. The fixed-checkpoint step-scaling analysis reports both generator evaluations and wall time under one consistent inference path.

Dual-path deployment. ForgeWM uses the native low-step student for online interaction and optionally applies a larger denoising budget to the saved rollout after interaction. The replay path starts from the realized draft rather than from fresh noise and reuses the recorded actions and refined causal prefix. The empirical motivation, formulation, and evaluation of this replay path are presented together later under *Inference Scaling and Replay-Time Refinement* (Section 4.3).

4 EXPERIMENTS

4.1 IMPLEMENTATION DETAILS

Training. We initialize ForgeWM from the public Matrix-Game 2.0 lineage (He et al., 2025) and train on 40,000 clips constructed from GF-Minecraft (Yu et al., 2025a) at 640×352. The final stage produces students with steady-state budgets of one, two, and four denoising updates. We conduct a stage-wise comparison of checkpoints from Stages 1–3 under a shared four-step inference schedule; full optimization settings, training details, and ablation results are provided in Appendix A.

Evaluation. We compare ForgeWM with the released Matrix-Game 2.0 (He et al., 2025) and HY-WorldPlay (Sun et al., 2025) checkpoints using 77-frame rollouts with shared initial frames and controls. Both baselines use four denoising updates per chunk; ForgeWM uses four updates for the initial chunk and its native one-, two-, or four-step budget thereafter. ForgeWM and Matrix-Game 2.0 share a keyboard-and-mouse interface, while a deterministic adapter maps the same controls to HY-WorldPlay's parameterization. Reference-aligned metrics use 1,000 paired trajectories; no-reference metrics use 462 constant-action videos from 77 initial states. Chunk time is the mean sampling time over non-initial chunks, excluding loading, VAE decoding, and writing; FPS accounts for the 12-frame ForgeWM/Matrix-Game chunks and 16-frame HY-WorldPlay chunks. Replay uses frozen ForgeWM-1 at $r = 0.3$ with four updates; the direct four-step baseline uses native ForgeWM-4 generation from noise. The user study includes 41 participants and 615 selections; CrossFPS uses 25 clips from each of seven games. Full protocols are provided in Appendix B.

Metrics. We report VBench Imaging Quality (IQ), Aesthetic Quality (AQ), and Subject Consistency (Huang et al., 2024); LPIPS (Zhang et al., 2018); optical-flow profile similarity; and Mouse Accuracy based on GameWorld Score (Zhang et al., 2025). For keyboard control, we report KCtrl, a camera-trajectory sign test on opposite-action pairs, in place of GameWorld Score Keyboard Accuracy, whose real-frame-trained inverse-dynamics evaluator can conflate control response with visual-domain similarity. For replay, Replay LPIPS and D_{draft} use LPIPS with an AlexNet backbone on the same 16 evaluated frames, excluding the initial frame. Replay LPIPS compares the output with its paired reference, whereas D_{draft} compares it with the saved draft. A blind three-way study measures visual, action, and spatiotemporal preferences.

4.2 INTERACTIVE WORLD MODEL COMPARISON

Table 1 shows that ForgeWM variants attain the best reported values in six of seven quality/control columns, while ForgeWM-1 achieves the highest measured generation throughput. Performance is not monotone in the denoising budget, motivating the fixed-checkpoint study below. Subject Consistency should be interpreted cautiously because it can favor conservative, low-motion videos; the reference-aligned Flow Profile metric better reflects whether the requested motion pattern is reproduced.

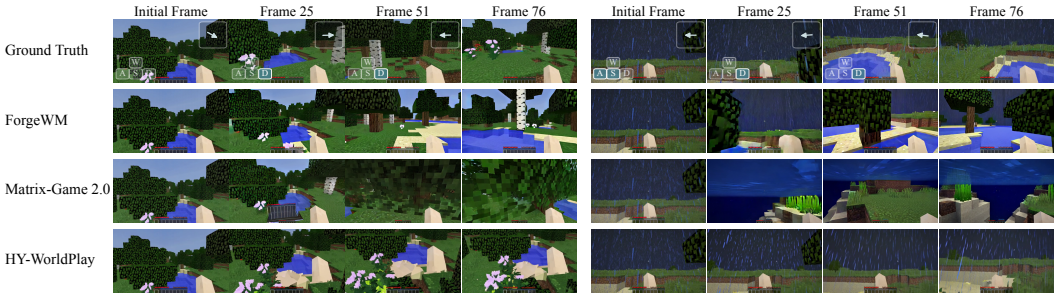

Figure 3: **Qualitative comparison.** Rollouts at frames 0, 25, 51, and 76. Rows show the reference and three models. Controls are annotated once on the reference: each overlay shows dominant WASD and accumulated mouse-look to the next frame; the final frame has no outgoing control overlay. Left/right: daytime forest stream/rainy riverbank at night.

Model	Visual Quality			Temporal Quality		Action Controllability		Efficiency	
	IQ↑	LPIPS↓	AQ↑	Subj. Cons.↑	Flow Prof.↑	KCtrl↑	Mouse Acc.↑	Latency (ms)↓	FPS↑
Matrix-Game 2.0	0.6282	0.6443	0.4583	0.7349	0.9343	0.9156	0.7061	370.9	32.35
HY-WorldPlay	0.6133	0.6172	0.4855	**0.9466**	0.8288	0.9286	0.5818	2164.3	7.54
ForgeWM-1 (1-step)	0.6776	0.6529	0.4807	0.8279	0.9403	0.9545	0.7848	168.2	72.10
ForgeWM-2 (2-step)	**0.6865**	0.6171	0.4814	0.8349	**0.9429**	**0.9740**	**0.8268**	239.7	50.31
ForgeWM-4 (4-step)	0.6788	**0.6168**	**0.4860**	0.7613	0.9420	**0.9740**	0.8102	369.6	32.47

Table 1: **Comparison with interactive world models.** Bold marks the best quality/control value in each column; efficiency entries are not bolded.

ForgeWM-2 and ForgeWM-4 jointly rank first on KCtrl, indicating the highest action-sign accuracy under counterfactual opposite-action pairs. Figure 3 shows aligned rollouts from the reference and the three models under a shared initial frame and control trace; Appendix D presents 22 s rollouts.

Human preference study. Figure 4 reports a blind three-way study comparing the four-step ForgeWM student with Matrix-Game 2.0 and HY-WorldPlay. In each comparison, the three clips shared the same initial state and control trace; model identities were hidden and their left-to-right order was randomized. ForgeWM-4 receives 68.8% of visual-quality preferences, 57.6% of action-accuracy preferences, and 55.6% of spatiotemporal-consistency preferences. Its pooled preference share is 60.7%, with the largest margin in visual quality. Appendix B.3 provides recruitment, interface, and instruction details.

