HeroWorld: Long-Horizon Action-Conditioned World Models for Third-Person Games

Mingjia Huo [1] **Kaiqin Kong** [1] **Minshen Zhang** [1] **Shaoxiong Duan** [1] **Junda Su** [1] **Will Lin** [1] **Hao Zhang** [1]

Abstract

Interactive world models turn video generation models into controllable simulators, but most game-based systems are developed in first-person settings where actions are expressed mainly through camera motion. Third-person control is more challenging because the model must jointly synthesize scene evolution, camera movement, visible character dynamics, and character-environment interaction. We introduce HeroWorld, a third-person action-conditioned world model adapted from a pretrained game video generator using curated human gameplay videos and agent-generated Minecraft rollouts. HeroWorld uses bidirectional finetuning, causal finetuning, and progressive horizon distillation, which gradually expands autoregressive rollout length to mitigate the train-short–test-long mismatch and enable stable 30-second controllable Zelda rollouts. We further study out-of-distribution transfer across unseen scenes, characters, and game domains, finding that robustness depends on preserving character-background separation and action-dependent interaction with surrounding geometry. Action-vector scaling, attention injection, and multi-game training improve robustness under domain shifts, offering practical guidance for building reliable third-person world models from pretrained video generators.

1. Introduction

Third-person interactive world models predict how a visual environment evolves in response to user actions, turning video generation models into controllable simulators. Unlike first-person models, where actions are often expressed through egocentric camera motion, third-person models must maintain a visible agent, its pose, and its contact with terrain and objects over long rollouts. This makes third-person control a useful stress test for long-horizon video generation: a forward action on flat ground, a downhill slope, or a climbing surface should induce different character-environment interactions rather than merely plausible motion.

Existing game-based world models are still dominated by first-person settings such as Minecraft (Decart & Etched, 2024; Savva et al., 2026) and shooter games (Valevski et al., 2024), where the controllable character is often off-screen. Recent general-domain models such as Hunyuan-GameCraft (Li et al., 2025), HY-WorldPlay (Sun et al., 2025), and Matrix-Game (Wang et al., 2026) include third-person gameplay, but our evaluation shows that their third-person behavior remains unstable, often losing character-background separation or character-environment interaction. Our work targets this gap and studies third-person action following and generalization directly.

We introduce HeroWorld, a third-person action-conditioned world model trained on human gameplay videos curated from NitroGen (NVIDIA, 2025) and agent-generated Minecraft rollouts (Fan et al., 2022). To adapt existing video world models to third-person games, we use a three-stage training pipeline: bidirectional finetuning adapts the visual distribution and trains the keyboard action module; causal finetuning with Diffusion Forcing enables streaming generation under block-causal attention; and progressive horizon distillation gradually expands the autoregressive rollout length during distillation. Unlike fixed-horizon distillation, this curriculum exposes the student to increasingly long self-generated contexts, helping it preserve action-conditioned motion, character identity, and scene geometry over extended rollouts. For out-of-domain (OOD) transfer, we further study action-vector scaling, attention injection, and multi-game training as lightweight action-enhancement techniques for improving robustness under unseen third-person scenes and characters.

We evaluate HeroWorld through long-horizon and OOD rollout studies. On Zelda, HeroWorld sustains 30-second rollouts across diverse behaviors, and progressive horizon distillation reduces appearance drift and improves action

[1]University of California San Diego. Correspondence to: Hao Zhang <haozhang@ucsd.edu>.

Proceedings of the 43^{rd} International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026.

following over vanilla Self Forcing and fixed-horizon long tuning. OOD rollouts reveal failures beyond ordinary video drift: actions must be grounded in visible pose, contact, terrain constraints, and surrounding geometry. Multi-game training and action-enhancement techniques improve robustness under these shifts.

In summary, our contributions are threefold:

1. We study long-horizon generalization for third-person interactive world models, a setting where action following requires visible character-environment interaction rather than camera motion alone.

2. We introduce HeroWorld, an action-conditioned world model trained on five third-person game domains with a three-stage training recipe combining bidirectional finetuning, Diffusion Forcing, and progressive horizon distillation.

3. We provide a fine-grained generalization study across OOD scenes, showing that third-person control benefits from multi-domain training and action-enhancement techniques.

2. Related Work

Game-based world models. Games provide action-conditioned video data with controllable dynamics, but many early world models focus on first-person or low-dimensional settings, including DOOM (Valevski et al., 2024), Atari (Alonso et al., 2024), and Minecraft (Decart & Etched, 2024; Yu et al., 2025; Savva et al., 2026). In these systems, the player body is often off-screen and action following reduces largely to camera motion. Recent foundation-style game models (Google DeepMind, 2025; Ye et al., 2025; Zhang et al., 2025; He et al., 2025; Xiang et al., 2025; Zhu et al., 2025; Mao et al., 2025; Li et al., 2025; Sun et al., 2025; Che et al., 2024) broaden game coverage and horizon length, but third-person character-centric control remains less systematically studied.

