A Large-Scale Game Video Dataset with Action and State Annotations

Zhen Li[1,2,4*§], Zian Meng[2,3*], Shuwei Shi[2], Wenshuo Peng[5],
Bo Zheng[2], Yunde Jia[4,1], Yuwei Wu[1,4†], Chuanhao Li[2†], and Kaipeng Zhang[2†]

[1]Beijing Key Laboratory of Intelligent Information Technology,
School of Computer Science & Technology, Beijing Institute of Technology,
[2]Alaya Studio, [3]Shanghai Innovation Institute,
[4]Guangdong Laboratory of Machine Perception and Intelligent Computing,
Shenzhen MSU-BIT University, [5]Tsinghua University

https://alaya-studio.github.io/wildworld-project/

Abstract. Dynamical systems theory and reinforcement learning view world evolution as the dynamics of latent states driven by actions, with visual observations providing partial information about those states. Recent video world models attempt to learn these action-conditioned dynamics from data. However, existing datasets rarely meet these requirements: they typically lack diverse and semantically meaningful action spaces, and actions are directly tied to visual observations rather than mediated by underlying states. As a result, actions are often entangled with pixel-level changes, making it difficult for models to learn structured world dynamics and maintain consistent evolution over long horizons. In this paper, we propose WildWorld, a large-scale action-conditioned world modeling dataset with explicit state annotations, automatically collected from a photorealistic AAA action role-playing game (*Monster Hunter: Wilds*). WildWorld contains over 108 million frames and features more than 450 actions, including movement, attacks, and skill casting, together with synchronized per-frame annotations of character skeletons, world states, camera poses, and depth maps. We further derive WildBench to evaluate models through Action Following and State Alignment. Extensive experiments reveal persistent challenges in modeling semantically rich actions and maintaining long-horizon state consistency, highlighting the need for state-aware video generation.

1 Introduction

Understanding and predicting how the world evolves from observations is one of the central goals of artificial intelligence [12,27,38]. Both dynamical systems the-

§ This work was done during the internship at Alaya Studio.
* Equal contribution.
† Corresponding authors: wuyuwei@bit.edu.cn; chuanhao.li@shanda.com; kaipeng.zhang@shanda.com

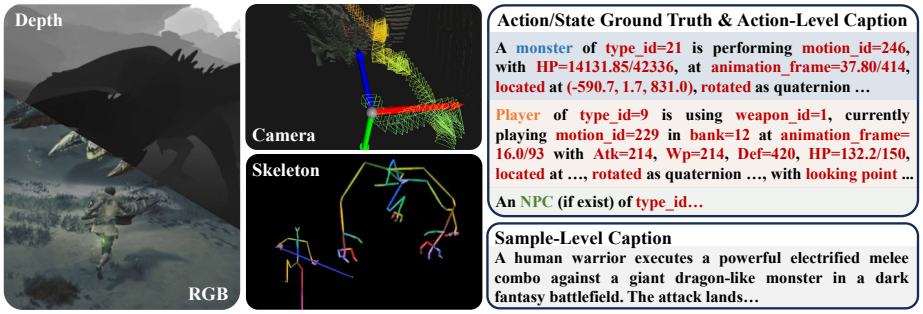

Fig. 1: We present a large-scale dataset curated from game engines for dynamic world modeling. It contains RGB frames with aligned depth maps, camera poses, skeletons, and action/state ground truth. We provide both fine-grained action-level captions and sample-level captions, making the dataset applicable to various experimental settings.

ory [2, 14] and reinforcement learning [43] typically model the world as a latent-state dynamical process, where the environment evolves through state transitions driven by actions. From this perspective, visual observations are merely partial and noisy projections of the true system state. Therefore, learning a predictive model of the world requires inferring latent states and modeling their action-conditioned state transitions. Such world models are crucial for enabling agents to plan, reason, and interact with complex environments over long horizons.

Recent years have witnessed significant progress in video generation and world models [13, 37, 45]. Many recent approaches [18, 25, 34] attempt to learn environment dynamics from large-scale video datasets by training generative models that predict future frames conditioned on past observations and actions. However, despite the increasing capability of such models, existing datasets remain insufficient for effectively learning structured action-conditioned dynamics. Most existing datasets provide only simple action annotations with limited semantic meaning, such as basic movements or camera rotations [31, 46]. Moreover, the effects of these actions are often directly visible in the resulting frames. For example, the action "move left" is typically reflected in the video as a corresponding change in viewpoint.

However, in many cases, actions do not manifest as explicit changes in observations but instead affect the world through implicit state transitions. For instance, the action "shoot" implicitly affects internal state variables such as the "remaining ammunition count". This state cannot be reliably inferred from visual observations alone, yet it plays a crucial role in determining future visual outcomes. When the remaining ammunition reaches zero, executing the shoot action will no longer produce firing effects or projectiles, leading to visual results that differ significantly from those observed when ammunition is available. Such coupling makes it difficult for models to disentangle state transitions from observation variations, thereby hindering the learning of stable and interpretable world dynamics. As a result, current models often perform poorly in long-horizon

prediction tasks, where small errors accumulate over time and eventually lead to noticeable inconsistencies or instability in the generated results.

In this paper, we propose WildWorld, a large-scale video dataset for action-conditioned world modeling with explicit state annotations. The dataset is automatically collected from the photorealistic AAA action role-playing game *Monster Hunter: Wilds*. As illustrated in Fig. 1, WildWorld features a rich and semantically meaningful action space containing over 450 actions, including movement, attacks, and skill casting. To facilitate data collection, we develop a bespoke toolchain capable of recording per-frame ground-truth annotations, including player actions, character skeletons, world states, camera poses, and depth maps. The toolchain is integrated with an automated gameplay pipeline, allowing the dataset to scale easily to over 108 million frames of gameplay footage while covering diverse interactive scenarios. By capturing complex interactions and the underlying state transitions, WildWorld enables the study of long-horizon compositional action sequences and their effects on evolving world states, providing a valuable foundation for building, training, and systematically evaluating state-aware interactive world models.

