ONLINEWM: CAUSALITY-AWARE ACTIVE ONLINE LEARNING FOR EFFECTIVE WORLD MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative world models aim to predict future states conditioned on actions, where action controllability is fundamental for reliable dynamics modeling. While recent efforts leverage simulator-generated data to enhance this capability, existing training pipelines face two fundamental limitations. First, static offline data collection leads to a distribution misalignment between training sets and the model's evolving error patterns, failing to resolve critical *long-tail* scenarios where dynamics predictions remain unreliable. Second, the standard objective of minimizing observational discrepancy often encourages the model to exploit spurious correlations instead of capturing the underlying action-effect causality. To address these limitations, we propose **OnlineWM**, an online training framework that continuously improves world modeling through active simulator interaction and causality-aware optimization. OnlineWM introduces two key innovations: (1) *Active Online Learning*: Instead of using fixed datasets, OnlineWM adaptively queries the simulator for new interaction sequences that target the model's current predictive weaknesses, ensuring high-utility data acquisition. (2) *Causality-Aware Fine-Tuning*: We propose a counterfactual learning strategy that contrasts the outcomes of different actions from identical states, forcing the model to attribute state transitions to specific actions rather than ambient environmental evolution, thereby grounding its predictions in reliable causal mechanisms. By integrating active data acquisition with causal optimization, OnlineWM establishes a closed-loop refinement process that ensures the model is both robust to diverse scenarios and precise in its causal attribution. Extensive experiments demonstrate that OnlineWM significantly enhances action controllability and generalizes effectively to unseen domains, suggesting the learning of physically-grounded causal dynamics rather than simple visual patterns.

1 INTRODUCTION

Recent advances in video generation have enabled the synthesis of controllable, high-fidelity, and temporally coherent visual content (Wan et al., 2025; Wu et al., 2025; Zheng et al., 2024; Lin et al., 2024), opening a promising avenue toward modeling the physical world directly in pixel space. Building upon this progress, generative world models (Bruce et al., 2024; Zhu et al., 2025a; Yu et al., 2025; He et al., 2025; Team et al., 2026; Ha & Schmidhuber, 2018) have emerged as a promising route for forward dynamics prediction, which formulates world modeling as an action-conditioned video prediction problem: given a history of frames and an executed action (*e.g.*, player movements or camera rotations), the model is expected to synthesize future frames that are consistent with the underlying transition dynamics. This paradigm shows broad applicability in robotics (Zhu et al., 2025b; Shou et al., 2026), autonomous driving (Agarwal et al., 2025), and game development (Yu et al., 2025; He et al., 2025).

Despite their potential, state-of-the-art world models still exhibit significant inaccuracies in action controllability (*i.e.*, the model's ability to precisely manifest the visual consequences of a specific control signal while maintaining temporal consistency), particularly when the model encounters actions that are underrepresented in its training distribution or represent out-of-distribution (OOD) pairings of actions and scene contexts. For example, as illustrated in Fig. 1, a world model may fail to faithfully simulate a `look up` command not because the action itself is inherently difficult to model, but because it is rarely co-observed with *walking trajectories* in the training distribution. Such failures are largely rooted in the inherent biases of large-scale real-world datasets used for

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.

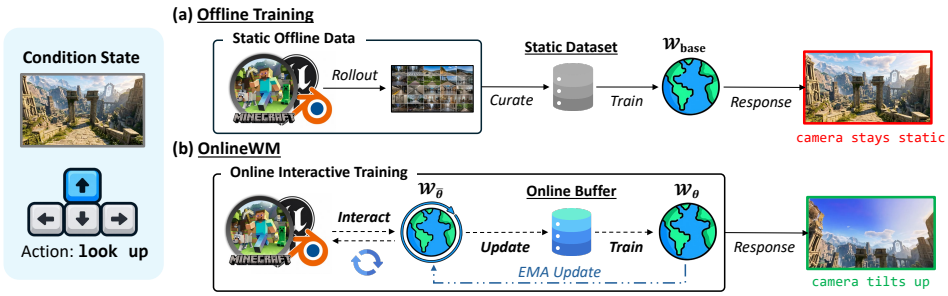


Fig. 1: **Offline *vs.* online training of action-controllable world models.** Given a condition state and an action (`look up`), a world model is tasked to predict the next observation. **(a) Prior work** trains $\mathcal{W}_{\text{base}}$ on a static dataset curated offline from simulators, which suffers from inherent action imbalance and leaves long-tail action–scene pairings undersampled, causing the prediction to ignore the action. **(b) OnlineWM (ours)** maintains an EMA copy $\mathcal{W}_{\bar{\theta}}$ that actively interacts with the simulator; the resulting rollouts are pushed into an online buffer that drives the training of $\mathcal{W}_\theta$, whose updates flow back to $\mathcal{W}_{\bar{\theta}}$ via EMA, yielding correct action grounding.

pre-training, where action distributions are highly imbalanced and sparse, and action annotations can be noisy or inaccurate. To alleviate these data-driven biases, recent methods have increasingly relied on supervised fine-tuning on controllable simulator-generated data (Team et al., 2026; He et al., 2025; Yu et al., 2025; Sun et al., 2025). However, this "collect-then-train" paradigm faces two inherent structural bottlenecks that limit its effectiveness in complex dynamics modeling.

First, existing data collection pipelines typically rely on offline datasets generated via fixed heuristics, such as uniform (Yu et al., 2025) or quality-biased sampling (He et al., 2025). While they provide broad coverage of collected trajectories, such a decoupling of data acquisition from the model's learning state leads to a misalignment between fixed training distributions and the model's evolving error patterns. Consequently, the training process suffers from redundant transitions while the critical long-tail scenarios—where the model's current dynamics prediction remains unreliable—remain undersampled. Second, the reliance on standard visual prediction loss functions induces a *causal ambiguity* during optimization. That is, the model may generate high-fidelity videos that nonetheless ignore the action control signals and follow common motion patterns, failing to achieve true action controllability. Ultimately, these limitations underscore a critical gap: the lack of an interactive mechanism that can adaptively align data acquisition with the model's evolving weaknesses while enforcing rigorous causal grounding.

To bridge the gap, we propose **OnlineWM**, an online training framework that transforms world model learning from static supervision to a closed-loop, active and causality-driven process. The core of OnlineWM lies in two synergistic components. First, we introduce *Active Online Learning* (AOL) that proactively queries simulators for high-gain samples with balanced difficulty (*i.e.*, novel to the current model while remaining learnable (Hughes et al., 2024)). To this end, we propose a difficulty decomposition strategy that separates the model's predictive errors into scene-level and action-level components, which allows the framework to explicitly identify and prioritize scenarios where learning is effective for both scene context modeling and transition dynamics. By maintaining a balanced contribution from both dimensions during data acquisition, OnlineWM ensures that the training process does not become biased toward pure visual complexity, and remains focused on the critical task of grounding actions within appropriate physical contexts. Second, to ensure accurate learning of causal dynamics from the crafted dataset, we develop *Causality-Aware Fine-Tuning* (CFT). By constructing explicit counterfactual scenarios and optimizing for the model's ability to discriminate (Miao et al., 2024) between divergent action outcomes within identical states, CFT ensures that the latent space disentangles action-induced perturbations from common ambient scene evolution. Together, these components enable OnlineWM to achieve superior action controllability and robust generalization in complex scenarios.

To conclude, our contributions can be summarized as:

- We propose OnlineWM, an online training framework that shifts world model training from passive supervision to an interactive, causality-driven refinement process. By establishing a dynamic

feedback loop between the world model's current state and the simulator, OnlineWM actively aligns the model's internal dynamics with the causal laws of the environment.

- We design two synergistic components that expose and rectify the model's causal deficiencies: AOL isolates action-specific failures from environmental complexity, enabling the targeted identification of samples with high epistemic value; and CFT leverages these samples for counterfactual training, forcing the model to learn the reliable causal relationship between actions and visual changes.
- Extensive experiments prove that OnlineWM effectively improves state-of-the-art world models (Sun et al., 2025) in both action controllability and visual quality. The results also demonstrate OnlineWM's effectiveness in general domains with long-tail or rare action-scene pairings, suggesting that it can learn generalizable knowledge about action-conditioned world dynamics.

2 Related Work

Generative World Models. Breakthroughs in visual generative modeling (Peebles & Xie, 2023; Lipman et al., 2022; Podell et al., 2023) have enabled the synthesis of high-quality visual content and motivated early attempts to simulate interactive environments with generative models (Valevski et al., 2024; Alonso et al., 2024; Bruce et al., 2024). Recent advances in video generation (Wu et al., 2025; Wan et al., 2025; Zheng et al., 2024; Lin et al., 2024) further improve temporal coherence, visual fidelity, and controllability, providing a strong foundation for generative world models to continuously improve generation quality (Bruce et al., 2024; Parker-Holder et al., 2024; Ball et al., 2025). Unlike general video generation, generative world models aim to predict future observations conditioned on both past visual states and agent actions, requiring the model to capture action-conditioned transition dynamics. More recent works (Yu et al., 2025; He et al., 2025; Team et al., 2026; Mao et al., 2025; Li et al., 2025a; Tang et al., 2025; Hong et al., 2025) further improve generative world models by constructing or selecting datasets with accurate action annotations, balanced action distributions, and high-quality action trajectories. These efforts substantially improve the data foundation for action-conditioned generation, but the resulting models still often struggle with precise action following under rare or out-of-distribution actions, or in visually complex scenes.

Improving Action Controllability for Generative World Models. Since generated future states should faithfully reflect the executed actions to support downstream applications (Zhu et al., 2025b), accurate action control is central to generative world modeling. Recent works improve action controllability either by strengthening the action representation or through dedicated post-training procedures. For instance, WorldCam (Nam et al., 2026) adopts camera pose as a unifying geometric representation to align user actions with 3D camera motion, while WorldCompass (Wang et al., 2026) introduces reward-based post-training built upon the DiffusionNFT framework (Zheng et al., 2025) to jointly improve action accuracy and visual quality. While effective, these methods treat the training data and learning process as fixed, leaving the model passive with respect to its own learning dynamics. A complementary perspective from active world model learning (Kim et al., 2020) emphasizes that accurate world modeling benefits from adapting data acquisition to the model's current learning state, directing exploration toward dynamics that are complex yet learnable. It is also relevant to generative world models: action-controllability errors can vary across scenes, actions, and training stages, making fixed offline data collection less effective for targeting the model's current deficiencies. Building on this perspective, OnlineWM combines closed-loop simulator interaction with causality-aware training to improve action-conditioned dynamics modeling.

