Adapting Vision-Language Models for Evaluating World Models

Mariya Hendriksen
Microsoft Research*
Cambridge, United Kingdom
mariya.hendriksen@gmail.com

Tabish Rashid
Microsoft Research
London, United Kingdom

David Bignell
Microsoft Research
Cambridge, United Kingdom

Raluca Georgescu
Microsoft Research
Cambridge, United Kingdom

Abdelhak Lemkhenter
Microsoft Research
Cambridge, United Kingdom

Katja Hoffman
Microsoft Research
Cambridge, United Kingdom

Sam Devlin†
Meta
London, United Kingdom

Sarah Parisot†
Microsoft Research
Cambridge, United Kingdom

Abstract

World models – generative models that simulate environment dynamics conditioned on past observations and actions – are gaining prominence in planning, simulation, and embodied AI. However, evaluating their rollouts remains a fundamental challenge, requiring fine-grained, temporally grounded assessment of action alignment and semantic consistency – capabilities not captured by existing metrics. Vision-Language Models (VLMs) have shown promise as automatic evaluators of generative content due to their strong multimodal reasoning abilities. Yet, their use in fine-grained, temporally sensitive evaluation tasks remains limited and requires targeted adaptation. We introduce a evaluation protocol targeting two recognition tasks – action recognition and character recognition – each assessed across binary, multiple-choice, and open-ended formats. To support this, we present UNIVERSE (UNIfied Vision-language Evaluator for Rollouts in Simulated Environments), a method for adapting VLMs to rollout evaluation under data and compute constraints. We conduct a large-scale study comparing full, partial, and parameter-efficient finetuning across task formats, context lengths, sampling strategies, and data compositions. The resulting unified evaluator matches the performance of task-specific baselines using a single checkpoint. Human studies confirm strong alignment with human judgments, establishing UNIVERSE as a scalable, semantics-aware evaluator for world models.

1 Introduction

World models are generative models trained to predict future observations conditioned on past observations and actions [5, 37, 41]. They offer a powerful abstraction for learning, reasoning, and planning in complex interactive environments, and are rapidly becoming foundational in domains such as neural game engines [21, 33, 35, 56], embodied AI [31, 106], and autonomous driving [47, 77, 84]. As their capabilities grow, a persistent challenge remains: evaluation.

*Work done during an internship at Microsoft Research.
†Equal contribution.

Preprint. Under review.

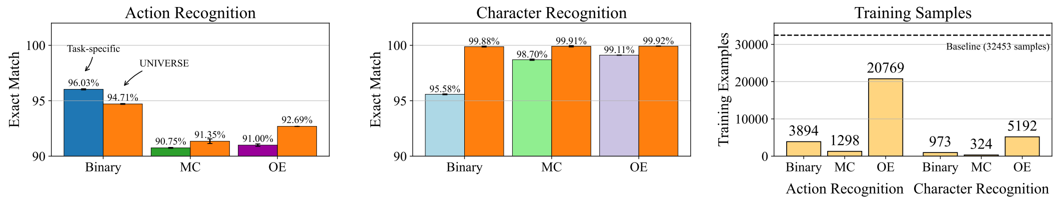

Figure 1: Performance and efficiency of UNIVERSE (orange bars throughout) compared to task-specific baselines (multiple colours). **Left and Center:** Action recognition and Character Recognition accuracy across binary, multiple-choice, and open-ended settings. **Right:** Sample efficiency – our adaptation recipe achieves strong performance with substantially fewer training samples per epoch. UNIVERSE matches the performance of task-specific baselines while using a single checkpoint, enabled by mixed supervision, efficient frame sampling, and lightweight partial fine-tuning.

Rollouts produced by world models are visually rich, temporally grounded, and semantically structured. Evaluating them requires: (i) precise alignment between generated frames and control sequences at the timestamp level [106], and (ii) consistent tracking of entities and their semantics over time [56].

Existing evaluation protocols offer limited insight in this regime. Traditional visual generation metrics are ill-suited to the structured and dynamic nature of rollouts: (i) early distributional metrics focus on static images and are sensitive to low-level variations [12, 46, 85], (ii) motion-aware metrics such as FVD [97] capture temporal patterns but lack semantic grounding, and (iii) multimodal metrics improve semantic alignment but ignore action conditioning and timestamp-level fidelity [52]. While human evaluation remains the gold standard [2, 6], it is expensive and difficult to scale. Similarly, emerging text-to-video benchmarks [50, 62, 66] focus on open-ended generation and neglect the fine-grained control and temporal coherence essential to evaluating world model outputs.

Vision-Language Models (VLMs) have demonstrated strong generalization across multimodal tasks by integrating visual and linguistic reasoning [1, 24, 26, 30, 61, 64, 72, 99], and are increasingly used to evaluate generative models [20, 59, 63, 70]. We extend this direction, hypothesizing that VLMs can serve as fine-grained evaluators of world model rollouts, capturing semantic consistency, temporal coherence, and alignment with control inputs. Virtual environments offer a favorable testbed for this purpose, as they expose full access to timestamped actions, object states, and agent behaviors, enabling controlled and systematic evaluation.

Fine-grained evaluation in this setting introduces new demands on VLMs. Unlike typical vision-language tasks, assessing world model rollouts requires precise temporal grounding, action sensitivity, and semantic tracking—often with limited supervision and constrained compute. Off-the-shelf VLMs lack domain-specific knowledge, and are not optimized for such temporally structured reasoning, particularly when text supervision is sparse (see Section 4, Zero-Shot Evaluation).

To address these limitations, we propose a structured evaluation protocol that targets two core axes of semantic fidelity: action alignment and character consistency, operationalized as two recognition tasks – Action Recognition (AR), and Character Recognition (CR) – each evaluated across prompts of varying complexity.

To support this protocol, we introduce UNIVERSE (UNIfied Vision-language Evaluator for Rollouts in Simulated Environments), a method for adapting VLMs to structured rollout evaluation. UNIVERSE emerges from a systematic study of adaptation strategies under realistic data and compute constraints. We analyze the impact of supervision regime, frame sampling strategy, visual context length, and training budget, arriving at an adaptation recipe that combines mixed supervision, efficient frame selection, and lightweight fine-tuning.

To assess the reliability of UNIVERSE, we conduct a human annotation study using rollouts generated by WHAM [56], a publicly available world model trained in a complex interactive environment. We evaluate the rollouts using UNIVERSE and collect human judgments on the same data to compare responses. We find that UNIVERSE's judgments exhibit strong alignment with human ratings, confirming its potential for automated evaluation world model rollouts, particularly when explicit ground truth is unavailable or costly to obtain.

In summary, this work makes three key contributions:

I We introduce a structured protocol for evaluating world model rollouts, based on action and character recognition tasks with increasing levels of complexity;

II We propose UNIVERSE (UNIfied Vision-language Evaluator for Rollouts in Simulated Environments), a method for adapting VLMs to fine-grained, temporally grounded evaluation tasks. UNIVERSE is the result of a comprehensive study comparing full, partial, and parameter-efficient fine-tuning strategies under realistic data and compute constraints enabled by an adaptation recipe that combines mixed supervision, frame sampling, and partial fine-tuning;

III We show that UNIVERSE matches the performance of task-specific baselines using a single checkpoint, demonstrating its ability to generalize across tasks with a unified model (Figure 1). Furthermore, its predictions align closely with human judgments, validating its effectiveness as a scalable, semantics-aware evaluator for world model rollouts.