Furthermore, we derive WildBench, a benchmark constructed from WildWorld for evaluating interactive world models. WildBench introduces two key evaluation metrics: Action Following and State Alignment. Specifically, Action Following measures the agreement between generated videos and ground-truth actions using a large vision-language model. State Alignment quantitatively measures the accuracy of state transitions by tracking skeletal keypoints in the generated videos and comparing them with the corresponding ground-truth annotations. We design several baseline models for state-aware video generation and compare them with existing approaches on WildBench. The experimental results reveal the limitations of current models and provide insights for future research, particularly in improving state transition modeling and long-horizon consistency.

To summarize, our contributions are threefold:

- We propose **WildWorld**, a large-scale video dataset comprising over 108 million frames, with a rich action space and diverse frame-level ground-truth annotations, including player actions, character skeletons, world states, camera poses, and depth maps.
- We curate **WildBench**, a benchmark for evaluating interactive world models, featuring two carefully designed metrics: Action Following and State Alignment.
- We conduct extensive experiments and analysis on WildBench, which provide insights into the future development of interactive world models.

2 Related Work

2.1 Interactive World Models

Recent advances in video generation models [6, 13, 41, 51] have enabled the development of interactive world generation models [1, 3, 45]. In the realm of video

generation, text-to-video [5, 29, 35] and image-to-video generation [39, 45, 49, 50] have achieved remarkable progress in visual quality and temporal consistency. As for interactive video generation, some works [11, 34, 37] enable interaction by switching prompts during the generation process, whereas others [9, 11, 18, 21, 24, 30, 52, 56] introduce actions via keyboard control [11, 18] and camera poses [16, 17] on top of image-to-video generation to control the generated video. Despite their promising results, these methods use restricted action spaces and tightly couple action control with pixel-level visual changes. In contrast, we focus on state-aware video generation controlled by actions, which features a rich action space with over 450 actions and uses states as an intermediate representation between actions and generated frames.

Some recent works [10, 32, 47, 48, 55] attempt to introduce latent state representations into video generation models to better capture environment dynamics. However, these approaches typically represent the world state as an implicit latent variable learned from visual observations. By contrast, we focus on explicit, semantically meaningful states and introduce WildWorld, a large-scale dataset with state annotations for learning and analyzing state dynamics.

2.2 Video Generation Datasets

Recent progress in video generation has been driven by several large-scale datasets, such as OpenVid-1M [36], MiraData [26], Open-Sora [33], and SpatialVID [46], which provide large collections of internet videos for training generative models. More recent works have begun to explore datasets for world modeling or interactive video generation, including OmniWorld [58], Sekai [31], GF-Minecraft [54], PLAICraft [19], GameGen-X [4], and GTASA [7]. While these datasets introduce gameplay videos or action signals to capture environment dynamics, they still primarily rely on visual observations and lack explicit, semantically meaningful state representations. MIND [53] proposes a benchmark for evaluating memory consistency and action control in world models. Compared with the aforementioned works, WildWorld provides explicit state annotations, including character skeletons, world states, camera poses, and depth maps. These annotations enable models to learn structured state dynamics and support direct evaluation of state alignment and action following.

3 WildWorld Dataset

The WildWorld curation process comprises three major components: a data acquisition platform, an automated gameplay pipeline, and a data processing and caption annotation pipeline, as illustrated in Fig. 2.

3.1 Data Acquisition Platform

In the data acquisition stage, we collect the interaction data required for training and evaluating interactive world models and organize them into three categories:

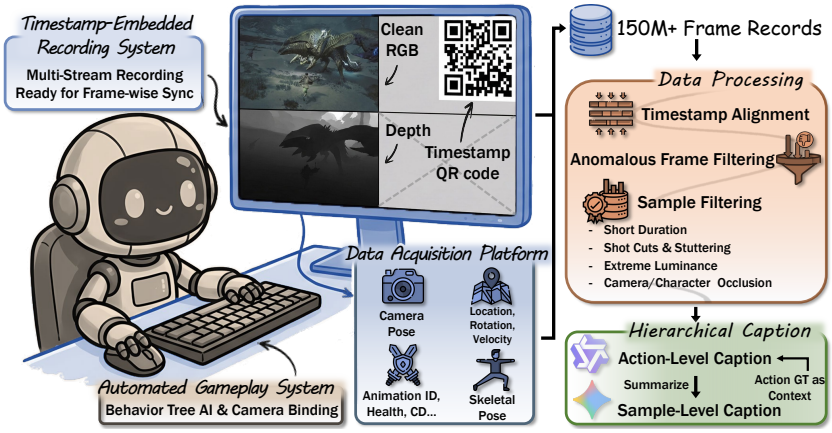

Fig. 2: The WildWorld dataset curation pipeline.

actions, states, and observations. Actions specify the control inputs that drive interactions, states describe the underlying evolution of the game world, and observations correspond to its visual manifestations. These three types of data can be recorded at different stages of modern game execution. In *Monster Hunter: Wilds*, the game engine processes player inputs, maintains and updates the world state, while the rendering pipeline consumes information from the game engine to produce the final imagery. Following this separation, we develop a dedicated game data acquisition platform engineered for high-fidelity recording of various categories of data.

Specifically, our data acquisition platform records player actions as they are executed and captures ground-truth world states, including the executed actions, the absolute locations, rotations, and velocities of the player character and monsters, their current animation IDs, and gameplay attributes such as health and resources analogous to stamina or mana. We additionally record the skeletal poses of both the player character and monsters. For world observations, we instrument the rendering pipeline to record RGB frames, depth maps, and the intrinsic and extrinsic parameters of the in-game camera. We further remove the HUD by disabling the corresponding late-stage shaders. This yields clean, HUD-free frames that better reflect the game world for training and evaluating interactive world models.