Long-horizon video generation. Autoregressive video generation and streaming distillation extend video models beyond short clips by generating under causal context or distilling bidirectional teachers into streaming students (Chen et al., 2024; Yin et al., 2025; Huang et al., 2025; Yang et al., 2025). Other work uses attention, positional, or memory mechanisms to extend rollouts (Liu et al., 2025; Xiao et al., 2025). These methods are usually validated where prompts or motions change smoothly; we adapt them to third-person gameplay, where action errors accumulate through visible character pose, contacts, terrain interaction, and scene geometry rather than only through egocentric motion.

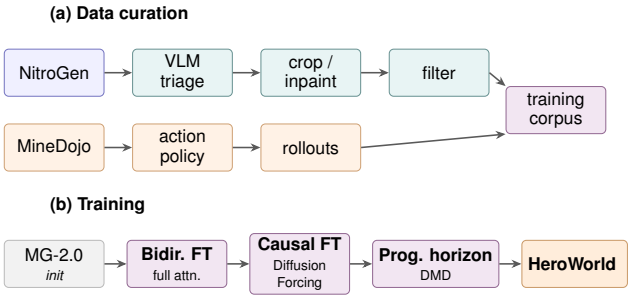

Figure 1. **Data and training pipeline.** Top: real gameplay is cleaned and filtered, while simulator rollouts provide action-labeled trajectories. Bottom: Matrix-Game 2.0 is adapted with bidirectional finetuning, causal finetuning, and progressive horizon distillation.

3. Method

3.1. Data Curation Pipeline

HeroWorld is trained on two complementary sources: curated third-person gameplay from NitroGen (NVIDIA, 2025) and procedurally generated third-person Minecraft trajectories from MineDojo (Fan et al., 2022). A vision–language model routes real clips through crop or inpaint cleanup, then a button-aware filter selects continuous-control segments for VAE encoding. This yields **Zelda-Main** (233,008 81-frame segments; 174.8 hours) and **Multi3D** (13,551 segments; $\sim$10.2 hours) for cross-game generalization. The simulator branch complements these videos with clean action-labeled rollouts that cover rare interaction cases. We use Zelda-Main for bidirectional pretraining, Multi3D for cross-game studies, and a Minecraft subset for character–environment OOD studies; details are in App. A.

3.2. Training

HeroWorld is trained in three stages, summarized in Fig. 1: bidirectional finetuning of a pretrained checkpoint, causal finetuning with Diffusion Forcing to enable streaming inference, and a progressive horizon distillation that aligns the streaming model with long-rollout test conditions.

3.2.1. Bidirectional and Causal Finetuning

We build HeroWorld on Matrix-Game 2.0 (He et al., 2025), re-initialize the keyboard action module, and keep the 3D VAE frozen. Following Solaris (Savva et al., 2026), bidirectional finetuning first adapts the visual distribution under full attention, and causal finetuning then applies Diffusion Forcing (Chen et al., 2024) with independent noise per chunk and a block-causal sliding-window mask. We skip the ODE-regression initialization used in CausVid (Yin et al., 2025), as simple causal finetuning is sufficient for Self Forcing in this setting.

3.2.2. Progressive Horizon Training

Our long-horizon distillation builds on LongLive's streaming long tuning (Yang et al., 2025): the student generates a chunk, reuses detached history and KV cache as causal context, and receives distillation loss only on the new chunk. We modify this with a *progressive horizon* curriculum for third-person control. Instead of distilling immediately at the final long horizon, we start from the native short horizon of 9 latents and add 6 latents per stage until reaching 39 or 81 latents. This exposes the model to self-generated contexts while avoiding early training on highly OOD long-rollout states, where character identity, scene geometry, and action response can fail before the short-horizon policy has stabilized.

3.3. Methods for Enhancing Action Following

We study three drop-in techniques that improve action following, particularly for OOD characters where the model otherwise produces a static subject under camera-only motion. Results are reported in Tab. 2.

Action scaling. We multiply the keyboard action vector by $K{=}3$ before cross-attention, amplifying the action's contribution in the attention readout while leaving the camera and image conditions unchanged, since camera actions do not need to decouple the character from the background.

Attention injection. Following FREE-Edit (Li et al., 2026) and AnyV2V (Ku et al., 2024), we run a source rollout (Zelda first frame) and an edit rollout (OOD first frame) under identical noise, action, and camera. For the first $\tau{=}2$ denoise steps, we copy the source self-attention Q, K *inside* the character bounding box of the edit branch. We additionally enable an attention sink of size 1 (Xiao et al., 2024) so that the block-0 source-derived K remains in every later causal block's local-attention window, instead of being squeezed out by the sliding window.

Multi-game training. We finetune on a small mixture of non-Zelda gameplay chunks alongside the Zelda data, sharing the action vocabulary through the keyboard module from Sec. 3.2.1. Same keyboard semantics under different visual priors forces the model to decouple action semantics from a single character identity.

3.4. Quality Trade-Off under Limited Domain Diversity

Multi-game training improves OOD action grounding by exposing the same keyboard semantics under different visual priors. Since our current mixture still contains only a few games, it may also bias long rollouts toward training-domain character and background styles. Thus, we use it as a lightweight action-enhancement strategy, while recognizing that broader domain coverage is needed for stronger visual consistency.