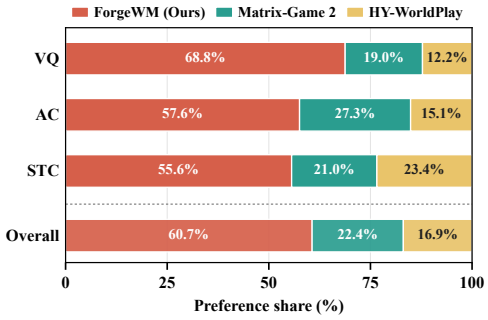

Figure 4: **Human preferences.** 41 participants, five triplets per criterion, 615 selections.

4.3 Inference Scaling and Replay-Time Refinement

Test-time step scaling. To separate inference-time compute from checkpoint specialization, we freeze ForgeWM-1 and vary only its denoising schedule. Figure 5 shows that performance remains stable beyond the native one-step budget. Imaging Quality peaks at two steps, Subject Consistency at four, and reference LPIPS improves through four to eight steps before slightly regressing. Flow Profile and KCtrl remain stable. More steps primarily increase motion magnitude, latency, and computational cost rather than directional control.

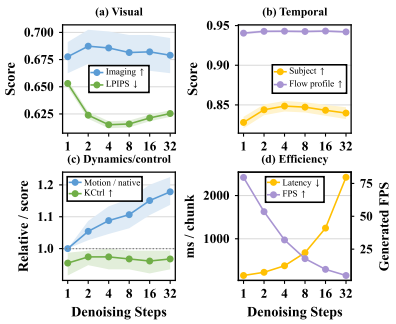

Figure 5: **Test-time step scaling.** ForgeWM-1 under different denoising budgets; shading denotes bootstrap 95% confidence intervals.

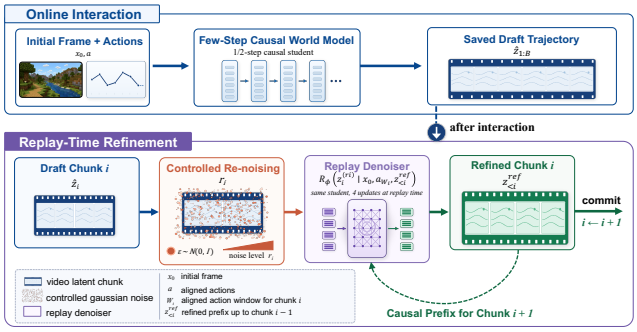

Figure 6: **Replay-Time Refinement.** The deployed student re-noises and denoises its own saved chunks under aligned actions and the refined causal prefix; each refined chunk is committed before the next is processed.

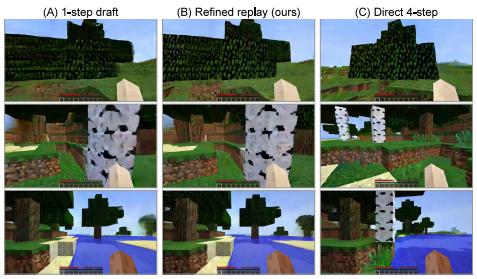

Figure 7: **Replay refinement versus direct generation.** Each row shares the same initial frame and controls. **(A)** Saved ForgeWM-1 draft; **(B)** replay-refined result; **(C)** direct ForgeWM-4 generation from noise.

Method	Replay LPIPS↓	D_{draft} ↓
Draft-preserving methods		
One-step draft (no refinement)	0.6532	—
Replay refinement (ours)	0.6155	0.1970
w/ four-step companion refiner	0.6157	0.1960
w/ four-step draft and refiner†	0.6099	0.2361
From-noise baselines		
Direct ForgeWM-4 (from noise)	0.6168	0.6187

Table 2: **Replay-Time Refinement.** Replay LPIPS is computed against the paired reference; D_{draft} is LPIPS to the saved draft. The † row is evaluated against its own four-step draft and is therefore not directly comparable with the remaining rows.

From scaling to refinement. These trends reveal a deployment asymmetry: one step is preferable for online interaction, whereas additional denoising can improve reference-aligned quality without sacrificing directional control. Rerunning from noise, however, produces a new trajectory rather than improving the rollout the user experienced. Replay-Time Refinement instead applies the additional compute directly to the saved draft.

Draft-preserving replay. Given a saved rollout $\hat{z}_{1:B}$, we retain its initial observation, actions, and latent chunks. Figure 6 summarizes the resulting sequential procedure. Each draft chunk is re-noised at an intermediate flow time r_i:

$$z_i^{(r_i)} = (1 - r_i)\hat{z}_i + r_i\epsilon_i, \qquad \epsilon_i \sim \mathcal{N}(0, I). \tag{9}$$

The deployed student then denoises it under the recorded action window and the already-refined causal prefix:

$$z_i^{\text{ref}} = \mathcal{R}_\phi^{\mathcal{S}(r_i)}\left(z_i^{(r_i)}; x_0, a_{\mathcal{W}_i}, z_{<i}^{\text{ref}}\right). \tag{10}$$

In the default setting, $\mathcal{R}_\phi$ is the frozen ForgeWM-1 draft model, and $\mathcal{S}(r_i)$ consists of four updates from $r_i = 0.3$ to zero. Replay therefore requires no parameter updates or separately trained refiner. Each refined chunk is committed before processing the next. The noise level r_i controls the trade-off between draft retention and visual revision. Because replay occurs after interaction, it adds no online computation.

Replay results. Figure 7 illustrates the distinction between refining a realized rollout and resampling a new one. Starting from the one-step draft in **(A)**, replay in **(B)** refines local visual detail



Figure 8: **CrossFPS rollouts.** ForgeWM-CrossFPS on **(A)** Halo Infinite, **(B)** Modern Warfare, and **(C)** a science-fiction shooter. Colored overlays indicate active gamepad controls.

while preserving the draft's viewpoint and scene layout. Direct ForgeWM-4 generation in **(C)** also produces a high-quality rollout, but its fresh-noise initialization can lead to a different realization, visibly changing the viewpoint and object layout. Table 2 quantifies this distinction: replay achieves an LPIPS of 0.6155, comparable to 0.6168 for direct ForgeWM-4, while reducing D_{draft} from 0.6187 to 0.1970. Using the four-step companion as the refiner yields nearly the same trade-off, indicating that replay does not require a separate checkpoint.

4.4 Recipe Transfer to FPS Gameplay

To test whether the training recipe extends to a different control space, we apply the same four-stage progression to first-person-shooter (FPS) gameplay. This produces ForgeWM-CrossFPS, a separately trained checkpoint. We adopt the cross-game FPS setting of SCOPE (Tong et al., 2026), in which controls are gamepad signals: two continuous analog sticks for movement and look, together with discrete action buttons (fire, aim, jump, reload, weapon switch, and melee). Using a similarly structured action interface and 4-stage recipe without architectural changes yields a four-step causal student.

Figure 8 shows representative rollouts from three games. The macro-average paired LPIPS is 0.656, while the generated-to-reference motion ratio averages 1.45, suggesting a tendency toward stronger motion than the reference. These measurements use a different domain and protocol from the Minecraft comparison; Appendix C reports the per-game breakdown.