3.2 Automated Gameplay Recording Pipeline

Turning the captured raw streams into a usable and scalable dataset requires solving several challenges at the system level. On the one hand, to enable long-running data collection with minimal human intervention, we implement an automated gameplay system that handles in-game menu navigation and executes player actions. On the other hand, to record the different types of interaction data

captured by separate tools, we design a robust recording system with embedded timestamps to facilitate subsequent synchronization and alignment across sources.

Automated Gameplay System. *Monster Hunter: Wilds* follows a quest-based structure. In each session, a party of up to four characters, comprising one player-controlled protagonist and three NPC companions, hunts one or two large monsters. Our automation consists of two components. For quest selection, we invoke the game engine's UI components to programmatically navigate in-game menus and randomly sample combinations of quests and NPCs, ensuring diverse coverage of maps, monsters, and team compositions. For automated combat, we use the behavior trees that drive NPC companions to fight autonomously and adjust the in-game camera binding correspondingly so that the entire party can act without human input. A natural concern is whether rule-based AI yields overly repetitive behavior. We argue that the resulting trajectories remain sufficiently diverse for two reasons. First, the combinatorial action space is large: the AI must select among dozens of moves and continuously adjust timing and positioning in response to monster behavior, which itself is stochastic. Second, the interaction between multiple AI-controlled characters and a reactive monster creates a high-dimensional dynamical system whose trajectories vary substantially across sessions, even under the same scripted logic. During automated combat, the camera is managed by the game's native target-lock system, which dynamically adjusts the camera position and angle to keep the engaged monster within the field of view while maintaining visual stability.

Timestamp-Embedded Recording System. We develop a recording system that simultaneously captures interaction data from multiple sources. For structured information represented in text form, such as actions and states, recording is straightforward. At each engine tick, these interaction data are uniformly recorded, serialized in JSON format, and written to a local file. Given that the full screen is typically occupied by the RGB frame in standard rendering setups, image data such as RGB and depth require a different strategy for simultaneous recording. To achieve this, we develop a dedicated system based on OBS Studio and ReShade. Specifically, a custom ReShade shader partitions the full display into four sub-windows, two of which present the RGB and depth frames from the rendering buffer. In practice, we set the full display resolution to 2K, yielding sub-windows of 720p. We further adapt a modified version of OBS Studio to simultaneously record different sub-windows of the screen as separate recording streams, allowing RGB and depth to be recorded with different encoding settings. Specifically, RGB is recorded with lossy HEVC compression under variable bitrate control, using a target bitrate of 16 Mbps and a maximum bitrate of 20 Mbps to reduce storage costs while maintaining high visual quality. In contrast, depth is recorded losslessly to preserve geometric precision and avoid discontinuities caused by lossy compression. In practice, we use the HEVC encoder with B frames enabled, and the resulting bitrate of the depth stream remains around 20 Mbps. In addition, we embed timestamps in recordings from multiple sources, providing a common basis for synchronization and subsequent process-

ing. Specifically, for image frames, we develop a simple program that renders the timestamp as a QR code at fixed intervals and displays it in one of the two unused sub-windows.

3.3 Data Processing and Annotation Pipeline

The temporally tagged, multi-source recordings collected in the previous stage contain rich action, state, and observation data. However, occasional runtime instability can cause misalignment and duplicated or dropped frames. The recordings may also contain low-quality or uninformative content, such as occlusions and cutscenes, making them unsuitable for direct use by interactive world models. We therefore build a two-stage processing pipeline. We first align the recordings by timestamp and filter anomalous frames to construct samples that are synchronized at the frame level. We then apply multiple filters to remove low-quality samples. Based on the resulting samples, we further annotate hierarchical captions to support fine-grained modeling and evaluation.

Timestamp Alignment and Anomalous Frame Filtering. To obtain aligned samples from multiple recordings, we first align them by timestamp and then filter them for temporal continuity. We take the text recordings, which contain action and state records synchronized with the game engine update, as the reference timeline, and align the RGB frames to them according to their recorded timestamps. Rather than applying simple nearest-neighbor matching, we use a bipartite matching algorithm that enforces one-to-one, order-preserving matches across recordings. The algorithm maximizes the number of matched frames while ensuring that the temporal deviation of each matched pair remains within a prescribed range, making the alignment robust to duplicated or missing RGB frames. In practice, under the 30 FPS recording setting, the prescribed range is asymmetric and set to $[-45, 15]$ milliseconds. This means that an RGB frame may lag behind the reference timeline by up to 45 milliseconds but may lead it by at most 15 milliseconds. This design ensures that moderate capture delay (one frame) is tolerated, while preventing the alignment from associating the current reference state with future visual content. In this sense, the matching procedure is not only timestamp based, but also causality aware. After alignment, we identify discontinuities caused by unmatched frames and use them to partition each recording into individual samples.

Sample Filtering. We filter the samples along the following dimensions to improve the overall data quality.

- **Duration Filtering**. Very short samples provide limited value for interactive world modeling. We therefore discard samples shorter than 81 frames.
- **Temporal Continuity Filtering**. Given the state record for every frame in a sample has a timestamp, we can directly measure the temporal gap between adjacent frames. Excessive gaps typically indicate either stuttering in the game or recording system, or transitions into non-combat content such as cutscenes. The latter can be identified in our data, since our platform only records data during combat or travel. We discard any sample in which the

gap between two adjacent frames exceeds 1.5 times the target frame interval, *i.e.*, approximately 50 ms at 30 FPS.
- **Luminance Filtering**. Overly bright or dark visuals in games can create visually distinctive gameplay experiences, for example in combat effects or in nighttime scenes. However, such samples are less suitable for stable model training. We apply a simple filter based on the luma channel of the RGB frames in YUV color space and remove samples with more than 15 consecutive frames of extremely high or low average brightness.
- **Camera Occlusion Filtering**. We remove samples with foreground occlusions, such as rocks, trees, or other scene geometry blocking the character. We detect such cases using the spring-arm behavior of the third-person camera. When occlusion occurs, the arm contracts, leading to an abnormally small camera-to-character distance. We therefore discard samples whose recorded distances fall below a threshold for a sustained number of frames. We further exclude samples with abrupt player position changes, such as fast travel, as they break visual continuity.
- **Character Occlusion Filtering**. Severe character overlap in the first frame can introduce ambiguity into image-to-video generation. We identify overlap between characters by projecting 3D skeletal keypoints onto screen coordinates in the first frame and discarding samples in which the overlap area exceeds 30% of either character's projected area.