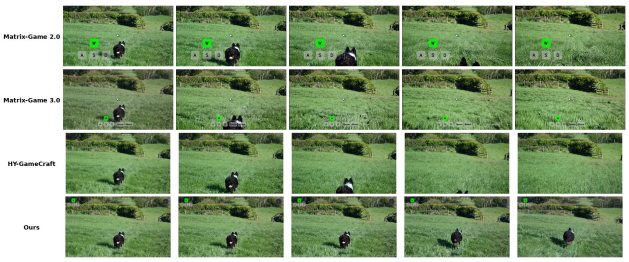

Figure 2. **Third-person control comparison under W-only input.** Rows are, from top to bottom: Matrix-Game 2.0, Matrix-Game 3.0, Hunyuan-GameCraft, and HeroWorld . Baselines often collapse control into camera motion, treating the dog as background until it disappears, while HeroWorld preserves character-background separation and action-conditioned character motion.

4. Experiments

4.1. Setup

Baselines and metrics. We compare with publicly released video world models, including Matrix-Game 2.0 (He et al., 2025), Matrix-Game 3.0 (Wang et al., 2026), and Hunyuan-GameCraft (Li et al., 2025). Our evaluation uses DPFlow (Morimitsu et al., 2025) EPE and Fl-all (Geiger et al., 2012) for action following against matched gameplay, synthetic-flow cosine for OOD prompts without reference videos, FVD-CLIP for temporal-window distributional fidelity, and VBench-Quality (Huang et al., 2024) for visual quality and consistency. Column arrows indicate whether higher or lower is better.

Evaluation set. We evaluate on 64 Zelda gameplay scenes, each 801 frames at 25 fps with aligned actions and diverse behaviors including climbing, gliding, running, standing, swimming, and walking. Each method conditions on the first frame and rolls out the remaining 800 frames using the recorded action stream.

4.2. Long-Horizon Performance

Temporal windows and methods. We split each rollout into 48 non-overlapping 16-frame clips and report Short, Mid, and Long windows covering frames 0–255, 256–511, and 512–767. We compare five checkpoints with the same backbone, optimizer, and dataset: SF, fixed-horizon LongLive at $39/81$ latents, and our progressive variants extending 9 latents to $39/81$.

Results. Tab. 1 shows that distillation reduces appearance and action drift relative to SF, especially in the Mid and Long windows. At equal target context, progressive distil-

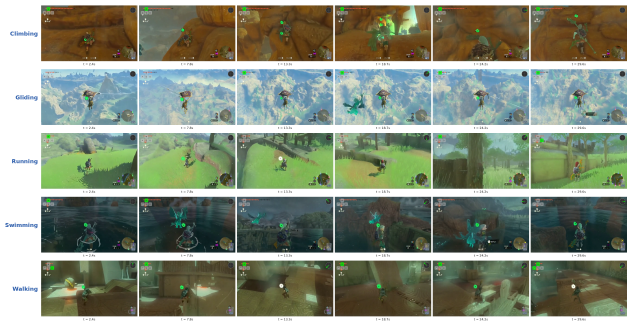

Figure 3. **Action following and long-horizon consistency.** Three 32-second generations from **Ours** (9 → 39) with input indicators overlaid.

Table 1. **Long-horizon Zelda evaluation.** Metrics are aggregated over 64 generated 32-second rollouts.

	FVD-CLIP ↓			Action ↓		VBench ↑
Method	Short	Mid	Long	EPE	Fl-all	Quality
SF (vanilla)	4.49	6.87	7.10	5.54	0.786	0.7906
LongLive (39-latent)	3.30	4.26	4.78	4.70	0.782	0.8007
LongLive (81-latent)	4.09	4.85	5.38	5.22	0.786	0.8042
Ours (9 → 39)	**3.20**	**4.16**	**4.48**	**4.55**	**0.776**	0.7941
Ours (9 → 81)	3.95	4.76	5.08	5.23	0.798	**0.8098**

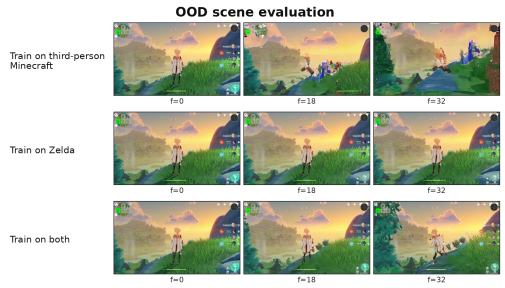

Figure 4. **Training games and OOD scene generalization.** Joint Minecraft+Zelda training better balances action response, visual quality, and terrain-aware interaction.

lation improves the appearance- and consistency-focused metrics: **Ours** (9 → 39) wins all FVD-CLIP windows and both action metrics, while **Ours** (9 → 81) wins FVD-CLIP over LongLive (81) and the VBench aggregate. The complete table, including FVD-I3D, is in App. 6.

Fig. 3 illustrates the qualitative behavior: held inputs produce the expected visual response while scene composition and subject identity remain coherent through t=32 s.