4.5 Evaluation Scope

Our primary evaluation intentionally focuses on a controlled Minecraft setting, where models can be assessed from matched initial states and mapped control traces. Broader out-of-distribution generalization is therefore beyond the scope of the current study. Because HY-WorldPlay uses a different control parameterization, its effective motion scale depends on the deterministic adapter. We consequently restrict direct quantitative comparisons to Minecraft and present CrossFPS separately as a complementary evaluation of recipe transfer to another gameplay domain.

5 Conclusion

We presented ForgeWM, a progressive causal training framework for budget-specialized few-step world models that preserves frame-aligned game-native controls. Across its budget-specialized students, ForgeWM delivers the strongest overall quality–control profile among the evaluated systems on paired Minecraft trajectories. ForgeWM further exposes distinct 1-, 2-, and 4-step operating points and a replay-refinement path that separates latency-critical interaction from optional quality-oriented processing, where the deployed one-step student refines its own saved draft and matches four-step reference quality without a second checkpoint. The same training procedure also transfers to a gamepad-controlled FPS setting, while our controlled quantitative comparison remains on Minecraft. Together, these components provide an effective training and deployment framework for controllable few-step video world models.

AI USE STATEMENT

In this work, we used generative AI tools to assist with writing and debugging portions of the training code and figure-generation scripts, to provide feedback on experimental design and result presentation, to search for and summarize relevant literature, and to edit portions of the manuscript for clarity, concision, and structure. All AI-assisted code was reviewed, executed, and tested by the authors, and the resulting figures were checked against the underlying experimental outputs. Generative AI tools did not directly supply any dataset, experimental measurement, model output, or human-study response reported in this paper. The authors checked citations against their original sources and independently verified all equations, methodological claims, and reported results. The authors take responsibility for the final content of this work, including text, code, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

The user study in Appendix B.3 involved 41 student volunteers who evaluated generated gameplay clips without being shown model identities. Participation was voluntary, and participants consented to the use of their responses for research. The study collected only anonymous clip preferences; no personally identifiable or sensitive information was recorded, and results are reported only in aggregate. Training and evaluation use gameplay data from GF-Minecraft (Yu et al., 2025a) and the cross-game FPS setting of SCOPE (Tong et al., 2026) under their respective terms of use.

REPRODUCIBILITY STATEMENT

Section 4 and Appendix A specify the training data, model initialization, and per-stage optimization settings for every reported checkpoint. Appendix B defines the evaluation metrics, the action conversion used for the HY-WorldPlay baseline, and the user-study protocol; Appendix C documents the CrossFPS adaptation. Together, these materials document the training and evaluation procedures used to produce the reported results.

REFERENCES

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 58757–58791. Curran Associates, Inc., 2024. doi: 10.52202/079017-1873. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/6bdde0373d53d4a501249547084bed43-Paper-Conference.pdf.

Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Bechtle, Feryal Behbahani, Stephanie Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando de Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 4603–4623, 2024.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 24081–24125. Curran Associates, Inc., 2024. doi: 10.52202/079017-0759. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/2aee1c4159e48407d68fe16ae8e6e49e-Paper-Conference.pdf.

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*, 2026.

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.

Yuchao Gu, Guian Fang, Yuxin Jiang, Weijia Mao, Song Han, Han Cai, and Mike Zheng Shou. Anyflow: Any-step video diffusion model with on-policy flow map distillation. *arXiv preprint arXiv:2605.13724*, 2026.

Junliang Guo, Yang Ye, Tianyu He, Haoyu Wu, Yushu Jiang, Tim Pearce, and Jiang Bian. Mineworld: a real-time and open-source interactive world model on minecraft. *arXiv preprint arXiv:2504.08388*, 2025.

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.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. In *Advances in Neural Information Processing Systems*, 2025.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Ruilong Li, Brent Yi, Junchen Liu, Hang Gao, Yi Ma, and Angjoo Kanazawa. Cameras as relative positional encoding. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 15984–16009. Curran Associates, Inc., 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/17a7075094632c88cccdd86270ad715b-Paper-Conference.pdf.

Shanchuan Lin, Ceyuan Yang, Hao He, Jianwen Jiang, Yuxi Ren, Xin Xia, Yang Zhao, Xuefeng Xiao, and Lu Jiang. Autoregressive adversarial post-training for real-time interactive video generation. *arXiv preprint arXiv:2506.09350*, 2025.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The eleventh international conference on learning representations*, 2023.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022.

NVIDIA, Aditi, Niket Agarwal, Arslan Ali, Jon Allen, Martin Antolini, Adeline Aubame, Alisson Azzolini, Junjie Bai, Maciej Bala, Yogesh Balaji, Josh Bapst, Aarti Basant, Mukesh Beladiya, Mohammad Qazim Bhat, Zaid Pervaiz Bhat, Dan Blick, Vanni Brighella, Han Cai, Tiffany Cai, Eric Cameracci, Jiaxin Cao, Yulong Cao, Mark Carlson, Carlos Casanova, Ting-Yun Chang, Yan Chang, Yu-Wei Chao, Prithvijit Chattopadhyay, Roshan Chaudhari, Chieh-Yun Chen, Junyu Chen, Ke Chen, Qizhi Chen, Wenkai Chen, Xiaotong Chen, Yu Chen, An-Chieh Cheng, Click Cheng, Xiu Chia, Jeana Choi, Chaeyeon Chung, Wenyan Cong, Yin Cui, Magdalena Dadela, Nalin Dadhich, Wenliang Dai, Joyjit Daw, Alperen Degirmenci, Rodrigo Vieira Del Monte, Robert Denomme, Sameer Dharur, Marco Di Lucca, Ke Ding, Wenhao Ding, Yifan Ding, Yuzhu Dong, Nicole Drumheller, Yilun Du, Aigul Dzhumamuratova, Aleksandr Efitorov, Hamid Eghbalzadeh, Naomi Eigbe, Imad El Hanafi, Hassan Eslami, Benedikt Falk, Jiaojiao Fan, Jim Fan, Amol Fasale, Sergiy Fefilatyev, Liang Feng, Francesco Ferroni, Sanja Fidler, Xiao Fu, Vikram Fugro, Prashant Gaikwad, TJ Galda, Katelyn Gao, Yihuai Gao, Wenhang Ge, Sreyan Ghosh, Arushi Goel, Vivek Goel, Akash Gokul, Rama Govindaraju, Jinwei Gu, Miguel Guerrero, Elfie Guo,