Hierarchical Caption Annotations. Fine-grained captions are important for capturing interaction details and enabling the training of more precisely controllable models, for example through prompt switching [25, 51]. Leveraging the action annotations provided in WildWorld, we divide each sample into action segments according to the frame-level action ID annotations, such that the action remains unchanged within each segment, *e.g.*, walking forward or charging a heavy attack. For each segment, we sample RGB frames at 1 FPS, resize them to 480p, and use Qwen3-VL-235B-A22B-Instruct served with vLLM to generate detailed captions. To compensate for the model's limited familiarity with game-specific scenarios, we additionally include the corresponding action and state ground truth in the prompt context. We further provide sample-level captions by summarizing the action-segment captions within each sample using Gemini 3 Flash.

3.4 Dataset Statistics

After processing and filtering, we obtain WildWorld with over 108 million frames. **Entity Diversity**. The dataset covers 29 unique monster species, 4 player characters, and 4 weapon types (Great Sword, Long Sword, Bow, Dual Blades). As shown in Fig. 3 (a), character types and weapon types are near-uniformly distributed, while monster species follow a long-tailed distribution dominated by a few frequent targets. Multi-monster encounters also appear, with 7 secondary species present in the data. This diversity supports the training of world models that generalize across entities and interaction patterns.

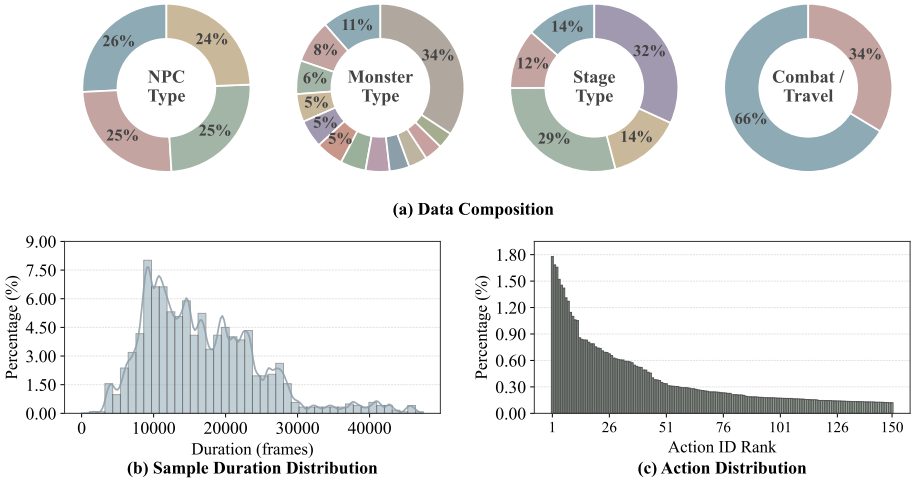

Fig. 3: Overview of WildWorld dataset statistics. (a) Data composition by character type, monster species, stage, and combat/travel ratio. (b) Distribution of sample durations in frames. (c) Frequency distribution of the top-150 action IDs, exhibiting a long-tail pattern.

Scene Complexity. Gameplay spans 5 distinct stages set in an open-world map with diverse environments including deserts, snowy mountains, forests, swamps, and wastelands, under varying weather (sunny, rainy) and time-of-day (day, night) conditions. As shown in Fig. 3 (a), approximately 66% of samples capture active combat, while the remaining 34% depict traversal on mounts, providing a broad range of interaction contexts for training and evaluating world models.

Temporal and Spatial Dynamics. Fig. 3 (b) shows the distribution of sample durations. The majority of samples span 4,000 to 28,000 frames, while a smaller subset exceeds 40,000 frames (over 30 minutes of gameplay), capturing extended combat sequences or exploration that demand long-horizon consistency. Spatially, the camera-to-character distance has a median of 15.69 units and the character-to-monster distance a median of 12.63 units. These close proximities ensure that the character and monster are prominently featured in the video frames, with clear visibility of their actions and state changes.

Action Richness. Each frame's character state is encoded as a (weapon type, bank ID, motion ID) triplet, yielding 5,960 unique character action triplets across 24 banks and 455 motion IDs. These actions span movement, attacks, evasion, defense, item usage, and transitions between actions, covering the full range of in-game interactions. Monsters exhibit 2,132 unique action pairs across 13 banks and 527 motion IDs. Fig. 3 (c) shows the frequency of the top-150 character action IDs, which account for 58.49% of all samples and follow a long-tail distribution, indicating rich behavioral variety.

4 WildBench Benchmark

Evaluating interactive world models requires measuring not only visual plausibility but also how well the model follows input actions and produces aligned states. Leveraging the action and state ground truth provided by WildWorld, we evaluate generated videos from four perspectives: video quality, camera control, action following, and state alignment. Among these, action following and state alignment directly evaluate how well a model follows input actions and produces the corresponding state transitions, both of which are central to interactive world modeling. This distinguishes our benchmark from existing ones [8, 28, 53], which mainly emphasize perceptual quality, controllability, or physics plausibility.