4.3. Generalization

Third-person control requires character-environment interaction. We analyze OOD rollouts across Minecraft, Zelda, and their union. In third-person control, an action stream must drive visible pose, contacts, terrain constraints, and interaction with surrounding geometry; in Fig. 4, correct action following means moving left while descending along the hill, not flat image translation. Minecraft-only training reacts to the action but breaks character quality and physics, Zelda-only training is stable but under-reacts, and joint training best preserves controllability, video quality, and terrain-aware motion.

Table 2. **Action enhancement techniques.** Results on 88 OOD synthetic-action prompts.

	Action	VBench ↑
Method	OOD cos ↑	Quality
Zelda Training	0.145	0.7731
Action Scaling	0.450	0.8299
NitroGen 4-game Training	0.250	0.7911
NitroGen + Minecraft Training	**0.772**	**0.8521**

Table 3. **Attention injection for OOD character transfer.** Vanilla denotes i2v on the composited first frame without injection.

	Action	VBench ↑	
Method	OOD cos ↑	Dyn. Deg.	Quality
Vanilla	0.263	0.025	0.7102
+ Attention inj.	**0.362**	**0.200**	**0.7258**

Multi game training improves OOD generalization. We compare Zelda-only training, action scaling, additional NitroGen games, and Zelda+Minecraft simulator rollouts under the same backbone, optimizer, and total token budget. Tab. 2 shows that action scaling sharpens control without new data, while synthetic Minecraft rollouts give the largest gain, likely because they cover rare behaviors such as walking into walls or stepping off ledges. Tab. 3 further shows that attention injection improves OOD character transfer, with the gain driven mainly by dynamic degree.

5. Conclusion

We introduced HeroWorld, a third-person action-conditioned world model for long-horizon interactive game simulation. Unlike first-person settings where control can be expressed mainly through camera motion, this setting requires persistent character identity, visible action response, and character-environment interaction. Our three-stage training pipeline improves 30-second Zelda rollouts, while OOD studies show that failures often involve character-background separation and terrain-aware interaction rather than generic video drift. Action scaling, attention injection, and multi-game training improve robustness under these shifts, offering practical guidance for reliable third-person world models.

References

Alonso, E., Jelley, A., Micheli, V., Kanervisto, A., Storkey, A., Pearce, T., and Fleuret, F. Diffusion for world modeling: Visual details matter in atari. *Advances in Neural Information Processing Systems*, 2024.

Che, H., Pan, X., Lin, H., Liu, Y., Niu, Q., Zhu, T., Zhang, T., Liu, W., Wei, Y., Hu, J., et al. Gamegen-x: Interactive open-world game video generation. *arXiv preprint arXiv:2411.00769*, 2024.

Chen, B., Martí Monsó, D., Du, Y., Simchowitz, M., Tedrake, R., and Sitzmann, V. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024.

Decart and Etched. Oasis: A universe in a transformer, 2024. https://oasis-model.github.io/.

Fan, L., Wang, G., Jiang, Y., Mandlekar, A., Yang, Y., Zhu, H., Tang, A., Huang, D.-A., Zhu, Y., and Anandkumar, A. MineDojo: Building open-ended embodied agents with internet-scale knowledge. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2022.

Geiger, A., Lenz, P., and Urtasun, R. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

Google DeepMind. Genie 3: A new frontier for world models, 2025. https://deepmind.google/discover/blog/genie-3-a-new-frontier-for-world-models/.

HaCohen, Y., Chiprut, N., Brazowski, B., Shalem, D., Moshe, D., Richardson, E., Levin, E., Shiran, G., Zabari, N., Gordon, O., Panet, P., Weissbuch, S., Kulikov, V., Bitterman, Y., Melumian, Z., and Bibi, O. LTX-Video: Realtime video latent diffusion, 2024. URL https://arxiv.org/abs/2501.00103.

He, X., Peng, C., Liu, Z., Wang, B., Zhang, Y., Cui, Q., Kang, F., Jiang, B., An, M., Ren, Y., et al. Matrix-game 2.0: An open-source real-time and streaming interactive world model. *arXiv preprint arXiv:2508.13009*, 2025.

Huang, X., Li, Z., He, G., Zhou, M., and Shechtman, E. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *arXiv preprint arXiv:2506.08009*, 2025.

Huang, Z., He, Y., Yu, J., Zhang, F., Si, C., Jiang, Y., Zhang, Y., Wu, T., Jin, Q., Chanpaisit, N., et al. VBench: Comprehensive benchmark suite for video generative models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Ku, M., Wei, C., Ren, W., Yang, H., and Chen, W. AnyV2V: A tuning-free framework for any video-to-video editing tasks. *Transactions on Machine Learning Research (TMLR)*, 2024. arXiv:2403.14468.

Li, J., Yang, J., Liu, Z., Yang, Y., Wang, S., Zhao, Y., Cao, P., Liu, Y., Zou, X., Wang, J., et al. Hunyuan-gamecraft: High-dynamic interactive game video generation with hybrid history condition. *arXiv preprint arXiv:2506.17201*, 2025.