2 Related Work

Challenges in Evaluating World Models. World models are generative systems that learn predictive representations of environment dynamics [37], originally proposed for model-based RL [92] and now central to domains such as neural game engines [21, 33, 35, 56], embodied AI [31, 106], and autonomous driving [47, 77, 84]. Recent models such as Dreamer v1–3 [38–41], MuZero [86], IRIS [74], UniSim [106], and DIAMOND [5] have improved rollout fidelity and controllability. Yet evaluation largely focuses on downstream success metrics—e.g., game score or goal completion [7–9, 36, 54]—which provide only coarse, indirect signals of rollout quality. Genie [17, 78] decouples world model learning and agent training, but its evaluation still emphasizes visual quality and control, without probing semantic or causal fidelity. Cosmos [2] proposes a structured protocol that combines FID/FVD with structure-from-motion-based 3D consistency checks and human ratings on instruction following, object permanence, and visual verity. While insightful, this approach is tied to simulator-specific infrastructure and requires costly manual comparison. Human-in-the-loop protocols such as the Video Generation Arena [6] also rely on pairwise comparison to assess rollout quality. These methods, though informative, are expensive and hard to scale.

Evaluation Metrics and Protocols for Visual Generation. Early evaluations of generative models relied on full-reference metrics such as PSNR and SSIM [100], which capture pixel-level and perceptual similarity but are sensitive to spatial misalignments and fail to reflect semantic fidelity. To address this, distributional metrics like Inception Score (IS) [85], Fréchet Inception Distance (FID) [46], and Kernel Inception Distance (KID) [12]. Other proposals such as PPL [57], Parzen likelihoods [34], and HYPE [110] attempt to quantify perceptual smoothness or human realism, but remain focused on static images. For video generation, FVD [97] generalizes FID using I3D features [18], introducing a motion-aware distributional baseline. Yet, FVD also lacks semantic grounding and does not account for causal structure or goal alignment. To improve semantic grounding, metrics based on text-image alignment have been proposed. CLIPScore [45] and CLIPSIM [102] compute similarity between generated visuals and textual or visual references using CLIP embeddings [82], while Jayasumana et al. [52] extend this to distributional comparisons via MMD. However, all operate at the frame level. Structured evaluation protocols using vision-language reasoning have also emerged. VQA Accuracy [70] uses LLMs to score answers on static image questions, and VQAScore [63] probes alignment via templated binary queries. Lee et al. [59] propose VLM evaluator to evaluate other VLMs responses given user criteria. These approaches introduce task structure but remain limited to single-frame evaluation. Recent text-to-video (T2V) benchmarks such as EvalCrafter [66], VBench [50], and DEVIL [62] introduce curated prompts and metrics covering text alignment, motion realism, and perceptual quality. While these protocols push forward evaluation of open-ended video generation, they lack timestamp-level action grounding.

Vision-Language Model Adaptation. VLMs have emerged as powerful tools for multimodal understanding, demonstrating strong performance across tasks such as captioning, retrieval, visual question answering, and instruction following [1, 24, 26, 30, 61, 64, 72, 99]. Adaptation approaches can be broadly categorized into prompt-level and weight-level methods. One prominent prompt-level adaptation techniques is prompt tuning, which injects task information directly into the input space [75, 103, 111], and in-context learning (ICL), where models such as GPT-3 [16] and Flamingo [4] condition on task demonstrations at inference time without updating parameters. Retrieval-augmented generation (RAG) [60] combines parametric models with non-parametric memory, and multimodal variants incorporate external visual or auditory context [22, 49]. While lightweight, these approaches are limited in their ability to model temporal dependencies or align with structured rollouts. Weight-level adaptation enables stronger domain alignment but incurs higher computational cost. Full finetuning remains effective yet costly, while partial finetuning [107] offers a trade-off by updating only selected layers. Parameter-efficient finetuning (PEFT) provides a scalable alternative and can be grouped into low-rank and adapter-based strategies [42]. Low-rank methods, such as LoRA [48], inject rank-constrained updates into frozen layers. Recent extensions improve upon this via weight decomposition [65], quantization-aware adaptation [27, 105], mixture-of-experts routing [104], and long-context support [25]. Adapter-based methods insert lightweight modules between frozen layers to enable modular adaptation with minimal overhead [69, 109]. A parallel line of work investigates multimodal few-shot learning. Frozen [95] was among the first to explore this setting, followed by works combining prompting and ICL for improved sample efficiency [53, 89], and works introducing a learnable meta-mapper to bridge frozen VLM components for few-shot meta-learning [76].

Our Focus. While prior efforts have explored related challenges, none directly address the evaluation of the structured, action-conditioned fidelity and semantics of world model rollouts using adapted VLM. To this end, we introduce: (i) an evaluation protocol for world model rollouts, targeting fine-grained, temporally grounded assessment of semantic fidelity; (ii) UNIVERSE, a VLM-based method to support the protocol. We validate its alignment with human judgments and demonstrate its scalability and semantic sensitivity across rollout conditions.

3 Methodology

We consider the problem of evaluating rollouts generated by *world models* in interactive environments. A world model W is trained to predict the next observation o_t given the past observations $o_{<t}$ and actions $a_{<t}$: $W : (o_{<t}, a_{<t}) \to o_t$, where $o_t \in \mathcal{O}$ represents the sensory observation at timestep t, typically an RGB image. Rollouts consist of temporally grounded sequences that reflect the causal effects of control inputs. These outputs are semantically rich and visually complex, requiring timestamp-level assessment of correctness.

To enable automatic evaluation, we propose UNIVERSE, an adapted Vision-Language Model (VLM) that serves as a structured evaluator for world model rollouts. Formally, it operates as a function: $E : (V, Q) \to \hat{A}$, where $V = (o_{t_1}, \ldots, o_{t_k}) \in \mathcal{O}^k$ is a sequence of frames from a rollout, $Q \in \mathcal{L}$ is a natural language question, and $\hat{A} \in \mathcal{L}$ is the predicted answer. Evaluation quality is measured by comparing $\hat{A}$ to the reference answer A using semantic similarity metrics.

Evaluation Protocol. We define two structured recognition tasks: (i) *Action Recognition (AR)*: Assesses whether generated sequences accurately reflect the effects of agent actions at each timestep; (ii) *Character Recognition (CR)*: Evaluates whether entities maintain consistent identity and appearance across time. Each task is framed as a visual QA problem: the evaluator receives a sequence of frames and a natural language prompt (binary, multiple-choice, or open-ended), and generates a textual response. Outputs are scored using Exact Match (EM) and ROUGE-F_1 (ROUGE), capturing both literal and semantic alignment with the reference answer. Metric details are in Appendix E.2.

Dataset Construction. Effective VLM adaptation for rollout evaluation requires a dataset that (i) captures realistic human behavior in interactive environments, and (ii) aligns with prior work in simulated settings to support comparability and reproducibility. Additionally, since WHAM – the world model used in our evaluation – is trained on *Bleeding Edge*, it is critical that the adaptation dataset match its distribution. To satisfy these constraints, we partnered with Ninja Theory and curated a dataset from both internal and public *Bleeding Edge* gameplay recordings, focusing on the *Skygarden* map used in WHAM [56]. This dataset provides high visual and behavioral diversity [79], includes a publicly available evaluation split, and is closely aligned with prior work in the domain [28, 56, 79, 88, 94], enabling cross-method comparison.

Data preparation proceeds in three stages: (i) *Preprocessing:* Segment gameplay into 14-frame clips with synchronized video, control logs, and metadata; (ii) *Description Generation:* Convert structured annotations (e.g., actions, agent states) into natural language summaries; (iii) *Question-Answer Pair Construction:* Generate six QA pairs per clip (binary, multiple-choice, and open-ended) spanning both AR and CR tasks. The final dataset contains 32.453 training clips and 8.113 validation clips, yielding 194.718 and 48.678 QA pairs, respectively. See Appendix D for details.