4.1 Evaluation Metrics

We comprehensively evaluate interactive world models from four perspectives:
Video Quality characterizes the overall perceptual quality of generated videos in terms of both motion and appearance. We evaluate this dimension using four VBench metrics [23]: Motion Smoothness (MS) assesses the smoothness and physical plausibility of generated motion; Dynamic Degree (DD) measures the magnitude of motion to penalize overly static videos; Aesthetic Quality (AQ) reflects the perceived artistic and visual appeal of the generated content; Image Quality (IQ) evaluates low-level visual distortions, such as overexposure, noise, and blur.
Camera Control is essential for interactive world models, as inaccurate viewpoint control can prevent the intended observations from being properly presented. We quantitatively evaluate camera control by measuring the discrepancy between the ground-truth camera trajectories and the camera trajectories estimated from generated videos using a structure-from-motion model, following CameraCtrl [15]. To reduce the impact of scale mismatch between trajectories from the game engine and those estimated from video, we apply a scalar alignment factor to the estimated translations before evaluation. We then compute Absolute Trajectory Error (ATE) and Relative Pose Error (RPE) for both translation and rotation [42]. ATE measures the absolute deviation from the ground-truth trajectory, reflecting the overall accuracy of camera control, while RPE measures the discrepancy between relative motions and is therefore more sensitive to local consistency and accumulated drift along the trajectory. In practice, we estimate camera trajectories using ViPE [20].
Action Following evaluates whether the model responds to input actions with the corresponding behaviors in generated videos. Since each sample may contain multiple actions, we evaluate it at the action segment level for a more detailed assessment. Based on the frame-level action ID annotations in WildWorld, we divide each sample into action segments within which the action remains unchanged. For each segment, we extract the corresponding frame range from both the generated video and the ground-truth video, and use Gemini 3 Flash to judge whether they express the same action. We further group actions into three categories, namely movement, fast displacement, and attack, based on their action

IDs, and design detailed prompts for each category. Each segment is assigned a score of 1 if the generated and ground-truth clips are judged to be consistent, and 0 otherwise. The final score is the average over all segments.

State Alignment. We use the poses of the player character and monsters as proxies for state because they directly reflect many underlying world states and can indirectly reveal others, such as death when health reaches zero. Using the ground-truth skeletons in WildWorld, we extract key skeletal points and project them onto screen coordinates to obtain 2D trajectories for each sample. For generated videos, we focus on image-to-video generation settings in which the first frame is the ground-truth frame. Thus, we initialize keypoints from the first frame and track them in generated videos with TAPNext [57]. Then, we define the State Alignment score as the mean coordinate accuracy between the predicted and ground-truth trajectories over all keypoints. For each keypoint, the coordinate accuracy is computed as the average fraction of frames whose predicted locations fall within thresholds of 4, 8, 16, and 32 pixels from the ground truth. We note that while state evolution may be stochastic due to factors such as random events, alignment with the ground truth remains statistically meaningful across a sufficiently large set of samples.

4.2 Data Curation

It is worth noting that all samples in the WildWorld dataset are, in principle, available for evaluation, allowing users to flexibly construct custom test sets based on scenario, difficulty, and other factors. In this paper, we manually curate a representative set of 200 samples covering diverse difficulty levels, combat scenarios, character and monster types, and events such as skill usage, knockdowns, deaths, and critical hits. Among them, 100 samples involve cooperation between the player and NPCs against monsters, while the other 100 consist of one-on-one combat between the player and a monster.

5 Experiments and Analysis

In this section, we first assess the reliability of the proposed benchmark metrics and their alignment with human preference. We then train different interactive world modeling approaches on WildWorld and evaluate them on WildBench using the ground-truth annotations provided in the WildWorld dataset.

5.1 Compared Approaches

Camera-Conditioned Video Generation. In this setting, the model takes a camera trajectory, an initial image, and a text prompt as inputs to generate a video that follows the specified camera motion. We fine-tune the Wan2.2-Fun-5B-Control-Camera [45] model using the ground-truth camera trajectories in WildWorld and dub the resulting model ***CamCtrl***. The baseline model uses a rule-based approach to convert discrete camera control action inputs into camera

poses, from which it computes Plücker embeddings [40] for each frame and injects them into the model. In contrast, we directly use the ground-truth per-frame camera poses in WildWorld as inputs for fine-tuning.

Skeleton-Conditioned Video Generation. Skeletal pose provides a direct and fine-grained representation of character motion. We introduce a skeleton-conditioned setting that takes the first frame and a skeleton video as inputs. We fine-tune Wan2.2-Fun-5B-Control [45], a video-to-video model that supports skeleton-based pose videos as a control signal, and dub the resulting model ***SkelCtrl***. To construct the skeleton video, we use the 3D skeletal keypoints and skeletal hierarchy annotated for each frame in WildWorld. We project the keypoints onto screen coordinates using the ground-truth camera pose and render them as a color-coded skeleton video that matches the input format expected by the baseline model.

State-Conditioned Video Generation. Based on ***CamCtrl***, we design a state-aware model: ***StateCtrl***, which injects states into the video generation process. We first model the states in a structured manner, dividing them into discrete states (*e.g.*, monster type and weapon category) and continuous states (*e.g.*, coordinates and health). Discrete states are mapped to vector representations through trainable embeddings, while continuous states are encoded into the same feature space using an MLP. At the encoding stage, we adopt a hierarchical modeling strategy with entity-level and global-level representations. We encode the state of each entity (*e.g.*, a monster) and also incorporate global states such as recording time. We use the Transformer architecture [44] to model the relationships between entities, producing a unified state embedding. The resulting embedding is aligned with video frames and injected into the intermediate layers of DiT as a conditioning signal to guide the video generation process. In addition, we introduce a state decoder to recover state information from the state embedding and a state predictor to predict the state of the next frame. During training, a decoder loss is used to ensure that the embedding preserves the original states; a predictor loss is used to supervise the state predictor, enhancing the temporal consistency and predictability of the state representation. During inference, ***StateCtrl*** supports using only the ground-truth state of the first frame, while the states of subsequent frames are autoregressively predicted by the state predictor. We denote this model as ***StateCtrl-AR***.

Across all settings, models are trained on 81-frame samples at a resolution of 544×960 and a frame rate of 16 FPS, using a batch size of 1 and a learning rate of 1×10^{-5}. Training is performed on 8 GPUs for 250,000 iterations with the Adam optimizer. During inference, we adopt the same resolution and frame rate, and use 50 sampling steps.