3 Methodology

3.1 Preliminaries

World Modeling Following representative works such as Genie (Bruce et al., 2024), we frame world modeling as an autoregressive video prediction problem $p(\mathbf{z}_{1:N}) = \prod_{t=1}^{N} p(\mathbf{z}_t|\mathbf{z}_{\leq t-1}, \mathbf{a}_t, c)$, where $\mathbf{z}_t$ and $\mathbf{a}_t$ represent the latent representation of the world state and the action taken at time t, while c represents shared global conditioning information such as the textual prompt.

Autoregressive Generation with Flow-based Models We adopt the autoregressive Diffusion Transformer (DiT) (Peebles & Xie, 2023) architecture of HY-World 1.5 (Sun et al., 2025; Wu et al.,

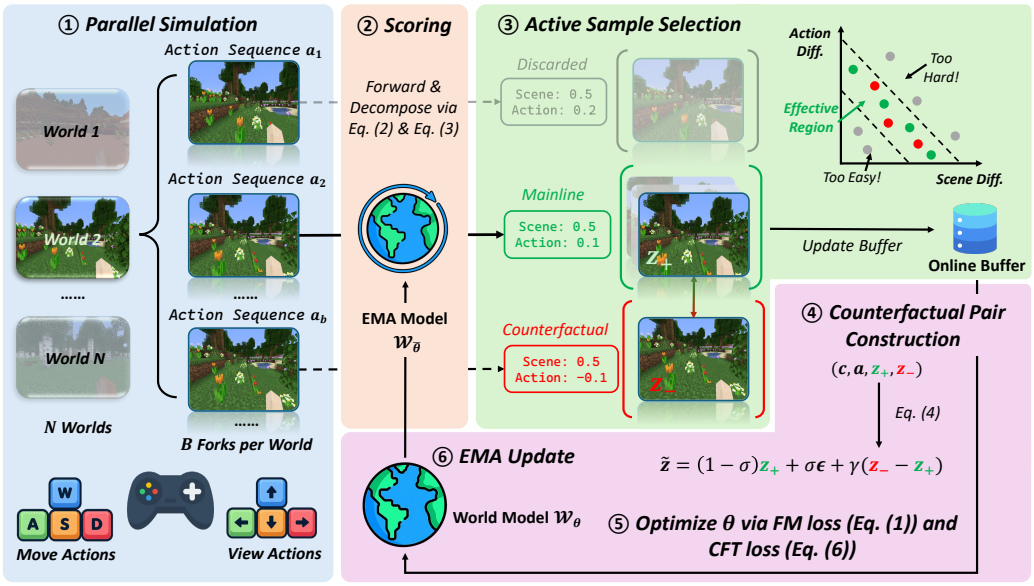


Fig. 2: **OnlineWM** trains a world model with active, causality-aware online interaction with a simulator. In each collection round, all N simulators generate B forks from the current state. Then, the world model evaluates each fork sample and decomposes the difficulties into scene- and action-level components to select the most effective one in each world along with its most similar negative sample. The selected fork is then continued for L_{main} steps and pushed into the buffer. In each training round, the world model draws samples from the buffer and optimizes with standard generation and causality-aware losses.

2025) as our base world model, which learns to generate the latent video chunk by chunk (4 latent frames per chunk, corresponding to 16 raw frames) with chunk-wise causal attention. The model is optimized by the flow matching (FM) objective (Lipman et al., 2022) with velocity prediction:

$$\mathcal{L}_{\text{FM}}(\theta) \;=\; \mathbb{E}_{t,\mathbf{z}_t,\boldsymbol{\epsilon},k,\mathbf{c}_t}\Big[\,\big\|\,\mathbf{v}_\theta(\mathbf{z}_t^k, k, \mathbf{c}_t) - (\boldsymbol{\epsilon} - \mathbf{z}_t)\,\big\|_2^2\,\Big], \quad \mathbf{z}_t^k = (1-\sigma_k)\,\mathbf{z}_t + \sigma_k\,\boldsymbol{\epsilon}. \tag{1}$$

In Eq. 1, $t \in \{1, \ldots, N\}$ denotes the chunk index and $\mathbf{z}_t$ denotes the clean video latent chunk encoded by the 3D VAE (Kingma & Welling, 2013) at index t. The clean latent is corrupted by Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and σ_k is drawn from a shifted logit-normal schedule (Wu et al., 2025) with timestep k. The condition $\mathbf{c}_t = (\mathbf{z}_{\leq t-1}, \mathbf{a}_t, c)$ includes previous latent chunks $\mathbf{z}_{\leq t-1}$, current action $\mathbf{a}_t$, and the shared global conditioning c. Here, action is represented by both discrete tokens (movement actions `w, a, s, d`, view actions `left, right, up, down`, or any combination) and continuous camera poses. More detailed architecture information can be found in Appendix A.1.

3.2 Active Online Learning

Active online learning (AOL) aims to actively acquire the most effective training samples by directing the simulator to generate rollouts according to the world model's current state. We characterize such samples as both *novel* and *learnable* (Hughes et al., 2024), which allows the model to continuously acquire new information about the world while avoiding the "white noise problem" (Schmidhuber, 2010), *i.e.*, endlessly fixating on unlearnable stimuli (Kim et al., 2020). As mentioned, the learning value of a training sample for generative world models depends on two complementary aspects: the scene context and the action-conditioned transition, both of which should provide learnable novelty for the current model. Motivated by this observation, we decompose sample acquisition into scene- and action-level components and select samples that maintain a balanced contribution from both.

Building AOL Samples As shown in Fig. 2, we launch N parallel interactive simulators during training. At each collection round, we randomly sample B candidate action sequences of length L_{fork} from the action space $\mathcal{A}$ to create forks in each world. For each fork in world n, we then run the

simulator for L_{fork} chunks with the corresponding action and obtain the ground-truth rollouts $\mathbf{z}_{n,b}$. Note that these rollouts share the same conditioning state $\mathbf{c}$ but differ in the resulting outcomes due to executed actions. We then score the difficulty of each sample using the standard FM loss at a fixed noise timestep with one forward pass. An exponential moving average (EMA) copy $\bar{\theta}$ of the world model is used for scoring to stabilize online training:

$$s_{n,b} \;=\; \left\| \mathbf{v}_{\bar{\theta}}(\tilde{\mathbf{z}}_{n,b},\, k^\star,\, \mathbf{c}_{n,b}) - (\boldsymbol{\epsilon} - \mathbf{z}_{n,b}) \right\|_2^2, \quad \tilde{\mathbf{z}}_{n,b} = (1-\sigma^\star)\,\mathbf{z}_{n,b} + \sigma^\star\,\boldsymbol{\epsilon}. \tag{2}$$

Here, $\sigma^\star$ is a fixed scoring noise level with scheduler index $k^\star$. The per-fork condition $\mathbf{c}_{n,b}$ extends the shared scene state $\mathbf{c}$ with the fork-specific action $\mathbf{a}_{n,b}$, and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is a single noise tensor shared across all B forks within the scoring round, so that the resulting ranking is driven purely by content differences. The score variability arises from differences in average scores across scenes and action-dependent variation within each scene. By the law of total variance, the score variability decomposes as $\mathrm{Var}_{n,b}[s_{n,b}] = \mathrm{Var}_n[\mathbb{E}_b[s_{n,b} \mid n]] + \mathbb{E}_n[\mathrm{Var}_b[s_{n,b} \mid n]]$, corresponding to between-scene variation and within-scene action variation, respectively. This motivates us to decompose $s_{n,b}$ into a scene-level mean d_n and an action-level residual $\delta_{n,b}$:

$$d_n = \mathrm{mean}\Big(\{s_{n,b}\}_{b=1}^{B}\Big), \quad \delta_{n,b} = s_{n,b} - d_n. \tag{3}$$

Within each world, we select the fork *whose action-level difficulty rank is inversely aligned with the world's scene-level difficulty rank*, pairing visually demanding scenes with relatively easier actions and visually simpler scenes with harder actions. This complementary pairing balances the contributions of the two difficulty dimensions in each acquired sample, preventing the training signal from being dominated by pure visual complexity while keeping every selected transition novel yet learnable for the model's current state. Starting from the selected fork, we roll out an additional L_{main} chunks in the simulator to obtain a full training trajectory from this high-utility branch.

3.3 Causality-Aware Fine-Tuning

Building on the trajectories acquired by AOL, we further construct counterfactual branches for causality-aware fine-tuning (CFT), which aims to force the model to attribute observed state transitions to the executed action rather than to common ambient scene evolution. For the selected branch b_+ in each world, we choose the most visually similar branch as the counterfactual branch b_-, yielding rollouts $\mathbf{z}_+ \leftarrow \mathbf{z}_{n,b_+}$ and $\mathbf{z}_- \leftarrow \mathbf{z}_{n,b_-}$ that share an identical conditioning state $\mathbf{c}$.

Learning Objectives Following WorldCompass (Wang et al., 2026), we adopt DiffusionNFT (Zheng et al., 2025), a *forward-process* post-training objective to optimize the world model with counterfactual supervision. Unlike reward-based optimization, our formulation operates on explicit positive–negative sample pairs, where the positive $\mathbf{z}_+$ denotes the ground-truth continuation under the executed action and the negative $\mathbf{z}_-$ denotes a counterfactual alternative from the same conditioning state $\mathbf{c}$. To enhance causality awareness during optimization, we perturb the forward-noised latent state by injecting the negative sample with a noise-level-adaptive weight:

$$\tilde{\mathbf{z}}^k = (1-\sigma_k)\,\mathbf{z}_+ + \sigma_k\,\boldsymbol{\epsilon} + \gamma_k\,(\mathbf{z}_- - \mathbf{z}_+), \tag{4}$$

where $\gamma_k = \lambda\, h\, \sigma_k(1-\sigma_k)$ controls the weight that tilts the noise toward $\mathbf{z}_-$ while keeping the boundary unchanged at $\sigma_k \in \{0, 1\}$, and h is a similarity metric (*e.g.*, cosine similarity, SSIM) to modulate sample hardness. From this counterfactually perturbed state, the model is then trained to recover the positive trajectory $\mathbf{z}_+$ and repel $\mathbf{z}_-$, forcing it to attribute the predicted dynamics to the executed action rather than to the visual cues shared between $\mathbf{z}_+$ and $\mathbf{z}_-$.