Aryaman Gupta, Siddharth Gururani, Hugo Hadfield, Song Han, Ankur Handa, Zekun Hao, Mohammad Harrim, Ali Hassani, Nathan Hayes-Roth, Yufan He, Chris Helvig, Cyrus Hogg, Madison Huang, Michael Huang, Sophia Huang, Yufan Huang, Jacob Huffman, DeLesley Hutchins, Suneel Indupuru, Boris Ivanovic, Arihant Jain, Joel Jang, Ryan Ji, Yanan Jian, Dongfu Jiang, Jingyi Jin, Atharva Joshi, Nikhilesh Joshi, Pranjali Joshi, Andy Ju, Jaehun Jung, Weiwei Kang, Scott Kassekert, Jan Kautz, Ashna Khetan, Julia Kiczka, Slawek Kierat, Gwanghyun Kim, Kuno Kim, Sunny Kim, Kezhi Kong, Xin Kong, Zhifeng Kong, Tomasz Kornuta, Egor Krivov, Hui Kuang, Saurav Kumar, Chia-Wen Kuo, George Kurian, Wojciech Kutak, JF Lafleche, Himangshu Lahkar, Omar Laymoun, Jayjun Lee, Sanggil Lee, Gabriele Leone, Boyi Li, Freya Li, Jiajun Li, Jinfeng Li, Ling Li, Pengcheng Li, Shangru Li, Tingle Li, Xiaolong Li, Xuan Li, Zhaoshuo Li, Zhiqi Li, Hao Liang, Maosheng Liao, Chen-Hsuan Lin, Tsung-Yi Lin, Ming-Yu Liu, Sifei Liu, Zihan Liu, Hai Loc Lu, Xiangyu Lu, Alice Luo, Ruipu Luo, Wenjie Luo, Jiangran Lyu, Martin Ding Ma, Nic Ma, Qianli Ma, Dawid Majchrowski, Louis Marcoux, Miguel Martin, Qing Miao, Ashkan Mirzaei, Shreyas Misra, Kaichun Mo, Durra Mohsin, Hyejin Moon, Pawel Morkisz, Saeid Motiian, Kirill Motkov, Seungjun Nah, Yashraj Narang, Deepak Narayanan, Thabang Ngazimbi, Julian Ouyang, Shubham Pachori, David Page, Yatian Pang, Sehwi Park, Mahesh Patekar, Mostofa Patwary, Marco Pavone, Trung Pham, Wei Ping, Soha Pouya, Shrimai Prabhumoye, Varun Praveen, Delin Qu, Hesam Rabeti, Morteza Ramezanali, Marilyn Reeb, Xuanchi Ren, Kristen Rumley, Wojciech Rymer, Jun Saito, Yeongho Seol, John Shao, Piyush Shekdar, Tianwei Shen, Humphrey Shi, Min Shi, Stella Shi, Kevin Shih, Mohammad Shoeybi, Mateusz Sieniawski, Shuran Song, Alexander Sotelo, Amir Sotoodeh, Sunil Srinivasa, Vignesh Srinivasakumar, Bartosz Stefaniak, Rahul Heinrich Steiger, Shangkun Sun, Jiaxiang Tang, Shitao Tang, Yangyang Tang, Yue Tang, Tolou Tavakkoli, Kayley Ting, Krzysztof Tomala, Wei-Cheng Tseng, Jibin Varghese, Sergei Vasilev, Thomas Volk, Raju Wagwani, Roger Waleffe, Andrew Z. Wang, Boxiang Wang, Haoxiang Wang, Qiao Wang, Shihao Wang, Shijie Wang, Ting-Chun Wang, Yan Wang, Yu Wang, Rohit Watve, David Wehr, Fangyin Wei, Xinshuo Weng, Jay Zhangjie Wu, Kedi Wu, Hongchi Xia, Summer Xiao, Tianjun Xiao, Kevin Xie, Daguang Xu, Jiashu Xu, Mengyao Xu, Ruqing Xu, Xingqian Xu, Yao Xu, Dinghao Yang, Dong Yang, Hans Yang, Xiaodong Yang, Xuning Yang, Yichu Yang, Yurong You, Zhiding Yu, Hao Yuan, Simon Yuen, Xiaohui Zeng, Pengcuo Zeren, Cindy Zha, Haotian Zhang, Jenny Zhang, Jing Zhang, Liangkai Zhang, Paris Zhang, Shun Zhang, Xuanmeng Zhang, Zhizheng Zhang, Ann Zhao, Yilin Zhao, Yuliya Zhautouskaya, Charles Zhou, Fengzhe Zhou, Shilin Zhu, Yuke Zhu, Dima Zhylko, and Artur Zolkowski. Cosmos 3: Omnimodal world models for physical ai, 2026. URL `https://arxiv.org/abs/2606.02800`.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning*, 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.

Zizhao Tong, Yeying Jin, Hongfeng Lai, Zeqing Wang, Zhaohu Xing, Kexu Cheng, Haoran Xu, Zhao Pu, Shangwen Zhu, Ruili Feng, Jian Zhao, Yan Zhang, Hao Tang, and Ling Shao. SCOPE: Simulating cross-game operations in playable environments for FPS world models. *arXiv preprint arXiv:2605.23345*, 2026.

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

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

Xunzhi Xiang, Zixuan Duan, Guiyu Zhang, Haiyu Zhang, Zhe Gao, Junta Wu, Shaofeng Zhang, Tengfei Wang, Qi Fan, and Chunchao Guo. Pathwise test-time correction for autoregressive long video generation. *arXiv preprint arXiv:2602.05871*, 2026.

Yongqi Yang, Huayang Huang, Xu Peng, Xiaobin Hu, Donghao Luo, Jiangning Zhang, Chengjie Wang, and Yu Wu. Towards one-step causal video generation via adversarial self-distillation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=P3O0fNmnWa.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Jiwen Yu, Yiran Qin, Xintao Wang, Pengfei Wan, Di Zhang, and Xihui Liu. Gamefactory: Creating new games with generative interactive videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11590–11599, 2025a.

Zhengyang Yu, Akio Hayakawa, Masato Ishii, Qingtao Yu, Takashi Shibuya, Jing Zhang, and Yuki Mitsufuji. Autorefiner: Improving autoregressive video diffusion models via reflective refinement over the stochastic sampling path. *arXiv preprint arXiv:2512.11203*, 2025b.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Yifan Zhang, Chunli Peng, Boyang Wang, Puyi Wang, Qingcheng Zhu, Fei Kang, Biao Jiang, Zedong Gao, Eric Li, Yang Liu, and Yahui Zhou. Matrix-game: Interactive world foundation model. *arXiv preprint arXiv:2506.18701*, 2025.

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

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

Kaiwen Zheng, Guande He, Min Zhao, Jintao Zhang, Huayu Chen, Jianfei Chen, Chen-Hsuan Lin, Ming-Yu Liu, Jun Zhu, and Qianli Ma. Causal-rcm: A unified teacher-forcing and self-forcing open recipe for autoregressive diffusion distillation in streaming video generation and interactive world models. *arXiv preprint arXiv:2606.25473*, 2026.

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

Hongzhou Zhu, Min Zhao, Guande He, Hang Su, Chongxuan Li, and Jun Zhu. Causal forcing: Autoregressive diffusion distillation done right for high-quality real-time interactive video generation. In *International Conference on Machine Learning*, 2026b.

A Training Details

A.1 Optimization and Stage-Wise Settings

Shared setup. All stages train the Matrix-Game 2.0 image-to-video lineage (He et al., 2025) (a Wan2.1-T2V-1.3B backbone with our action-conditioning module) on GF-Minecraft clips (Yu et al., 2025a) at 640×352 and 12 frames per second. A VAE with $4\times$ temporal compression maps each clip to a latent sequence of 21 latent frames ($16\times44\times80$), so a three-latent causal chunk spans twelve video frames. We optimize with AdamW ($\beta = (0.0, 0.999)$), mixed-precision `bf16`, gradient checkpointing, and fully-sharded data parallelism across eight GPUs; the seed is fixed at 0. The flow-matching path uses 1000 discretization steps with a timestep shift of 5.0; the consistency and distribution-matching stages apply classifier-free guidance with weight 3.0 to the frozen teacher. Learning rates are held constant within each stage.