5.2 Overall Evaluation

Evaluation of the Proposed Benchmark Metrics. We validate the reliability of the proposed Action Following and State Alignment metrics, as well as their alignment with human preference. For Action Following, we use the same evaluation protocol with human judgments instead of model-based assessment.

Table 1: Comparison of different interactive video generation approaches trained on WildWorld and evaluated on WildBench. Lower is better for ATE and RPE. Higher is better for all other metrics.

Method	Video Quality				Camera Control		Action	State
	MS	DD	AQ	IQ	ATE(↓)	RPE(↓)	Following	Alignment
Baseline	96.38	99.00	50.81	65.62	4.63	0.18	53.77	11.29
CamCtrl	97.85	97.00	48.29	62.88	2.02	0.13	83.46	15.18
SkelCtrl	97.85	95.00	47.92	62.43	2.55	0.10	92.81	22.03
StateCtrl	97.45	99.00	50.86	67.78	0.94	0.07	85.66	16.06
StateCtrl-AR	97.43	99.00	50.90	67.76	1.01	0.08	74.66	16.13

Specifically, 10 volunteers are recruited, and each segment is annotated by three volunteers. We discard segments on which the annotators disagree, accounting for about 5% of the full set. We then measure the agreement between human judgments and model scores. For State Alignment, we run keypoint tracking directly on ground-truth videos and evaluate the resulting trajectories using the same protocol to verify the reliability of the metric.

The experimental results show that human judgments and the proposed model-based Action Following metric achieve 85% agreement on WildBench, indicating that the metric can reliably reflect human evaluation of action consistency. Moreover, the State Alignment metric achieves 43.23% coordinate accuracy relative to the skeleton ground truth, demonstrating its effectiveness in measuring alignment between generated and ground-truth state evolution.

Evaluation of Interactive Video Generation. Table 1 presents the WildBench evaluation results for different interactive world modeling approaches trained on the WildWorld dataset. Here, Baseline refers to Wan2.2-TI2V-5B [45], and Video Quality is reported as a percentage. We observe that

- **All approaches improve over the baseline on interaction-related metrics**. For example, the camera-conditioned video generation model ***CamCtrl*** reduces ATE and RPE on Camera Control by $\triangle 2.61$ and $\triangle 0.05$, respectively. Taking the skeleton video as a conditioning input, ***SkelCtrl*** achieves nearly 100% average improvements in Action Following and State Alignment. Moreover, by directly conditioning on discrete and continuous state information, ***StateCtrl*** and ***StateCtrl-AR*** improve performance across all three interaction-related dimensions. This highlights the utility of WildWorld across diverse modeling approaches.
- **VBench metrics appear saturated on WildWorld.** We observe that all methods achieve over 95% on the MS (Motion Smoothness) and DD (Dynamic Degree) metrics under Video Quality. In contrast, their actual abilities to produce reasonable motion and dynamics still differ substantially, as evidenced by the large performance differences in Action Following and State Alignment. This suggests that interactive world models require more

Fig. 4: Qualitative comparisons of different interactive world modeling approaches trained on the WildWorld dataset.

fine-grained and nuanced evaluation metrics to assess highly dynamic video generation, which aligns with the design goal of WildBench.

- **Using visual signals directly as conditional inputs yields a trade-off.** We find that ***SkelCtrl***, which uses visual signals as interactive control inputs, achieves larger gains on interaction-related metrics than ***StateCtrl***, which learns soft embeddings from the same information. However, these gains come at the cost of lower video quality, as reflected by lower AQ (Aesthetic Quality) and IQ (Image Quality) scores. We further analyze this pattern in the qualitative evaluation below.
- **Autoregressive interactive world models show promise**. Using only the first-frame state and autoregressively predicting subsequent states as control inputs, ***StateCtrl-AR*** achieves performance comparable to ***StateCtrl***, but exhibits a noticeable drop in Action Following. We attribute this degradation to error accumulation in iterative next-state prediction, a phenomenon also observed in autoregressive video generation [22, 25, 51]. We believe that combining this paradigm with autoregressive video generation may lead to further advances.

5.3 Qualitative Evaluation

Fig. 4 presents visual comparisons of different interactive world modeling approaches on two test examples. The top example shows that ***CamCtrl*** produces camera motion consistent with the ground truth, but fails to capture the monster's dynamics. In particular, for the bottom example, we observe that ***StateCtrl*** generates a clearer foreground subject, whereas the subject in the ground truth is partially occluded by splashing sand and gravel. In contrast,

SkelCtrl better reproduces this effect. This observation is consistent with the stronger image-quality performance of ***StateCtrl***, as clearer frames are typically perceived as having higher image quality.

6 Conclusion

In this paper, we have presented WildWorld, a large-scale video dataset with explicit state annotations to facilitate the study of action-conditioned world modeling. The dataset is automatically collected from a photorealistic AAA action role-playing game, *Monster Hunter: Wilds*, through a scalable data collection pipeline. WildWorld provides a rich and meaningful action space with over 450 actions, and each video is annotated with frame-level annotations including character skeletons, world states, camera poses, and depth. Furthermore, we have introduced WildBench, a benchmark derived from WildWorld, which enables the quantitative evaluation of action following and state alignment. Experimental results demonstrate that existing models still face significant challenges in modeling semantically rich actions and maintaining long-horizon state consistency. These findings highlight the importance of incorporating explicit state information for advancing action-conditioned video generation and world modeling.

