GAMEHORIZON SUITE: MULTI-HORIZON DATA AND EVALUATION IN GAMEPLAY

Anonymous authors
Paper under double-blind review

ABSTRACT

Video games provide a testbed for AI models, combining abilities of visual understanding, instruction decomposition, goal planning, and action execution over multiple temporal horizons. Prior data and benchmarks either have narrow game coverage, lack language instructions, or rely on high-variance online rollouts. To address these challenges, we introduce GameHorizon, a unified data and evaluation suite measuring gameplay abilities at different horizons for diverse model families. GameHorizon Suite contains three components. First, GameHorizon-Annotator is a scalable and automated annotation pipeline for multi-horizon instructions. Second, using the pipeline, we construct GameHorizon-Data, a large-scale AAA gameplay dataset with temporally aligned videos, player actions, and multi-horizon instructions. It comprises $5{,}000$ hours of recordings from 21 games. Third, we build GameHorizon-Bench with reproducible offline and stepwise online testing. The offline track enables reproducible evaluation via thousands of standardized questions in three primary tasks and ten diagnostic variants, while the online track tests whether offline scores reflect actual gameplay capabilities and localizes failures to specific steps within long-horizon gameplay. We evaluate 47 models through more than one million model invocations, showing a meaningful hierarchy of task difficulty and pronounced differences in model performance. Our work can provide a unified yardstick for gameplay abilities across models and horizons. We will release our data, annotator, and benchmark for future research.

1 INTRODUCTION

Empowering AI models to play modern video games provides a measurable testbed for understanding, decision-making, and acting within complex environments. Game objectives span varying temporal scales, *e.g.*, collecting an item within seconds, winning a fight lasting several minutes, and executing a strategy that unfolds over an entire game session, as shown in Fig. 1. Whatever the horizon, these objectives are all reflected in the same stream of primitive actions, such as keystrokes and mouse movements. Thus, a growing number of dedicated game agents (Magne et al., 2026; Yue et al., 2026; Tan et al., 2025; Wang et al., 2025c) have emerged. General vision-language models (VLMs) and agents (Team Gemini, 2023; Achiam et al., 2023; Bolton et al., 2025; Qin et al., 2025; Wang et al., 2025b; 2023; Tan et al., 2024) have also begun to treat video games as a capability target, *e.g.*, SIMA 2 (Bolton et al., 2025) equips Gemini to follow instructions in open-world games.

Playing games well requires two abilities: (1) based on an understanding of the environment, planning subsequent operations by decomposing long-horizon objectives into subgoals; and (2) turning the subgoals into actions. In practice, the abilities are split across two model types. Dedicated game agents are optimized for acting. For high-frequency control, they use lightweight vision-language-action (VLA) models (Kim et al., 2024) under action supervision, at the cost of the capacity to reason and plan. General-purpose models excel at planning, yet they have not been systematically tested on actions. The few cases, *e.g.*, Gemini and Claude playing Pokémon, rely on bespoke harnesses and cannot be compared across models. What is needed, therefore, is one yardstick for both categories: one that measures whether a model can align vision, actions, and multi-horizon goals. The yardstick should be low-cost, standardized, and reproducible, independent of harnesses or environments.

Prior work falls short of these requirements. First, annotation costs limit game coverage. GameWorld (Ouyang et al., 2026) targets simple mini-games. VPT (Baker et al., 2022) is confined to

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.

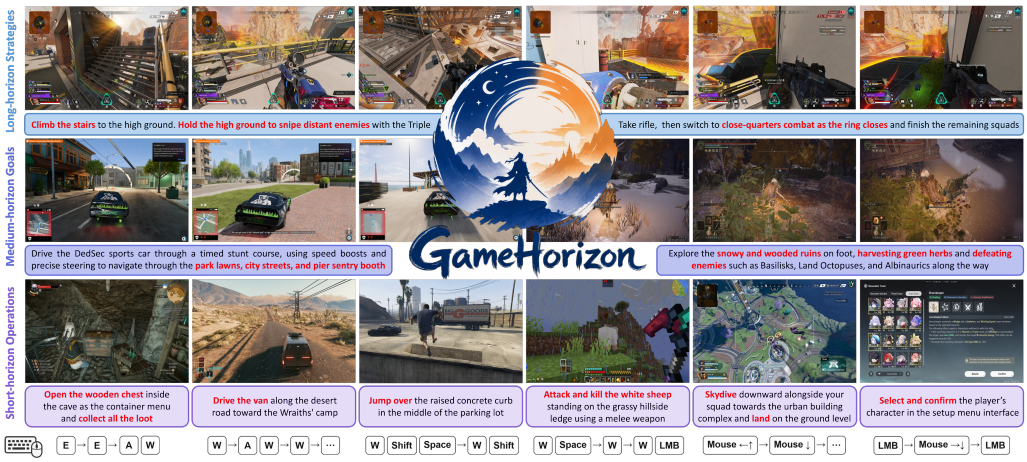

Figure 1: **GameHorizon Suite** spans diverse games with a pyramid of primitive actions (e.g., Left Mouse Button as LMB), short-horizon operations, medium-horizon goals, and long-horizon strategies, as a unified yardstick across models. Red text highlights key actions or objects in video frames.

Minecraft. Conclusions drawn from mini-games or a single title cannot generalize to heterogeneous AAA games. Second, human–agent interaction is mediated through language, *e.g.*, instruction following and goal planning. Previous attempts (Magne et al., 2026) lack annotations of text instructions. Open-P2P (Yue et al., 2026) only provides sparse annotations. Third, prior work mainly relies on online testing with few rollouts. Lumine (Tan et al., 2025) reports success rates from three trials per scene. Small samples yield low-confidence comparisons. Online results are sensitive to game environments and harnesses, hindering reproducibility. An aggregate success rate collapses distinct failure modes into a single scalar. When a model fails, it remains unclear whether it misidentifies current actions, infers the next goal incorrectly, or fails to map the goal to the right future controls.

To address these challenges, we introduce **GameHorizon**, a data and evaluation suite spanning multiple horizons and AAA games. It serves as a unified yardstick across diverse models and comprises three components. First, **GameHorizon-Annotator** is a scalable annotation pipeline for multi-horizon instructions. Unlike the manual annotation (Wang et al., 2025c), it automatically produces an instruction pyramid with short-horizon operations, medium-horizon goals, and long-horizon strategies. It operates bottom-up, abstracting fine-grained instructions into higher levels. Second, we construct **GameHorizon-Data**, the first large-scale AAA gameplay dataset with aligned videos, actions, and multi-horizon instructions. It contains 5,000 hours of recordings across 21 games, exceeding prior corpora such as gaming-500-hours (Markov AI, 2026) in scale. Third, we build **GameHorizon-Bench** with reproducible offline and stepwise online evaluation. The offline track comprises three primary tasks: single-horizon action, multi-horizon decomposition, and cross-horizon consistency. Additional variants enable finer diagnosis. The offline track is reliable based on thousands of standardized questions. Besides, online track evaluates long-horizon gameplay via short-horizon subtasks, with order-dependent causal and order-flexible thematic tasks. Stepwise verification enables failure localization. It tests whether offline scores reflect actual gameplay abilities.

The evaluations comprise over one million model inferences. In the offline track, we test 47 models, *e.g.*, VLMs, unified multimodal models (UMMs), coding, GUI, and game agents. Results reveal a meaningful hierarchy of task difficulty and pronounced differences in model abilities. In the online track, we observe a positive association between success rates and offline scores, suggesting that offline accuracy is a valid proxy for gameplay ability. Beyond aggregate performance, we analyze bottlenecks of current models. Planning future actions and decomposing complex goals are harder than identifying current actions. In future-action planning, multi-horizon instructions surpass vision-only inputs by 7.2%, showing the efficacy of our design. Our contributions are listed as follows:

- We introduce GameHorizon, a data and evaluation suite with multiple horizons and AAA games.
- GameHorizon-Annotator works as a scalable annotation pipeline for multi-horizon instructions.
- GameHorizon-Data is a large-scale game dataset with the aligned videos, actions, and instructions.
- GameHorizon-Bench provides reproducible offline and stepwise online testing for diverse models.

2 Related Work

Gameplay Data and Benchmarks. Prior work suffers from narrow game coverage, limited instruction annotations, and high-variance evaluations. First, many datasets are confined to a single game or simplified mini-games. WildWorld (Li et al., 2026) is collected from Monster Hunter Wilds. MineDojo (Fan et al., 2022), VPT (Baker et al., 2022), STEVE-1 (Lifshitz et al., 2023), MineRL (Guss et al., 2019), and MCU (Zheng et al., 2025) provide data and evaluation in Minecraft. Some efforts attempt to encompass multiple titles. However, due to annotation costs, they either remain limited in scale (*e.g.*, 300 hours for D2E by Choi et al., 2026 and 500 hours for gaming-500-hours by Markov AI, 2026) or fall back on mini-games (Ouyang et al., 2026). Second, existing corpora lack text instructions. NitroGen (Magne et al., 2026), GameVerse (Zhang et al., 2026), D2E (Choi et al., 2026), and VPT (Baker et al., 2022) are devoid of instructions, whereas Open-P2P (Yue et al., 2026) only offers sparse annotations. Game-TARS (Wang et al., 2025c) relies on costly manual annotations, which hinders scalability and remains unreleased. Third, previous benchmarks mainly adopt online evaluations with a limited number of rollouts. GameVerse (Zhang et al., 2026) conducts rollouts on 3–20 cases. VideoGameBench (Zhang et al., 2025) tests each model with a single run per game. In contrast, GameHorizon provides a unified data and evaluation suite, featuring $5{,}000$ hours of recordings, diverse AAA game genres, multi-horizon instructions, as well as a reliable benchmark.

Game-Playing Models. Games serve as a testbed for models to perceive, plan, and act in complex environments. On the one hand, dedicated game agents are tailored for high-frequency action control, at the expense of planning abilities. Open-P2P (Yue et al., 2026) employs a lightweight action decoder, whereas JARVIS-VLA (Li et al., 2025) instantiates a VLA policy with a short context window. On the other hand, general-purpose models, *e.g.*, VLMs, UMMs, and agents, exhibit stronger planning capabilities, yet they have not been systematically tested on action execution. Cradle (Tan et al., 2024) couples GPT-4V (Achiam et al., 2023) with a multi-module agent harness for commercial games. Gemini (Comanici et al., 2025) and Claude (Anthropic, 2026) are tested on *Pokémon* through bespoke agent loops. These results remain incomparable due to specialized harnesses. To bridge this gap, GameHorizon presents a standardized yardstick for different models, enabling evaluations across goal planning, instruction following, and action execution at multiple horizons.

3 GameHorizon Suite

As shown in Fig. 2, we present GameHorizon, a unified data and evaluation suite spanning multiple horizons, diverse model families, and AAA games. The GameHorizon-Annotator, GameHorizon-Data, as well as GameHorizon-Bench are described in Sec. 3.1, Sec. 3.2, and Sec. 3.3, respectively.

3.1 GameHorizon-Annotator

Given aligned videos and actions, GameHorizon-Annotator constructs a three-level instruction pyramid, *i.e.*, short-horizon operations, medium-horizon goals, and long-horizon strategies. The workflow comprises action-aware segmentation, bottom-up temporal merging, and instruction annotation.