Stage	Initialization	Training Context	Objective	Output
0	Base	Full-clip bidirectional	FM	Domain teacher
1	Base	Clean causal history	Causal FM	Causal teacher
2	Stage 1	Clean causal history	Online CD	Few-step initializer
3	Stage 2	Self-generated history	DMD	1/2/4-step students

Table A1: **Training stages.** Each stage changes either the temporal execution pattern, sampling objective, or history distribution.

	Stage 0	Stage 1	Stage 2	Stage 3
Objective	$\mathcal{L}_{\mathrm{FM}}$	$\mathcal{L}_1$	$\mathcal{L}_2$	$\mathcal{L}_3$
Trainer	FM	causal FM	consist. distill.	DMD
Attention	bidir.	causal	causal	causal
Init	base	base	Stage 1	Stage 2
Blk. (latents)	21	3	3	3
Trainer iterations	4k	20k	6k	4k
Gen. lr	$2\mathrm{e}{-6}$	$2\mathrm{e}{-5}$	$2\mathrm{e}{-6}$	$2\mathrm{e}{-6}$
Critic lr	–	–	–	$4\mathrm{e}{-7}$
Global batch	8	8	8	8
EMA	–	–	0.99/200	0.99/200

Table A2: **Per-stage optimization for the reported Minecraft ForgeWM lineage.** "Blk." is the number of latent frames per attention block (21 = full-clip bidirectional; 3 = one causal chunk). "EMA" is the generator exponential-moving-average decay and the step it starts ("–" when disabled). Global batch is the effective batch after data-parallel accumulation. All learning rates are constant.

Stage-specific settings. Table A2 lists the per-stage optimization. *Stage 0* adapts the base generator with full-clip bidirectional attention over 21-latent clips and no EMA; we freeze its $4{,}000$-update checkpoint and reuse it as the real denoiser $\hat{z}_{\mathrm{real}}$ in Stage 3. *Stage 1* replaces full attention with block-wise causal attention (three latent frames per block) and trains the teacher-forced objective from the same base initialization; the reported lineage uses the $20{,}000$-update checkpoint. *Stage 2* initializes the generator, its EMA copy, and the frozen teacher from that Stage 1 checkpoint and runs online causal consistency distillation for $6{,}000$ updates over a discrete grid of $N = 48$ noise levels, sampling one adjacent pair per step. *Stage 3* initializes from the Stage 2 checkpoint and comprises three separate $4{,}000$-trainer-iteration runs for $K \in \{1, 2, 4\}$. The four-step student uses the denoising schedule $[1000, 750, 500, 250]$ for every generated chunk. Following Yang et al. (2026), the one- and two-step students use the same fixed four-step schedule for the first generated chunk and their respective budget-matched K-step schedules thereafter. Each run uses a local attention window of six latent frames, equivalent to two three-latent chunks. The critic is updated at every trainer iteration, whereas the generator is updated once every five iterations. Figure A1 plots the per-stage training losses recorded during these runs. The reported ForgeWM-1/2/4 models are the corresponding Stage 3 checkpoints.

The CrossFPS checkpoint reported in the main paper follows the same sequence of four training objectives on gamepad-controlled data, using the widened continuous-control interface described in Appendix C.2.

A.2 Stage-Wise Inference Ablation

The body compares checkpoints from Stages 1–3 under a shared four-step inference schedule; this section gives the protocol and results, and adds the Stage 0 domain teacher as a non-causal reference point.

Protocol. For Stages 1–3, we hold the inference path fixed and vary only the loaded generator weights. All three checkpoints use the same chunked causal pipeline, model configuration, four-step schedule $[1000, 750, 500, 250]$, unrestricted causal attention used for the main-table results, and identical per-trajectory noise seeds. Stage 0 is bidirectional and has no chunked few-step deployment; we therefore evaluate it in its native regime, using a single full-clip block and a four-step

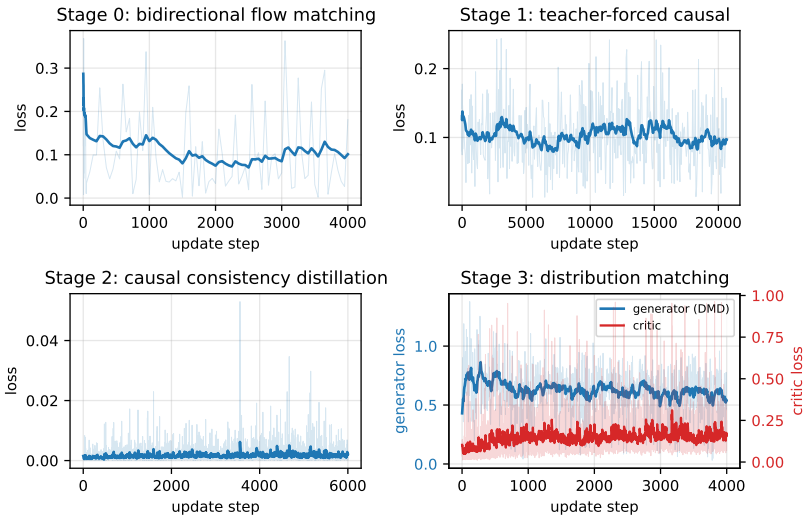

Figure A1: **Training losses across stages.** Raw values are shown as light traces, with exponential-moving-average trends overlaid; the horizontal axis denotes trainer iterations. In Stage 3, the critic is updated every iteration and the generator every fifth iteration. *Top-left*: Stage 0 bidirectional flow matching. *Top-right*: Stage 1 teacher-forced causal flow matching. *Bottom-left*: Stage 2 online causal consistency distillation. *Bottom-right*: Stage 3 distribution matching, with the generator (DMD) loss on the left axis and the critic (fake-denoiser) loss on the right axis. Because the stages optimize different objectives, absolute loss magnitudes are not comparable across panels; the curves document optimization behavior within each stage only.

Stage	Inference regime	LPIPS↓	IQ↑	AQ↑	SC↑
0	bidirectional teacher (ref.)	$0.814_{[.809,.819]}$	0.455	0.463	0.677
1	teacher-forced causal	$0.806_{[.799,.812]}$	0.508	0.454	0.700
2	causal consistency	$\mathbf{0.605}_{[.600,.610]}$	0.659	0.483	**0.760**
3	distribution matching	$0.617_{[.613,.620]}$	**0.716**	**0.489**	**0.760**

Table A3: **Stage-wise inference ablation on** 1,000 **paired trajectories.** LPIPS uses the AlexNet backbone on 16 evaluated frames; subscripts denote trajectory-level bootstrap 95% confidence intervals. IQ, AQ, and SC are the corresponding VBench metrics. Stage 0 is a non-causal domain-teacher reference evaluated in its native full-clip regime.