References

1. Agarwal, N., Ali, A., Bala, M., Balaji, Y., Barker, E., Cai, T., Chattopadhyay, P., Chen, Y., Cui, Y., Ding, Y., et al.: Cosmos world foundation model platform for physical ai. arXiv preprint arXiv:2501.03575 (2025)
2. Bertsekas, D.: Dynamic programming and optimal control: Volume I, vol. 4. Athena scientific (2012)
3. Bruce, J., Dennis, M.D., Edwards, A., Parker-Holder, J., Shi, Y., Hughes, E., Lai, M., Mavalankar, A., Steigerwald, R., Apps, C., et al.: Genie: Generative interactive environments. In: Proceedings of the 41st International Conference on Machine Learning (2024)
4. Che, H., He, X., Liu, Q., Jin, C., Chen, H.: Gamegen-x: Interactive open-world game video generation. arXiv preprint arXiv:2411.00769 (2024)
5. Chen, H., Zhang, Y., Cun, X., Xia, M., Wang, X., Weng, C., Shan, Y.: Videocrafter2: Overcoming data limitations for high-quality video diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7310–7320 (2024)
6. Cudlenco, N., Masala, M., Leordeanu, M.: Agentic video generation: From text to executable event graphs via tool-constrained llm planning. arXiv preprint arXiv:2604.10383 (2026)
7. Cudlenco, N., Masala, M., Leordeanu, M.: [tiny paper] GEST-engine: Controllable multi-actor video synthesis with perfect spatiotemporal annotations. In: Proceedings of the 2nd Workshop on World Models: Understanding, Modelling and Scaling at the International Conference on Learning Representations (2026), https://openreview.net/forum?id=uUofPYVMZH
8. Duan, H., Yu, H.X., Chen, S., Fei-Fei, L., Wu, J.: Worldscore: A unified evaluation benchmark for world generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 27713–27724 (2025)

9. Gao, J., Chen, Z., Liu, X., Zhuang, J., Xu, C., Feng, J., Qiao, Y., Fu, Y., Si, C., Liu, Z.: Longvie 2: Multimodal controllable ultra-long video world model. arXiv preprint arXiv:2512.13604 (2025)
10. Garrido, Q., Nagarajan, T., Terver, B., Ballas, N., LeCun, Y., Rabbat, M.: Learning latent action world models in the wild. arXiv preprint arXiv:2601.05230 (2026)
11. Genie 3 Contributors: Genie 3. `https://deepmind.google/models/genie/` (2025)
12. Ha, D., Schmidhuber, J.: World models. arXiv preprint arXiv:1803.10122 **2**(3), 440 (2018)
13. HaCohen, Y., Brazowski, B., Chiprut, N., Bitterman, Y., Kvochko, A., Berkowitz, A., Shalem, D., Lifschitz, D., Moshe, D., Porat, E., et al.: Ltx-2: Efficient joint audio-visual foundation model. arXiv preprint arXiv:2601.03233 (2026)
14. Hafner, D., Pasukonis, J., Ba, J., Lillicrap, T.: Mastering diverse domains through world models. arXiv preprint arXiv:2301.04104 (2023)
15. He, H., Xu, Y., Guo, Y., Wetzstein, G., Dai, B., Li, H., Yang, C.: Cameractrl: Enabling camera control for text-to-video generation. arXiv preprint arXiv:2404.02101 (2024)
16. He, H., Xu, Y., Guo, Y., Wetzstein, G., Dai, B., Li, H., Yang, C.: Cameractrl: Enabling camera control for video diffusion models. In: Proceedings of the International Conference on Learning Representations (2025)
17. He, H., Yang, C., Lin, S., Xu, Y., Wei, M., Gui, L., Zhao, Q., Wetzstein, G., Jiang, L., Li, H.: Cameractrl ii: Dynamic scene exploration via camera-controlled video diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13416–13426 (October 2025)
18. 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)
19. He, Y., Weilbach, C.D., Wojciechowska, M.E., Zhang, Y., Wood, F.: Plaicraft: Large-scale time-aligned vision-speech-action dataset for embodied ai. arXiv preprint arXiv:2505.12707 (2025)
20. Huang, J., Zhou, Q., Rabeti, H., Korovko, A., Ling, H., Ren, X., Shen, T., Gao, J., Slepichev, D., Lin, C.H., et al.: Vipe: Video pose engine for 3d geometric perception. arXiv preprint arXiv:2508.10934 (2025)
21. Huang, T., Zheng, W., Wang, T., Liu, Y., Wang, Z., Wu, J., Jiang, J., Li, H., Lau, R.W., Zuo, W., et al.: Voyager: Long-range and world-consistent video diffusion for explorable 3d scene generation. arXiv preprint arXiv:2506.04225 (2025)
22. Huang, X., Li, Z., He, G., Zhou, M., Shechtman, E.: Self forcing: Bridging the train-test gap in autoregressive video diffusion. arXiv preprint arXiv:2506.08009 (2025)
23. 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: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21807–21818 (2024)
24. HunyuanWorld Team: Hy-world 1.5: A systematic framework for interactive world modeling with real-time latency and geometric consistency. arXiv preprint (2025)
25. Ji, S., Chen, X., Yang, S., Tao, X., Wan, P., Zhao, H.: Memflow: Flowing adaptive memory for consistent and efficient long video narratives. arXiv preprint arXiv:2512.14699 (2025)
26. Ju, X., Gao, Y., Zhang, Z., Yuan, Z., Wang, X., Zeng, A., Xiong, Y., Xu, Q., Shan, Y.: Miradata: A large-scale video dataset with long durations and structured captions. Advances in Neural Information Processing Systems **37**, 48955–48970 (2024)