Action-Aware Video Segmentation. Constructing multi-horizon instructions in a top-down manner is challenging, as VLMs struggle to resolve extended temporal contexts, *e.g.*, an hour-long gameplay session. We proceed bottom-up, partitioning long videos into short clips. Off-the-shelf segmentation tools (Castellano, 2024) rely on inter-frame visual similarity and tend to over-segment continuous actions under rapid camera motion or abrupt viewpoint shifts. To address this issue, we perform action-aware segmentation using keyboard-mouse traces to determine clip boundaries. Specifically, we map keyboard-mouse events to action semantics, *e.g.*, `Shift` as sprinting in Cyberpunk 2077. We then scan the mapped action stream, grouping consecutive actions within the same sustained event, such as alternating between walking and running. Discrete events define initial boundaries. Since the same keyboard-mouse input may carry different semantics, *e.g.*, a left click may indicate an attack or item selection, we employ a VLM (Google DeepMind, 2026a) to examine video frames, disambiguate action semantics, and refine the initial segmentation into final short-horizon clips.

Bottom-Up Temporal Merging. Based on the short-horizon (L_1) clips and instructions, we merge adjacent segments into medium-horizon (L_2) clips. The VLM determines whether neighboring L_1 operations form a continuous progression toward the same medium-horizon goal. Similarly, to

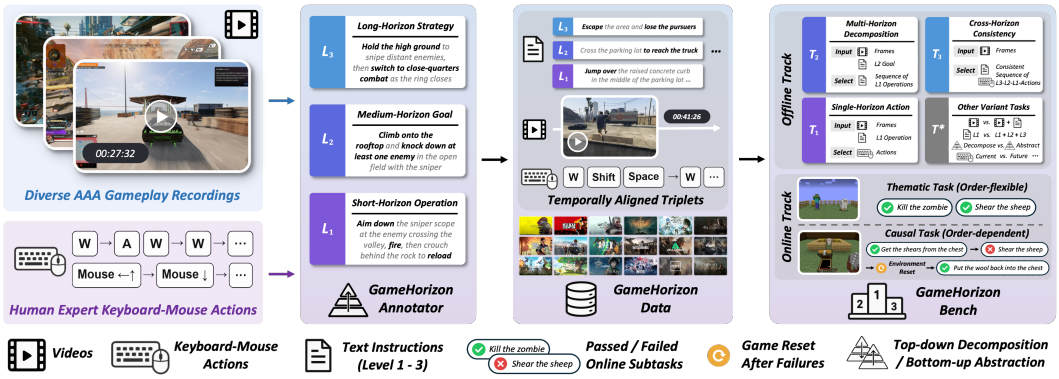

Figure 2: **Overview of GameHorizon Suite.** GameHorizon-Annotator automatically produces a pyramid of short-horizon operations L_1, medium-horizon goals L_2, and long-horizon strategies L_3. GameHorizon-Data contains 5,000 hours of gameplay across 21 game titles, with temporally aligned videos, actions, and multi-horizon instructions. GameHorizon-Bench provides reproducible offline and stepwise online testing. The offline track contains thousands of standardized questions across three primary tasks T_1–T_3 and diagnostic variants T^*. The online track tests order-dependent causal tasks and order-flexible thematic tasks via verifiable subtasks and game resets for failure localization.

construct long-horizon (L_3) clips, the VLM assesses whether neighboring L_2 goals follow the same gameplay strategy. Besides, we apply a dynamic programming algorithm (Bellman, 1966) to enforce level-specific duration ranges of 1–5 seconds for L_1, 1–2 minutes for L_2, and 5–8 minutes for L_3. The short range helps avoid fragmenting primitive actions, while the long range keeps clip durations within the capacity of current models. We term the temporal merging process bottom-up because it starts from numerous fine-grained L_1 clips and progressively merges them into higher-level clips.

Multi-Horizon Instruction Annotation. The VLM receives level-specific prompts tailored to input modalities and output granularities. For an L_1 clip, it takes sampled video frames and aligned actions as input. The prompt asks the VLM to generate an action-grounded instruction describing the current operation of the player with sufficient details, *e.g.*, coordinates, object descriptions, and spatial relations, to guide a virtual agent in game sessions. For an L_2 clip, the VLM synthesizes frames, actions, and constituent L_1 instructions into a medium-horizon instruction describing the goal pursued by the player. The output retains only necessary details while omitting excessive local information. For an L_3 clip, the inputs comprise video frames and the constituent L_2 instructions, without action traces. The VLM generates a long-horizon instruction that captures the high-level gameplay strategy. The prompt emphasizes the long-term intent while suppressing local details.

For L_1 and L_2, the action inputs help identify visually ambiguous operations, *e.g.*, determining whether the player detaches from the squad during skydiving in Apex Legends. By comparison, we omit actions at L_3 to keep the instructions focused on high-level strategies rather than detailed operations. All prompts include the game title, clip duration, frame rate, and sampled frame indices. At higher levels, the constituent lower-level instructions provide compact semantic context and help maintain consistency across horizons. The L_1 and L_2 prompts also include game-specific keybinds that map actions to in-game semantics, helping the VLM interpret keyboard-mouse controls. See the appendix for more details, *e.g.*, prompts, segmentation, instructions, keybinds, and overall design.

3.2 GameHorizon-Data

Using our annotator, we build GameHorizon-Data, a large corpus with game videos, actions, and multi-horizon instructions across diverse games. We report statistics and comparisons in this section.

Dataset Statistics. First, regarding data scale and proportion, the corpus comprises 5,000 hours of human gameplay across 21 titles, spanning diverse genres such as open-world, action role-playing, competitive shooter, sandbox survival, and creature-collecting adventure games. Valorant contributes the largest share with 617.5 hours (12.35%), while even the smallest entry, Honor of Kings: World, contains 35.9 hours (0.72%). All recordings are collected from 100 experienced players who receive extensive training on the recording and upload procedures. We filter out low-quality or uninformative content via automated rules and VLM assessments, *e.g.*, prolonged action-free cutscenes

Table 1: **Dataset comparison.** Large-scale denotes at least 3,000 hours of gameplay. AAA-focused indicates that more than half of the included titles are AAA games. Direct human actions require that the majority of actions are recorded from human players rather than inferred, extracted, or generated.

Dataset	Data Scope		Annotation Properties				Notes
	Large Scale	AAA-Focused	Direct Human Actions	Instruction Annotations	Dense Instructions	Multi-Horizon	
Single-Game Datasets							
MineRL (Guss et al., 2019)	✗	✗	✓	✗	✗	✗	Minecraft; human actions; no textual instructions.
MineDojo (Fan et al., 2022)	✓	✗	✗	✗	✗	✗	Web Minecraft videos; no actions or instructions.
VPT (Baker et al., 2022)	✓	✗	✗	✗	✗	✗	Web Minecraft videos; IDM actions; no instructions.
STEVE-1 (Lifshitz et al., 2023)	✗	✗	✓	✓	✗	✗	Minecraft; only 10K short-term text instructions.
PLAICraft (He et al., 2025)	✓	✗	✓	✗	✗	✗	Minecraft; human actions; no textual instructions.
WildWorld (Li et al., 2026)	✗	✓	✗	✗	✗	✗	Monster Hunter Wilds; AI actions; no instructions.
EgoCS-400K (Guo et al., 2026)	✓	✓	✗	✗	✗	✗	CS; replay-derived action labels; no instructions.
Multi-Game Datasets							
NitroGen (Magne et al., 2026)	✓	✗	✗	✗	✗	✗	Gamepad-overlay actions without text instructions.
Open-P2P (Yue et al., 2026)	✓	✗	✓	✓	✗	✗	Majority non-AAA games; sparse text instructions.
D2E (Choi et al., 2026)	✗	✗	✗	✗	✗	✗	Mostly IDM-inferred actions; no text instructions.
Gaming-500-hours (Markov AI, 2026)	✗	✗	✓	✗	✗	✗	Majority non-AAA games; no textual instructions.
GameHorizon-Data (Ours)	✓	✓	✓	✓	✓	✓	**AAA; human actions; multi-horizon instructions.**

and switches away from the game window. After filtering, 4,341 hours (86.8%) are retained for instruction annotation, while keyboard-mouse actions remain available for all 5,000 hours. Second, in terms of annotation counts, our data contains 411.03 million keyboard-mouse actions, averaging 22.84 events per second. The action stream combines mouse inputs (movements and button presses) sampled at 20 Hz with an average of 2.84 keyboard events per second. The retained footage is annotated with 6,184,036 instructions, including 5,947,588 short-horizon operations, 189,158 medium-horizon goals, and 47,290 long-horizon strategies. Third, turning to instruction spans, the average duration increases across horizons, *i.e.*, 2.63 seconds for L_1, 82.6 seconds for L_2, and 330.4 seconds for L_3. Each span measures the temporal coverage of one distinct instruction, with all frames in the corresponding clip sharing the same instruction at that horizon. Thus, every frame is aligned with instructions at all horizons, yielding a dense instruction pyramid across diverse AAA scenarios.

Comparisons with Other Datasets. We compare different datasets in Table 1. GameHorizon-Data is the first corpus that jointly provides large-scale AAA gameplay, direct human actions, and dense multi-horizon instructions. First, for data scope, most datasets are restricted to a single game. The few multi-game datasets span predominantly non-AAA games. In contrast, our data contains 5,000 hours of gameplay across 21 titles, with a primary focus on AAA titles. Second, for action annotations, several datasets lack directly recorded human controls. For example, MineDojo (Fan et al., 2022) provides web Minecraft videos without action annotations, while some other datasets obtain action labels using Inverse Dynamics Models (IDMs) (Choi et al., 2026), gamepad segmentation (Magne et al., 2026), in-game AI (Li et al., 2026), or game replays (Guo et al., 2026). The inferred labels can deviate from the actual controls from players. By comparison, GameHorizon-Data captures keyboard-mouse actions from 100 experienced players across the entire dataset. Authentic human gameplay recordings can establish a reliable foundation for faithful evaluations in complex game worlds. Third, most existing datasets omit instructions. Among the few exceptions, STEVE-1 (Lifshitz et al., 2023) contains 10K short-term instructions, while Open-P2P (Yue et al., 2026) provides one instruction every few minutes. Our dataset contains 6,184,036 distinct multi-horizon instructions with dense per-frame coverage. Refer to the appendix for more statistics on our dataset.

3.3 GameHorizon-Bench

Leveraging our data, we develop GameHorizon-Bench, a unified benchmark that evaluates multi-horizon gameplay abilities across diverse models via reproducible offline and stepwise online tracks.

Reproducible Offline Track. The offline track formulates gameplay evaluation as multiple-choice questions (MCQs) derived from videos, instructions, and actions in GameHorizon-Data. It contains thousands of questions in three primary and ten variant tasks. High-quality annotations, standardized inputs and options, as well as independence from environments and harnesses enable unified and reproducible comparisons across the VLMs, UMMs, GUI agents, coding agents, and game agents.