Model Architecture. We adapt a model from the PaliGemma family [10, 91], consisting of a vision encoder $\mathcal{M}_V$, a projection head $\mathcal{M}_P$, and a language decoder $\mathcal{M}_L$. Based on initial zero-shot evaluations (Appendix G.1), we use a single configuration for all experiments—PaliGemma 2 3b, which includes a 2B-parameter Gemma 2 decoder pretrained on 2T tokens. Input frames are resized to 224×224 and tokenized into 256 patches each. Model achitecture details are in Appendix E.1.

Each model input sequence $S = \{S_{\mathcal{I}}, S_{\mathcal{T}}^{\text{PREF}}, S_{\mathcal{T}}^{\text{SUFF}}\}$ consists of: visual tokens $S_{\mathcal{I}}$ from k frames, a textual prefix $S_{\mathcal{T}}^{\text{PREF}}$ containing the task-language cue and question, and a suffix $S_{\mathcal{T}}^{\text{SUFF}}$ with the expected answer (used only during training). This format allows the decoder to attend jointly over visual and textual context. Full prompt details are provided in Appendix E.1.

Training Objective. We optimize a causal language modeling loss on the answer suffix:

$$\mathcal{L}(S) = -\sum_{t=1}^{T_{\text{SUFF}}} \log P(s_t^{\text{SUFF}} \mid S_{<t'}) \tag{1}$$

where s_t^{SUFF} is the t-th token in the suffix, and $t' = T_{\mathcal{I}} + T_{\text{PREF}} + t$ is the token position in the flattened sequence.

Adaptation Strategies. We explore a broad design space for adapting pretrained VLMs to temporally grounded rollout evaluation. Our study spans three core dimensions: *fine-tuning configurations*, *frame sampling policy*, and *supervision composition*.

Fine-Tuning Configurations. We compare five adaptation strategies varying in parameter count and modularity: (i) *Zero-shot prompting:* No tuning; model is prompted directly. (ii) *Full fine–tuning:* All parameters $\theta = \theta_V \cup \theta_P \cup \theta_L$ are updated end-to-end. (iii) *Dual-component fine–tuning:* Two of three modules are trained (e.g., $\theta_P \cup \theta_L$). (iv) *Single-component fine-tuning:* Only one module—vision, projection, or language—is updated. (v) *Parameter-efficient fine-tuning:* We apply LoRA [48] adapters to attention and MLP layers in vision and language components: $\mathbf{W} \leftarrow \mathbf{W} + \frac{\alpha}{r}\mathbf{AB}, \quad \alpha = 8,\ r \in \{8, 16, 32, 48, 64\}$.

Frame Sampling Policy. We vary both the number of input frames and their sampling strategy. Specifically, we sweep over $k \in [1, 8]$, and evaluate two selection methods: (i) *First-n*: selecting the first k frames from each rollout; (ii) *Uniform-n*: sampling k frames uniformly across the full clip.

Supervision Composition. To support generalization across QA formats and tasks, we construct a multi-task dataset covering binary, multiple-choice, and open-ended prompts across both AR and CR. We perform a three-stage grid search to optimize the data mixture: (i) Varying AR/CR task ratios ($\alpha_{\texttt{AR}}, \alpha_{\texttt{CR}}$) while fixing QA type proportions ($\beta_{\texttt{Binary}}, \beta_{\texttt{MC}}, \beta_{\texttt{OE}}$); (ii) Tuning the proportion of open-ended supervision ($\beta_{\texttt{OE}}$) for best performance; (iii) Adjusting $\beta_{\texttt{Binary}}$ and $\beta_{\texttt{MC}}$.

UNIVERSE: UNIfied Vision-language Evaluator for Rollouts in Simulated Environments. We distill our empirical findings into UNIVERSE, a lightweight and scalable adaptation method for temporally grounded evaluation of world model rollouts using VLMs. Designed for constrained compute and limited supervision, UNIVERSE delivers strong generalization across our evaluation protocol using a single, partially tuned model. The method combines three key components:

I *Partial fine-tuning:* We update only the projection head (θ_P), training just 0.07% of model parameters. Despite this minimal footprint, it achieves the second-best performance among all strategies—trailing only vision encoder tuning, which requires $\sim$11% of parameters and incurs significantly higher compute cost.

II *Efficient frame sampling:* Each input sequence includes $k = 8$ frames sampled uniformly from a 14-frame rollout. This sparsity-aware strategy maintains long-range temporal structure while reducing token count and enabling efficient batching.

III *Mixed supervision:* We train on a hierarchical mixture of tasks and QA formats. The task distribution favors Action Recognition ($\alpha_{\texttt{AR}} = 0.8$) due to its stronger causal grounding. Within each task, we emphasize open-ended questions ($\beta_{\texttt{OE}} = 0.8$), while maintaining smaller proportions of binary ($\beta_{\texttt{binary}} = 0.15$) and multiple-choice ($\beta_{\texttt{MC}} = 0.05$) examples.

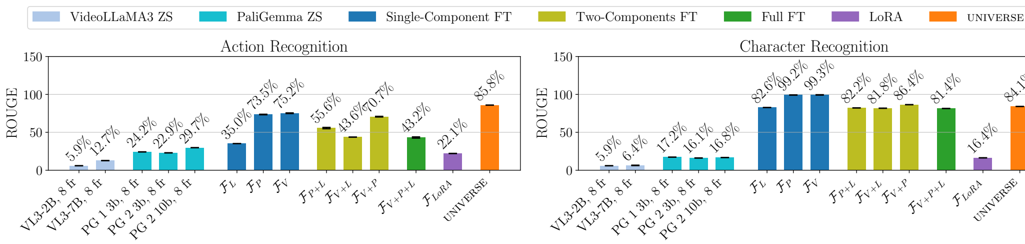

Figure 2: Comparison of UNIVERSE and baseline models on Action and Character Recognition. **Left**: UNIVERSE outperforms all baselines on AR. **Right**: On CR, it ranks third, behind models with either full vision encoder tuning or task-specific training with greater supervision. Trained under a unified protocol with minimal parameter updates (0.07%) and reduced per-task data, UNIVERSE delivers strong performance across both tasks, highlighting its efficiency and generalization.

4 Experiments

We evaluate UNIVERSE on ground truth video data, focusing on AR and CR across binary, multiple-choice, and open-ended formats. Our goals are twofold: (i) to benchmark performance against zero-shot and fine-tuned baselines, and (ii) to assess the trade-offs between adaptation strategies under constrained supervision and compute.

Baselines. We compare UNIVERSE against two classes of baselines: (i) *Zero-shot VLMs:* Seven off-the-shelf models, including VideoLLaMA3 (2B, 7B) [14] and three PaliGemma models: version 1 (3B) and version 2 (3B and 10B) [10, 91], evaluated without domain adaptation using an 8-frame visual context window.[3] (ii) *Fine-tuned PaliGemma 2:* Variants adapted via full, partial, and parameter-efficient tuning. This backbone is selected based on a sweep over PaliGemma variants, using zero-shot performance as a guide (Appendix G.1). The adaptation space includes 8 primary baselines: (i) *Single-component fine-tuning*: tuning only the vision encoder ($\mathcal{F}_{\text{V}}$), the multimodal projector ($\mathcal{F}_{\text{P}}$), or the language head ($\mathcal{F}_{\text{L}}$); (ii) *Two-component fine-tuning*: jointly tuning pairs of components—$\mathcal{F}_{\text{V+P}}$, $\mathcal{F}_{\text{V+L}}$, and $\mathcal{F}_{\text{P+L}}$; (iii) *Full-model fine-tuning*: tuning all components simultaneously ($\mathcal{F}_{\text{V+P+L}}$); (iv) *LoRA-based tuning*: Parameter-efficient adaptation with rank $r = 8$, selected after observing minimal performance variation across $r \in \{8, 16, 32, 48, 64\}$ (see Appendix G.3 for details). All models are trained using 8-frame clips and a single epoch.