Following DiffusionNFT, we denote $\mathbf{v}_\theta$ and $\mathbf{v}_{\text{old}}$ as the current and EMA model velocity predictions (conditioned on $\mathbf{c}$ and timestep k), and define implicit positive and negative policies along with their predictions for causality-aware fine-tuning:

$$\begin{cases} \mathbf{v}^+ = (1-\beta)\,\mathbf{v}_{\text{old}} + \beta\,\mathbf{v}_\theta, & \hat{\mathbf{z}}_+ = \tilde{\mathbf{z}}^k - \sigma_k\,\mathbf{v}^+, \\ \mathbf{v}^- = (1+\beta)\,\mathbf{v}_{\text{old}} - \beta\,\mathbf{v}_\theta, & \hat{\mathbf{z}}_- = \tilde{\mathbf{z}}^k - \sigma_k\,\mathbf{v}^-. \end{cases} \tag{5}$$

The final CFT loss combines the two reconstructions with a similarity-based weight that allocates more supervision to similar negatives and avoids learning from samples with large discrepancies:

$$\mathcal{L}_{\text{CFT}}(\theta) = (1-\alpha)\left\|\hat{\mathbf{z}}_+ - \mathbf{z}_+\right\|_w^2 + \alpha\left\|\hat{\mathbf{z}}_- - \mathbf{z}_-\right\|_w^2,\ \alpha = h/2. \tag{6}$$

To stabilize the training process across noise levels and pair difficulties, each reconstruction term adopts adaptive normalization with the stop-gradient operation (Wang et al., 2026; Zheng et al., 2025):

$$\left\|\hat{\mathbf{z}}-\mathbf{z}\right\|_w^2 \;=\; \mathrm{mean}\left(\frac{(\hat{\mathbf{z}}-\mathbf{z})^2}{\mathrm{sg}\big[\,\mathrm{mean}\big(|\hat{\mathbf{z}}-\mathbf{z}|\big)\,\big]+\varepsilon}\right). \tag{7}$$

Remark Ignoring scalar factors from the reconstruction norm, the gradient of $\mathcal{L}_{\mathrm{CFT}}$ with respect to the current velocity can be written as

$$\nabla_{\mathbf{v}_\theta}\mathcal{L}_{\mathrm{CFT}} \;\propto\; -\sigma_k\beta\Big[(1-\alpha)(\hat{\mathbf{z}}_+-\mathbf{z}_+)-\alpha(\hat{\mathbf{z}}_--\mathbf{z}_-)\Big]. \tag{8}$$

The two branches provide complementary learning signals. The positive branch pulls the predicted velocity toward reconstructing $\mathbf{z}_+$ from the augmented noisy state $\tilde{\mathbf{z}}^k$, which is tilted toward the counterfactual outcome. In contrast, the negative branch provides a repulsive signal that pushes the velocity away from $\mathbf{z}_-$. Consequently, CFT converts counterfactual rollouts into causal learning signals, forcing the model to distinguish the causally correct outcome from its counterfactual alternative.

3.4 Overall Learning Objective

Training Buffer To improve sample efficiency for online training, we maintain a fixed-capacity replay buffer $\mathcal{D}$ of size $S_{\max}$ and only run the collection round every N_{collect} iterations. Each sample is assigned a priority p initialized as its FM score $s_{n,b}$ at collection time, together with a usage counter n_{train} that increments after each draw. When $\mathcal{D}$ exceeds its capacity, we evict the sample with the smallest effective priority $\tilde{p}=p\cdot\max(0.01,\,1-n_{\text{train}}/N_{\max})$ to retain high-score but under-trained samples. Training samples are drawn *uniformly* from $\mathcal{D}$.

For each sample drawn from the training buffer, we calculate the total loss by the weighted combination of FM (Eq. 1) and CFT (Eq. 6) losses and perform joint optimization:

$$\mathcal{L}_{\text{total}}=\mathcal{L}_{\text{FM}}+\lambda_{\text{CFT}}\cdot\mathcal{L}_{\text{CFT}}. \tag{9}$$

4 Experiments

4.1 Experimental Setup

Overall Setting We use HY-World 1.5 (Sun et al., 2025; Wu et al., 2025) (with 8B parameters) as our base world model. We initialize it from the WorldCompass (Wang et al., 2026) RL post-training checkpoint. The simulator backend uses MineStudio (Cai et al., 2024) built on MineRL (Guss et al., 2019), which offers flexible Minecraft environment control and distributed interaction based on Ray (Moritz et al., 2018). We use 4 compute nodes, each equipped with 8 GPUs, running $N=32$ parallel Minecraft worlds to generate rollouts of resolution 832×480 at 20 FPS with per-frame GT camera poses. The valid action space $\mathcal{A}$ contains 9 keyboard movements (`idle`, `w`, `a`, `s`, `d`, and the four diagonals), 5 camera primitives (`idle`, `left`, `right`, `up`, `down`) and their combinations, resulting in 45 discrete compound action tokens per tick. Each sampled action is held fixed for a short duration to prevent overly abrupt visual changes (see Appendix A.2 for details).

Training Parameters We train the world model for 3,000 steps with the Muon optimizer (Liu et al., 2025a). The peak learning rate is set to 2×10^{-5} with a `one-cycle` schedule. We use sequence parallelism (SP) with 8 SP groups and 2 gradient accumulation steps, equivalent to an effective batch size of 16 per iteration. In each AOL collection round, we fork $B=8$ candidate branches with random override probability $p_{\text{rand}}=0.2$ to prevent overfitting fixed patterns. The selected fork of length $L_{\text{fork}}=1$ is unrolled for another $L_{\text{main}}=8$ chunks. The maximum size of the replay buffer is set to $S_{\max}=96$ with $N_{\max}=N_{\text{collect}}=6$. Both the scoring and old-policy networks for CFT are EMA copies with decay factor 0.99. We set the scoring noise level to $\sigma^\star=0.5$ and $\lambda=0.1,\beta=1,\lambda_{\text{CFT}}=0.2$ for the CFT loss. Full training details are listed in Appendix A.3.

Tab. 1: Evaluation on *visual generation quality* between different methods. Best results are in **bold**.

Method	SSIM ↑	LPIPS ↓	VBench					
			Subject ↑	Background ↑	Motion ↑	Aesthetic ↑	Imaging ↑	Overall ↑
GT	*1.000*	*0.000*	*0.901*	*0.961*	*0.974*	*0.545*	*0.695*	*0.815*
Base Model[1]	0.458	0.506	*0.895*	*0.965*	*0.977*	*0.549*	*0.713*	*0.820*
+ *Random*	0.531	0.453	0.876	0.967	0.981	0.537	0.639	0.800
+ *AOL*	**0.538**	0.419	**0.890**	**0.970**	**0.982**	0.541	0.644	0.805
Full Model (*w/ CFT*)	0.531	**0.411**	**0.890**	0.968	0.979	**0.543**	**0.681**	**0.812**

Evaluation Metrics We evaluate all methods on a held-out Minecraft set, consisting of manually curated challenging scenes of 61 frames each, paired with action trajectories in which actions change more frequently than in training. We evaluate the world model from two perspectives: (i) *visual quality* and (ii) *action controllability*. For *visual quality*, we use both VBench (Huang et al., 2024) metrics and conventional visual metrics, including SSIM and LPIPS (Zhang et al., 2018), to comprehensively evaluate generation quality. For *action controllability*, we first use Depth Anything V3 (Lin et al., 2025) with Sim(3) Umeyama alignment (Umeyama, 1991) to estimate normalized camera poses, and then compute discrete-token control accuracy and relative pose error (RPE) against the ground truth following Nam et al. (2026); Wang et al. (2026) (see Appendix A.4 for details).

Inference Settings At inference time, the world model autoregressively generates the latent video in chunks of 4 latent frames. Within each chunk, we run 50 Euler steps under the flow-matching scheduler with time-shift 5, and apply classifier-free guidance with scale 6 using a precomputed null-prompt embedding for the unconditional branch. Across chunks, the key-value features of past chunks are cached and a temporally aligned 12-frame context is selected from a 20-frame sliding memory window for additional conditioning. All evaluations share the same inference configuration to ensure a fair comparison across method variants.

4.2 Main Results

We compare our proposed method against *Random*, which samples training data without active acquisition or counterfactual supervision. Tab. 1 reports the visual generation results. Visual-quality metrics are relatively close across model variants, but clear trends can still be observed. Compared with *Random*, *AOL* improves reconstruction fidelity and perceptual similarity, achieving higher SSIM and lower LPIPS. It also brings consistent gains across all VBench dimensions. Adding CFT further improves LPIPS and gives the best VBench Imaging and Overall scores. This suggests that counterfactual supervision complements reconstruction-based learning by improving perceptual and imaging quality. In our experiments, we observed that selecting the most difficult fork in each scene leads to significant instability during training, which confirms the need for selecting samples with balanced difficulty.

The improvement is clearer in terms of action controllability, which is the central objective of generative world modeling. Compared with *Random*, *AOL* improves both Combined and Fine-grained action accuracy, while reducing both rotational and translational trajectory errors. This shows that active acquisition selects samples that are more useful for learning action-conditioned transitions. With CFT, the full model achieves the best results on all controllability metrics. Relative to *Random*, it improves Combined accuracy by 19.4% and Fine-grained accuracy by 12.5%, while reducing $\mathrm{RPE}_{\mathrm{rot}}$ by 36.3% and $\mathrm{RPE}_{\mathrm{trans}}$ by 20.1%. These gains indicate that AOL provides more effective supervision, and CFT further uses counterfactual samples to effectively strengthen the model's understanding of action transition dynamics.

4.3 Qualitative Results and Analysis

We further provide a qualitative evaluation of OnlineWM. Fig. 3 shows a challenging scenario where the agent moves into tree leaves, causing strong occlusion and a large visual distribution shift.

[1]As the base model cannot effectively follow actions or produce Minecraft-style outputs, general visual metrics without reference videos (*i.e.*, VBench scores) are not directly comparable and are listed for reference only.

Tab. 2: Evaluation on *action controllability* between different methods. Best results are in **bold**.

Method	Action Accuracy		Camera Control	
	Combined ↑	Fine-grained ↑	RPE_{rot} ↓	RPE_{trans} ↓
Base Model	0.240	0.490	0.6906	0.0235
+ *Random*	0.356	0.569	0.6393	0.0199
+ *AOL*	0.393	0.610	0.5033	0.0166
Full Model (*w/ CFT*)	**0.425**	**0.640**	**0.4072**	**0.0159**

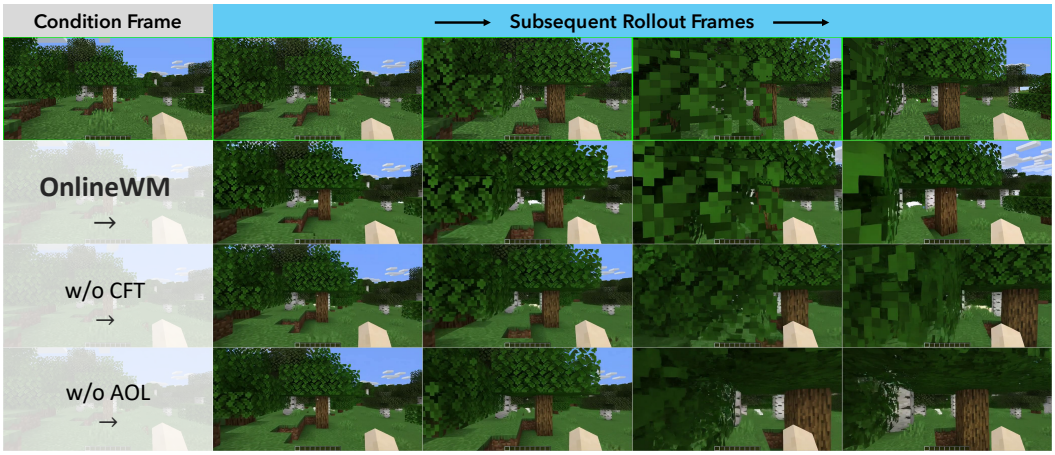


Fig. 3: Visual comparison of all method variants in a challenging scenario. The first row shows the ground-truth video, and subsequent rows denote the generation results of three variants: *OnlineWM*, *OnlineWM w/o the CFT loss* and *OnlineWM* ***removing the whole AOL and CFT design***.