UniPC schedule. Its row serves as a non-causal domain-teacher reference and is not directly comparable to the causal rows. All rows are scored on the same 1,000 trajectories using the paired LPIPS protocol of Appendix B; intervals are trajectory-level bootstrap 95% confidence intervals.

Findings. Table A3 isolates the contribution of each training stage under a shared four-step evaluation protocol. Stages 0 and 1 are trained as full-trajectory flow-matching models and are not intended for four-step sampling; under this budget, they obtain LPIPS scores of 0.814 and 0.806, respectively. Stage 2 produces the largest improvement, reducing LPIPS from 0.806 to 0.605 with non-overlapping confidence intervals. This result shows that causal consistency distillation, rather than teacher-forced causalization alone, is the stage that enables effective few-step sampling.

Stage 3 introduces training on self-generated autoregressive histories. At four steps, it obtains an LPIPS of 0.617, with Stage 2 retaining a small paired advantage of -0.012 ($[-0.015, -0.008]$). Its effect is more visible in the no-reference metrics: Imaging Quality increases from 0.659 to 0.716, while Subject Consistency remains unchanged at 0.760 and Aesthetic Quality changes only slightly. Thus, at the four-step budget, Stage 3 shifts the trade-off toward sharper per-frame appearance rather than improving paired reconstruction fidelity. Its incremental effect at one- and two-step budgets is not isolated by this ablation.

B Evaluation Protocol

Checkpoint scope. We evaluate Matrix-Game 2.0 and HY-WorldPlay using their publicly released checkpoints, without additional training on the 40,000 GF-Minecraft clips used for ForgeWM. Each system is evaluated under its native inference configuration with the shared initial frames and mapped control traces described below.

B.1 Action Conversion Across Control Interfaces

The three compared systems do not expose the same control interface, so a single recorded control trace is mapped onto each system's own action parameterization and every model is then driven from that mapped trace. The source trace is the frame-rate stream used throughout this work: a discrete keyboard state and a continuous two-dimensional mouse delta per video frame, recorded at 640×352 and 12 frames per second over the 77-frame comparison window. ForgeWM and Matrix-Game 2.0 share this parameterization, so for those two the mapping is the identity up to the sign convention of the vertical mouse axis, which differs between the source recording and the model interface. HY-WorldPlay additionally accepts a continuous camera parameterization, and the conversion for that pathway integrates the per-frame mouse deltas into the angular quantities its interface expects.

B.2 Metric Definitions

Table 1 reports nine columns. Let $X_n = (x_{n,0}, \dots, x_{n,T-1})$ be generated clip n, let $R_n = (r_{n,0}, \dots, r_{n,T-1})$ be its temporally aligned reference when a reference is required, and let N be the number of evaluated clips. All reported dataset scores first average within a clip and then across clips. IQ and AQ use the 462 constant-action clips (77 scenes times six actions), whereas SC, LPIPS, and Flow Profile use the $1{,}000$ shared-action rollouts. Every metric reads exactly the first 77 frames.

VBench metrics. Imaging Quality (IQ), Aesthetic Quality (AQ), and Subject Consistency (SC) use the standard VBench implementation (Huang et al., 2024). Denote the MUSIQ predictor by q_{MUSIQ}, the LAION aesthetic linear predictor by q_{aes}, the normalized CLIP ViT-L/14 image feature by $c(\cdot)$, and the normalized DINO ViT-B/16 feature by $d(\cdot)$. Their dataset-level scores are

$$\mathrm{IQ} = \frac{1}{NT} \sum_{n=1}^{N} \sum_{t=0}^{T-1} \frac{q_{\mathrm{MUSIQ}}(x_{n,t})}{100}, \tag{11}$$

$$\mathrm{AQ} = \frac{1}{NT} \sum_{n=1}^{N} \sum_{t=0}^{T-1} \frac{q_{\mathrm{aes}}(c(x_{n,t}))}{10}. \tag{12}$$

For Subject Consistency, let $[z]_+ = \max(0, z)$ and define the local and first-frame similarities

$$u_{n,t} = [d(x_{n,t-1})^\top d(x_{n,t})]_+, \qquad v_{n,t} = [d(x_{n,0})^\top d(x_{n,t})]_+. \tag{13}$$

Then

$$\mathrm{SC} = \frac{1}{2N(T-1)} \sum_{n=1}^{N} \sum_{t=1}^{T-1} (u_{n,t} + v_{n,t}). \tag{14}$$

Thus, SC rewards both local frame-to-frame consistency and retention of the first-frame subject appearance; it is not an identity-classification accuracy.

Reference-aligned visual and motion metrics. Let

$$\mathcal{I} = \{1, 6, 11, \dots, 76\} \tag{15}$$

be the 16 evaluated frame indices, excluding the given first frame, and let ℓ_{Alex} denote LPIPS with the AlexNet backbone (Zhang et al., 2018). We report

$$\mathrm{LPIPS} = \frac{1}{N|\mathcal{I}|} \sum_{n=1}^{N} \sum_{t\in\mathcal{I}} \ell_{\mathrm{Alex}}(x_{n,t}, r_{n,t}), \tag{16}$$

where lower is better.

For Flow Profile similarity, let $F(y_t, y_{t+1}) \in \mathrm{R}^{H\times W\times 2}$ denote the fixed dense optical-flow estimator. For a video Y, define its temporal motion-magnitude profile by

$$p_k(Y) = \frac{1}{|\Omega|} \sum_{(u,v)\in\Omega} \left\| F(\tilde{Y}_{4k}, \tilde{Y}_{4(k+1)})_{u,v} \right\|_2, \tag{17}$$

for $k = 0, \ldots, 18$, and let $\mathbf{p}(Y) = (p_0(Y), \ldots, p_{18}(Y))$. Here $\tilde{Y}_t$ is frame t converted to grayscale and resized to 160×88 using area interpolation. The spatial evaluation region is $\Omega = \Omega_h \times \Omega_w$, where $\Omega_h = \{\lfloor 0.08H \rfloor, \ldots, \lfloor 0.82H \rfloor - 1\}$ and $\Omega_w = \{\lfloor 0.08W \rfloor, \ldots, \lfloor 0.92W \rfloor - 1\}$. This region suppresses the image borders and the HUD-heavy lower portion. We compute F using OpenCV Farneback flow with pyramid scale 0.5, four pyramid levels, window size 21, four iterations, polynomial neighborhood size 7, polynomial standard deviation 1.5, and flags set to zero.

The reported score is the mean paired cosine similarity

$$\mathrm{FlowProf} = \frac{1}{N} \sum_{n=1}^{N} \frac{\mathbf{p}(X_n)^\top \mathbf{p}(R_n)}{\|\mathbf{p}(X_n)\|_2 \|\mathbf{p}(R_n)\|_2}. \tag{18}$$

For each summand, the cosine score is defined as zero when $\|\mathbf{p}(X_n)\|_2 \|\mathbf{p}(R_n)\|_2 \le 10^{-12}$. Thus, Flow Profile compares the temporal pattern of motion magnitude rather than optical-flow direction or absolute visual similarity.