Results. Figure 2 (left, center) summarizes performance across Action Recognition (AR) and Character Recognition (CR). Zero-shot models perform poorly across both tasks: VideoLLaMA3 variants fall below 12.7% on AR and 6.4% on CR; PaliGemma variants reach up to 29.7% on AR and 17.2% on CR. These results confirm that general-purpose VLMs lack the temporal grounding and domain-specific semantics required for structured rollout evaluation. In contrast, UNIVERSE achieves strong performance—outperforming all models on AR and ranking third on CR. The top two CR models either fine-tune the full vision encoder, requiring updates to $\sim$400M parameters, or fine-tune the multimodal projector with 5 $\times$ more CR supervision, each trained in isolation on a single task and prompt format. UNIVERSE, by comparison, tunes only the 2.66M-parameter projector (0.07% of the model), under a unified protocol spanning both tasks, all prompt formats, and reduced per-task supervision. Its performance under these constraints underscores the efficiency and generality of our adaptation strategy for temporally grounded evaluation.

5 Analysis

In the previous section, we observed that all models exhibit a consistent performance gap between AR and CR, underscoring the greater temporal and causal complexity of action understanding. While CR relies primarily on static visual features and identity persistence, AR requires fine-grained temporal grounding. This disparity motivates our focus on AR as the more challenging and diagnostic task. In this section, we examine how key adaptation choices influence UNIVERSE performance on AR.

[3] We also experimented with CLIPScore-based evaluation (Appendix G.2); results underperformed relative to selected baselines and were constrained to predefined candidate sets, further underscoring the need for model adaptation.

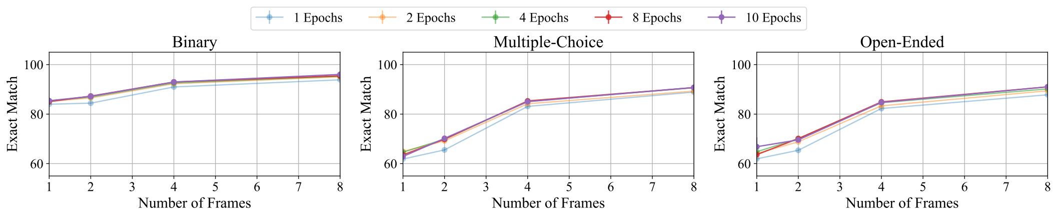

Figure 4: Action Recognition performance as a function of training supervision (epochs) and temporal context (number of frames), evaluated across all formats. Performance improves along both axes, with highest accuracy achieved when both dimensions are scaled.

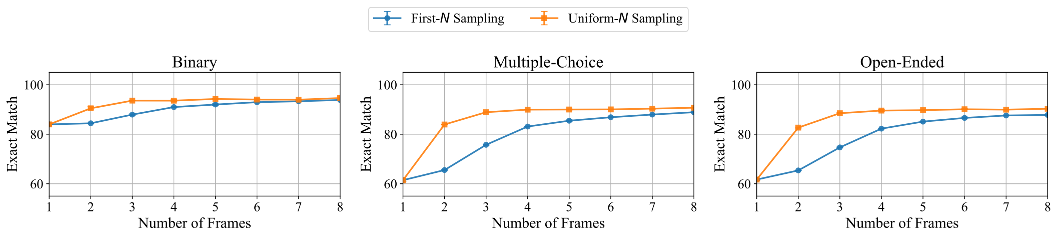

Figure 5: Effect of frame sampling strategy on Action Recognition performance across all formats. Uniform-n sampling (orange) consistently outperforms first-n (blue), with especially large gains at low frame counts, and maintains an advantage as temporal context increases.

Supervision and Temporal Context. We begin by analyzing how supervision (training budget) and temporal input (number of frames) influence UNIVERSE performance. By independently and jointly varying the number of training epochs and input frames, we disentangle the contributions of model capacity and temporal context to task success.

Results. CR converges rapidly, achieving over 97% exact match after only 12.5% of an epoch (~4K samples; Figure 3, bottom), and shows minimal improvement with further training, indicating low dependence on supervision or temporal context. In contrast, AR improves only modestly under extended training when limited to a single frame (Figure 3, top), suggesting that supervision alone is insufficient in the absence of temporal information. Motivated by this, we jointly scale both supervision and input length, varying the number of frames and epochs. As shown in Figure 4, performance on AR improves consistently across all formats, with the best results achieved under combined scaling. This insight informed the temporally diverse sampling strategy used in training UNIVERSE.

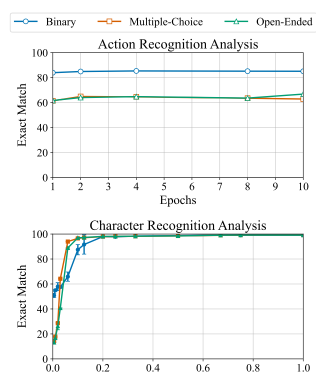

Figure 3: Exact Match accuracy over training epochs for Action Recognition (AR, top) and Character Recognition (CR, bottom). AR improves gradually with supervision, while CR converges to high accuracy with minimal training.

Temporal Sampling Strategies. Following the observation that AR requires both extended supervision and temporally rich input, we examine how frame selection impacts performance. We compare first-n sampling, which selects the first n consecutive frames from each rollout, to uniform-n sampling, which draws n evenly spaced frames across the entire sequence. We conduct experiments at varying context lengths, using $n \in \{1, 2, \dots, 8\}$ frames. to evaluate the impact of both sampling method and input horizon.

Results. As shown in Figure 5, uniform-n consistently outperforms first-n across all evaluation formats. The effect is most pronounced at low frame counts. With only 2 input frames, uniform sampling improves exact match accuracy from 84.42% to 90.47% in Binary, from 65.53% to 83.93% in Multiple-Choice, and from 65.38% to 82.68% in Open-Ended formats. Gains persist even at 8 frames, where uniform sampling maintains an advantage across formats.

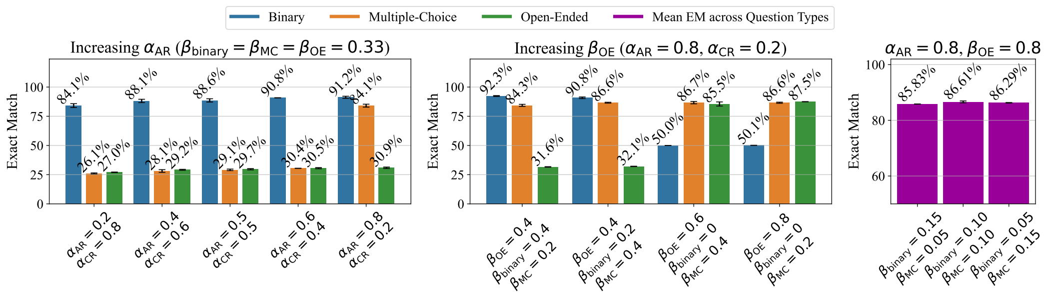

Figure 6: Hierarchical ablation of training data composition for UNIVERSE. **Left**: Varying task-level ratio α (AR vs. CR) with uniform format distribution ($\beta = 1/3$) shows that increasing $\alpha_{\texttt{AR}}$ improves AR performance, especially for multiple-choice, while open-ended remains flat. **Center**: Sweeping format-level ratio $\beta_{\texttt{OE}}$ with fixed $\alpha_{\texttt{AR}} = 0.8$ reveals that oversampling open-ended data ($\beta_{\texttt{OE}} = 0.8$) improves AR-OE performance. **Right**: Fine-tuning binary and MC proportions under $\beta_{\texttt{OE}} = 0.8$ shows performance is stable across mixes, with slight gains from $\beta_{\texttt{binary}} = 0.15$, $\beta_{\texttt{MC}} = 0.05$.