OnlineWM handles this case accurately: it preserves the spatial relation between the camera, the tree trunk, and the surrounding leaves, and follows the commanded action to produce the correct collision-like transition. Removing CFT weakens this spatial reasoning, leading to less accurate alignment between the predicted motion and the scene geometry. When AOL is also removed, the model struggles with this rare and difficult transition, and the rollout collapses to a more common trajectory under the tree instead of correctly modeling the interaction with the leaves. This comparison shows that AOL helps expose the model to informative failure-prone cases, while CFT improves the learning of action-conditioned spatial dynamics. Refer to Appendix D for more qualitative examples.

4.4 Generalization to General Domains

OnlineWM learns action control through online interaction with simulators. An important question is whether the learned action-conditioned dynamics are specific to the simulator domain, or can transfer to visually different general domains. To examine this, we directly evaluate the world model trained with OnlineWM on Minecraft using general-domain images as the conditioning input, without any further training or adaptation. We compare it with the RL post-training checkpoint from WorldCompass (Wang et al., 2026). To better stress-test action controllability, we use an action sequence with faster view changes than the default setting in WorldCompass. We report camera RPE and present the generated video alongside 3D Gaussian reconstruction results from WorldMirror (Liu et al., 2025b), annotated with the estimated camera trajectory.

As shown in Fig. 4, the baseline model struggles to follow the conditioning actions, particularly the `up` action, which is relatively uncommon in the training data. In contrast, the model trained with OnlineWM produces view changes that are more consistent with the input actions. This improvement is reflected in both the qualitative visual comparison and the quantitative action-control metrics, where OnlineWM achieves lower translational and rotational RPE. These results demonstrate that OnlineWM does not merely improve in-domain simulator performance, but also learns action-conditioned transition knowledge that can transfer to general visual domains. It also suggests that

learning effective causality-aware action control from simulators is a promising direction towards building effective general world models. Refer to Appendix D for more qualitative examples on general domains.

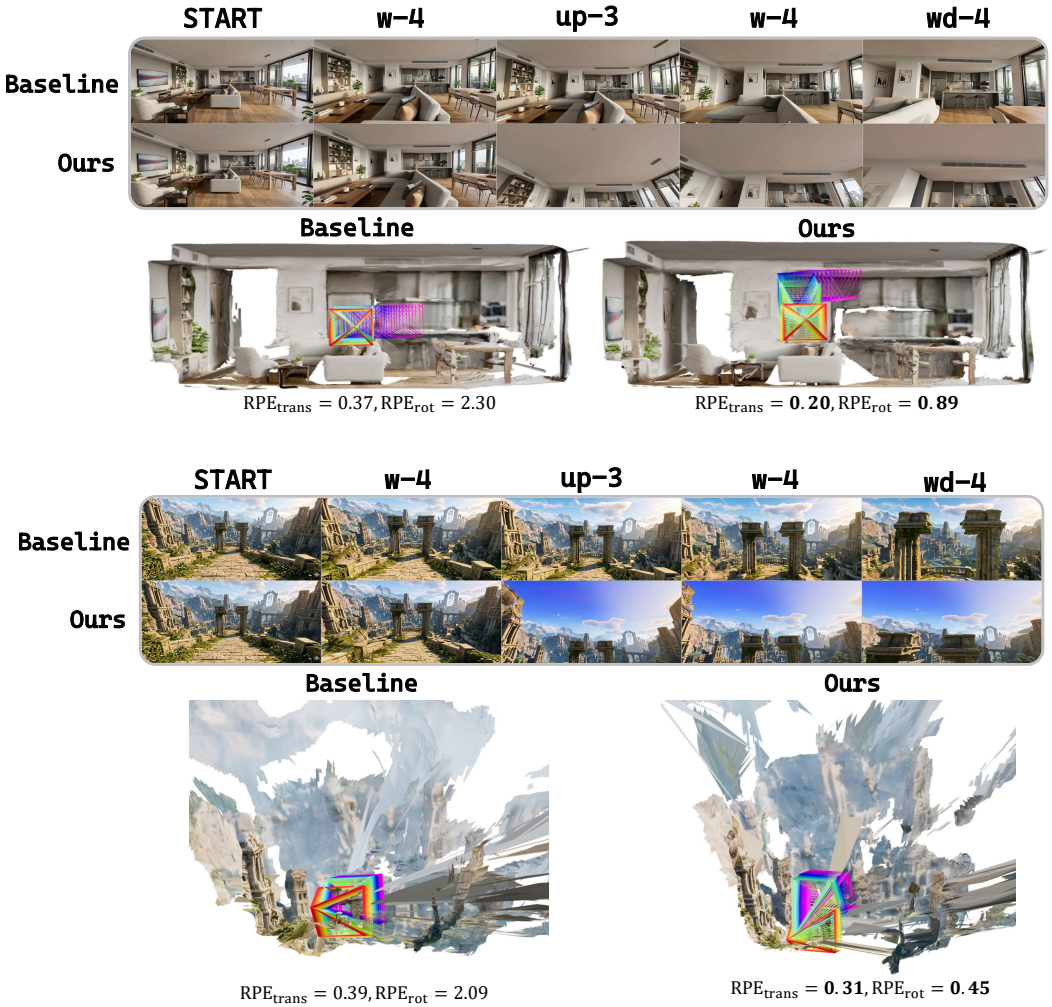


Fig. 4: Visual comparison of action controllability on general domains. For each scene, the top rows show rollouts conditioned on the same action sequence (denoted by the labels at the top of each column, *e.g.*, `w-4`, `up-3`) under the baseline and our model. The bottom row visualizes the corresponding 3D Gaussian Splatting reconstruction and the recovered camera trajectory, with relative pose errors RPE$_{trans}$ and RPE$_{rot}$ reported below.

5 Conclusion

In conclusion, OnlineWM presents a new paradigm for world model training, shifting from passive supervision to a closed-loop, active, and causality-driven refinement process. It delivers notable improvements in both action controllability and visual quality, demonstrating the fundamental benefits of online training over existing offline paradigms for world models. The generalization results further indicate that OnlineWM can serve as an effective and universal training paradigm for distilling action control from simulators into various domains. We believe this closed-loop, causality-aware perspective offers a promising path toward world models whose dynamics are grounded in reliable causal mechanisms rather than spurious visual correlations, and we hope OnlineWM can inspire future research on interactive and causally grounded learning for generative world models.

AI USE STATEMENT

Generative AI was used to synthesize general-domain evaluation images, assist with draft revision, generate icons, and clean and reformat code. AI-generated text served only as a reference for the author's revision, and all citations were added manually. All other activities requiring disclosure were either not AI-assisted or not applicable. The authors reviewed all AI-assisted work and take responsibility for the final manuscript and artifacts.

REPRODUCIBILITY STATEMENT

We disclose all experimental details to ensure reproducibility. The experimental setup and inference settings are described in Sec. 4.1. Appendix A further provides the model architecture, simulator and data-processing conventions, training hyperparameters, distributed setup, and evaluation protocols; Algorithms 1–3 specify the training procedure. We will also release all code and checkpoints upon acceptance of this paper.

REFERENCES

Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari. *Advances in Neural Information Processing Systems*, 37:58757–58791, 2024.

Philip J. Ball, Jakob Bauer, Frank Belletti, Bethanie Brownfield, Ariel Ephrat, Shlomi Fruchter, Agrim Gupta, Kristian Holsheimer, Aleksander Holynski, Jiri Hron, Christos Kaplanis, Marjorie Limont, Matt McGill, Yanko Oliveira, Jack Parker-Holder, Frank Perbet, Guy Scully, Jeremy Shar, Stephen Spencer, Omer Tov, Ruben Villegas, Emma Wang, Jessica Yung, Cip Baetu, Jordi Berbel, David Bridson, Jake Bruce, Gavin Buttimore, Sarah Chakera, Bilva Chandra, Paul Collins, Alex Cullum, Bogdan Damoc, Vibha Dasagi, Maxime Gazeau, Charles Gbadamosi, Woohyun Han, Ed Hirst, Ashyana Kachra, Lucie Kerley, Kristian Kjems, Eva Knoepfel, Vika Koriakin, Jessica Lo, Cong Lu, Zeb Mehring, Alex Moufarek, Henna Nandwani, Valeria Oliveira, Fabio Pardo, Jane Park, Andrew Pierson, Ben Poole, Helen Ran, Tim Salimans, Manuel Sanchez, Igor Saprykin, Amy Shen, Sailesh Sidhwani, Duncan Smith, Joe Stanton, Hamish Tomlinson, Dimple Vijaykumar, Luyu Wang, Piers Wingfield, Nat Wong, Keyang Xu, Christopher Yew, Nick Young, Vadim Zubov, Douglas Eck, Dumitru Erhan, Koray Kavukcuoglu, Demis Hassabis, Zoubin Gharamani, Raia Hadsell, Aäron van den Oord, Inbar Mosseri, Adrian Bolton, Satinder Singh, and Tim Rocktäschel. Genie 3: A new frontier for world models. 2025.

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

Shaofei Cai, Zhancun Mu, Kaichen He, Bowei Zhang, Xinyue Zheng, Anji Liu, and Yitao Liang. Minestudio: A streamlined package for minecraft ai agent development. *arXiv preprint arXiv:2412.18293*, 2024.

William H Guss, Brandon Houghton, Nicholay Topin, Phillip Wang, Cayden Codel, Manuela Veloso, and Ruslan Salakhutdinov. Minerl: A large-scale dataset of minecraft demonstrations. *arXiv preprint arXiv:1907.13440*, 2019.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

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

Yicong Hong, Yiqun Mei, Chongjian Ge, Yiran Xu, Yang Zhou, Sai Bi, Yannick Hold-Geoffroy, Mike Roberts, Matthew Fisher, Eli Shechtman, et al. Relic: Interactive video world model with long-horizon memory. *arXiv preprint arXiv:2512.04040*, 2025.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21807–21818, 2024.