Action controllability. KCtrl uses constant-action counterfactual pairs from the same initial scene. We sample frames $0, 4, \ldots, 76$, yielding 20 sampled frames and 19 consecutive relative poses. For action a, let $v_{s,a,k} \in \mathrm{R}^3$ denote the translation component of relative pose k in scene s, where $k = 0, \ldots, 18$. Define the requested axis and sign as $j(\text{forward}) = j(\text{back}) = z$, $j(\text{left}) = j(\text{right}) = x$, and $\eta_{\text{f}} = \eta_{\text{r}} = +1$, $\eta_{\text{b}} = \eta_{\text{l}} = -1$. The net-direction indicator is

$$C_{s,a} = \mathrm{I}\left[\eta_a \sum_{k=0}^{18} (v_{s,a,k})_{j(a)} > 0 \right], \tag{19}$$

where $\mathrm{I}[\cdot]$ denotes the indicator function. With $S = 77$ scenes, the main-table score averages the two opposite translation pairs per scene:

$$\mathrm{KCtrl} = \frac{1}{2S} \sum_{s=1}^{S} \left(C_{s,\text{f}} C_{s,\text{b}} + C_{s,\text{l}} C_{s,\text{r}} \right). \tag{20}$$

Thus a pair receives credit only if both opposite commands yield the requested net camera-motion sign; no test-time motion-magnitude threshold is used.

Mouse Accuracy follows the GameWorld/VPT inverse-dynamics evaluator (Zhang et al., 2025). Let $g_{\mathrm{IDM}}(X)_t \in \mathrm{R}^2$ be its predicted camera action at frame t, and let q be GameWorld's nine-way camera-direction quantizer, $q : \mathrm{R}^2 \to \{0, \ldots, 8\}$. Write TL and TR for turn-left and turn-right, respectively; their expected labels are $y_{\mathrm{TL}} = 3$ and $y_{\mathrm{TR}} = 4$. Over the two turn actions, $S = 77$ scenes, and $T_{\mathrm{IDM}} = 76$ evaluated predictions per clip, define the per-clip accuracy as $\hat{y}_{s,a,t} = q(g_{\mathrm{IDM}}(X_{s,a})_t)$ and

$$A_{s,a} = \frac{1}{T_{\mathrm{IDM}}} \sum_{t=0}^{T_{\mathrm{IDM}}-1} \mathrm{I}[\hat{y}_{s,a,t} = y_a]. \tag{21}$$

The reported score is

$$\mathrm{MouseAcc} = \frac{1}{2S} \sum_{s=1}^{S} \left(A_{s,\mathrm{TL}} + A_{s,\mathrm{TR}} \right). \tag{22}$$

This is a nine-way direction-classification accuracy under the commanded constant mouse action, not a continuous regression error or a pixel-reference metric.

Efficiency. For measured chunk j, let L_j be synchronized sampling time in milliseconds and C_j the number of newly generated frames (12 for ForgeWM and Matrix-Game 2.0, and 16 for HY-WorldPlay). Over the J timed chunks,

$$\text{Latency} = \frac{1}{J}\sum_{j=1}^{J} L_j, \qquad \text{FPS} = \frac{1}{J}\sum_{j=1}^{J} \frac{1000 C_j}{L_j}. \tag{23}$$

Loading, VAE decoding, and file writing are excluded. Efficiency columns are reported for context and are not bolded in the table.

Profiling and resampling. Efficiency is profiled on NVIDIA GPUs over 30 clips after three warm-ups; the four-step ForgeWM/Matrix-Game comparison pools 90 measurements from three GPUs, and efficiency reporting follows the same 30-clip protocol. No-reference metrics use scene-level bootstrap and paired metrics use trajectory-level bootstrap; plots report means with bootstrap 95% confidence intervals. The fixed-checkpoint step-scaling study freezes ForgeWM-1 and evaluates $1/2/4/8/16/32$ steps on identical scene–action–seed tuples. Replay-refinement evaluation starts from a saved one-step draft and refines it with the same frozen student under the recorded controls ($r = 0.3$, four-step schedule), comparing against the draft and against from-noise controls.

B.3 User Study Protocol

We recruited 41 student volunteers to participate in the study. Each was shown a sequence of comparisons; in every comparison three clips appeared side by side, one from the four-step ForgeWM student, one from Matrix-Game 2.0, and one from HY-WorldPlay. All clips in a comparison share the same source initial state and recorded control trace; each model consumes that trace through its native or converted control interface described above. Model identities were hidden and the left-to-right placement was randomized independently for each comparison, so position carried no information about which system produced a clip.

Each participant judged five comparisons per criterion under three criteria, giving $41 \times 5 = 205$ selections per criterion and 615 in total. The criteria were described to participants as follows. *Visual Quality*: sharpness, naturalness, level of detail, and absence of artifacts. *Action Accuracy*: whether the motion in the clip follows the direction and the persistence of the control input. *Spatiotemporal Consistency*: whether scene identity and geometry are preserved over time, and whether the clip is free of flicker and warping. For each comparison and criterion the participant selected exactly one of the three clips; there was no tie option, which is why the three percentages for a criterion sum to 100%.

C CrossFPS Evaluation and Adaptation

C.1 Cross-Domain FPS Evaluation

We sample 25 clips from each of the seven games in the official CrossFPS evaluation split (Tong et al., 2026) with a fixed seed, giving 175 clips. Equal counts per game matter here: the released split is heavily unbalanced (its largest title carries roughly $65\times$ the clips of its smallest), so an unweighted mean over clips would report the largest title's score under a cross-game label. We therefore both sample equally and report a macro-average over games. Since the released archive does not separately label its validation and test partitions, the sample is drawn from the combined evaluation split and these values are not directly comparable to the benchmark's partition-specific published results.

Each clip is replayed under its own recorded gamepad trace. Generation and scoring follow the same protocol as the Minecraft experiments: 81 pixel frames at 640×352 from the clip's first frame, and paired LPIPS (AlexNet backbone) on 16 frames sampled across the rollout, excluding the given first frame. PSNR is computed and averaged over the same temporally aligned frame pairs. Reference footage is temporally resampled to the same frame count. Motion is measured as the mean dense optical-flow magnitude, the same estimator as the main table's flow column; *flow ratio* is the generated magnitude divided by the reference magnitude, so 1.0 means the rollout moves as much as the reference.

Game	LPIPS↓	PSNR↑	Flow	Ratio
Xonotic	0.5828	11.21	13.88	1.66
Modern Warfare III	0.6352	11.61	8.15	1.35
Modern Warfare	0.6479	10.99	8.29	1.46
Warzone	0.6695	10.66	9.13	1.78
Halo Infinite	0.6730	9.75	11.23	1.46
Halo	0.6920	9.31	13.28	1.29
Call of Duty	0.6933	9.67	10.40	1.16
Macro-average	0.6562	10.46	10.62	1.45

Table C1: **Per-game CrossFPS results,** 25 clips per game. Flow is the mean dense optical-flow magnitude in pixels; ratio is generated over reference, where 1.0 matches the reference motion scale.