To be specific, as shown in Fig. 3, the single-horizon action task (T_1) asks models to make the correct action decision based on input frames and the short-horizon instruction (L_1). The correct option is constructed by mapping ground-truth keyboard-mouse controls to textual action sequences using game-specific keybinds, which enables unified evaluation across games and models without

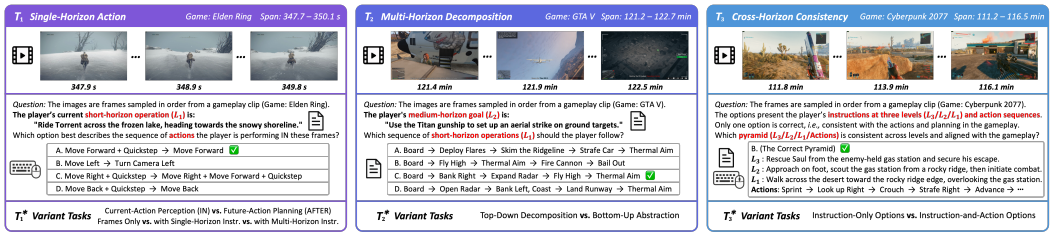

Figure 3: **The offline track of GameHorizon-Bench.** It comprises three primary and ten variant tasks. Given frames and the short-horizon instruction L_1, single-horizon action T_1 requires models to determine the correct action sequence. Multi-horizon decomposition T_2 evaluates whether models can decompose a medium-horizon goal L_2 into the correct sequence of short-horizon operations L_1. Cross-horizon consistency T_3 assesses the overall consistency across frames, L_1–L_3 instructions, and actions. Variant tasks T^* provide diagnostics and insights into different gameplay abilities. Due to space constraints, L_1 instructions are abridged in T_2. Only the correct option is showcased in T_3.

being constrained by heterogeneous action spaces. T_1 provides a rigorous test centered on action perception. Next, the multi-horizon decomposition task (T_2) requires models to identify the ordered sequence of short-horizon operations for realizing a medium-horizon goal (L_2). The correct option presents the L_1 sequence corresponding to the target L_2 goal within the pyramid. T_2 evaluates top-down goal decomposition and temporal planning from abstract goals to concrete operations. Finally, the cross-horizon consistency task (T_3) assesses the overall consistency across videos, instructions, and actions. The correct option consists of the aligned L_3, L_2, L_1, and action sequence. T_3 focuses on integrated understanding across visual observations, textual instructions, and gameplay actions.

Moreover, a series of variant tasks (T^*) provides diagnostics and analyses of different gameplay abilities. For T_1^*, we compare current-action perception and future-action planning under three input settings, including frames only, frames with a short-horizon instruction, and frames with multi-horizon instructions. The comparison reveals how text instructions influence action decisions and planning. For T_2^*, we compare top-down decomposition from L_2 goals to L_1 operations with bottom-up abstraction in the reverse direction. The comparison clarifies the rationale for our bottom-up annotation and top-down benchmark. As for T_3^*, we compare instruction-only versus instruction-and-action options. The comparison can identify whether the bottlenecks arise from planning or action decoding.

Stepwise Online Track. As a complementary evaluation, our online track tests whether offline scores are positively associated with gameplay capabilities. It evaluates long-horizon gameplay through order-dependent causal and order-flexible thematic tasks, each comprising 2–6 verifiable short-horizon subtasks. When a subtask fails, the game is reset to the success state to prevent the failure from affecting subsequent steps. Beyond the aggregate success rates (Tan et al., 2025; Bolton et al., 2025), our stepwise protocol can localize failures to specific stages in long-horizon objectives.

Specifically, the track comprises ten causal and ten thematic tasks with 62 short-horizon subtasks. Due to the inaccessibility of underlying states in most AAA games, we implement the online track in Minecraft, as in prior work (Ju et al., 2026; Zheng et al., 2025). The steps within each causal task have sequential dependencies, *e.g.*, first collecting milk, then crafting a cake, and finally placing it on the dining table. For thematic tasks, the subtasks share a common theme and can be completed in any order, *e.g.*, crafting a wooden pickaxe, axe, and shovel. Regarding failure localization, some models do not understand the task, whereas others understand the intent but commit execution errors, *e.g.*, placing objects in wrong positions or performing a recipe step incorrectly when making a cake.

Comparisons with Other Benchmarks. First, with a primary focus on AAA titles, GameHorizon-Bench evaluates 47 models on 21 games, spanning VLMs, UMMs, GUI agents, coding agents, and game agents. Prior work tests limited model types mainly on Minecraft (Fan et al., 2022; Zheng et al., 2025; Ju et al., 2026) or non-AAA games (Paglieri et al., 2025; Zhang et al., 2026; Ouyang et al., 2026). Second, multi-horizon instructions and human actions enable reliable evaluation across nested horizons, which is a characteristic absent from existing benchmarks. Third, beyond online success rates (Fan et al., 2022; Ouyang et al., 2026; Tan et al., 2025) or progress scores (Paglieri et al., 2025; Ju et al., 2026), we provide the reproducible offline track with thousands of MCQs and the stepwise online track with failure localization. Refer to the appendix for more details about our GameHorizon-Bench, including the online tasks, qualitative results, and benchmark comparisons.

Table 2: **Offline results of the primary tasks T_1–T_3.** The table reports results for general VLMs, UMMs, GUI agents, and coding agents with question-answering abilities. Accuracies are in percentages. Overall denotes the mean across three tasks. Models are divided into four performance tiers. Best and second-best are bold and underlined. The last row reports the averages for all models.

Tier 1						Tier 2					
Rank	**Model**	T_1	T_2	T_3	**Overall**	**Rank**	**Model**	T_1	T_2	T_3	**Overall**
🥇 **1**	GPT-6-Astra	**69.4**	79.6	**91.5**	**80.2**	**12**	Doubao-Seed-2.0-Pro	59.2	74.9	80.0	71.4
🥈 **2**	Gemini 3.8 Flash	65.9	**81.2**	<u>84.8</u>	<u>77.3</u>	**13**	Claude Fable 5	62.9	72.7	78.1	71.2
🥉 **3**	Gemini 3.7 Flash	<u>66.2</u>	<u>80.1</u>	83.9	76.7	**14**	GPT-5.6 Terra	62.1	75.6	74.4	70.7
4	Gemini 3.6 Flash	62.3	79.1	84.6	75.3	**15**	Qwen3.7-Plus	58.8	75.9	71.5	68.7
5	GPT-5.6 Sol	65.3	77.5	81.5	74.8	**16**	Qwen3.8-27B	59.9	77.4	68.1	68.5
6	Kimi-K3	64.4	79.3	79.7	74.5	**17**	GPT-5.6 Luna	61.6	70.7	71.1	67.8
7	Gemini 3.5 Flash	64.9	77.4	80.8	74.4	**17**	Kimi-K2.6	61.2	73.2	69.0	67.8
8	GPT-5.5	64.6	78.9	76.2	73.2	**19**	Doubao-Seed-2.0-Lite	55.2	69.0	76.3	66.8
9	Gemini 3.1 Pro	64.5	75.8	78.2	72.8	**20**	Gemma 4 31B-IT	53.8	68.6	76.6	66.3
10	Doubao-Seed-2.1-Turbo	63.1	79.4	75.5	72.7	**21**	MiniMax-M3	57.8	68.5	70.6	65.6
11	Doubao-Seed-2.1-Pro	62.9	76.7	76.6	72.1	**22**	Qwen3-VL-235B-A22B-Thinking	58.2	71.6	65.9	65.2
Tier 3						**Tier 4**					
Rank	**Model**	T_1	T_2	T_3	**Overall**	**Rank**	**Model**	T_1	T_2	T_3	**Overall**
23	Claude Opus 4.8	59.1	62.1	73.1	64.8	**34**	Qwen3.6-35B-A3B	53.3	52.9	66.3	57.5
24	Claude Sonnet 5	54.4	57.7	82.0	64.7	**35**	Qwen3-Omni-30B-A3B-Instruct	52.5	55.6	63.7	57.3
25	Step-3.7-Flash	57.5	71.3	63.4	64.1	**36**	GELab-Zero-4B-Preview	51.8	46.8	72.7	57.1
26	Qwen3-VL-235B-A22B-Instruct	55.1	62.1	73.1	63.4	**37**	Qwen2.5-VL-7B-Instruct	52.0	54.3	63.7	56.7
27	GLM-5V-Turbo	53.4	65.4	70.8	63.2	**38**	InternVL3.5-8B	51.8	50.2	65.5	55.8
28	GPT-5.2	58.9	58.9	70.6	62.8	**39**	GLM-4.1V-9B-Thinking	54.9	54.8	55.3	55.0
29	Qwen3.5-397B-A17B	53.3	60.1	73.5	62.3	**40**	GPT-4o	55.7	46.3	61.3	54.4
30	BAGEL-7B-MoT	51.7	57.3	75.1	61.4	**41**	UI-TARS-1.5-7B	49.1	47.1	62.9	53.0
31	Qwen2.5-VL-32B-Instruct	54.8	58.0	69.7	60.8	**42**	Ovis-U1-3B	44.7	39.6	59.2	47.8
32	Step3-VL-10B	55.4	64.7	61.7	60.6	**43**	InternVL3.5-2B	45.1	40.3	52.7	46.0
33	SenseNova-U1-8B-MoT	48.2	59.2	67.5	58.3	**44**	InternVL-U-4B	46.1	37.9	49.8	44.6
Average over all models:		$T_1 = 57.3$		$T_2 = 65.1$	$T_3 = 71.6$	**Overall** $= 64.7$					

4 EXPERIMENTS

We conduct systematic evaluations. Results on offline primary tasks and variant tasks are in Sec. 4.1 and Sec. 4.2. Besides, Sec. 4.3 reports the online track, examining relations between offline scores and gameplay abilities. See the appendix for details on evaluation protocols and implementations.

4.1 OFFLINE PRIMARY TASKS

Benchmark Discriminability. We first test 44 models with question-answering abilities in Table 2. The mean accuracy across our benchmark is 64.7%, substantially above the 25% random baseline, suggesting that the designed tasks are solvable yet challenging for meaningful evaluation. Among models, the accuracy ranges from 44.6% to 80.2%, spanning 35.6% and clearly distinguishing models in different tiers. Among tasks, the results reveal a difficulty hierarchy, *i.e.*, 57.3% on T_1, 65.1% on T_2, and 71.6% on T_3. T_1 is the most difficult task because it requires fine-grained action prediction under strict criteria from keyboard-mouse trajectories. T_3 emphasizes cross-horizon understanding and exhibits the highest accuracy because the options provide rich information from instructions and actions. T_2 lies between T_1 and T_3, focusing on the planning of complex goals. Our GameHorizon-Bench shows an appropriate task-difficulty gradient and clear model discriminability.

General-Purpose VLMs. Proprietary models dominate the top of the leaderboard. GPT-6-Astra ranks first with 80.2% overall accuracy, followed by Gemini 3.8 Flash and Gemini 3.7 Flash with 77.3% and 76.7%. Kimi-K3 is the only open-weight model in Tier 1, ranking sixth with 74.5%. Recent open-weight models show competitive performance, *e.g.*, Qwen3.8-27B, Kimi-K2.6, and Gemma 4 31B-IT rank 16th, 17th, and 20th, surpassing several proprietary models. Within the same model family, newer or larger variants generally perform better. For example, overall accuracy increases from 62.8% for GPT-5.2 to 73.2% for GPT-5.5, 74.8% for GPT-5.6 Sol, and 80.2% for GPT-6-Astra. Similar trends are observed in other families, such as Gemini, Kimi, Qwen, and Doubao.

Unified Multimodal Models. We evaluate BAGEL-7B-MoT (Deng et al., 2025), SenseNova-U1-8B-MoT (Diao et al., 2026), Ovis-U1-3B (Wang et al., 2025a), and InternVL-U-4B (Tian et al., 2026). All four models fall into Tier 3 or Tier 4, indicating that current UMMs remain less competitive in gameplay tasks. The larger models perform better, *e.g.*, BAGEL-7B-MoT achieves 61.4% and ranks 30th. In comparison, Ovis-U1-3B and InternVL-U-4B achieve only 47.8% and 44.6%. To examine potential interference between understanding and generation, we compare InternVL-U-4B with its understanding backbone, InternVL3.5-2B. Despite having additional generation parameters, InternVL-U-4B underperforms InternVL3.5-2B by 1.4%. The result suggests that unifying understanding and generation could introduce interference, offsetting gains from the increased capacity.

Table 3: **Offline results of game agents.** Gemma 4 31B-IT is presented for reference. Game agents are trained for actions without question answering or reasoning abilities. We test them on the action task T_1. In-domain denotes model fine-tuning, *e.g.*, JARVIS-VLA on VPT. Zero-shot is the opposite.