Edward Hughes, Michael Dennis, Jack Parker-Holder, Feryal Behbahani, Aditi Mavalankar, Yuge Shi, Tom Schaul, and Tim Rocktaschel. Open-endedness is essential for artificial superhuman intelligence. *arXiv preprint arXiv:2406.04268*, 2024.

Kuno Kim, Megumi Sano, Julian De Freitas, Nick Haber, and Daniel Yamins. Active world model learning with progress curiosity. In *International conference on machine learning*, pp. 5306–5315. PMLR, 2020.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

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

Ruilong Li, Brent Yi, Junchen Liu, Hang Gao, Yi Ma, and Angjoo Kanazawa. Cameras as relative positional encoding. *arXiv preprint arXiv:2507.10496*, 2025b.

Bin Lin, Yunyang Ge, Xinhua Cheng, Zongjian Li, Bin Zhu, Shaodong Wang, Xianyi He, Yang Ye, Shenghai Yuan, Liuhan Chen, et al. Open-sora plan: Open-source large video generation model. *arXiv preprint arXiv:2412.00131*, 2024.

Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, et al. Muon is scalable for llm training. *arXiv preprint arXiv:2502.16982*, 2025a.

Yifan Liu, Zhiyuan Min, Zhenwei Wang, Junta Wu, Tengfei Wang, Yixuan Yuan, Yawei Luo, and Chunchao Guo. Worldmirror: Universal 3d world reconstruction with any-prior prompting. *arXiv preprint arXiv:2510.10726*, 2025b.

Xiaofeng Mao, Zhen Li, Chuanhao Li, Xiaojie Xu, Kaining Ying, Tong He, Jiangmiao Pang, Yu Qiao, and Kaipeng Zhang. Yume-1.5: A text-controlled interactive world generation model. *arXiv preprint arXiv:2512.22096*, 2025.

Yikun Miao, Meiqing Wu, Siew-Kei Lam, Changsheng Li, and Thambipillai Srikanthan. Hierarchical object-aware dual-level contrastive learning for domain generalized stereo matching. *Advances in Neural Information Processing Systems*, 37:132050–132076, 2024.

Philipp Moritz, Robert Nishihara, Stephanie Wang, Alexey Tumanov, Richard Liaw, Eric Liang, Melih Elibol, Zongheng Yang, William Paul, Michael I Jordan, et al. Ray: A distributed framework for emerging {AI} applications. In *13th USENIX symposium on operating systems design and implementation (OSDI 18)*, pp. 561–577, 2018.

Jisu Nam, Yicong Hong, Chun-Hao Paul Huang, Feng Liu, JoungBin Lee, Jiyoung Kim, Siyoon Jin, Yunsung Lee, Jaeyoon Jung, Suhwan Choi, et al. Worldcam: Interactive autoregressive 3d gaming worlds with camera pose as a unifying geometric representation. *arXiv preprint arXiv:2603.16871*, 2026.

Jack Parker-Holder, Philip Ball, Jake Bruce, Vibhavari Dasagi, Kristian Holsheimer, Christos Kaplanis, Alexandre Moufarek, Guy Scully, Jeremy Shar, Jimmy Shi, Stephen Spencer, Jessica Yung, Michael Dennis, Sultan Kenjeyev, Shangbang Long, Vlad Mnih, Harris Chan, Maxime Gazeau, Bonnie Li, Fabio Pardo, Luyu Wang, Lei Zhang, Frederic Besse, Tim Harley, Anna Mitenkova, Jane Wang, Jeff Clune, Demis Hassabis, Raia Hadsell, Adrian Bolton, Satinder Singh, and Tim Rocktäschel. Genie 2: A large-scale foundation world model. 2024. URL https://deepmind.google/discover/blog/genie-2-a-large-scale-foundation-world-model/.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

Jürgen Schmidhuber. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE transactions on autonomous mental development*, 2(3):230–247, 2010.

Quanxin Shou, Fangqi Zhu, Shawn Chen, Puxin Yan, Zhengyang Yan, Yikun Miao, Xiaoyi Pang, Zicong Hong, Ruikai Shi, Hao Huang, et al. Halo: A unified vision-language-action model for embodied multimodal chain-of-thought reasoning. *arXiv preprint arXiv:2602.21157*, 2026.

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.

Junshu Tang, Jiacheng Liu, Jiaqi Li, Longhuang Wu, Haoyu Yang, Penghao Zhao, Siruis Gong, Xiang Yuan, Shuai Shao, Linfeng Zhang, et al. Hunyuan-gamecraft-2: Instruction-following interactive game world model. *arXiv preprint arXiv:2511.23429*, 2025.

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

S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, 1991. doi: 10.1109/34.88573.

Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. *arXiv preprint arXiv:2408.14837*, 2024.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Zehan Wang, Tengfei Wang, Haiyu Zhang, Xuhui Zuo, Junta Wu, Haoyuan Wang, Wenqiang Sun, Zhenwei Wang, Chenjie Cao, Hengshuang Zhao, et al. Worldcompass: Reinforcement learning for long-horizon world models. *arXiv preprint arXiv:2602.09022*, 2026.

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

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, 2025.

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*, pp. 586–595, 2018.

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

Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all. *arXiv preprint arXiv:2412.20404*, 2024.

Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. Irasim: A fine-grained world model for robot manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9834–9844, 2025a.

Fangqi Zhu, Zhengyang Yan, Zicong Hong, Quanxin Shou, Xiao Ma, and Song Guo. Wmpo: World model-based policy optimization for vision-language-action models. *arXiv preprint arXiv:2511.09515*, 2025b.

A More Implementation Details

A.1 Model Architecture

We provide additional details on the HY-World 1.5 backbone (Sun et al., 2025; Wu et al., 2025) used as our base world model, focusing on how discrete actions and continuous camera poses are injected into the DiT. Tab. 3 summarizes the key architectural and VAE parameters; the rest of this section describes the conditioning mechanisms.

Tab. 3: Architecture parameters of the HY-World 1.5 backbone (Sun et al., 2025; Wu et al., 2025).

Component	Configuration
Diffusion transformer	
Block depth (dual-stream)	54
Hidden dim / heads / head dim	2048 / 16 / 128
MLP expansion ratio	4×
Patch size (3D conv)	$1\times1\times1$
Positional encoding	3-axis RoPE on (T, H, W), dims $(16, 56, 56)$
3D causal VAE	
Spatial / temporal compression factor	16× / 4×
Latent channels	32
Latent length	$T_z = \lfloor (T-1)/4 \rfloor + 1$

Action Injection Each latent frame is associated with a compound action token from the 45-token vocabulary defined in our experimental setup. The token is embedded through the same sinusoidal-then-2-layer-MLP module that lifts the diffusion timestep, summed into the global AdaLN conditioning vector together with the timestep and pooled text embeddings, and broadcast across the visual tokens of the frame so that action information modulates every dual-stream block via AdaLN rather than entering the attention sequence as a token. The final MLP layer of the action embedder is zero-initialized, so action conditioning behaves as a no-op when adapting from a pretrained text-to-video backbone.

Camera Injection Per-frame camera poses (a 4×4 extrinsic and a 3×3 intrinsic) are injected through projective rotary positional encoding (PRoPE) (Li et al., 2025b), which HY-World 1.5 adopts as a parallel visual attention branch attached to each dual-stream block. Conceptually, PRoPE conditions attention on the per-frame camera projection so that geometrically corresponding rays across frames are aligned within the camera-conditioned attention space, complementing the standard 3-axis RoPE that encodes spatio-temporal position. The PRoPE branch is fused back into the visual stream through a zero-initialized linear projection and is applied only to the visual stream, while text tokens retain the standard RoPE attention.

Autoregressive Rollout with History The noisy latent chunk to be denoised is channel-wise concatenated with the latents of previously generated chunks together with a binary mask indicating observed positions, and the concatenated tensor is processed by the patch-embedding 3D convolution whose conditioning-side input channels are zero-initialized. The attention mask is bidirectional within a chunk and strictly causal across chunks, enforcing autoregressive generation while preserving full intra-chunk dependencies, and the key-value features of past chunks are cached at inference time so that only the current chunk is recomputed per rollout step. Text conditioning is processed through a separate stream within each dual-stream block and joins the visual tokens via concatenated joint attention, with separate output projections and MLPs for each modality.

A.2 Online Data Collection and Processing Details

Simulator and Action Sampling Each parallel Minecraft world is initialized in `creative` mode with a unique random map seed and a spawn biome uniformly drawn from a fixed pool. To keep visual transitions smooth and temporally consistent with the model's chunked latent representation, the

action agent samples a compound action token and holds it for a contiguous block of latent frames—roughly 0.8–3.2 seconds at 20 FPS—before re-sampling. Longer trajectories can be composed from these action primitives. Tab. 4 summarizes the simulator and data conventions adopted throughout our online data collection.

Tab. 4: Simulator and data conventions for online data collection in OnlineWM.

Setting	Value
World and rendering	
Game mode	`creative`
Spawn biomes (uniform draw)	{`plains`, `forest`, `desert`, `taiga`, `savanna`}
Field of view (FoV)	70° horizontal
Action and timing	
View delta per simulator tick	3°
Action hold duration	4–16 latent frames (0.8–3.2 s at 20 FPS)
World reset cadence	every $\sim$96 latent frames
Pose and conditioning	
Camera extrinsic	4×4 world-to-camera, first-frame normalized
Camera intrinsic	3×3, derived from 70° FoV at 832×480
Caption (fixed)	`"Minecraft first-person gameplay."`

Sample Format Each collected sample stores everything required to compute both the flow-matching and the CFT losses without re-running the simulator. Raw RGB frames are encoded into VAE latents on the fly and discarded, and every buffer entry consists of three parts: (i) the long mainline trajectory of L_{main} chunks together with its per-latent-frame action tokens and camera poses, used for the flow-matching loss; (ii) the short positive–negative fork pair $(\mathbf{z}_+, \mathbf{z}_-)$ of L_{fork} chunks each, sharing the same conditioning state $\mathbf{c}$ and used for the causality-aware loss; and (iii) the conditioning first-frame latent together with the precomputed text-encoder features for the fixed caption and vision-encoder features for the conditioning frame. Camera extrinsics are normalized to the first-frame pose so that every rollout shares a canonical origin, while intrinsics stay constant within a sample. Tab. 5 summarizes the corresponding tensor shapes.

Tab. 5: Per-sample replay buffer layout in OnlineWM. Shapes use the latent spatial size $H_z \times W_z = 30\times52$ (at 832×480 raw resolution) and 4 latent frames per chunk.