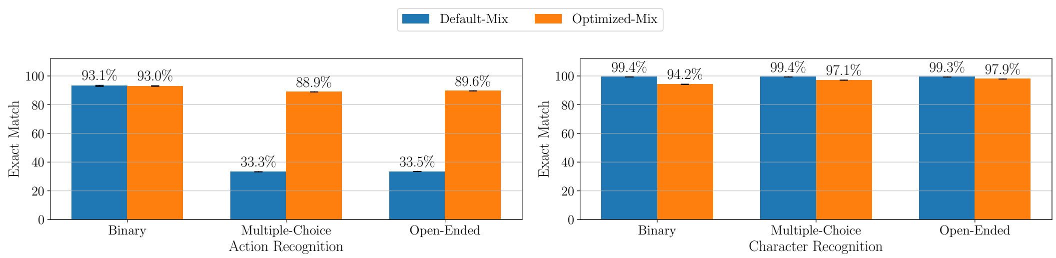

Figure 7: Comparison of two training regimes: a default data mix (equal task and format proportions) and the optimized mix derived from hierarchical tuning. The optimized configuration yields substantial gains on AR, while maintaining strong CR performance.

Optimizing Data Mix for Unified Multi-Task Evaluation. We analyze how training data composition affects multi-task performance in UNIVERSE, with the goal of enabling a single model to generalize across AR and CR. Specifically, we study how the task-level ratio α (AR vs. CR) and format-level ratio β (binary, multiple-choice, open-ended) influence performance across evaluation settings. We first conduct a hierarchical ablation to identify an optimized data mixture, then assess its impact by comparing against a default task mix with uniform sampling.

Data Mix Optimization. To determine an effective training mixture for UNIVERSE, we perform a hierarchical ablation over task-level and format-level data ratios. We begin by varying the task-level proportion α (AR vs. CR), holding the format distribution fixed at $\beta_{\texttt{binary}} = \beta_{\texttt{MC}} = \beta_{\texttt{OE}} = 1/3$. As shown in Figure 6 (left), increasing $\alpha_{\texttt{AR}}$ improves AR performance—especially for multiple-choice—while CR remains stable, with a favorable tradeoff reached at $\alpha_{\texttt{AR}} = 0.8$. However, open-ended accuracy shows little change, motivating format-specific rebalancing. Fixing $\alpha_{\texttt{AR}} = 0.8$, we sweep the format ratio $\beta_{\texttt{OE}}$, and observe in Figure 6 (center) that AR-OE accuracy improves substantially with increased open-ended coverage, peaking at $\beta_{\texttt{OE}} = 0.8$, albeit at the cost of binary performance. To restore balance, we fix $\beta_{\texttt{OE}} = 0.8$ and allocate the remaining budget across binary and multiple-choice formats. As shown in Figure 6 (right), performance remains robust across configurations, with a slight preference for $\beta_{\texttt{binary}} = 0.15$ and $\beta_{\texttt{MC}} = 0.05$. Based on these findings, we adopt the following optimized data composition: $\alpha_{\texttt{AR}} = 0.8$, $\alpha_{\texttt{CR}} = 0.2$; $\beta_{\texttt{binary}} = 0.15$, $\beta_{\texttt{MC}} = 0.05$, and $\beta_{\texttt{OE}} = 0.8$.

Effectiveness of the Optimized Mix. Having identified an optimized training mixture through hierarchical ablation, we now evaluate its impact in practice. We compare the final UNIVERSE model—trained with this optimized mix—to a baseline trained with a default task and format distribution. We train both models on 4 epochs. As shown in Figure 7, the optimized configuration yields substantial gains on AR, particularly for multiple-choice and open-ended formats, while maintaining competitive

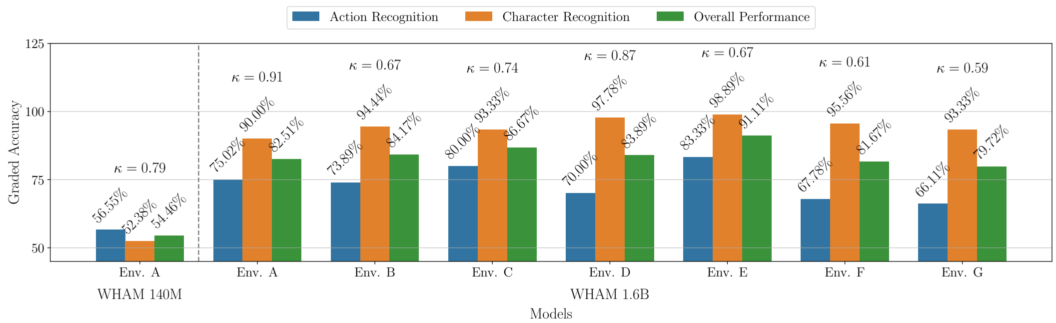

Figure 8: Graded accuracy of UNIVERSE across rollouts from WHAM-140M (Env. A) and WHAM-1.6B (Envs. A–G). Performance improves markedly with higher-fidelity rollouts from WHAM-1.6B, even in out-of-domain settings (B–G). Cohen's κ (above bars) reflects inter-rater agreement.

performance on CR. These results underscore the importance of data composition in enabling robust multi-task learning within a unified evaluator.

6 Evaluating World Model Rollouts with UNIVERSE

We evaluate UNIVERSE as an automated evaluator of world model rollouts using the WHAM benchmark [56], which provides pretrained world models and an evaluation set. Our analysis focuses on two axes: (i) *in-domain accuracy*, measured on Skygarden—the environment used during fine-tuning—and (ii) *generalization* to six unseen environments.

We compare UNIVERSE's predictions on samples from two world models: (i) *WHAM-140M*, trained on Skygarden with lower-quality rollouts (128×128 resolution), and (ii) *WHAM-1.6B*, trained on a diverse environment suite with higher-resolution output (300×180). We prepare 30 rollouts for each model-environment pair, yielding a total of 240 rollouts. Each rollout is segmented into 14-frame clips and paired with six natural language questions, following our evaluation protocol. UNIVERSE answers each question using majority voting over five greedy decoding samples. Human annotators then rate each response on a four-point ordinal scale: *Correct*, *Partially Correct*, *Incorrect*, and *Unclear*. Each response is independently rated by two annotators; in cases of disagreement, a third annotator serves as adjudicator. Inter-annotator agreement is quantified using Cohen's κ. Full details of the annotation protocol are provided in Appendix F.

Results. Figure 8 summarizes graded accuracy across models and environments. We observe a significant performance gap between rollouts from WHAM-140M and WHAM-1.6B. Despite being in-domain, WHAM-140M yields lower evaluation accuracy likley due to a resolution mismatch: its 128×128 frames are upsampled to 224×224 for UNIVERSE input. In contrast, WHAM-1.6B produces higher-quality inputs, enabling more reliable evaluation. On WHAM-1.6B Skygarden rollouts, UNIVERSE achieves 75.02% graded accuracy on AR and 90.00% on CR. When applied to six previously unseen environments, AR accuracy remains relatively stable with the lowest accuracy of 66.11% for environment G, and highest accuracy of 83.33% for environment E. Overall, the results indicate strong generalization, especially for identity-focused recognition tasks. Cohen's κ scores further reflect the interpretability of UNIVERSE's outputs: agreement is substantial overall ($\kappa = 0.73$), with the lowest in Environment G ($\kappa = 0.59$) and highest in Environment A on WHAM-1.6B rollouts ($\kappa = 0.91$). These results highlight that UNIVERSE remains aligned with human judgments.