Model	T_1	In-Domain	Zero-Shot
NitroGen	22.0	✗	✓
Open-P2P	24.4	✗	✓
JARVIS-VLA	**28.1**	✗	✓

(a) **GameHorizon-Bench (AAA games)**

Model	T_1	In-Domain	Zero-Shot
Open-P2P	26.8	✗	✓
Gemma 4 31B-IT	54.1	✗	✓
JARVIS-VLA	**54.2**	✓	✗

(b) **VPT (Minecraft)**

Table 4: **Results of the offline variant tasks T_1^*.** The table presents six variants of the primary task T_1, including current-action perception and future-action planning under three input settings, *i.e.*, frames only, frames with short-horizon instructions, and frames with multi-horizon instructions. The parenthesized values represent differences in percentage points relative to the preceding column.

Model	Current-Action Perception			Future-Action Planning		
	Frames Only	+ Short	+ Multi-Horizon	Frames Only	+ Short	+ Multi-Horizon
Qwen2.5-VL-32B-Instruct	39.0	53.5 (+14.5)	54.5 (+1.0)	33.5	29.5 (−4.0)	33.5 (+4.0)
Qwen3-VL-235B-A22B-Instruct	41.0	52.0 (+11.0)	54.5 (+2.5)	37.0	34.0 (−3.0)	39.5 (+5.5)
Gemini 3.1 Pro	50.0	67.0 (+17.0)	67.0 (+0.0)	34.0	36.5 (+2.5)	46.5 (+10.0)
Gemini 3.5 Flash	48.0	62.0 (+14.0)	64.0 (+2.0)	34.0	39.0 (+5.0)	47.5 (+8.5)
Overall	44.5	58.6 (+14.1)	60.0 (+1.4)	34.6	34.8 (+0.2)	41.8 (+7.0)

Table 5: **Offline variant tasks T_2^* and T_3^*.** The left table compares the top-down decomposition with bottom-up abstraction. The right table quantifies the accuracy loss from action decoding in T_3.

Model	Top-Down Decomposition	Bottom-Up Abstraction
Qwen2.5-VL-32B-Instruct	61.0	98.0 (+37.0)
Qwen3-VL-235B-A22B-Instruct	61.5	98.0 (+36.5)
Gemini 3.1 Pro	76.5	99.0 (+22.5)
Gemini 3.5 Flash	79.0	98.5 (+19.5)
Overall	69.5	98.4 (+28.9)

(a) **Goal Decomposition and Abstraction (T_2^*)**

Model	$L_3 - L_2 - L_1$ w/o Actions	$L_3 - L_2 - L_1$ w/ Actions
Qwen2.5-VL-32B-Instruct	69.0	68.5 (−0.5)
Qwen3-VL-235B-A22B-Instruct	72.0	71.0 (−1.0)
Gemini 3.1 Pro	77.5	77.5 (+0.0)
Gemini 3.5 Flash	77.5	76.5 (−1.0)
Overall	74.0	73.4 (−0.6)

(b) **Cross-Horizon Consistency (T_3^*)**

GUI Agents. We test GELab-Zero-4B-Preview (GELab-Team, StepFun, 2025) and UI-TARS-1.5-7B (Qin et al., 2025). Both models fall into Tier 4, indicating limited transfer from computer-use environments to AAA gameplay. GELab-Zero-4B-Preview outperforms UI-TARS-1.5-7B by 4.1%. UI-TARS-1.5-7B achieves 53.0%, trailing its general-purpose baseline Qwen2.5-VL-7B-Instruct by 3.7%. The results suggest that domain-specific post-training weakens cross-domain generalization.

Coding Agents. We evaluate Claude Fable 5, Opus 4.8, and Sonnet 5. Fable 5 performs best, ranking 13th in Tier 2 with 71.2% overall accuracy. Opus 4.8 and Sonnet 5 fall into Tier 3, ranking 23rd and 24th with 64.8% and 64.7%. Despite the focus on coding abilities, these models generalize well to gameplay and remain competitive on our benchmark. Their advantage over the GUI agents may stem from broader training data, stronger foundation models, and better reasoning capabilities.

Game Agents. In Table 3, three game agents perform near the 25% random baseline on unseen AAA games. The scores can fall below chance because action outputs that fail to match any option are counted as incorrect, *e.g.*, empty actions. JARVIS-VLA (Li et al., 2025) shows strong domain dependence. After fine-tuning on VPT (Baker et al., 2022), it achieves 54.2% on Minecraft, comparable to the zero-shot Gemma 4 31B-IT. However, the accuracy drops to 28.1% on unseen games. NitroGen (Magne et al., 2026) and Open-P2P (Yue et al., 2026) are similar, as neither is trained on AAA data with multi-horizon instructions and actions. The results highlight the value of our dataset.

4.2 Offline Variant Tasks

Given the evaluation costs, we test four models on 200 random questions for each of the ten variants.

Action Perception and Planning. As shown in Table 4, all four models consistently perform better on current-action perception than future-action planning across the three input settings. On average, perception outperforms planning by 9.9 percentage points with frames only, 23.8 points with short-horizon instructions, and 18.2 points with multi-horizon instructions. The gap indicates that models more reliably recognize actions in the observed frames than determine future actions. Future-action planning is challenging because it requires inferring subsequent behavior from temporal context. Nevertheless, the planning capability is essential to gameplay, especially for the long-horizon tasks.

Table 6: **Online results of GameHorizon-Bench.** We test 12 models in online track, with three models per offline tier. Each entry reports the success rate as a percentage, followed by the number of passed tasks out of the total in parentheses. Offline and online rankings show a positive association.

Tier 1							Tier 2						
Online Rank	Model	Offline Rank	Short-Horizon Subtasks	Long-Horizon Tasks	Causal	Thematic	Online Rank	Model	Offline Rank	Short-Horizon Subtasks	Long-Horizon Tasks	Causal	Thematic
1	GPT-6-Astra	1	**66.1** (41/62)	**45.0** (9/20)	**40.0** (4/10)	**50.0** (5/10)	**4**	GPT-5.6 Terra	14	37.1 (23/62)	10.0 (2/20)	20.0 (2/10)	0.0 (0/10)
2	Gemini 3.6 Flash	4	56.5 (35/62)	30.0 (6/20)	30.0 (3/10)	30.0 (3/10)	**5**	GPT-5.6 Luna	17	33.9 (21/62)	10.0 (2/20)	10.0 (1/10)	10.0 (1/10)
3	Kimi-K3	6	46.8 (29/62)	10.0 (2/20)	20.0 (2/10)	0.0 (0/10)	**6**	MiniMax-M3	21	29.0 (18/62)	10.0 (2/20)	10.0 (1/10)	10.0 (1/10)
Tier 3							**Tier 4**						
Online Rank	Model	Offline Rank	Short-Horizon Subtasks	Long-Horizon Tasks	Causal	Thematic	Online Rank	Model	Offline Rank	Short-Horizon Subtasks	Long-Horizon Tasks	Causal	Thematic
7	GLM-5V-Turbo	27	27.4 (17/62)	5.0 (1/20)	10.0 (1/10)	0.0 (0/10)	**10**	Qwen3.6-35B-A3B	34	11.3 (7/62)	5.0 (1/20)	10.0 (1/10)	0.0 (0/10)
8	Qwen3.5-397B-A17B	29	19.4 (12/62)	5.0 (1/20)	10.0 (1/10)	0.0 (0/10)	**11**	InternVL3.5-8B	38	3.2 (2/62)	0.0 (0/20)	0.0 (0/10)	0.0 (0/10)
9	Step3-VL-10B	32	14.5 (9/62)	5.0 (1/20)	10.0 (1/10)	0.0 (0/10)	**12**	UI-TARS-1.5-7B	41	1.6 (1/62)	0.0 (0/20)	0.0 (0/10)	0.0 (0/10)

Visual Inputs and Textual Instructions. Table 4 shows that short-horizon instructions consistently improve action perception across all four models. Average accuracy increases by 14.1%. The improvement is achieved solely by adding instructions to model inputs without fine-tuning on our data. The results indicate that visual observations alone (Magne et al., 2026) may not fully specify task intent, whereas instructions help models identify the correct operations for specified goals. The finding underscores the value of our instructions as inputs for training and evaluation in gameplay.

Multi-Horizon Pyramid. In Table 4, instruction horizons have distinct effects on current-action perception and future-action planning. For action perception, short-horizon instructions increase the average accuracy by 14.1%, whereas multi-horizon instructions yield only a further 1.4% improvement. For future planning, short-horizon instructions produce only a 0.2% improvement, while multi-horizon instructions further increase accuracy by 7.0%. Short-horizon instructions directly describe immediate operations and align more closely with current actions. Medium- and long-horizon instructions provide sustained goals and strategies that better guide future planning. These results demonstrate the advantage of our multi-horizon pyramid over the prior simple instructions. Our multi-horizon framework can support both immediate decision-making and future-task planning.

Top-Down and Bottom-Up. In Table 5(a), models perform much better on bottom-up abstraction from L_1 to L_2 than top-down decomposition. For abstraction, concrete low-level operations contain sufficient information to identify higher-level goals, resulting in near-ceiling accuracy. Although this saturation limits its discriminability, the reliable bottom-up mapping is well suited to our annotator. Decomposition starts from abstract goals that may correspond to multiple valid operations. The ambiguity makes it unsuitable for annotation but turns it into a challenging task. The results justify our designs, *i.e.*, bottom-up abstraction for annotation and top-down decomposition for evaluation.

Action Decoding. In Table 5(b), adding actions reduces accuracy by 0.6%, showing that action decoding introduces little error beyond matching instructions across horizons. Most errors arise from planning and aligning multi-horizon goals. This helps locate the bottleneck for current models.

4.3 Stepwise Online Track

In Table 6, online rankings align with offline rankings. The positive association indicates that our offline track captures relevant abilities and provides an effective indicator of online gameplay performance. The long-horizon online tasks are difficult, *e.g.*, GPT-6-Astra has a 45.0% success rate. A single failed step can invalidate the entire task. The results expose existing limitations in long-horizon game planning and strategy formulation. See the appendix for more results and analysis.

5 Conclusion

We propose GameHorizon, a unified data and evaluation suite that measures gameplay capabilities at different horizons across diverse models. GameHorizon-Annotator automatically builds a dense instruction pyramid spanning short-horizon operations, medium-horizon goals, and long-horizon strategies. GameHorizon-Data is a large-scale AAA gameplay dataset with aligned videos, actions, and multi-horizon instructions in 21 titles. GameHorizon-Bench combines reproducible offline evaluation with stepwise online testing, enabling standardized model comparisons. Experiments reveal capability differences and key bottlenecks among current models. Our work can serve as a data foundation and unified yardstick for examining gameplay performance across models and horizons.

AI USE STATEMENT

We use generative AI tools solely for language polishing of the manuscript, including checking for grammatical errors, improving sentence structure, and refining word choice. These tools are not used to generate research ideas, methods, experiments, data, or results. All AI-assisted revisions are reviewed and verified by the authors. We take full responsibility for the final content of this work.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Anthropic. Claude model system cards. `https://www.anthropic.com/system-cards`, 2026. Accessed: 2026-08-20.

Bowen Baker, Ilge Akkaya, Peter Zhokov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. Video pretraining (vpt): Learning to act by watching unlabeled online videos. In *Advances in neural information processing systems*, volume 35, pp. 24639–24654, 2022.

Richard Bellman. Dynamic Programming. *Science*, 153(3731):34–37, 1966.