Field	Shape	Description
Mainline trajectory ($L_{main}=8$ chunks)		
`latents`	$[32,\ 4L_{\text{main}},\ H_z,\ W_z]$	post-fork mainline VAE latents
`action_tokens`	$[4L_{\text{main}}]$	one compound token per latent frame
`w2c`	$[4L_{\text{main}},\ 4,\ 4]$	first-frame-normalized world-to-camera
`intrinsic`	$[4L_{\text{main}},\ 3,\ 3]$	derived from 70° FoV
Counterfactual fork pair ($L_{fork}=1$ chunk each)		
`short_latents`	$[32,\ 4L_{\text{fork}},\ H_z,\ W_z]$	chosen-fork prefix, used as $\mathbf{z}_+$ in CFT
`neg_latents`	$[32,\ 4L_{\text{fork}},\ H_z,\ W_z]$	counterfactual fork, used as $\mathbf{z}_-$ in CFT
Other `neg_`$\star$ fields	shapes as in mainline	per-frame action token and pose for $\mathbf{z}_-$
Shared conditioning		
`image_cond`	$[32,\ 1,\ H_z,\ W_z]$	VAE latent of the conditioning frame
`prompt_embed`	[1,000, 3584]	LLM embedding of the fixed caption
`vision_states`	[729, 1152]	SigLIP embedding of the conditioning frame

A.3 TRAINING

We expand on the training procedure of OnlineWM in three parts: the hyperparameter configuration, the algorithmic skeleton of the closed-loop training pipeline, and the distributed architecture used to scale the framework across multiple nodes.

Hyperparameters Tab. 6 reports the full hyperparameter configuration used throughout our experiments, organized into optimization, distributed setup, active online learning, replay buffer, and causality-aware fine-tuning.

Tab. 6: Training hyperparameters and distributed setup of OnlineWM.

Hyperparameter	**Value**
Optimization	
Optimizer	Muon (Liu et al., 2025a) (AdamW for $\leq$1D parameters)
Peak learning rate	2×10^{-5}
LR schedule / warmup steps	`one-cycle` with cosine annealing / 100
Total training steps	3,000
Weight decay / max grad norm	10^{-4} / 1.0
Mixed precision (forward / master)	`bf16` / `fp32`
Distributed setup	
Compute	4 nodes $\times$ 8 GPUs = 32 GPUs
Sequence-parallel groups (size per group)	8 (4)
Gradient accumulation steps	2
Effective batch size	16
Parallel simulators (N)	32
Active online learning	
Fork branches per world (B)	8
Fork length / main rollout (L_{fork} / L_{main})	1 / 8 chunks
Random override probability (p_{rand})	0.2
Scoring noise level ($\sigma^\star$)	0.5
EMA decay (scoring & old policy)	0.99
AOL warmup steps	100
Replay buffer	
Capacity ($S_{\max}$)	96
Collection interval (N_{collect})	6
Max usage per sample ($N_{\max}$)	6
Causality-aware fine-tuning	
Counterfactual mixing weight (λ)	0.1
NFT advantage scaling (β)	1.0
CFT loss weight (λ_{CFT})	0.2
Training noise schedule	*shifted logit-normal* ($\mu=0,\ \sigma=1$, time-shift 3)

Algorithmic Skeleton We summarize the OnlineWM training procedure as three interleaved algorithms. Algorithm 1 describes the closed-loop main pipeline, which alternates between an active online collection round and a gradient update; the collection round invokes **AOL_Select** (Algorithm 2) to score all $N\times B$ candidate forks under the EMA model, decompose their difficulty into scene- and action-level components, and select a chosen fork together with a counterfactual fork for each world. At each gradient step, **CFT_Loss** (Algorithm 3) is computed alongside the standard flow-matching loss on the long mainline trajectory and combined as $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{FM}} + \lambda_{\text{CFT}}\mathcal{L}_{\text{CFT}}$.

Distributed Architecture Training runs on 4 compute nodes with 8 GPUs each, for a total of 32 GPUs. The 8B-parameter backbone is distributed via hybrid sharded data parallelism (HSDP)—sharded within each node and replicated across nodes—and combined with sequence parallelism over 8 groups, each spanning 4 GPUs; together with 2 gradient accumulation steps, this yields an effective

Algorithm 1 OnlineWM: closed-loop active online training.

Require: Pretrained world model θ with EMA copy $\bar{\theta}$, parallel simulators $\{\mathrm{Env}_n\}_{n=1}^{N}$ at conditioning state $\mathbf{c}_n$, replay buffer $\mathcal{D} \leftarrow \emptyset$, total training steps T

for $t = 1, 2, \ldots, T$ **do**
  if $|\mathcal{D}| < S_{\max}$ **or** $t \bmod N_{\text{collect}} = 0$ **then** ▷ collection round
    for $n = 1, \ldots, N$ **do**
      sample B candidate compound action sequences and roll out for L_{fork} chunks
      encode each fork to a latent $\mathbf{z}_{n,b}$ sharing the conditioning state $\mathbf{c}_n$
    end for
    $\{(b_+^{(n)}, b_-^{(n)})\}_{n=1}^{N}, \{s_{n,b}\} \leftarrow$ **AOL_Select** (Algorithm 2)
    for $n = 1, \ldots, N$ **do**
      $\mathbf{z}_+^{(n)} \leftarrow \mathbf{z}_{n,\, b_+^{(n)}}, \quad \mathbf{z}_-^{(n)} \leftarrow \mathbf{z}_{n,\, b_-^{(n)}}$ ▷ short positive / negative forks
      continue $b_+^{(n)}$ in Env_n for L_{main} chunks $\rightarrow \mathbf{z}_{\text{main}}^{(n)}$ ▷ long mainline rollout
      push $\big(\mathbf{z}_{\text{main}}^{(n)}, \mathbf{z}_+^{(n)}, \mathbf{z}_-^{(n)}, \mathbf{c}_n\big)$ to $\mathcal{D}$ with priority $s_{n,\, b_+^{(n)}}$, evicting low-priority entries
    end for
  end if
  draw $(\mathbf{z}_{\text{main}}, \mathbf{z}_+, \mathbf{z}_-, \mathbf{c}) \sim \mathrm{Uniform}(\mathcal{D})$ and increment its usage counter ▷ training step
  $\mathcal{L}_{\text{FM}} \leftarrow$ flow-matching loss on $\mathbf{z}_{\text{main}}$ (Eq. 1)
  $\mathcal{L}_{\text{CFT}} \leftarrow$ **CFT_Loss**$\big(\mathbf{z}_+, \mathbf{z}_-, \mathbf{c}\big)$ (Algorithm 3)
  $\theta \leftarrow \theta - \eta\, \nabla_\theta \big(\mathcal{L}_{\text{FM}} + \lambda_{\text{CFT}}\, \mathcal{L}_{\text{CFT}}\big)$
  $\bar{\theta} \leftarrow \gamma\, \bar{\theta} + (1 - \gamma)\, \theta$ ▷ EMA update
end for
return θ

Algorithm 2 **AOL_Select**: scoring, scene/action decomposition, and rank-inverted fork selection.

Require: fork latents $\{\mathbf{z}_{n,b}\}_{n,b}$ with shared conditioning $\{\mathbf{c}_n\}$, EMA model $\bar{\theta}$, scoring noise $\sigma^\star$ at timestep index $k^\star$, override probability p_{rand}

sample shared noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for all (n, b) **do** ▷ single EMA forward per fork
  $\tilde{\mathbf{z}}_{n,b} \leftarrow (1 - \sigma^\star)\, \mathbf{z}_{n,b} + \sigma^\star\, \boldsymbol{\epsilon}$
  $s_{n,b} \leftarrow \big\| \mathbf{v}_{\bar{\theta}}\big(\tilde{\mathbf{z}}_{n,b},\, k^\star,\, \mathbf{c}_{n,b}\big) - (\boldsymbol{\epsilon} - \mathbf{z}_{n,b}) \big\|_2^2$
end for
$d_n \leftarrow \mathrm{mean}_b\, s_{n,b}, \quad \delta_{n,b} \leftarrow s_{n,b} - d_n$ ▷ scene / action decomposition
for $n = 1, \ldots, N$ **do**
  if $\mathrm{rand}() < p_{\text{rand}}$ **then**
    $b_+^{(n)} \leftarrow$ uniform random in $\{1, \ldots, B\}$
  else
    $r_n \leftarrow$ percentile rank of d_n in $\{d_1, \ldots, d_N\}$ ▷ lower $\Leftrightarrow$ easier scene
    $b_+^{(n)} \leftarrow$ fork in world n whose action rank (by $\delta_{n,b}$) matches $1 - r_n$
  end if
  $b_-^{(n)} \leftarrow \arg\max_{b \neq b_+^{(n)}} \mathrm{sim}\big(\mathbf{z}_{n,\, b_+^{(n)}},\, \mathbf{z}_{n,b}\big)$ ▷ counterfactual fork
end for
return $\{(b_+^{(n)}, b_-^{(n)})\}_{n=1}^{N}$ and the score table $\{s_{n,b}\}$

batch size of 16 per optimizer update. The $N = 32$ Minecraft simulators are evenly distributed across the four nodes, where the leader rank on each node drives its assigned simulators and broadcasts the resulting raw rollout frames to its peers, so that VAE encoding and EMA scoring run in parallel on every rank within an SP group. After each collection round, the new samples are merged into a globally synchronized replay buffer, ensuring that the uniform draw at the training step sees a consistent buffer view across all nodes.

Algorithm 3 CFT_Loss: causality-aware fine-tuning loss on a counterfactual sample pair.