7 Conclusion

In this paper, we investigate the use of Vision-Language Models (VLMs) as automated evaluators for world model rollouts, addressing the fundamental challenge of fine-grained, temporally grounded evaluation. We introduce a structured evaluation protocol centered on action and character recognition tasks across binary, multiple-choice, and open-ended formats. To support this, we propose UNIVERSE, a unified method for adapting VLMs to this setting through mixed supervision, efficient frame sampling, and lightweight fine-tuning. Our large-scale study demonstrates that UNIVERSE matches the performance of task-specific baselines using a single checkpoint and aligns closely with human judgments, establishing it as a scalable, semantics-aware evaluator for evaluating world models, particularly when explicit ground truth is unavailable or costly to obtain.

Limitations While UNIVERSE performs well in simulation, its generalization beyond simulated environments remains an open challenge. The evaluation protocol targets two fidelity axes, which, while comprehensive, omit higher-order reasoning over goals, causality, and multi-agent dynamics. Our experiments focus on short to medium context lengths; scaling to long-horizon rollouts remains an open challenge, especially under limited supervision. Although compute-efficient, training could be further improved with adaptive curricula or progressive tuning. Finally, like all pretrained VLMs, UNIVERSE may reflect dataset biases and underperform on rare or ambiguous behaviors.

References

[1] Marah I. Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat S. Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Caio César Teodoro Mendes, Weizhu Chen, Vishrav Chaudhary, Parul Chopra, Allie Del Giorno, Gustavo de Rosa, Matthew Dixon, Ronen Eldan, Dan Iter, Amit Garg, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Jamie Huynh, Mojan Javaheripi, Xin Jin, Piero Kauffmann, Nikos Karampatziakis, Dongwoo Kim, Mahoud Khademi, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Chen Liang, Weishung Liu, Eric Lin, Zeqi Lin, Piyush Madan, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Xia Song, Masahiro Tanaka, Xin Wang, Rachel Ward, Guanhua Wang, Philipp Witte, Michael Wyatt, Can Xu, Jiahang Xu, Sonali Yadav, Fan Yang, Ziyi Yang, Donghan Yu, Chengruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.

[2] Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, Daniel Dworakowski, Jiaojiao Fan, Michele Fenzi, Francesco Ferroni, Sanja Fidler, Dieter Fox, Songwei Ge, Yunhao Ge, Jinwei Gu, Siddharth Gururani, Ethan He, Jiahui Huang, Jacob Samuel Huffman, Pooya Jannaty, Jingyi Jin, Seung Wook Kim, Gergely Klár, Grace Lam, Shiyi Lan, Laura Leal-Taixé, Anqi Li, Zhaoshuo Li, Chen-Hsuan Lin, Tsung-Yi Lin, Huan Ling, Ming-Yu Liu, Xian Liu, Alice Luo, Qianli Ma, Hanzi Mao, Kaichun Mo, Arsalan Mousavian, Seungjun Nah, Sriharsha Niverty, David Page, Despoina Paschalidou, Zeeshan Patel, Lindsey Pavao, Morteza Ramezanali, Fitsum Reda, Xiaowei Ren, Vasanth Rao Naik Sabavat, Ed Schmerling, Stella Shi, Bartosz Stefaniak, Shitao Tang, Lyne Tchapmi, Przemek Tredak, Wei-Cheng Tseng, Jibin Varghese, Hao Wang, Haoxiang Wang, Heng Wang, Ting-Chun Wang, Fangyin Wei, Xinyue Wei, Jay Zhangjie Wu, Jiashu Xu, Wei Yang, Lin Yen-Chen, Xiaohui Zeng, Yu Zeng, Jing Zhang, Qinsheng Zhang, Yuxuan Zhang, Qingqing Zhao, and Artur Zólkowski. Cosmos world foundation model platform for physical AI. *arXiv preprint arXiv:2501.03575*, 2025.

[3] Ibrahim M Alabdulmohsin, Xiaohua Zhai, Alexander Kolesnikov, and Lucas Beyer. Getting vit in shape: Scaling laws for compute-optimal model design. *Advances in Neural Information Processing Systems*, 36:16406–16425, 2023.

[4] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L. Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

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

[6] Artificial Analysis. Video generation arena leaderboard, 2024. URL `https://huggingface.co/spaces/ArtificialAnalysis/Video-Generation-Arena-Leaderboard`. Accessed: 24 March 2025.

[7] 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. *Advances in Neural Information Processing Systems*, 35: 24639–24654, 2022.

[8] Charles Beattie, Joel Z. Leibo, Denis Teplyashin, Tom Ward, Marcus Wainwright, Heinrich Küttler, Andrew Lefrancq, Simon Green, Víctor Valdés, Amir Sadik, Julian Schrittwieser, Keith Anderson, Sarah York, Max Cant, Adam Cain, Adrian Bolton, Stephen Gaffney, Helen King, Demis Hassabis, Shane Legg, and Stig Petersen. Deepmind lab. *arXiv preprint arXiv:1612.03801*, 2016.

[9] Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of artificial intelligence research*, 47:253–279, 2013.

[10] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey A. Gritsenko, Neil Houlsby, Manoj Kumar, Keran Rong, Julian Eisenschlos, Rishabh Kabra, Matthias Bauer, Matko Bosnjak, Xi Chen, Matthias Minderer, Paul Voigtlaender, Ioana Bica, Ivana Balazevic, Joan Puigcerver, Pinelopi Papalampidi, Olivier J. Hénaff, Xi Xiong, Radu Soricut, Jeremiah Harmsen, and Xiaohua Zhai. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024.

[11] Lukas Biewald. Experiment tracking with weights and biases, 2020. URL `https://www.wandb.com/`. Software available from wandb.com.

[12] Mikolaj Binkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD gans. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

[13] Steven Bird and Edward Loper. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[14] Zesen Cheng Zhiqiang Hu Yuqian Yuan Guanzheng Chen Sicong Leng Yuming Jiang Hang Zhang Xin Li Peng Jin Wenqi Zhang Fan Wang Lidong Bing Deli Zhao Boqiang Zhang, Kehan Li. Videollama 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*, 2025. URL `https://arxiv.org/abs/2501.13106`.

[15] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[16] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[17] Jake Bruce, Michael D. Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Bechtle, Feryal M. P. Behbahani, Stephanie C. Y. Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott E. Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando de Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

[18] João Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 4724–4733. IEEE Computer Society, 2017.

[19] Soravit Changpinyo, Doron Kukliansky, Idan Szpektor, Xi Chen, Nan Ding, and Radu Soricut. All you may need for VQA are image captions. *arXiv preprint arXiv:2205.01883*, 2022.

[20] Dongping Chen, Ruoxi Chen, Shilin Zhang, Yaochen Wang, Yinuo Liu, Huichi Zhou, Qihui Zhang, Yao Wan, Pan Zhou, and Lichao Sun. Mllm-as-a-judge: Assessing multimodal llm-as-a-judge with vision-language benchmark. In *Forty-first International Conference on Machine Learning*, 2024.

[21] Jingye Chen, Yuzhong Zhao, Yupan Huang, Lei Cui, Li Dong, Tengchao Lv, Qifeng Chen, and Furu Wei. Model as a game: On numerical and spatial consistency for generative games. *arXiv preprint arXiv:2503.21172*, 2025.

[22] Wenhu Chen, Hexiang Hu, Xi Chen, Pat Verga, and William Cohen. Murag: Multimodal retrieval-augmented generator for open question answering over images and text. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5558–5570, 2022.

[23] Xi Chen, Xiao Wang, Soravit Changpinyo, A. J. Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Nan Ding, Keran Rong, Hassan Akbari, Gaurav Mishra, Linting Xue, Ashish V. Thapliyal, James Bradbury, and Weicheng Kuo. Pali: A jointly-scaled multilingual language-image model. *arXiv preprint arXiv:2209.06794*, 2022.