Li, M., Liu, Y., Li, Y., et al. FREE-Edit: Using editing-aware injection in rectified flow models for zero-shot image-driven video editing. *arXiv preprint arXiv:2603.01164*, 2026.

Liu, K., Hu, W., Xu, J., Shan, Y., and Lu, S. Rolling forcing: Autoregressive long video diffusion in real time. *arXiv preprint arXiv:2509.25161*, 2025.

Mao, X., Li, Z., Li, C., Xu, X., Ying, K., He, T., Pang, J., Qiao, Y., and Zhang, K. Yume-1.5: A text-controlled interactive world generation model. *arXiv preprint arXiv:2512.22096*, 2025.

Morimitsu, H., Zhu, X., Cesar Jr., R. M., Ji, X., and Yin, X.-C. DPFlow: Adaptive optical flow estimation with a dual-pyramid framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. URL https://arxiv.org/abs/2503.14880.

NVIDIA. NitroGen: Large-scale gameplay dataset, 2025. URL https://huggingface.co/datasets/nvidia/NitroGen. Hugging Face dataset.

Samuel, D., Levy, M., Darshan, N., Chechik, G., and Ben-Ari, R. OmnimatteZero: Training-free video matting and compositing via latent diffusion models, 2025. URL https://arxiv.org/abs/2503.18033.

Savva, G., Michel, O., Lu, D., Waiwitlikhit, S., Meehan, T., Mishra, D., Poddar, S., Lu, J., and Xie, S. Solaris: Building a multiplayer video world model in minecraft. *arXiv preprint arXiv:2602.22208*, 2026.

Sun, W., Zhang, H., Wang, H., Wu, J., Wang, Z., Wang, Z., Wang, Y., Zhang, J., Wang, T., and Guo, C. Worldplay: Towards long-term geometric consistency for real-time interactive world modeling. *arXiv preprint arXiv:2512.14614*, 2025.

Valevski, D., Leviathan, Y., Arar, M., and Fruchter, S. Diffusion models are real-time game engines. *arXiv preprint arXiv:2408.14837*, 2024.

Wang, Z., Liu, Z., Li, J., Huang, K., Xu, B., Kang, F., An, M., Wang, P., Jiang, B., Wei, Y., et al. Matrix-game 3.0:

Real-time and streaming interactive world model with long-horizon memory. *arXiv preprint arXiv:2604.08995*, 2026.

Xiang, J., Gu, Y., Liu, Z., Feng, Z., Gao, Q., Hu, Y., Huang, B., Liu, G., Yang, Y., Zhou, K., et al. Pan: A world model for general, interactable, and long-horizon world simulation. *arXiv preprint arXiv:2511.09057*, 2025.

Xiao, G., Tian, Y., Chen, B., Han, S., and Lewis, M. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations (ICLR)*, 2024.

Xiao, Z., Lan, Y., Zhou, Y., Ouyang, W., Yang, S., Zeng, Y., and Pan, X. Worldmem: Long-term consistent world simulation with memory. *arXiv preprint arXiv:2504.12369*, 2025.

Yang, S., Huang, W., Chu, R., Xiao, Y., Zhao, Y., Wang, X., Li, M., Xie, E., Chen, Y., Lu, Y., et al. Longlive: Real-time interactive long video generation. *arXiv preprint arXiv:2509.22622*, 2025.

Ye, D., Zhou, F., Lv, J., Ma, J., Zhang, J., Lv, J., Li, J., Deng, M., Yang, M., Fu, Q., et al. Yan: Foundational interactive video generation. *arXiv preprint arXiv:2508.08601*, 2025.

Yin, T., Zhang, Q., Zhang, R., Freeman, W. T., Durand, F., Shechtman, E., and Huang, X. From slow bidirectional to fast autoregressive video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22963–22974, 2025.

Yu, J., Qin, Y., Wang, X., Wan, P., Zhang, D., and Liu, X. Gamefactory: Creating new games with generative interactive videos. *arXiv preprint arXiv:2501.08325*, 2025.

Zhang, Y., Liu, C. P. Z., Wang, B., He, X., Cui, Q., Kang, F., Jiang, B., An, M., Ren, Y., et al. Matrix-game: Interactive world foundation model. *arXiv preprint arXiv:2506.18701*, 2025.

Zhu, Y., Feng, J., Zheng, W., Gao, Y., Tao, X., Wan, P., Zhou, J., and Lu, J. Astra: General interactive world model with autoregressive denoising. In *arXiv preprint arXiv:2512.08931*, 2025.

A. Data Curation

This appendix expands the data pipeline summarized in Sec. 3.1. We give per-set statistics, then detail the three non-trivial stages: VLM bbox triage, OmnimatteZero inpainting, and 81-frame button-aware filtering.

A.1. Dataset Statistics

Tab. 4 summarizes the two real-data sets curated through the pipeline. Zelda-Main is the bidirectional pretraining corpus; Multi3D is a cross-game generalization set spanning three additional third-person games processed through the same cut/inpaint pipeline.