Adrian Bolton, Alexander Lerchner, Alexandra Cordell, Alexandre Moufarek, Andrew Bolt, Andrew Lampinen, Anna Mitenkova, Arne Olav Hallingstad, Bojan Vujatovic, Bonnie Li, et al. Sima 2: A generalist embodied agent for virtual worlds. *arXiv preprint arXiv:2512.04797*, 2025.

Brandon Castellano. PySceneDetect. `https://www.scenedetect.com`, 2024. Software; accessed August 20, 2026.

Suhwan Choi, Jaeyoon Jung, Haebin Seong, Minchan Kim, Minyeong Kim, Yongjun Cho, Yoonshik Kim, Yu Park, Youngjae Yu, and Yunsung Lee. D2e: Scaling vision-action pretraining on desktop data for transfer to embodied ai. In *International Conference on Learning Representations*, 2026.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, et al. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

Haiwen Diao, Penghao Wu, Hanming Deng, Jiahao Wang, Shihao Bai, Silei Wu, Weichen Fan, Wenjie Ye, Wenwen Tong, Xiangyu Fan, et al. Sensenova-u1: Unifying multimodal understanding and generation with neo-unify architecture. *arXiv preprint arXiv:2605.12500*, 2026.

Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. Minedojo: Building open-ended embodied agents with internet-scale knowledge. 35:18343–18362, 2022.

GELab-Team, StepFun. Step-GUI technical report. *arXiv preprint arXiv:2512.15431*, 2025.

Google DeepMind. Gemini 3.5 Flash: Model card. `https://deepmind.google/models/model-cards/gemini-3-5-flash/`, 2026a. Accessed: 2026-09-05.

Google DeepMind. Gemini 3.6 Flash: Model card. `https://deepmind.google/models/model-cards/gemini-3-6-flash/`, 2026b. Accessed: 2026-09-05.

Rongjin Guo, Dong Liang, Yuhao Liu, Fang Liu, Tianyu Huang, Gerhard P Hancke, and Rynson WH Lau. Egocs-400k: An egocentric gameplay dataset for world models. *arXiv preprint arXiv:2606.18180*, 2026.

William H. Guss, Brandon Houghton, Nicholay Topin, Phillip Wang, Cayden Codel, Manuela Veloso, and Ruslan Salakhutdinov. Minerl: a large-scale dataset of minecraft demonstrations. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 2442–2448, 2019.

Yingchen He, Christian D Weilbach, Martyna E Wojciechowska, Yuxuan Zhang, and Frank Wood. Plaicraft: Large-scale time-aligned vision-speech-action dataset for embodied ai. *arXiv preprint arXiv:2505.12707*, 2025.

Tianjie Ju, Yueqing Sun, Zheng Wu, Wei Zhang, Yaqi Huo, Xi Su, Qi Gu, Xunliang Cai, Gongshen Liu, and Zhuosheng Zhang. Mineexplorer: Evaluating open-world exploration of mllm agents in minecraft, 2026.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Muyao Li, Zihao Wang, Kaichen He, Xiaojian Ma, and Yitao Liang. Jarvis-vla: Post-training large-scale vision language models to play visual games with keyboards and mouse. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 17878–17899, 2025.

Zhen Li, Zian Meng, Shuwei Shi, Wenshuo Peng, Yuwei Wu, Bo Zheng, Chuanhao Li, and Kaipeng Zhang. Wildworld: A large-scale dataset for dynamic world modeling with actions and explicit state toward generative arpg. 2026.

Shalev Lifshitz, Keiran Paster, Harris Chan, Jimmy Ba, and Sheila McIlraith. Steve-1: A generative model for text-to-behavior in minecraft. 36:69900–69929, 2023.

Loïc Magne, Anas Awadalla, Guanzhi Wang, Yinzhen Xu, Joshua Belofsky, Fengyuan Hu, Joohwan Kim, Ludwig Schmidt, Georgia Gkioxari, Jan Kautz, Yisong Yue, Yejin Choi, Yuke Zhu, and Linxi Fan. Nitrogen: An open foundation model for generalist gaming agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21511–21521, 2026.

Markov AI. gaming-500-hours. Hugging Face Datasets, `https://huggingface.co/datasets/markov-ai/gaming-500-hours`, 2026. Accessed: 2026-08-20.

Mingyu Ouyang, Siyuan Hu, Kevin Qinghong Lin, Hwee Tou Ng, and Mike Zheng Shou. Gameworld: Towards standardized and verifiable evaluation of multimodal game agents. 2026.

Davide Paglieri, Bartł omiej Cupiał, Samuel Coward, Ulyana Piterbarg, Maciej Woł czyk, Akbir Khan, Eduardo Pignatelli, Ł ukasz Kuciński, Lerrel Pinto, Rob Fergus, Jakob Foerster, Jack Parker-Holder, and Tim Rocktaeschel. BALROG: Benchmarking agentic LLM and VLM reasoning on games. In *International Conference on Learning Representations*, 2025.

Dongmin Park, Minkyu Kim, Beongjun Choi, Junhyuck Kim, Keon Lee, Jonghyun Lee, Inkyu Park, ByeongUk Lee, Jaeyoung Hwang, JAEWOO AHN, Ameya Mahabaleshwarkar, Bilal Kartal, Pritam Biswas, Yoshi Suhara, Kangwook Lee, and Jaewoong Cho. Orak: A foundational benchmark for training and evaluating LLM agents on diverse video games. In *International Conference on Learning Representations*, 2026.

Yujia Qin, Yining Ye, Junjie Fang, Haoming Wang, Shihao Liang, Shizuo Tian, Junda Zhang, Jiahao Li, Yunxin Li, Shijue Huang, et al. Ui-tars: Pioneering automated gui interaction with native agents. *arXiv preprint arXiv:2501.12326*, 2025.

Weihao Tan, Wentao Zhang, Xinrun Xu, Haochong Xia, Ziluo Ding, Boyu Li, Bohan Zhou, Junpeng Yue, Jiechuan Jiang, Yewen Li, et al. Cradle: Empowering foundation agents towards general computer control. *arXiv preprint arXiv:2403.03186*, 2024.

Weihao Tan, Xiangyang Li, Yunhao Fang, Heyuan Yao, Shi Yan, Hao Luo, Tenglong Ao, Huihui Li, Hongbin Ren, Bairen Yi, et al. Lumine: An open recipe for building generalist agents in 3d open worlds. *arXiv preprint arXiv:2511.08892*, 2025.

Team Gemini. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Changyao Tian, Danni Yang, Guanzhou Chen, Erfei Cui, Zhaokai Wang, Yuchen Duan, Penghao Yin, Sitao Chen, Ganlin Yang, Mingxin Liu, et al. Internvl-u: Democratizing unified multimodal models for understanding, reasoning, generation and editing. *arXiv preprint arXiv:2603.09877*, 2026.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models, 2023. *arXiv preprint arXiv:2305.16291*, 2023.

Guo-Hua Wang, Shanshan Zhao, Xinjie Zhang, Liangfu Cao, Pengxin Zhan, Lunhao Duan, Shiyin Lu, Minghao Fu, Xiaohao Chen, Jianshan Zhao, et al. Ovis-u1 technical report. *arXiv preprint arXiv:2506.23044*, 2025a.

Haoming Wang, Haoyang Zou, Huatong Song, Jiazhan Feng, Junjie Fang, Junting Lu, Longxiang Liu, Qinyu Luo, Shihao Liang, Shijue Huang, et al. Ui-tars-2 technical report: Advancing gui agent with multi-turn reinforcement learning. *arXiv preprint arXiv:2509.02544*, 2025b.

Zihao Wang, Xujing Li, Yining Ye, Junjie Fang, Haoming Wang, Longxiang Liu, Shihao Liang, Junting Lu, Zhiyong Wu, Jiazhan Feng, et al. Game-tars: Pretrained foundation models for scalable generalist multimodal game agents. *arXiv preprint arXiv:2510.23691*, 2025c.

Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. 34:12077–12090, 2021.

Yuguang Yue, Irakli Salia, Samuel Hunt, Chris Green, Wenzhe Shi, and Jonathan J Hunt. Scaling behavior cloning improves causal reasoning: An open model for real-time video game playing. *arXiv preprint arXiv:2601.04575*, 2026.

Alex L Zhang, Thomas L Griffiths, Karthik R Narasimhan, and Ofir Press. Videogamebench: Can vision-language models complete popular video games? *arXiv preprint arXiv:2505.18134*, 2025.

Kuan Zhang, Dongchen Liu, Qiyue Zhao, Jinkun Hou, Xinran Zhang, Qinlei Xie, Miao Liu, and Yiming Li. Gameverse: Can vision-language models learn from video-based reflection? *arXiv preprint arXiv:2603.06656*, 2026.

Xinyue Zheng, Haowei Lin, Kaichen He, Zihao Wang, Qiang Fu, Haobo Fu, Zilong Zheng, and Yitao Liang. MCU: An evaluation framework for open-ended game agents. In *International conference on machine learning*, pp. 78221–78259. PMLR, 2025.

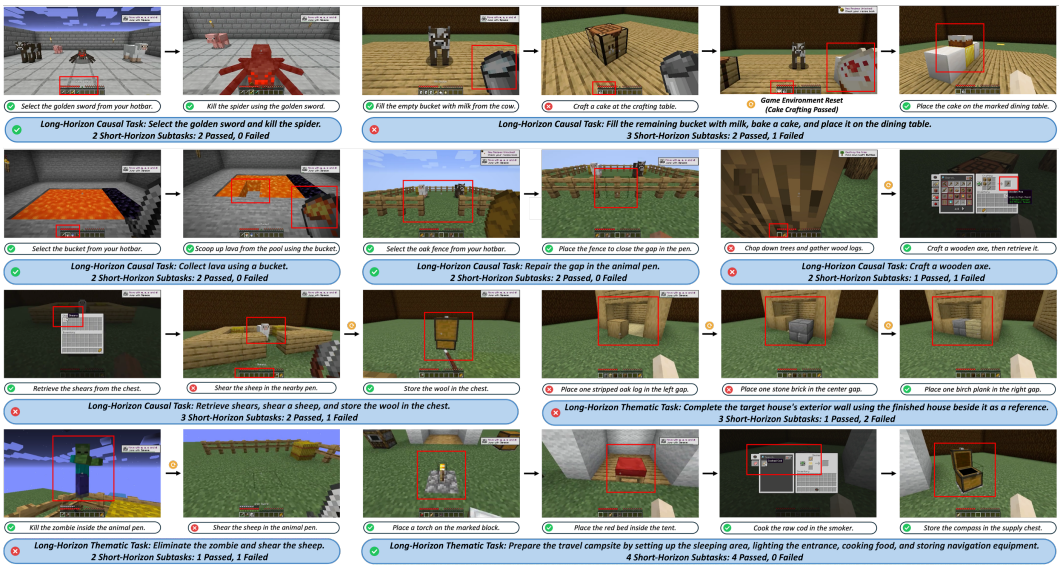

Figure 4: **The online track of GameHorizon-Bench.** It contains long-horizon causal and thematic tasks, each comprising multiple verifiable short-horizon subtasks. The causal tasks can only be completed in a prescribed order because of dependencies between successive subtasks. The thematic tasks contain subtasks that share a common theme but can be performed in any order. The figure shows results from Gemini 3.6 Flash (Google DeepMind, 2026b), with arrows indicating its actual execution order. After a subtask fails, the environment is reset to the success state so that evaluation can continue. A long-horizon task is considered passed only when all constituent subtasks succeed.

This appendix contains the following contents:

- More details of our GameHorizon-Bench, including evaluation protocols, implementation details, benchmark comparisons, more offline quantitative metrics, as well as online qualitative examples.
- More details on GameHorizon-Data, *e.g.*, data statistics across all games and instruction horizons.
- More details of the GameHorizon-Annotator, *e.g.*, prompt, segmentation, keybind, and workflow.