Require: positive–negative pair $(\mathbf{z}_+, \mathbf{z}_-)$ sharing condition $\mathbf{c}$, current model $\mathbf{v}_\theta$, EMA model $\mathbf{v}_{\text{old}}$, hyperparameters λ and β

$h \leftarrow \text{sim}(\mathbf{z}_+, \mathbf{z}_-), \quad \alpha \leftarrow h/2$ ▷ visual similarity (e.g., SSIM) and reconstruction weight

sample $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and timestep k from the shifted logit-normal schedule, with noise level σ_k

$\gamma_k \leftarrow \lambda\, h\, \sigma_k(1-\sigma_k)$

$\tilde{\mathbf{z}}^k \leftarrow (1-\sigma_k)\,\mathbf{z}_+ + \sigma_k\,\boldsymbol{\epsilon} + \gamma_k\,(\mathbf{z}_- - \mathbf{z}_+)$ ▷ counterfactual perturbation, Eq. 4

forward both networks at $(\tilde{\mathbf{z}}^k, k, \mathbf{c})$ to obtain $\mathbf{v}_\theta$ and $\mathbf{v}_{\text{old}}$

$\mathbf{v}^+ \leftarrow (1-\beta)\,\mathbf{v}_{\text{old}} + \beta\,\mathbf{v}_\theta, \qquad \mathbf{v}^- \leftarrow (1+\beta)\,\mathbf{v}_{\text{old}} - \beta\,\mathbf{v}_\theta$

$\hat{\mathbf{z}}_+ \leftarrow \tilde{\mathbf{z}}^k - \sigma_k\,\mathbf{v}^+, \qquad \hat{\mathbf{z}}_- \leftarrow \tilde{\mathbf{z}}^k - \sigma_k\,\mathbf{v}^-$

$\mathcal{L}_{\text{CFT}} \leftarrow (1-\alpha)\,\|\hat{\mathbf{z}}_+ - \mathbf{z}_+\|_w^2 + \alpha\,\|\hat{\mathbf{z}}_- - \mathbf{z}_-\|_w^2$ ▷ Eq. 6

return $\mathcal{L}_{\text{CFT}}$

A.4 Inference and Evaluation

Action Accuracy via Estimated Camera Poses To assess action controllability, we recover discrete action tokens from the generated video and compare them against the ground truth. We first apply Depth Anything V3 (Lin et al., 2025) to both the generated and the ground-truth videos to estimate per-frame world-to-camera poses, subsample the estimates to the latent rate, and compute the relative camera motion between consecutive latent frames. The relative translation and rotation are then independently quantized—using fixed magnitude thresholds—into a 9-way movement label and a 5-way view label, and combined into a single compound token $\hat{a}_n$ in the same 45-entry space as our action vocabulary. Action controllability is measured by two complementary token-level accuracies:

$$\text{Acc}_{\text{combined}} = \frac{1}{N-1}\sum_{n=1}^{N-1} \mathbb{1}[\hat{a}_n = a_n], \qquad \text{Acc}_{\text{fine}} = \tfrac{1}{2}\big(\text{Acc}_{\text{move}} + \text{Acc}_{\text{view}}\big), \tag{10}$$

where $\text{Acc}_{\text{combined}}$ requires both the movement and the view label to be recovered correctly (Combined in Tab. 2), while Acc_{fine} averages the two axis-wise accuracies (Fine-grained in Tab. 2). For each method, we sweep the translation threshold over a small candidate set and report the configuration with the highest $\text{Acc}_{\text{combined}}$ following WorldCompass (Wang et al., 2026), which compensates for the per-scene scale ambiguity inherited from the depth estimator.

Relative Pose Error For trajectory-level controllability, we report the relative pose error (RPE) between the predicted and the ground-truth camera trajectories. The predicted trajectory is first registered to the ground truth through a closed-form Sim(3) Umeyama alignment (Umeyama, 1991) on the camera centers, which absorbs the estimator's per-scene scale and any global frame offset. We then compare the predicted and ground-truth incremental motions of consecutive frame pairs, and report two trajectory-averaged numbers: the translational RPE, computed as the mean Euclidean norm of the residual translation between aligned and ground-truth motions, and the rotational RPE, computed as the mean geodesic angle of the residual rotation, reported in degrees.

B Additional Experiments

B.1 Additional Ablation Experiments for AOL and CFT

We extend Tab. 2 with two variants: *Random + CFT*, which applies CFT to randomly acquired training branches, and *AOL + CFT (random negative)*, which replaces the most visually similar counterfactual branch with a random alternative from the same conditioning state. All variants use the same optimizer settings, EMA decay, buffer size, inference settings, and evaluation protocol reported in Sec. 4.1 and Tab. 6.

CFT without Active Acquisition CFT improves all four metrics under random acquisition, while yielding larger action-accuracy gains with AOL. The full model performs best across all four metrics, supporting the complementary roles of active acquisition and counterfactual supervision.

Tab. 7: Additional ablations on *action controllability*. Best results are in **bold**.

Method	Action Accuracy		Camera Control	
	Combined ↑	Fine-grained ↑	RPE_{rot} ↓	RPE_{trans} ↓
Base Model	0.240	0.490	0.6906	0.0235
+ *Random*	0.356	0.569	0.6393	0.0199
+ *Random* + *CFT*	0.369	0.596	0.5023	0.0172
+ *AOL*	0.393	0.610	0.5033	0.0166
+ *AOL* + *CFT (random negative)*	0.410	0.622	0.4670	0.0163
Full Model (*w/ CFT*)	**0.425**	**0.640**	**0.4072**	**0.0159**

Counterfactual Negative Selection Random negatives improve all four metrics over *AOL*, while visually similar negatives yield further gains. This supports using similar counterfactual outcomes to better distinguish the effects of different actions.

B.2 Sensitivity to the Scoring Noise Level

The scoring noise level $\sigma^\star$ in Eq. 2 directly affects the difficulty estimates used for AOL selection. We vary $\sigma^\star$ from 0.3 to 0.7 and evaluate the stability of the resulting branch rankings relative to the default $\sigma^\star = 0.5$. Tab. 8 reports Kendall's τ_b and the percentages of rank differences within one or two positions. The ranking is stable around the default setting, with most changes occurring among branches with very similar scores, suggesting that our ranking mechanism is not overly sensitive to the particular choice of $\sigma^\star$.

Tab. 8: Stability of AOL acquisition rankings across scoring noise levels, measured relative to the default $\sigma^\star = 0.5$.

$\sigma^\star$	Kendall's τ_b ↑ (p-value)	Rank difference ≤ 1 ↑	Rank difference ≤ 2 ↑
0.3	0.8867 ($\approx 10^{-5}$)	86.72%	99.22%
0.4	0.9386 ($\approx 10^{-5}$)	92.97%	100.00%
0.5	1.0000 (N.A.)	100.00%	100.00%
0.6	0.9263 ($\approx 10^{-5}$)	92.97%	99.22%
0.7	0.8320 ($\approx 10^{-5}$)	75.78%	91.41%

B.3 Using the Most Difficult Fork for AOL

We also conduct a separate sampling ablation comparing random acquisition with *Hardest*, which selects the **highest-scoring training fork** in each scene, $b_+^{(n)} = \arg\max_b s_{n,b}$. However, it significantly underperforms random acquisition on all six metrics and shows high training instability at early training stage (see Tab. 9 for validation results), suggesting that prioritizing prediction error alone is insufficient for determining useful training data.

Tab. 9: Comparison of random and hardest-fork acquisition.

Method	SSIM ↑	LPIPS ↓	Combined ↑	Fine-grained ↑	RPE_{rot} ↓	$\text{RPE}_{\text{trans}}$ ↓
Random	0.485	0.480	0.486	0.693	1.3642	0.2191
Hardest	0.457	0.561	0.404	0.604	2.0324	0.3530

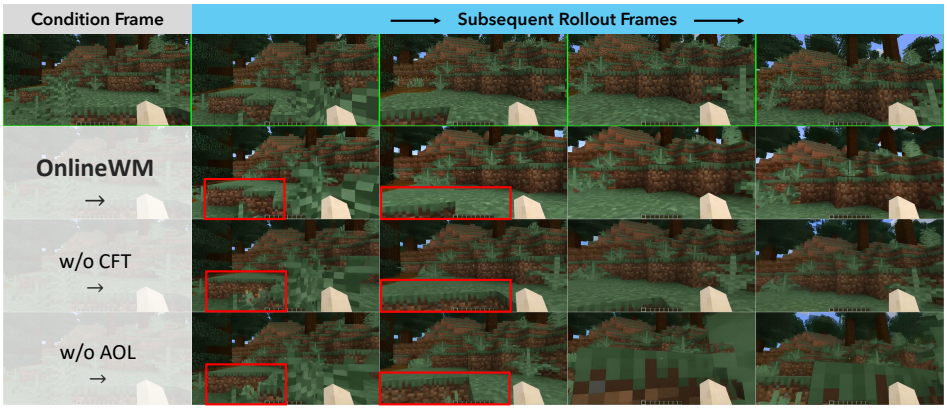
C Limitations and Future Directions

In this work, we build OnlineWM on top of Minecraft, which offers flexibility, efficiency, and rich action coverage well-suited for studying causally grounded action control. While Minecraft already supports the diverse scenarios and interactions explored in this work, modern game engines such as Unity and Unreal Engine 5 further provide photorealistic rendering, finer-grained physics, and richer motion patterns including articulated character dynamics, deformable objects, and complex environmental interactions. Bringing OnlineWM into these engines holds substantial promise for scaling up the learning of world transition dynamics and improving visual fidelity. However, it's worth noting that efficient engineering design of the online training infrastructure is also crucial to support the practicability of OnlineWM when using these advanced game engines.

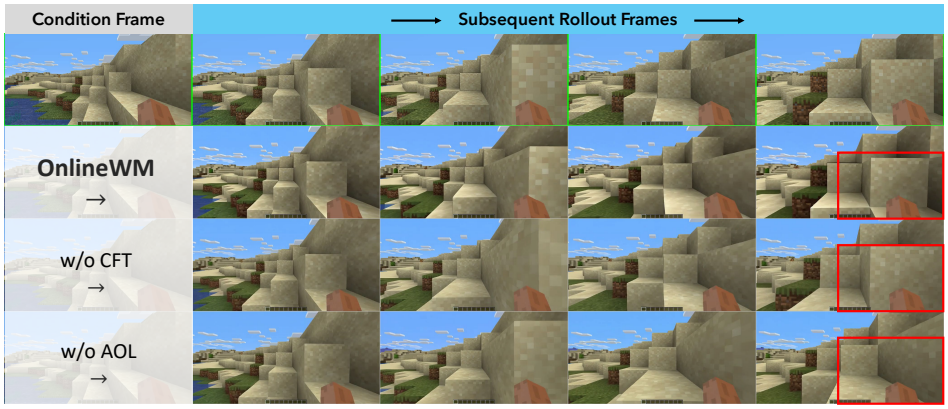
Beyond modeling scene-level transition dynamics, the closed-loop and causality-driven nature of OnlineWM also points toward learning *physical dynamics* through embodied interaction. Since the active online learning loop and causality-aware fine-tuning are agnostic to the specific form of action, they could in principle be extended to settings where **custom agents continuously interact with the environment in pursuit of informative experience**. For instance, a robotic manipulator might autonomously probe a simulated workspace, actively seeking states whose contact outcomes are novel yet learnable for the current world model, and constructing counterfactual rollouts—*e.g.*, grasping versus pushing, or applying different forces from the same configuration—to anchor its predictions in genuine physical causality rather than visual co-occurrence. We see such an extension as a promising avenue for moving from passive observation of dynamics toward an interactive, embodied paradigm of physical world understanding, and leave its concrete realization to future work.