[24] Xi Chen, Josip Djolonga, Piotr Padlewski, Basil Mustafa, Soravit Changpinyo, Jialin Wu, Carlos Riquelme Ruiz, Sebastian Goodman, Xiao Wang, Yi Tay, Siamak Shakeri, Mostafa Dehghani, Daniel Salz, Mario Lucic, Michael Tschannen, Arsha Nagrani, Hexiang Hu, Mandar Joshi, Bo Pang, Ceslee Montgomery, Paulina Pietrzyk, Marvin Ritter, A. J. Piergiovanni, Matthias Minderer, Filip Pavetic, Austin Waters, Gang Li, Ibrahim Alabdulmohsin, Lucas Beyer, Julien Amelot, Kenton Lee, Andreas Peter Steiner, Yang Li, Daniel Keysers, Anurag Arnab, Yuanzhong Xu, Keran Rong, Alexander Kolesnikov, Mojtaba Seyedhosseini, Anelia Angelova, Xiaohua Zhai, Neil Houlsby, and Radu Soricut. Pali-x: On scaling up a multilingual vision and language model. *arXiv preprint arXiv:2305.18565*, 2023.

[25] Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. Longlora: Efficient fine-tuning of long-context large language models. In *The Twelfth International Conference on Learning Representations*.

[26] Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, Jiasen Lu, Taira Anderson, Erin Bransom, Kiana Ehsani, Huong Ngo, Yen-Sung Chen, Ajay Patel, Mark Yatskar, Chris Callison-Burch, Andrew Head, Rose Hendrix, Favyen Bastani, Eli VanderBilt, Nathan Lambert, Yvonne Chou, Arnavi Chheda, Jenna Sparks, Sam Skjonsberg, Michael Schmitz, Aaron Sarnat, Byron Bischoff, Pete Walsh, Chris Newell, Piper Wolters, Tanmay Gupta, Kuo-Hao Zeng, Jon Borchardt, Dirk Groeneveld, Jen Dumas, Crystal Nam, Sophie Lebrecht, Caitlin Wittlif, Carissa Schoenick, Oscar Michel, Ranjay Krishna, Luca Weihs, Noah A. Smith, Hannaneh Hajishirzi, Ross B. Girshick, Ali Farhadi, and Aniruddha Kembhavi. Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models. *arXiv preprint arXiv:2409.17146*, 2024.

[27] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115, 2023.

[28] Sam Devlin, Raluca Georgescu, Ida Momennejad, Jaroslaw Rzepecki, Evelyn Zuniga, Gavin Costello, Guy Leroy, Ali Shaw, and Katja Hofmann. Navigation turing test (ntt): Learning to evaluate human-like navigation. In *International Conference on Machine Learning*, pages 2644–2653. PMLR, 2021.

[29] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021.

[30] Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence.

PaLM-E: An embodied multimodal language model. In *International Conference on Machine Learning*, pages 8469–8488. PMLR, 2023.

[31] Yilun Du, Sherry Yang, Pete Florence, Fei Xia, Ayzaan Wahid, Brian Ichter, Pierre Sermanet, Tianhe Yu, Pieter Abbeel, Joshua B. Tenenbaum, Leslie Pack Kaelbling, Andy Zeng, and Jonathan Tompson. Video language planning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*.

[32] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.

[33] Shenyuan Gao, Siyuan Zhou, Yilun Du, Jun Zhang, and Chuang Gan. Adaworld: Learning adaptable world models with latent actions. *arXiv preprint arXiv:2503.18938*, 2025.

[34] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2672–2680, 2014.

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

[36] William H Guss, Mario Ynocente Castro, Sam Devlin, Brandon Houghton, Noboru Sean Kuno, Crissman Loomis, Stephanie Milani, Sharada Mohanty, Keisuke Nakata, Ruslan Salakhutdinov, et al. The minerl 2020 competition on sample efficient reinforcement learning using human priors. *arXiv preprint arXiv:2101.11071*, 2021.

[37] David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. *Advances in Neural Information Processing Systems*, 31:2451–2463, 2018.

[38] Danijar Hafner, Timothy P Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *International Conference on Learning Representations*.

[39] Danijar Hafner, Timothy P. Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.

[40] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

[41] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, pages 1–7, 2025.

[42] Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey. *Transactions on Machine Learning Research*, 2024.

[43] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[44] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[45] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7514–7528, 2021.

[46] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing*

Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, pages 6626–6637, 2017.

[47] Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. Gaia-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023.

[48] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 2022.

[49] Ziniu Hu, Ahmet Iscen, Chen Sun, Zirui Wang, Kai-Wei Chang, Yizhou Sun, Cordelia Schmid, David A Ross, and Alireza Fathi. Reveal: Retrieval-augmented visual-language pre-training with multi-source multimodal knowledge memory. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 23369–23379, 2023.

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

[51] John D Hunter. Matplotlib: A 2d graphics environment. *Computing in science & engineering*, 9(03):90–95, 2007.

[52] Sadeep Jayasumana, Srikumar Ramalingam, Andreas Veit, Daniel Glasner, Ayan Chakrabarti, and Sanjiv Kumar. Rethinking FID: Towards a better evaluation metric for image generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 9307–9315. IEEE, 2024.

[53] Woojeong Jin, Yu Cheng, Yelong Shen, Weizhu Chen, and Xiang Ren. A good prompt is worth millions of parameters: Low-resource prompt-based learning for vision-language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2763–2775, 2022.

[54] Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H. Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, Afroz Mohiuddin, Ryan Sepassi, George Tucker, and Henryk Michalewski. Model based reinforcement learning for atari. *ICLR*, 1:2, 2020.

[55] Anssi Kanervisto, Christian Scheller, and Ville Hautamäki. Action space shaping in deep reinforcement learning. In *2020 IEEE conference on games (CoG)*, pages 479–486. IEEE, 2020.

[56] Anssi Kanervisto, David Bignell, Linda Yilin Wen, Martin Grayson, Raluca Georgescu, Sergio Valcarcel Macua, Shan Zheng Tan, Tabish Rashid, Tim Pearce, Yuhan Cao, Abdelhak Lemkhenter, Chentian Jiang, Gavin Costello, Gunshi Gupta, Marko Tot, Shu Ishida, Tarun Gupta, Udit Arora, Ryen W. White, Sam Devlin, Cecily Morrison, and Katja Hofmann. World and human action models towards gameplay ideation. *Nature*, 638(8051):656–663, 2025.

[57] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 4401–4410. Computer Vision Foundation / IEEE, 2019.

[58] Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Eduardo Blanco and Wei Lu, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, pages 66–71. Association for Computational Linguistics, 2018.

[59] Seongyun Lee, Seungone Kim, Sue Park, Geewook Kim, and Minjoon Seo. Prometheus-vision: Vision-language model as a judge for fine-grained evaluation. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 11286–11315, 2024.

[60] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020.

[61] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.

[62] Mingxiang Liao, Qixiang Ye, Wangmeng Zuo, Fang Wan, Tianyu Wang, Yuzhong Zhao, Jingdong Wang, Xinyu Zhang, et al. Evaluation of text-to-video generation models: A dynamics perspective. *Advances in Neural Information Processing Systems*, 37:109790–109816, 2024.

[63] Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. Evaluating text-to-visual generation with image-to-text generation. In *European Conference on Computer Vision*, pages 366–384. Springer, 2024.

[64] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306, 2024.

[65] Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*.

[66] Yaofang Liu, Xiaodong Cun, Xuebo Liu, Xintao Wang, Yong Zhang, Haoxin Chen, Yang Liu, Tieyong Zeng, Raymond Chan, and Ying Shan. Evalcrafter: Benchmarking and evaluating large video generation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22139–22149, 2024.

[67] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

[68] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2022.