A More Details on GameHorizon-Bench

A.1 Stepwise Diagnosis of the Online Track

Examples of our online tasks are presented in Fig. 4 with the actual execution order and results of Gemini 3.6 Flash. As described in Sec. 3.3 of the main paper, each long-horizon causal or thematic task is decomposed into short-horizon subtasks. The stepwise protocol with game resets can localize failures to specific operations. *E.g.*, we can distinguish models that do not know how to craft a cake from those that understand the procedure but perform a certain step incorrectly. In the failure cases, Gemini 3.6 Flash understands intended operations yet commits execution errors, *e.g.*, placing objects in wrong positions or failing to follow the correct recipe. Besides, in Table 12, we list all 20 long-horizon tasks and their 62 constituent subtasks, grouped into causal and thematic categories.

A.2 Effects of Thinking on the Offline Track

As shown in Table 7, thinking does not uniformly improve offline performance. The effects vary with model reasoning quality. Enabling thinking improves the accuracy of GLM-5V-Turbo and Doubao-Seed-2.1-Pro by 5.6 and 9.3 percentage points. The gains are particularly pronounced on T_2, reaching 9.0 and 19.9 percentage points, suggesting that capable models can leverage thinking to decompose and plan complex goals. In contrast, thinking reduces the accuracy of UI-TARS-1.5-7B and Qwen2.5-VL-7B by 8.2 and 5.9 percentage points. Qualitative inspection indicates that lightweight models can produce hallucinated reasoning, thereby misleading action prediction rather

Table 7: **Effects of thinking on offline results.** Δ denotes the change in overall accuracy. For lightweight models, thinking often induces hallucinations, leading to performance degradation. More capable models can leverage thinking to plan and decompose complex goals in gameplay sessions, therefore improving the accuracy, such as the GLM-5V-Turbo and Doubao-Seed-2.1-Pro.

Model	Thinking	T_1	T_2	T_3	Overall	Δ
UI-TARS-1.5-7B	✗	49.1	47.1	62.9	53.0	–
	✓	45.2	45.4	43.9	44.8	**-8.2**
Qwen2.5-VL-7B	✗	52.0	54.3	63.7	56.7	–
	✓	44.4	52.2	55.7	50.8	**-5.9**

(a) Degradation with Thinking

Model	Thinking	T_1	T_2	T_3	Overall	Δ
GLM-5V-Turbo	✗	53.2	56.4	63.1	57.6	–
	✓	53.4	65.4	70.8	63.2	**+5.6**
Doubao-Seed-2.1-Pro	✗	56.4	56.8	75.2	62.8	–
	✓	62.9	76.7	76.6	72.1	**+9.3**

(b) Improvement with Thinking

Table 8: **Comparison with existing gameplay benchmarks.** Model diversity denotes evaluation across at least three model types, *e.g.*, VLMs, UMMs, GUI agents, coding agents, and game agents.

Benchmark	Benchmark Scope		Annotation Properties		Evaluation Settings		Notes
	Model Diversity	AAA-Focused	Direct Human Actions	Multi-Horizon	Reproducible Offline	Stepwise Online	
Single-Game Benchmarks							
MineDojo (Fan et al., 2022)	✗	✗	✗	✗	✗	✗	Online success rates in Minecraft; the game agents only.
MCU (Zheng et al., 2025)	✗	✗	✗	✗	✗	✗	Minecraft; VLM-based online scores; game agents only.
MineExplorer (Ju et al., 2026)	✗	✗	✗	✗	✗	✓	Online milestone rates in Minecraft; the VLMs only.
Multi-Game Benchmarks							
BALROG (Paglieri et al., 2025)	✗	✗	✗	✗	✗	✓	Non-AAA; the online progress scores; LLMs and VLMs.
VideoGameBench (Zhang et al., 2025)	✗	✗	✗	✗	✗	✗	Retro and classic games; online scores and VLMs only.
Orak (Park et al., 2026)	✗	✗	✗	✗	✗	✗	Majority non-AAA; the online scores; LLMs and VLMs.
GameWorld (Ouyang et al., 2026)	✗	✗	✗	✗	✗	✗	Browser mini-games only; online scores; VLM agents.
GameVerse (Zhang et al., 2026)	✗	✗	✗	✗	✗	✓	Mostly non-AAA; online milestone scores; VLM agents.
GameHorizon-Bench (Ours)	✓	✓	✓	✓	✓	✓	**AAA; reproducible offline; stepwise online; diverse models.**

than improving it. The results indicate that the efficacy of thinking in gameplay depends on whether it enables models to decompose and plan complex goals correctly. The findings reveal an inherent trade-off between the reasoning ability and the model efficiency in real-time gameplay sessions.

A.3 Comparisons with Other Benchmarks

In line 315 of the paper, we compare GameHorizon-Bench with other benchmarks. Here, Table 8 provides detailed comparisons in benchmark scopes, annotation properties, and evaluation settings.

A.4 Evaluation Protocols and Details

Offline Track. We test 47 models across five types, including 35 general-purpose VLMs, 4 UMMs, 2 GUI agents, 3 coding agents, and 3 dedicated game agents. The 44 models of the first four types natively support question answering. We adopt unified MCQs for all offline tasks. Models receive the same four-option questions and output selections for standardized and reproducible evaluation.

The dedicated game agents (Yue et al., 2026; Magne et al., 2026; Li et al., 2025) are trained to produce actions without supporting question answering or reasoning. We evaluate these models only on the single-horizon action task (T_1) by matching their action predictions against the four options via sequence alignment. The option with the highest alignment score is taken as the selected answer. If all four scores fall below a threshold, indicating no meaningful overlap with any option, the prediction is marked incorrect, *e.g.*, an empty action sequence. For NitroGen (Magne et al., 2026), we additionally map its gamepad actions to the keyboard-mouse trajectories using our keybinds.

Each of the three primary tasks (T_1–T_3) contains 1,000 MCQs, while each of the ten variants contains 200 MCQs. The offline track consists of 5,000 MCQs for high-confidence and reliable evaluation. Some models are evaluated through their official APIs, while others are served locally on GPUs. The inference of game agents follows their official checkpoints, rollout procedures, and context-window sizes, whereas the other four model categories are given the same number of input frames. We report the MCQ accuracy for all models and divide them into four performance tiers.

Online Track. Due to evaluation costs, we sample three models per offline tier for online testing. We construct ten causal and ten thematic tasks with 62 short-horizon subtasks. The thematic tasks span diverse themes, *e.g.*, resource gathering, crafting, farming, building, and combat. A long-horizon task is considered passed only when the model completes it end-to-end without intermediate failures

Table 9: **Statistics of GameHorizon-Data across 21 game titles.** We report the duration and share of recordings, counts of actions and instructions, and the average temporal span of each distinct instruction. Games are sorted by total duration. Valid rate denotes the fraction retained for instruction annotation after filtering. The actions are reported in units of 10^4, and instruction spans in seconds.

Game Title	Recording Duration and Share				Annotation Counts				Avg. Instruction Span (s)		
	Total (h)	Share (%)	Valid Rate (%)	Valid (h)	Actions (10^4)	L_1	L_2	L_3	L_1	L_2	L_3
Valorant	617.5	12.35	86.4	533.8	5,811	705,380	23,113	5,778	2.72	83.1	332.5
Minecraft	589.8	11.80	84.5	498.3	5,087	448,166	21,715	5,429	4.00	82.6	330.4
Grand Theft Auto V	566.1	11.32	86.5	489.9	3,585	382,302	21,644	5,411	4.61	81.5	325.9
Palworld	438.7	8.77	83.5	366.5	3,741	329,615	15,971	3,993	4.00	82.6	330.4
Delta Force	337.0	6.74	85.1	286.9	3,123	379,091	12,422	3,105	2.72	83.1	332.5
Roco Kingdom: World	329.2	6.58	91.9	302.4	2,677	786,446	13,054	3,263	1.38	83.4	333.6
Red Dead Redemption 2	304.4	6.09	85.9	261.6	1,992	202,428	11,553	2,888	4.65	81.5	326.1
Cyberpunk 2077	301.2	6.02	87.9	264.7	2,574	192,222	11,646	2,912	4.96	81.8	327.3
Genshin Impact	232.6	4.65	90.4	210.4	1,862	547,134	9,082	2,270	1.38	83.4	333.6
PUBG: Battlegrounds	197.4	3.95	84.6	167.0	1,818	220,726	7,233	1,808	2.72	83.1	332.5
Elden Ring Nightreign	168.9	3.38	90.0	152.0	1,348	250,877	6,630	1,658	2.18	82.6	330.2
The Witcher 3: Wild Hunt	158.7	3.17	89.4	141.8	1,323	287,433	6,290	1,573	1.78	81.2	324.6
Elden Ring	135.9	2.72	92.6	125.9	1,116	207,706	5,489	1,372	2.18	82.6	330.2
Escape from Tarkov	119.8	2.40	81.7	97.9	1,065	129,320	4,237	1,059	2.72	83.1	332.5
Apex Legends	116.5	2.33	85.3	99.4	1,082	131,381	4,305	1,076	2.72	83.1	332.5
Wuthering Waves	108.4	2.17	89.1	96.5	819	294,088	4,077	1,019	1.18	85.2	341.0
Neverness to Everness	101.3	2.03	89.3	90.4	800	235,145	3,903	976	1.38	83.4	333.6
Assassin's Creed	51.1	1.02	87.0	44.5	338	34,401	1,963	491	4.65	81.5	326.1
Black Myth: Wukong	51.1	1.02	91.3	46.6	414	76,971	2,034	509	2.18	82.6	330.2
Watch Dogs 2	38.5	0.77	85.0	32.7	249	25,324	1,445	361	4.65	81.5	326.1
Honor of Kings: World	35.9	0.72	87.3	31.3	277	81,429	1,352	338	1.38	83.4	333.6
Total / Avg.	5,000	100.00	86.8	4,341	41,103	5,947,588	189,158	47,290	2.63	82.6	330.4

and game resets. Each subtask is allocated a maximum budget of 400 model calls. Any subtask not completed within the budget is failed. The task results are automatically determined by the underlying game states. We report the success rates for both short-horizon subtasks and long-horizon tasks.

B More Details on GameHorizon-Data

In line 210 of the manuscript, we report and discuss the overall statistics of GameHorizon-Data. Here, as shown in Table 9, we provide detailed statistics across all 21 game titles, including recording durations, valid rates, action and instruction counts, and average instruction spans at each horizon.

C More Details on GameHorizon-Annotator

C.1 Overview of GameHorizon Suite

First, to facilitate understanding, we provide an additional overview of our GameHorizon Suite. As shown in Fig. 2 of the main paper, it comprises three components: GameHorizon-Annotator, GameHorizon-Data, and GameHorizon-Bench. The workflow begins with raw gameplay acquisition. Previous work collects web videos and recovers action pseudo-labels via Inverse Dynamics Models (IDMs) (Baker et al., 2022; Lifshitz et al., 2023; Choi et al., 2026) or gamepad segmentation (Magne et al., 2026; Xie et al., 2021). The inferred pseudo-labels can deviate from the actual controls executed by humans. In contrast, we recruit 100 experienced human players and deploy a dedicated recording-and-upload system to synchronously capture game videos at 2K resolution along with timestamped keyboard and mouse actions. Authentic human gameplay recordings and actions can establish a reliable foundation for the faithful evaluations in complex game worlds.