Table C1 gives the breakdown. LPIPS spans 0.583–0.693 across titles. Flow ratios range from 1.16 to 1.78, with a macro-average of 1.45, indicating systematic over-response relative to the reference motion scale.

C.2 CrossFPS Adaptation: Action Module and Training

The CrossFPS checkpoint reported in the main paper follows the same four-stage training recipe and retains the same backbone and action-module topology. Adapting to gamepad control requires only an interface-level change: the continuous-control input width is increased from two to four channels, while the data and control semantics differ from the Minecraft setting. This section specifies that interface adaptation and the initialization of the added input channels.

Action schema. Minecraft control uses a two-dimensional mouse delta and a six-state keyboard vector. CrossFPS is driven by a gamepad, so the continuous channel widens from two to four dimensions – the (x, y) deflection of the left analog stick (movement) and the (x, y) deflection of the right analog stick (camera look) – while the discrete channel stays six-dimensional and is reinterpreted as the six gamepad buttons (right trigger / fire, left trigger / aim-down-sights, and the south / west / north / right-thumb buttons for jump, reload, weapon switch, and melee). In the model configuration, this changes only `mouse_dim_in` from 2 to 4. The backbone, action-module topology, and all downstream dimensions remain unchanged; only the width of the first continuous-control input projection is enlarged.

Where the change lands. Continuous controls enter through the first layer of the action module's mouse branch, a linear layer over the windowed, temporally-grouped control vector concatenated with the visual hidden state (the `mouse_mlp` input projection). Widening the continuous input from two to four channels enlarges only this input projection – from $1536 + 2 \times 4 \times 3 = 1560$ to $1536 + 4 \times 4 \times 3 = 1584$ input units (1536 visual hidden units plus `mouse_dim_in` $\times$ VAE temporal compression $\times$ window control units) – leaving every downstream weight shape unchanged.

Initialization of the new channels. A naive warm-start fails in two ways that we observed directly. First, the shape filter used for checkpoint loading discards the *entire* input projection when its input width changes, including the 1536 columns that read the visual hidden state and the columns for the two original control channels; a model re-initialized this way collapses to a "use the hidden state, ignore the control" solution and does not recover action controllability during training. Second, keeping the visual columns but *zero-initializing* the two new control channels leaves those channels without gradient signal at the small Stage 0 learning rate: with no input variation reaching a zeroed column, the symmetry is never broken and the two added channels stay effectively dead.

We therefore *graft* the input projection from the base checkpoint. The 1536 visual columns and the two original control channels are copied verbatim, and each of the two *new* channels is initialized by copying one of the pretrained control channels rather than from zero, so every channel begins from a trained-scale, symmetry-broken state that gradient descent can move. The zero-action-residual-at-initialization property is preserved not by the input projection but by the action module's *output* projection: the base checkpoint ships the mouse and keyboard output projections zeroed, so at the

first update the widened action branch still contributes no residual and the base video prior is left intact while the new interface is learned.

Training. Stage 0 adapts the grafted generator on the merged CrossFPS corpus (65,246 sharded clips across the seven titles) for 12,000 updates. Stage 1 initializes from this Stage 0 checkpoint, which already contains the four-channel continuous-control projection. Stage 2 then initializes from the resulting Stage 1 checkpoint, and Stage 3 initializes from Stage 2; no further grafting is required. Stages 1–3 otherwise follow the corresponding Minecraft objectives and optimization settings in Table A2 (20,000 teacher-forced updates, 6,000 consistency-distillation updates, and 4,000 Stage 3 trainer iterations). The four-step CrossFPS student uses the $[1000, 750, 500, 250]$ schedule and is evaluated in Table C1.

D Extended Qualitative Rollouts

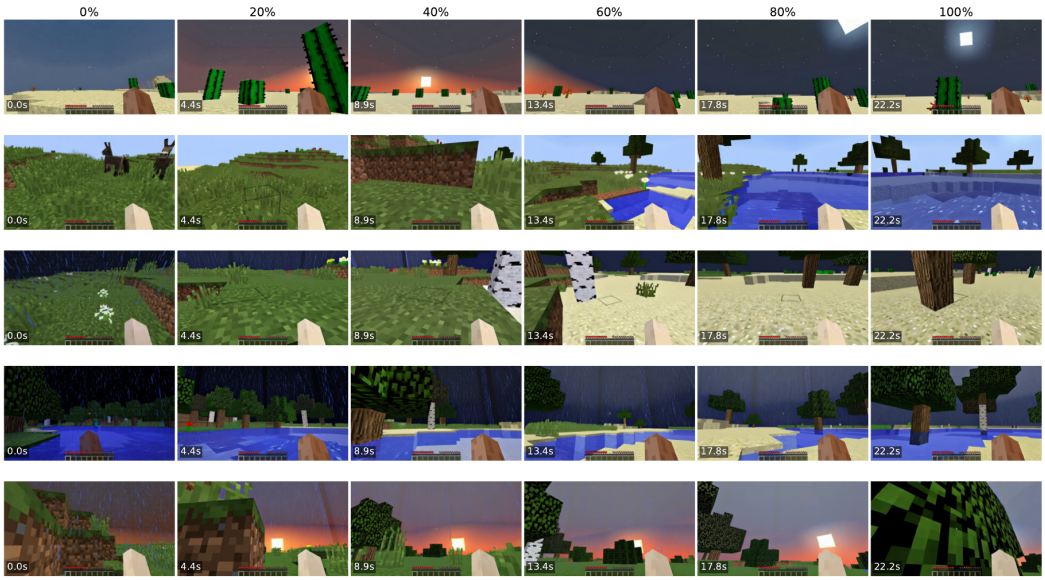

Figure D1: **Extended qualitative rollouts** from the four-step Minecraft checkpoint. Each row shows one 22 s causal rollout sampled at evenly spaced timestamps, with elapsed time marked per frame. Rows span biome and time of day and include both sustained forward motion and sustained turning.

Figure D1 illustrates the long-horizon degradation modes discussed in Appendix E.

E Limitations

We note three limitations that the results in this paper make visible. *Long-horizon drift.* The quantitative protocol evaluates a 77-frame window; the extended rollouts of Figure D1, roughly $3.5\times$ that horizon, show a slow loss of block structure and spreading color artifacts at later timestamps. Causalization and distillation reduce but do not eliminate the autoregressive accumulation of error, and we do not claim indefinite rollout stability. *Motion over-response.* On the cross-domain FPS split the generated flow magnitude exceeds the reference by a macro-average ratio of 1.45 (Table C1): the model tends to move more than the recorded control implies, so magnitude fidelity lags directional and persistence fidelity. *Budget-dependent stage value.* At the four-step budget, Stage 3 increases Imaging Quality but does not improve paired LPIPS relative to Stage 2. Its measured incremental effect at this budget is therefore metric-dependent; the stage-wise ablation does not isolate its incremental effect at one or two steps.