A.2. Asset Licenses and Terms

We credit the datasets, models, tools, and evaluation assets used in this work through citations. NitroGen releases action annotations and metadata for publicly available gameplay videos under CC BY-NC 4.0; the underlying gameplay videos remain subject to the original video-platform terms and relevant rightsholders. Our Minecraft simulator rollouts use MineDojo, while Minecraft assets remain subject to Mojang/Microsoft terms. We use or evaluate against external models, baselines, and evaluation tools under the licenses and terms specified in their respective repositories.

A.3. VLM-Annotated bbox Triage

Each candidate clip's first frame is fed to a vision–language model (GPT-4o-mini or Gemini, identical prompt) that returns (i) a bbox enclosing the gameplay viewport and (ii) a triage label in {`cut`, `inpaint`, `drop`}. We sample multiple chunks per video and aggregate via consensus: a video-level bbox is accepted when the sampled bboxes agree to IoU ≥ 0.95 *or* maximum coordinate delta ≤ 8 px, with up to one outlier removed; the accepted bbox is the median over the consistent cluster. For chunk-level robustness we resample each clip's bbox until two consistent samples are observed.

A.4. Inpainting with OmnimatteZero

For `inpaint`-classified clips we remove persistent overlays (HUD, face cam, controller widgets, streamer branding) using OmnimatteZero (Samuel et al., 2025), a training-free video matting and object-removal method that exploits the prior of a pretrained latent video diffusion model. We use `a-r-r-o-w/LTX-Video-0.9.7-diffusers` (HaCohen et al., 2024) as the backbone, run inference in `bf16`, and use the VLM-annotated overlay bboxes as the foreground mask. OmnimatteZero performs latent-space attention guidance during inversion and resampling to produce a clean background plate, which we composite back into the original chunk. Because no per-video training is required,

Set	Games	81-frame segments	Hours @ 30 fps
Zelda-Main	1 (Zelda)	233,008	174.8
Multi3D	3	13,551	∼10.2

Table 4. Real-data sets produced by our curation pipeline. Multi3D spans Horizon Forbidden West, Dark Souls Remastered, and Code Vein.

the backend trivially scales to new games.

A.5. Filtering: 81-Frame No-Near-Button

The filter takes a per-chunk frame-aligned action stream and returns a list of fixed-length 81-frame segments. The procedure is:

1. Compute the set of *event* frames $E = \{i :$ any controller button is pressed at frame $i\}$.
2. Form the *truncated* set $T = \{j : \exists i \in E, |j - i| \leq 5\}$, i.e., the 11-frame window around each button press.
3. A frame i is *clean* iff its button list is empty *and* $i \notin T$.
4. Maximal runs of clean frames are split into non-overlapping segments of length exactly 81.

The ± 5-frame mask prevents button-driven discrete events (jump, attack, equip) from contaminating segments that the model is meant to interpret as continuous joystick-driven motion. The 81-frame length matches the FastVideo / MatrixGame schema (one VAE latent block at the $4\times$ temporal compression of Wan2.1).

B. Training Details

This appendix expands the three-stage training pipeline (Fig. 1). All stages share the same backbone, optimizer family, and resolution; they differ in objective, learning rate, horizon, and starting checkpoint.

B.1. Backbone and Optimization

We build HeroWorld on Matrix-Game 2.0 (He et al., 2025). The 3D VAE is frozen throughout; the keyboard action module is re-initialized in Stage 1 and inherited in later stages. All stages train in `bf16` with AdamW (β_1=0.9, β_2=0.95, weight decay 0, max grad norm 1), at 480×832 resolution on 32 NVIDIA H100 GPUs under HSDP (shard dim 32, replicate dim 1) with per-GPU batch size 1. Hyperparameters that differ across stages are summarized in Tab. 5.

B.2. Stage 1: Bidirectional Finetuning

We finetune the full network with full attention and a shared per-frame noise level on Zelda-Main (App. A.1) for 60,000 steps. The starting weights are a 20k-step warmup checkpoint trained on a Zelda+Minecraft mixture; this stage adapts the model to our games and trains the freshly initialized keyboard action module. We checkpoint every 10k steps and select the best snapshot by the optical-flow validation criterion of App. C.3 for the next stage.

B.3. Stage 2: Causal Finetuning with Diffusion Forcing

Initialized from Stage 1, we run Self Forcing pretraining with Diffusion Forcing under a block-causal sliding-window mask, sampling independent noise levels per chunk. Following Solaris (Savva et al., 2026) we skip the ODE-regression initialization. Stage 2 runs for 6,000 steps with a $10\times$ smaller learning rate (3×10^{-6}), uses a stronger EMA (0.99) for stable streaming, and turns off CFG dropout (`training_cfg_rate`=0) since the goal is to align the student with the bidirectional teacher's marginal under matched action conditions. We keep the same 33-frame / 9-latent context as Stage 1.