Based on the recordings, we develop GameHorizon-Annotator, an annotation pipeline for textual instructions across multiple temporal horizons. Due to annotation costs or vision-only architectures, prior studies either omit instructions (Magne et al., 2026; Zhang et al., 2026) or provide sparse labels (Yue et al., 2026). Game-TARS (Wang et al., 2025c) relies on manual instruction labeling, which is expensive and remains unavailable to the community. However, instructions are vital for human-agent interaction, *e.g.*, instruction following and goal decomposition. When issuing requests to an agent, humans dictate not only primitive actions like reloading a weapon, but also long-term objectives such as defending a bridge. To this end, our GameHorizon-Annotator produces a three-level pyramid of instructions, including short-horizon operations (1–5 seconds), medium-horizon

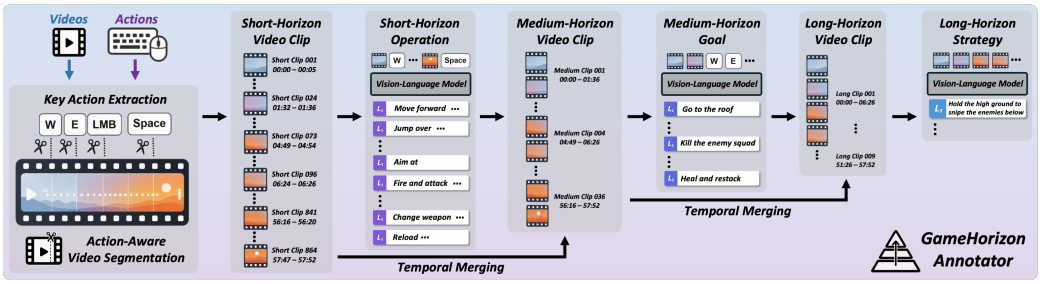

Figure 5: **Workflow of GameHorizon-Annotator.** Videos and actions are first processed by action-aware segmentation to produce short-horizon clips, with key actions determining their temporal boundaries. A VLM annotates each clip with an L_1 operation. Lower-level clips with their instructions are progressively merged into medium- and long-horizon clips based on action continuity and semantic coherence, from which the VLM derives L_2 goals and L_3 strategies. The bottom-up procedure abstracts fine-grained video-action trajectories into a pyramid of multi-horizon instructions.

goals (1–2 minutes), and long-horizon strategies (5–8 minutes). The annotator operates bottom-up, abstracting dense and action-grounded instructions into higher-level goals and strategies. The automated pipeline reduces annotation costs and enables scalable labeling across large collections.

Applying GameHorizon-Annotator to the collected trajectories, we construct GameHorizon-Data, the first large-scale AAA gameplay dataset that aligns videos, player actions, and multi-horizon instructions. It covers 5,000 hours of human gameplay from 21 game titles, spanning diverse genres such as open-world, action role-playing, competitive shooter, sandbox survival, and creature-collecting adventure titles. In total, GameHorizon-Data contains 4,571 videos recorded at 60 fps and 411.03 million keyboard-mouse action events. It is annotated with 6,184,036 distinct instructions, including 5,947,588 short-horizon operations, 189,158 medium-horizon goals, and 47,290 long-horizon strategies. On average, GameHorizon-Data provides one distinct short-horizon instruction every 2.63 seconds, while each frame is aligned with corresponding instructions at all three horizons. Our annotations are substantially denser than those in prior work. For example, Open-P2P (Yue et al., 2026) includes only one instruction every few minutes, with uneven temporal coverage. Combining scale, density, and diversity, GameHorizon-Data enables unified evaluations of multi-horizon gameplay tasks and capabilities across diverse models and AAA environments.

Finally, leveraging GameHorizon-Data, we introduce GameHorizon-Bench, a comprehensive benchmark featuring reproducible offline and stepwise online testing across diverse model families. Prior benchmarks predominantly rely on a small number of harness-dependent online rollouts (Zhang et al., 2025; Tan et al., 2025; Zhang et al., 2026), yielding low-confidence comparisons that are difficult to reproduce. Aggregate success rates also conflate different failure modes. In contrast, the offline track of our GameHorizon-Bench ensures reliable and reproducible evaluation through thousands of multiple-choice questions (MCQs) with standardized actions and instructions in three primary tasks, including single-horizon action, multi-horizon instruction decomposition, and cross-horizon consistency. Diagnostic variants further probe model capabilities along different dimensions, *e.g.*, current-action perception *vs.* future-action planning and top-down decomposition *vs.* bottom-up abstraction. Complementarily, the online track tests long-horizon gameplay through verifiable short-horizon subtasks. It covers causal tasks, whose subtasks follow a prescribed sequence, and thematic tasks, whose subtasks can be completed in any order. When an agent fails at a subtask, the environment is reset to the corresponding success state, allowing evaluation to continue. The stepwise protocol localizes errors to specific steps. Our online track further tests whether offline scores reflect actual gameplay abilities. Together, the two tracks establish GameHorizon-Bench as a unified, standardized, reproducible, and diagnostic yardstick across model families and horizons.

C.2 Overview of the Annotator

As illustrated in Fig. 5, action-aware segmentation first partitions a recording of a complete gameplay session into L_1 clips, each annotated with a short-horizon operation. Using the L_1 instructions as semantic cues, we merge adjacent L_1 clips into L_2 clips and annotate them with medium-horizon goals. We then merge L_2 clips into L_3 clips and annotate them with long-horizon strategies. The

GameHorizon-Annotator proceeds bottom-up via alternating temporal merging and instruction annotation across all temporal horizons, maintaining semantic consistency throughout the pyramid.

C.3 Annotation Prompts

As described in Sec. 3.1 of the main paper, we utilize a VLM to annotate the multi-horizon instructions. Here, we present the annotation prompts for short-horizon operations, medium-horizon goals, and long-horizon strategies in Fig. 6, Fig. 7, and Fig. 8, respectively. The short-horizon prompt takes sampled frames and temporally aligned keyboard-mouse actions as inputs. It requires the VLM to describe the current operation with sufficient critical details, such as object descriptions and spatial relations. The medium-horizon prompt also incorporates the constituent short-horizon instructions and asks the VLM to summarize the player's goal while avoiding excessive local details. Game-specific keybinds are additionally provided in the short- and medium-horizon prompts to help the VLM interpret keyboard and mouse actions in terms of their in-game semantics. The long-horizon prompt takes sampled frames and the constituent medium-horizon goals as inputs while omitting actions and keybinds. It requires the VLM to capture the player's overall gameplay strategy without local operational details. Across all three horizons, the prompts provide common video metadata, including the game type, clip duration, frame rate, and sampled frame indices. They also require the VLM to analyze the visual context and output a natural-language instruction without explanations.

C.4 Action-Aware Video Segmentation

We provide more details on the action-aware video segmentation. The segmentation follows four stages, including keyboard-mouse semantic calibration, primitive action classification, rule-based coarse segmentation, and VLM-based refinement. First, using game-specific keybinds, we map raw keyboard and mouse events to gameplay actions, such as movement, attack, interaction, skill activation, and menu operations. When one input can correspond to multiple actions, we retain all candidates instead of making an early decision. Second, we classify primitive actions as input-deterministic, input-ambiguous, or vision-dependent according to how reliably their semantics can be inferred from the inputs. Third, we replay the frame-aligned input logs to recover held keys, discrete triggers, mouse-button states, and camera movements. When multiple actions occur in the same frame, action-priority rules select the primary label, while the remaining actions are retained as concurrent evidence. Consecutive frames with identical labels are merged, short noisy segments are smoothed, and overly long segments are divided. Finally, ambiguous or vision-dependent segments are refined by a VLM using videos, candidate labels, action semantics, and trigger timestamps. We further remove temporal gaps and overlaps, producing segments with continuous action events. Our GameHorizon-Annotator is also compatible with segmentation tools such as PySceneDetect (Castellano, 2024), which detects boundaries based on inter-frame visual similarity and can therefore fragment continuous actions under large camera movements.

C.5 Game-Specific Keybinds

As mentioned in Sec. 3.1 and Sec. 3.3 of the main paper, we adopt game-specific keybinds for mapping keyboard and mouse inputs to in-game action semantics. We showcase the keybinds for Apex Legends and Cyberpunk 2077 as two representative examples in Table 10 and Table 11, respectively.

Annotation Prompt for Short-Horizon Operations

System Information

I am providing a sequence of *{sampled_frames_numbers}* game video **frames**. I am also providing the corresponding mouse (mouse_movement and mouse_key) and keyboard **action annotations** of the game player for each frame. The game type is *{game_type}*. Your task is to specify the current **short-horizon operation** of the game player as a short-term text instruction for a virtual agent.

Task

1. Analyze the visual context of the gameplay.
2. Identify key entities, UI elements, and environmental factors.
3. Correlate the player's actions with visual context to understand the intent and operation.
4. Output the short-horizon operation as a text instruction based on the visual evidence and action trajectory.

Goal

The instruction should include the **sufficient critical details** (e.g., coordinates, object descriptions, or spatial relations) to guide the agent in the gameplay.

Besides, you should output only the instruction text.

Important Rules

1. You should only output the textual instruction without any other text.
2. DO NOT output anything such as "The instruction is:" or "After analyzing the video."
3. Instructions should be written in natural language.

Information

1. Video Information
{sampled_frames_numbers} frames are sampled from the short-horizon clip. The duration is *{clip_duration}*. The frame rate is *{clip_fps}* fps. The frame index list is: *{sampled_frames_index_in_clip}*.

2. Action and Keybind Information
For better understanding, here is some keybind information about the *{game_type}* game.

For mouse movement [dx, dy], dx and dy represent mouse movement in the x and y directions.
dx > 0 means movement horizontally to the right. dx < 0 means movement horizontally to the left.
dy > 0 means movement vertically upward. dy < 0 means movement vertically downward.
If one term contains multiple actions, e.g., [[dx1,dy1],[dx2,dy2]], actions are listed in the order of occurrence.

Mouse-key and keyboard actions indicate the specific mouse buttons and keyboard keys that the player pressed or released. The mapping between keys and in-game actions is as follows: *{game_specific_keybinds}*.

Input

1. Video Input
The frames of the short-horizon clips are listed as: *{sampled_frames_input}*.
2. Action Input
For each sampled frame, the corresponding mouse and keyboard action annotations of the game player are provided below, which may help you understand the operation and intention: *{sampled_actions_input}*.

Figure 6: **The prompt for short-horizon operations.** Key differences across horizons are highlighted in red, such as the inputs, outputs, and required levels of detail in the instructions. Blue text denotes the code variables in the prompt. The short-horizon prompt takes sampled frames and temporally aligned keyboard-mouse actions as inputs. It requires the VLM to describe the current operation with sufficient details, such as coordinates, object descriptions, as well as spatial relations.

Annotation Prompt for Medium-Horizon Goals

System Information

I am providing a sequence of *{sampled_frames_numbers}* game video **frames**. I am also providing the corresponding mouse (mouse_movement and mouse_key), keyboard **actions**, and the **lower-level text instructions (short-horizon)** of the game player. The game type is *{game_type}*. Your task is to specify the **medium-horizon goal** of the game player as a medium-term text instruction for an agent.

Task

1. Analyze the visual context of the gameplay.
2. Identify key entities, UI elements, and environmental factors.
3. Correlate the player's actions and instructions with visual context to understand the intentions and goals.
4. Output the medium-horizon goal as a text instruction based on frames, actions, and short-term instructions.

Goal

The instruction should not focus excessively on local details. Instead, it must clearly state the player's goal throughout this video clip from *{min_L2_minute}* to *{max_L2_minute}* minutes. You can **include some details, only when necessary** to guide the agent.

Important Rules

1. You should only output the textual instruction without any other text.
2. DO NOT output anything such as "The instruction is:" or "After analyzing the video."
3. Instructions should be written in natural language.