D MORE VISUALIZATION

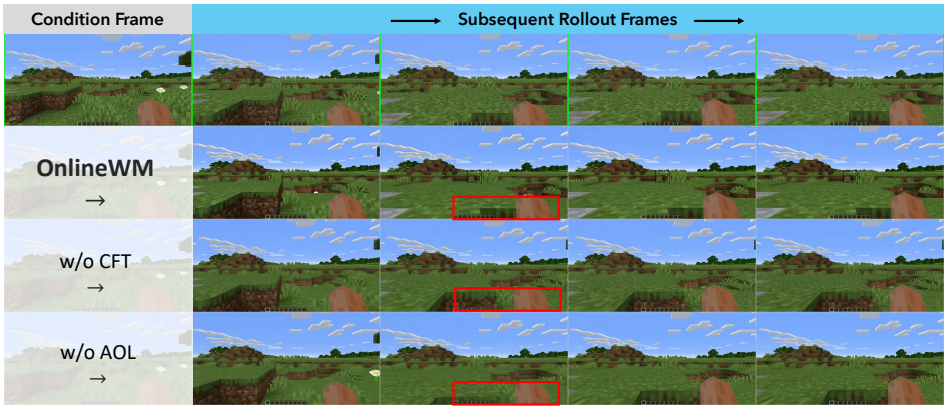
We include more qualitative visualization results in this appendix, including comparisons of action controllability for different method variants (Fig. 5) and generalization results to general domains (Fig. 6, 9, 7, 8).



(a)



(b)



(c)

Fig. 5: More visual comparison of all method variants for action controllability with environmental collisions. The first row shows the ground-truth video, and subsequent rows denote the generation results of three variants: *OnlineWM*, *OnlineWM w/o the CFT loss*, and *OnlineWM* ***removing the whole AOL and CFT design***.

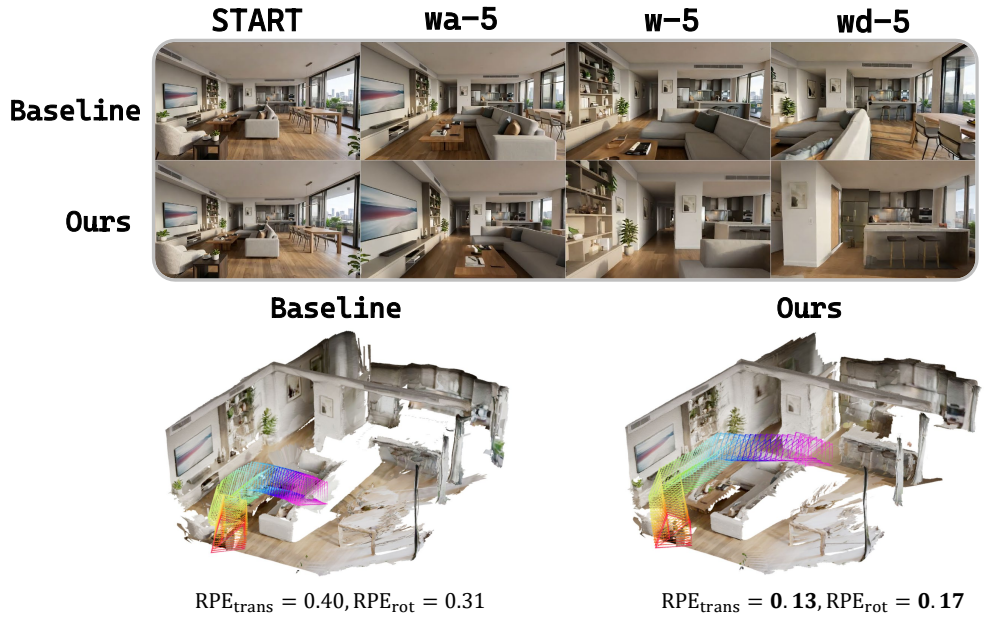


Fig. 6: Visual comparison of the generalization ability of learned action control on general domains.

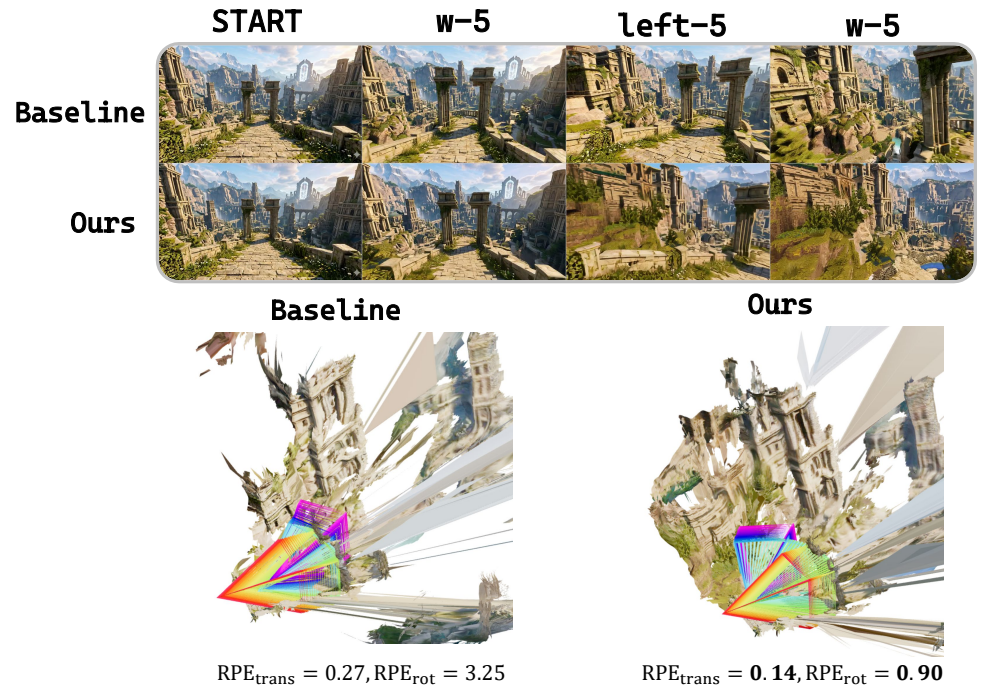


Fig. 7: Visual comparison of the generalization ability of learned action control on general domains.

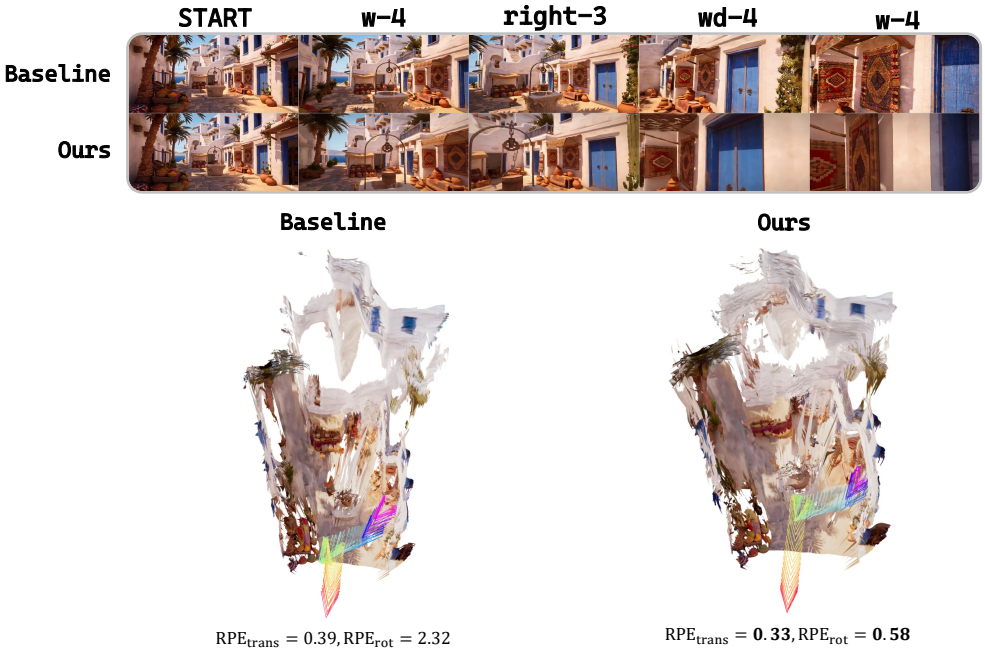


Fig. 8: Visual comparison of the generalization ability of learned action control on general domains.

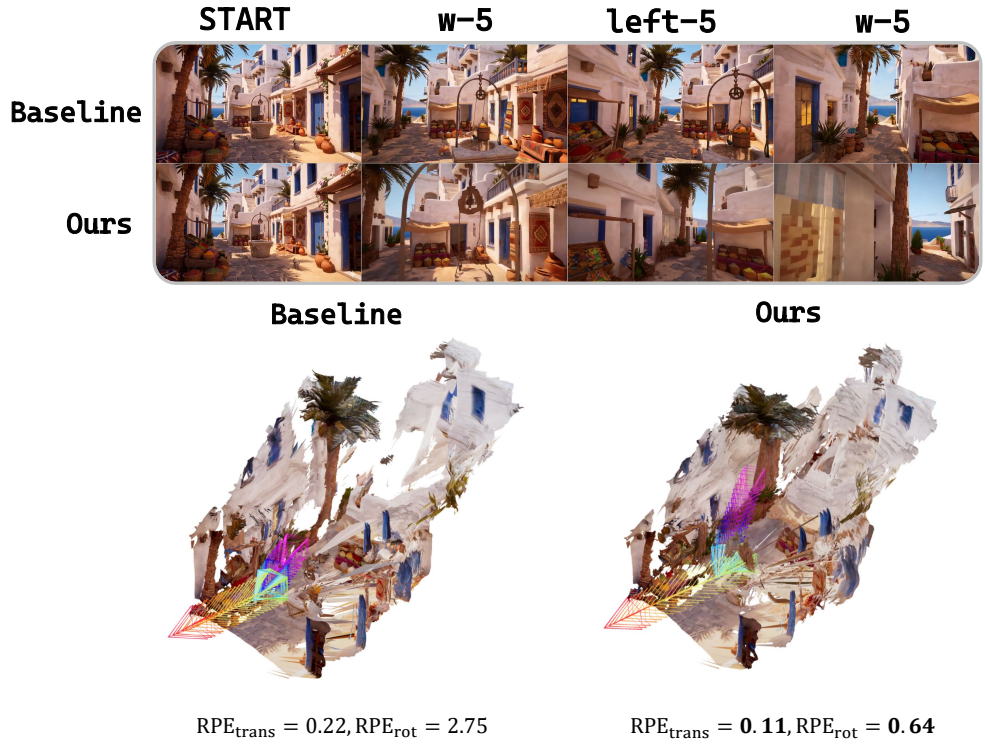


Fig. 9: Visual comparison of the generalization ability of learned action control on general domains.