B.4. Stage 3: Progressive Horizon Distillation

Stage 3 runs DMD (Yin et al., 2025) under the streaming long-tuning recipe of LongLive (Yang et al., 2025) with our progressive-horizon curriculum (Sec. 3.2). Each substage initializes from the best checkpoint of the previous substage and increases the maximum streaming horizon. We run substages at T=9 (matching the SF pretrain context) and T=21 (corresponding to the 81-frame long horizon). DMD-specific settings include a generator-to-fake-score update ratio of 1:5, real-score guidance scale 3.5, CFG dropout 0.1, and a min/max timestep ratio of $0.2/0.98$. Each substage budgets up to 60,000 DMD steps but is early-stopped when the top-5 checkpoint selector (by held-out FVD-CLIP / EPE) plateaus; for our final long-horizon model the selected checkpoint sits in the steps 22,500–26,500 window of the final substage.

B.5. Action-Enhancement Variants

For the action-vector scaling experiments (Sec. 3.3), we multiply the keyboard action vector by $K\in\{1, 3, 5\}$ before cross-attention; we refer to these variants as `kb`K. The attention-injection variant of Sec. 3.3 requires no additional training: at inference we run the source rollout on the original Zelda first frame and the edit rollout on the

	Stage 1 (Bidir. FT)	Stage 2 (SF / DF)	Stage 3 (DMD)
Init. weights	`zelda_mc_warmup_20k`	Stage 1 best ckpt	Stage 2 best ckpt
Objective	flow matching	Self Forcing (Huang et al., 2025)	DMD (Yin et al., 2025)
Attention	full / shared noise	block-causal / DF (Chen et al., 2024)	block-causal streaming
Frames per sample	33	33	$33 \rightarrow 81$ (progressive)
Latent length T	9	9	$9 \rightarrow 21$ (progressive)
Frames per block	3	3	3
Learning rate	2×10^{-5}	3×10^{-6}	2×10^{-5}
LR schedule	constant, 10-step warmup	constant	constant
Max train steps	60,000	6,000	60,000 (best $\leq$32k)
EMA decay	0.9999	0.99	0.999
CFG dropout rate	—	0	0.1
Real-score guidance	3.5	3.0	3.5
Generator/fake ratio	—	—	1:5
Validation guidance	1.0	6.0	6.0

Table 5. Per-stage training hyperparameters. Latent length T refers to `num_latent_t` (the number of temporal latent tokens after the Wan2.1 VAE's $4\times$ temporal compression); T=9 corresponds to a 33-frame clip and T=21 to an 81-frame clip. Stage 3 expands T progressively across substages.

Figure 5. **Additional third-person control comparison under W-only input.** Rows are, from top to bottom: Matrix-Game 2.0, Matrix-Game 3.0, Hunyuan-GameCraft, and HeroWorld . As in Fig. 2, prior methods often fail to separate the controllable character from the background, so the requested forward motion is absorbed by camera/background motion and the character fades or deforms. HeroWorld better preserves character-background separation and action-conditioned character motion.

OOD first frame under matched noise, then copy the source self-attention Q, K inside the character bbox for the first τ=2 denoise steps with attention-sink size 1 (Xiao et al., 2024).

C. Evaluation Metrics

C.1. Complete Long-Horizon Results

C.2. Synthetic Optical Flow

The flow metrics in Sec. 4.1 compare generated rollouts against paired ground-truth videos via DPFlow. Out-of-distribution evaluation drives the model with synthetic action streams (e.g. "hold W for 32 s") for which no paired GT video exists, so this recipe is unavailable. We instead replace the GT side with a flow field *predicted analytically from the action stream* and apply the identical mf-cos / mf-ang / EPE computation. OOD scores are therefore on the same scale as the in-distribution numbers up to the bias introduced by the analytic predictor, which we discuss below.

Predicted flow. We use a third-person camera-kinematics model with no depth dependence. Eight scalar parameters are fit per game: gains $\alpha_{\text{yaw}}, \alpha_{\text{pitch}}, \alpha_{\text{turn}}$ that map mouse and turn-key input to camera angular velocity ω, gains $\beta_{\text{fwd}}, \beta_{\text{strafe}}$ that map W/S/A/D into avatar translation $\mathbf{T}_{\text{avatar}}$, focal length f, and a rigid pivot offset $\mathbf{r} = (0, r_y, r_z)$ from the optical center to the avatar. Treating the orbit camera as a body rotating about the pivot gives $\mathbf{T}_{\text{total}} = \mathbf{T}_{\text{avatar}} - \omega \times \mathbf{r}$. With depth fixed at Z=1 the per-pixel flow follows the Longuet-Higgins linearization:

$$u = \tfrac{xy}{f}\omega_x - \Big(f + \tfrac{x^2}{f}\Big)\omega_y - f\,T_x + x\,T_z,$$
$$v = \Big(f + \tfrac{y^2}{f}\Big)\omega_x - \tfrac{xy}{f}\omega_y - f\,T_y + y\,T_z,$$

with image-plane coordinates (x, y) centered at the principal point. Discarding depth collapses the parallax magnitude on translation but preserves the radial *direction* of forward motion and the angular structure of every rotation, so direction-based metrics (mf-cos, mf-ang) remain meaningful while pixel-EPE picks up a parallax bias. We therefore lean on mf-cos as the primary OOD signal and do not report OOD pixel-EPE.