Information

1. Video Information
{sampled_frames_numbers} frames are sampled from medium-horizon clip. The duration is *{clip_duration}*. The frame rate is *{clip_fps}* fps. The frame index list is: *{sampled_frames_index_in_clip}*.

2. Action and Keybind Information
For better understanding, here is some keybind information about the *{game_type}* game.

For mouse movement [dx, dy], dx and dy represent mouse movement in the x and y directions.
dx > 0 means movement horizontally to the right. dx < 0 means movement horizontally to the left.
dy > 0 means movement vertically upward. dy < 0 means movement vertically downward.
If one term contains multiple actions, e.g., [[dx1,dy1],[dx2,dy2]], actions are listed in the order of occurrence.

Mouse-key and keyboard actions indicate the specific mouse buttons and keyboard keys that the player pressed or released. The mapping between keys and in-game actions is as follows: *{game_specific_keybinds}*.

Input

1. Video Input
The frames of the medium-horizon clips are listed as: *{sampled_frames_input}*.
2. Instruction (short-horizon operations, i.e., the lower level)
The short-horizon operations included in the medium-horizon clips are listed as: *{sampled_L1_Instruct_input}*.
3. Action Input
For each sampled frame, the corresponding mouse and keyboard action annotations of the game player are provided below, which may help you understand the operation and intention: *{sampled_actions_input}*.

Figure 7: **The prompt for medium-horizon goals.** Key differences across the three horizons are highlighted in red, such as the inputs, outputs, and required levels of detail in the instructions. Blue text denotes the code variables in the prompt. The medium-horizon prompt incorporates video frames, aligned actions, and the constituent short-horizon instructions, requiring the VLM to summarize the sustained goals of the player while avoiding excessive local spatial and temporal details.

Annotation Prompt for Long-Horizon Strategies

System Information

I am providing a sequence of *{sampled_frames_numbers}* game video **frames**. I am also providing the **lower-level text instructions (medium-horizon goals)** of the game player. The game type is *{game_type}*. Your task is to specify the **long-horizon game strategies** of the game player as a long-term text instruction for an agent.

Task

1. Analyze the visual context of the gameplay.
2. Identify key entities, UI elements, and environmental factors.
3. Correlate the player's instructions with visual context to understand the long-term game strategies.
4. Output the long-horizon strategy as a text instruction based on frames and medium-horizon instructions.

Goal

The instruction **should not include local details**. It must clearly state the player's gameplay strategy throughout the video clip from *{min_L3_minute}* to *{max_L3_minute}* minutes. Besides, you should only output the instruction text without any other text.

Important Rules

1. You should only output the textual instruction without any other text.
2. DO NOT output anything such as "The instruction is:" or "After analyzing the video."
3. Instructions should be written in natural language.

Information

1. Video Information
{sampled_frames_numbers} frames are sampled from long-horizon clip. The duration is *{clip_duration}*. The frame rate is *{clip_fps}* fps. The frame index list is: *{sampled_frames_index_in_clip}*.

Input

1. Video Input
The frames of the long-horizon clips are listed as: *{sampled_frames_input}*.
2. Instruction (medium-horizon goals, i.e., the lower level)
The medium-horizon goals included in the long-horizon clips are listed as: *{sampled_L2_Instruct_input}*.

Figure 8: **The prompt for long-horizon strategies.** Key differences across three horizons are highlighted in red, such as the inputs, outputs, and required levels of detail in the instructions. Blue text denotes the code variables in the prompt. The long-horizon prompt takes sampled frames and the constituent medium-horizon goals as inputs while omitting actions and keybinds. It requires the VLM to capture the overall gameplay strategy and long-term intent without local operational details.

Table 10: **The keybind for Apex Legends.** Keyboard and mouse inputs are mapped to in-game action semantics via keybinds. Multiple candidates for an ambiguous input are separated by slashes.

Input Event	In-Game Action Semantics
KEY_W	Move Forward
KEY_S	Move Back
KEY_A	Move Left
KEY_D	Move Right
SPACE	Jump
LSHIFT	Sprint
KEY_C	Crouch – Toggle
LCONTROL	Crouch – Hold
RI_MOUSE_LEFT_BUTTON_DOWN	Attack
RI_MOUSE_RIGHT_BUTTON_DOWN	Aim Down Sight
KEY_R	Reload
KEY_V	Melee
KEY_1	Equip Weapon 1
KEY_2	Equip Weapon 2
KEY_3	Holster Weapons
RI_MOUSE_WHEEL	Cycle Weapon
KEY_G	Equip Grenade
KEY_N	Inspect Weapon
KEY_Q	Tactical Ability
KEY_Z	Ultimate Ability
KEY_E	Interact / Pickup / Open Door / Revive / Use Zipline
KEY_4	Use Selected Health Item
KEY_H	Character Utility Action
RI_MOUSE_MIDDLE_BUTTON_DOWN	Smart Tag / Ping
TAB	Open/Close Inventory
KEY_M	Open/Close Map

Table 11: **The keybind for Cyberpunk 2077.** Keyboard and mouse inputs are mapped to in-game action semantics via keybinds. Multiple candidates for an ambiguous input are separated by slashes.

Input Event	In-Game Action Semantics
KEY_W	Move Forward / Vehicle Accelerate
KEY_S	Move Back / Vehicle Brake or Reverse
KEY_A	Move Left / Vehicle Turn Left
KEY_D	Move Right / Vehicle Turn Right
SPACE	Jump / Vehicle Handbrake / Pause Braindance
LSHIFT	Sprint – Toggle / Change Braindance Layer
RSHIFT	Sprint – Hold
KEY_C	Crouch – Toggle / Skip Dialogue
LCONTROL	Crouch – Hold / Vehicle Horn / Panzer Smokescreen
RI_MOUSE_LEFT_BUTTON_DOWN	Shoot / Fast Attack / Panzer Primary Cannon
RI_MOUSE_RIGHT_BUTTON_DOWN	Aim / Block / Panzer Missile Launcher
RI_MOUSE_MIDDLE_BUTTON_DOWN	Use Combat Gadget / Tag / Reverse Camera / Braindance Zoom
RI_MOUSE_WHEEL	Cycle Weapon / Next or Previous Weapon / Braindance Zoom
KEY_R	Reload / Vehicle Radio / Restart Braindance
KEY_Q	Quick Attack / Vehicle Cycle Camera / Rewind Braindance
KEY_E	Cyberware Systems / Fast-Forward Braindance
KEY_F	Interact / Pickup / Enter or Exit Vehicle
KEY_X	Use Consumable / Exit Braindance
KEY_Y	Skip Dialogue / Dialogue Confirm
KEY_Z	Open Notifications / Quickhack Details
KEY_T	Open Phone
TAB	Scan – Hold / Braindance Analysis Mode
KEY_1	Select First Weapon / Previous Item
KEY_2	Select Second Weapon
KEY_3	Select Third Weapon / Next Item
LMENU	Cycle Weapon / Weapon Wheel / Vehicle Cycle Lights
KEY_V	Call Vehicle
KEY_B	Draw or Withdraw Weapon
KEY_I	Open Main Menu / Inventory
KEY_M	Open Map
KEY_J	Open Journal
KEY_K	Open Crafting
ESCAPE	Pause Menu / Close Menu

Table 12: **Long-horizon online tasks and the constituent short-horizon subtasks.** The online track contains 20 long-horizon tasks and 62 short-horizon subtasks. The causal tasks impose a fixed order, whereas thematic tasks allow any order. Subtasks are verified through underlying game states.

No.	Long-Horizon Task	Steps	Short-Horizon Subtasks
	Causal Tasks		
1	Craft a wooden axe.	2	(1) Chop down trees and gather wood logs outside. (2) Use the crafting table to craft a wooden axe.
2	Craft an armor stand from workshop materials.	5	(1) Craft four oak planks from an oak log. (2) Craft eight sticks from the oak planks. (3) Smelt three stone blocks into smooth stone. (4) Craft six smooth stone slabs. (5) Craft an armor stand from the sticks and a stone slab.
3	Craft an item frame from workshop materials.	3	(1) Craft eight oak planks from two oak logs. (2) Craft eight sticks from four oak planks. (3) Craft an item frame from eight sticks and one leather.
4	Retrieve shears, shear a sheep, and store the wool in the chest.	3	(1) Retrieve the shears from the chest. (2) Shear the sheep in the nearby pen. (3) Store the wool in the chest.
5	Fill the bucket with milk, bake a cake, and place it on the dining table.	3	(1) Fill the empty bucket with milk from the cow. (2) Craft a cake at the crafting table. (3) Place the cake on the marked dining table.
6	Repair the gap in the animal pen.	2	(1) Select the oak fence from the hotbar. (2) Place the oak fence to close the gap in the pen.
7	Mine the missing obsidian block, repair the ruined portal, and ignite it.	3	(1) Mine the loose obsidian block using the diamond pickaxe. (2) Place the obsidian in the missing portal-frame position. (3) Ignite the completed Nether portal with flint and steel.
8	Collect lava using an empty bucket.	2	(1) Select the bucket from the hotbar. (2) Scoop up lava from the pool using the empty bucket.
9	Mine iron, forge an iron sword, and use it to defeat the zombie.	4	(1) Mine two iron ore blocks using the stone pickaxe. (2) Smelt the iron ore into two iron ingots. (3) Craft an iron sword. (4) Kill the zombie using the iron sword.
10	Select the golden sword and kill the spider.	2	(1) Select the golden sword from the hotbar. (2) Kill the spider using the golden sword.
	Thematic Tasks		
1	Complete the target house's exterior wall using the finished house beside it as a reference.	6	(1) Place a stripped oak log in the ground-row left gap. (2) Place a stone brick block in the ground-row center gap. (3) Place a birch plank in the ground-row right gap. (4) Place a dark oak plank in the second-row left gap. (5) Place a glass block in the second-row center gap. (6) Place a spruce plank in the second-row right gap.
2	Eliminate the zombie and shear the sheep.	2	(1) Kill the zombie inside the animal pen. (2) Shear the sheep in the animal pen.
3	Store items from the inventory into the chest.	2	(1) Store the iron ingots in the chest. (2) Store the bread in the chest.
4	Assemble a mixed-material toolkit at the workshop.	3	(1) Craft an iron pickaxe. (2) Craft a wooden axe. (3) Craft a diamond sword.
5	Secure and light the tunnel to mine coal and iron.	4	(1) Fill a gap in the passage using cobblestone. (2) Place a torch in the mine. (3) Mine coal ore. (4) Mine iron ore.

Continued on the next page

Table 12 continued

No.	Long-Horizon Task	Steps	Short-Horizon Subtasks
6	Complete routine farm chores by gathering food, planting crops, and managing livestock.	5	(1) Milk the cow using an empty bucket. (2) Pick up the egg lying on the ground. (3) Plant wheat seeds on the farmland. (4) Kill the sheep using the diamond sword. (5) Kill the chicken using the iron sword.
7	Collect raw blocks for building.	2	(1) Dig dirt blocks using an iron shovel. (2) Mine stone blocks using a diamond pickaxe.
8	Process building materials and food at the shelter.	2	(1) Smelt cobblestone in the furnace. (2) Cook raw mutton in the smoker.
9	Assemble a wooden toolkit at the workshop.	3	(1) Craft a wooden pickaxe. (2) Craft a wooden axe. (3) Craft a wooden shovel.
10	Prepare the travel campsite by setting up the sleeping area, lighting the entrance, cooking food, and storing navigation equipment.	4	(1) Place the red bed on the wooden sleeping pad in the tent. (2) Place a torch on the marked block at the entrance. (3) Cook the raw cod in the smoker. (4) Store the compass in the supply chest.