27. Kim, K., Sano, M., De Freitas, J., Haber, N., Yamins, D.: Active world model learning with progress curiosity. In: Proceedings of the International Conference on Machine Learning. pp. 5306–5315. PMLR (2020)
28. Li, D., Fang, Y., Chen, Y., Yang, S., Cao, S., Wong, J., Luo, M., Wang, X., Yin, H., Gonzalez, J.E., et al.: Worldmodelbench: Judging video generation models as world models. arXiv preprint arXiv:2502.20694 (2025)
29. Li, J., Feng, W., Fu, T.J., Wang, X., Basu, S., Chen, W., Wang, W.Y.: T2v-turbo: Breaking the quality bottleneck of video consistency model with mixed reward feedback. Advances in Neural Information Processing Systems **37**, 75692–75726 (2024)
30. Li, J., Tang, J., Xu, Z., Wu, L., Zhou, Y., Shao, S., Yu, T., Cao, Z., Lu, Q.: Hunyuan-gamecraft: High-dynamic interactive game video generation with hybrid history condition. arXiv preprint arXiv:2506.17201 (2025)
31. Li, Z., Li, C., Mao, X., Lin, S., Li, M., Zhao, S., Xu, Z., Li, X., Feng, Y., Sun, J., et al.: Sekai: A video dataset towards world exploration. arXiv preprint arXiv:2506.15675 (2025)
32. Lillemark, H.J., Huang, B., Zhan, F., Du, Y., Keller, T.A.: Flow equivariant world models: Memory for partially observed dynamic environments (2026), `https://arxiv.org/abs/2601.01075`
33. Lin, B., Ge, Y., Cheng, X., Li, Z., Zhu, B., Wang, S., He, X., Ye, Y., Yuan, S., Chen, L., et al.: Open-sora plan: Open-source large video generation model. arXiv preprint arXiv:2412.00131 (2024)
34. Mao, X., Li, Z., Li, C., Xu, X., Ying, K., Zhang, K.: Yume1.5: A text-controlled interactive world generation model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7752–7761 (June 2026)
35. Mao, X., Lin, S., Li, Z., Li, C., Peng, W., He, T., Pang, J., Chi, M., Qiao, Y., Zhang, K.: Yume: An interactive world generation model. arXiv preprint arXiv:2507.17744 (2025)
36. Nan, K., Xie, R., Zhou, P., Fan, T., Yang, Z., Chen, Z., Li, X., Yang, J., Tai, Y.: Openvid-1m: A large-scale high-quality dataset for text-to-video generation. arXiv preprint arXiv:2407.02371 (2024)
37. RobbyAnt Team, Gao, Z., Wang, Q., Zeng, Y., Zhu, J., Cheng, K.L., Li, Y., Wang, H., Xu, Y., Ma, S., et al.: Advancing open-source world models. arXiv preprint arXiv:2601.20540 (2026)
38. Schmidhuber, J.: On learning to think: Algorithmic information theory for novel combinations of reinforcement learning controllers and recurrent neural world models. arXiv preprint arXiv:1511.09249 (2015)
39. Shi, X., Huang, Z., Wang, F.Y., Bian, W., Li, D., Zhang, Y., Zhang, M., Cheung, K.C., See, S., Qin, H., et al.: Motion-i2v: Consistent and controllable image-to-video generation with explicit motion modeling. In: Proceedings of the ACM SIGGRAPH Conference. pp. 1–11 (2024)
40. Sitzmann, V., Rezchikov, S., Freeman, B., Tenenbaum, J., Durand, F.: Light field networks: Neural scene representations with single-evaluation rendering. Advances in Neural Information Processing Systems **34**, 19313–19325 (2021)
41. Sora 2 Contributors: Sora 2. `https://openai.com/index/sora-2/` (2025)
42. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of rgb-d slam systems. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 573–580. IEEE (2012)
43. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press (2018)

44. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
45. Wan Team, Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.W., Chen, D., Yu, F., Zhao, H., Yang, J., et al.: Wan: Open and advanced large-scale video generative models. arXiv preprint arXiv:2503.20314 (2025)
46. Wang, J., Yuan, Y., Zheng, R., Lin, Y., Gao, J., Chen, L.Z., Bao, Y., Zhang, Y., Zeng, C., Zhou, Y., et al.: Spatialvid: A large-scale video dataset with spatial annotations. arXiv preprint arXiv:2509.09676 (2025)
47. Wang, L., Chen, Z., Du, Y., Yan, D., Ge, W., Shen, G., Xu, X., Wu, L., Chen, M., Xu, T., et al.: A mechanistic view on video generation as world models: State and dynamics. arXiv preprint arXiv:2601.17067 (2026)
48. World Labs: 3d as code. `https://www.worldlabs.ai/blog/3d-as-code` (2025)
49. Xing, J., Xia, M., Zhang, Y., Chen, H., Yu, W., Liu, H., Liu, G., Wang, X., Shan, Y., Wong, T.T.: Dynamicrafter: Animating open-domain images with video diffusion priors. In: Proceedings of the European Conference on Computer Vision. pp. 399–417. Springer (2024)
50. Xu, J., Zou, X., Huang, K., Chen, Y., Liu, B., Cheng, M., Shi, X., Huang, J.: Easyanimate: A high-performance long video generation method based on transformer architecture. arXiv preprint arXiv:2405.18991 (2024)
51. 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)
52. 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)
53. Ye, Y., Lu, X., Jiang, Y., Gu, Y., Zhao, R., Liang, Q., Pan, J., Zhang, F., Wu, W., Wang, A.J.: Mind: Benchmarking memory consistency and action control in world models. arXiv preprint arXiv:2602.08025 (2026)
54. Yu, J., Qin, Y., Wang, X., Wan, P., Zhang, D., Liu, X.: Gamefactory: Creating new games with generative interactive videos. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11590–11599 (2025)
55. Yue, J., Huang, Z., Chen, Z., Wang, X., Wan, P., Liu, Z.: Simulating the visual world with artificial intelligence: A roadmap. arXiv preprint arXiv:2511.08585 (2025)
56. Zhang, Y., Peng, C., Wang, B., Wang, P., Zhu, Q., Kang, F., Jiang, B., Gao, Z., Li, E., Liu, Y., et al.: Matrix-game: Interactive world foundation model. arXiv preprint arXiv:2506.18701 (2025)
57. Zholus, A., Doersch, C., Yang, Y., Koppula, S., Patraucean, V., He, X.O., Rocco, I., Sajjadi, M.S., Chandar, S., Goroshin, R.: Tapnext: Tracking any point (tap) as next token prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9693–9703 (2025)
58. Zhou, Y., Wang, Y., Zhou, J., Chang, W., Guo, H., Li, Z., Ma, K., Li, X., Wang, Y., Zhu, H., et al.: Omniworld: A multi-domain and multi-modal dataset for 4d world modeling. arXiv preprint arXiv:2509.12201 (2025)