Calibration. The eight parameters are fit once per game on a held-out set of in-distribution clips by minimising RMSE between predicted and DPFlow-extracted mean flow over per-clip segments, with sign-stable initialisation on β_{fwd} and β_{strafe}.

Analysis. This proxy is intended to score whether the generated video moves in the direction implied by the action

Table 6. **Complete long-horizon Zelda evaluation.** This table expands Tab. 1 with FVD-I3D.

	FVD-CLIP ↓			FVD-I3D ↓			Action ↓		VBench ↑
Method	Short	Mid	Long	Short	Mid	Long	EPE	Fl-all	Quality
SF (vanilla)	4.49	6.87	7.10	360.7	506.1	528.2	5.54	0.786	0.7906
LongLive (39-latent)	3.30	4.26	4.78	**294.9**	434.6	504.7	4.70	0.782	0.8007
LongLive (81-latent)	4.09	4.85	5.38	298.5	**385.4**	**411.8**	5.22	0.786	0.8042
Ours ($9 \rightarrow 39$)	**3.20**	**4.16**	**4.48**	313.2	459.0	545.5	**4.55**	**0.776**	0.7941
Ours ($9 \rightarrow 81$)	3.95	4.76	5.08	303.0	390.2	460.5	5.23	0.798	**0.8098**

stream, not whether it exactly matches a missing ground-truth rollout. It is therefore most informative for controlled OOD prompts such as sustained W/A/S/D inputs, where the expected camera/avatar motion has a stable sign and can be summarized by mean-flow direction. The same simplification also explains its limits: because the predictor has no scene depth, collision state, or terrain geometry, it cannot decide whether a character should step around an obstacle, slide along a wall, or descend a slope. We use these synthetic-flow scores as a coarse action-following signal and pair them with visual inspection and VBench-Quality, which capture character identity, background stability, and other video-quality failures not represented by the analytic flow field.

C.3. Metrics as Stopping Criteria

Standard training-loss curves are a poor stopping criterion for action-conditioned world models: the per-step regression loss decreases smoothly long before the model has learned to actually *follow* actions, and a model that has converged on the loss can still produce action-agnostic camera dynamics. We instead monitor a held-out validation set with two optical-flow signals at every checkpoint, computed by DPFlow (Morimitsu et al., 2025) between generated and ground-truth rollouts:

- **Mean-flow cosine similarity** (mf-cos↑): cosine similarity between the spatially-averaged flow vectors of generated and reference videos. Captures whether the model moves *in the correct direction*, robust to per-pixel magnitude error.
- **Mean-flow angle error** (mf-ang↓, in degrees): angular distance between the same averaged flow vectors. Provides a complementary, magnitude-independent error signal.

These two are markedly less noisy than per-pixel EPE on a 96-prompt validation set: a single divergent prompt can spike pixel-EPE by $10\times$ while leaving cosine similarity intact, so we use mf-cos as the primary stop criterion and break ties with mf-ang.

Observation. Fig. 6 plots both metrics over Stage 1 (bidirectional finetuning on Zelda-Main) for the run that produced our released checkpoint. The trajectory has three phases: (a) a near-flat plateau through the first $\sim$30k steps where the model is still adapting to the keyboard module and shows essentially no action-following signal; (b) a slow, monotonic improvement from $\sim$30k to $\sim$55k; and (c) a final sharp gain in the last $\sim$5k steps, ending at mf-cos 0.593 and mf-ang 40.7° at step 60k. Stopping earlier (e.g. at the 20k or 40k checkpoint) would have shipped a model whose generations look visually plausible but follow actions noticeably worse.

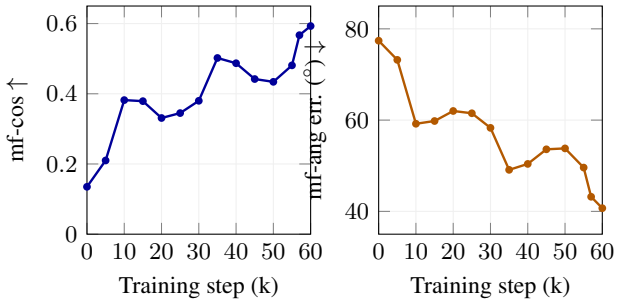

Figure 6. **Stage 1 stopping criteria.** Mean-flow cosine similarity (left, higher is better) and mean-flow angle error (right, lower is better) on the held-out 96-prompt Zelda validation set, as a function of bidirectional-finetuning step. Both curves are nearly flat through $\sim$30k steps and improve sharply only in the final $\sim$10k. We therefore train Stage 1 for the full 60k steps and select the checkpoint by mf-cos.

Stop criterion in practice. For Stage 1 we run a fixed budget of 60k steps and pick the argmax-mf-cos checkpoint as the input to Stage 2. For Stage 2 (Self Forcing) and Stage 3 (DMD) we additionally use validation FVD-CLIP and EPE on the long-horizon evaluation set; in both cases the streaming model converges much faster (best Stage 2 checkpoint by step $\sim$2–4k, best Stage 3 substage checkpoints in the 22,500–26,500 window). VBench-Quality is computed offline at submission time and is not used as a live stopping signal.