[69] Gen Luo, Yiyi Zhou, Tianhe Ren, Shengxin Chen, Xiaoshuai Sun, and Rongrong Ji. Cheap and quick: Efficient vision-language instruction tuning for large language models. *Advances in neural information processing systems (NeurIPS)*, 2023.

[70] Oscar Mañas, Benno Krojer, and Aishwarya Agrawal. Improving automatic vqa evaluation using large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 4171–4179, 2024.

[71] Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. Peft: State-of-the-art parameter-efficient fine-tuning methods, 2022. URL `https://github.com/huggingface/peft`.

[72] Brandon McKinzie, Zhe Gan, Jean-Philippe Fauconnier, Sam Dodge, Bowen Zhang, Philipp Dufter, Dhruti Shah, Xianzhi Du, Futang Peng, Floris Weers, Anton Belyi, Haotian Zhang, Karanjeet Singh, Doug Kang, Hongyu Hè, Max Schwarzer, Tom Gunter, Xiang Kong, Aonan Zhang, Jianyu Wang, Chong Wang, Nan Du, Tao Lei, Sam Wiseman, Mark Lee, Zirui Wang, Ruoming Pang, Peter Grasch, Alexander Toshev, and Yinfei Yang. MM1: Methods, analysis and insights from multimodal LLM pre-training. In *European Conference on Computer Vision*, pages 304–323. Springer, 2024.

[73] Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Cristian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, and et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

[74] Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. In *The Eleventh International Conference on Learning Representations*, 2023.

[75] Atsuyuki Miyai, Qing Yu, Go Irie, and Kiyoharu Aizawa. Locoop: Few-shot out-of-distribution detection via prompt learning. *Advances in Neural Information Processing Systems*, 36:76298–76310, 2023.

[76] Ivona Najdenkoska, Xiantong Zhen, and Marcel Worring. Meta learning to bridge vision and language models for multimodal few-shot learning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, 2023.

[77] Jingcheng Ni, Yuxin Guo, Yichen Liu, Rui Chen, Lewei Lu, and Zehuan Wu. Maskgwm: A generalizable driving world model with video mask reconstruction. *arXiv preprint arXiv:2502.11663*, 2025.

[78] Jack Parker-Holder, Philip Ball, Jake Bruce, Vibhavari Dasagi, Kristian Holsheimer, Christos Kaplanis, Alexandre Moufarek, Guy Scully, Jeremy Shar, Jimmy Shi, Stephen Spencer, Jessica Yung, Michael Dennis, Sultan Kenjeyev, Shangbang Long, Vlad Mnih, Harris Chan, Maxime Gazeau, Bonnie Li, Fabio Pardo, Luyu Wang, Lei Zhang, Frederic Besse, Tim Harley, Anna Mitenkova, Jane Wang, Jeff Clune, Demis Hassabis, Raia Hadsell, Adrian Bolton, Satinder Singh, and Tim Rocktäschel. Genie 2: A large-scale foundation world model, 2024. URL `https://deepmind.google/discover/blog/genie-2-a-large-scale-foundation-world-model/`. Accessed: 20 March 2025.

[79] Tim Pearce, Tabish Rashid, Dave Bignell, Raluca Georgescu, Sam Devlin, and Katja Hofmann. Scaling laws for pre-training agents and world models. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025.

[80] AJ Piergiovanni, Weicheng Kuo, and Anelia Angelova. Pre-training image-language transformers for open-vocabulary tasks. *arXiv preprint arXiv:2209.04372*, 2022.

[81] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[82] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *Proceedings of the International Conference on Machine Learning (ICML)*, 2021.

[83] Morgane Rivière, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozinska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucinska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju-yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjösund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, and Lilly McNealus. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

[84] Lloyd Russell, Anthony Hu, Lorenzo Bertoni, George Fedoseev, Jamie Shotton, Elahe Arani, and Gianluca Corrado. Gaia-2: A controllable multi-view generative world model for autonomous driving. *arXiv preprint arXiv:2503.20523*, 2025.

[85] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 2226–2234, 2016.

[86] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy P. Lillicrap, and David Silver. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

[87] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, 2018.

[88] Sugandha Sharma, Guy Davidson, Khimya Khetarpal, Anssi Kanervisto, Udit Arora, Katja Hofmann, and Ida Momennejad. Toward human-ai alignment in large-scale multi-player games. In *ACL 2024 Worldplay Workshop*, 2024.

[89] Haoyu Song, Li Dong, Weinan Zhang, Ting Liu, and Furu Wei. Clip models are few-shot learners: Empirical studies on vqa and visual entailment. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6088–6100, 2022.

[90] Krishna Srinivasan, Karthik Raman, Jiecao Chen, Michael Bendersky, and Marc Najork. Wit: Wikipedia-based image text dataset for multimodal multilingual machine learning. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 2443–2449, 2021.

[91] Andreas Steiner, André Susano Pinto, Michael Tschannen, Daniel Keysers, Xiao Wang, Yonatan Bitton, Alexey A. Gritsenko, Matthias Minderer, Anthony Sherbondy, Shangbang Long, Siyang Qin, R. Reeve Ingle, Emanuele Bugliarello, Sahar Kazemzadeh, Thomas Mesnard, Ibrahim Alabdulmohsin, Lucas Beyer, and Xiaohua Zhai. Paligemma 2: A family of versatile vlms for transfer. *arXiv preprint arXiv:2412.03555*, 2024.

[92] Richard S Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM Sigart Bulletin*, 2(4):160–163, 1991.

[93] Suramya Tomar. Converting video formats with ffmpeg. *Linux Journal*, 2006(146):10, 2006.

[94] Marko Tot, Shu Ishida, Abdelhak Lemkhenter, David Bignell, Pallavi Choudhury, Chris Lovett, Luis França, Matheus Ribeiro Furtado de Mendonça, Tarun Gupta, Darren Gehring, et al. Adapting a world model for trajectory following in a 3d game. In *ICLR 2025 Workshop on World Models*, 2025.

[95] Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. Multimodal few-shot learning with frozen language models. *Advances in Neural Information Processing Systems*, 34:200–212, 2021.

[96] P Umesh. Image processing in python. *CSI Communications*, 23, 2012.

[97] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric and challenges. *arXiv preprint arXiv:1812.01717*, 2018.

[98] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[99] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[100] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4): 600–612, 2004.

[101] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

[102] Chenfei Wu, Lun Huang, Qianxi Zhang, Binyang Li, Lei Ji, Fan Yang, Guillermo Sapiro, and Nan Duan. Godiva: Generating open-domain videos from natural descriptions. *arXiv preprint arXiv:2104.14806*, 2021.

[103] Mingrui Wu, Xinyue Cai, Jiayi Ji, Jiale Li, Oucheng Huang, Gen Luo, Hao Fei, Guannan Jiang, Xiaoshuai Sun, and Rongrong Ji. Controlmllm: Training-free visual prompt learning for multimodal large language models. *Advances in Neural Information Processing Systems*, 37:45206–45234, 2024.

[104] Xun Wu, Shaohan Huang, and Furu Wei. Mixture of lora experts. In *ICLR*, 2024.

[105] Yuhui Xu, Lingxi Xie, Xiaotao Gu, Xin Chen, Heng Chang, Hengheng Zhang, Zhengsu Chen, Xiaopeng Zhang, and Qi Tian. Qa-lora: Quantization-aware low-rank adaptation of large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*, 2024.

[106] Sherry Yang, Yilun Du, Seyed Kamyar Seyed Ghasemipour, Jonathan Tompson, Leslie Pack Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*, 2024.

[107] Peng Ye, Yongqi Huang, Chongjun Tu, Minglei Li, Tao Chen, Tong He, and Wanli Ouyang. Partial fine-tuning: A successor to full fine-tuning for vision transformers. *arXiv preprint arXiv:2312.15681*, 2023.

[108] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023.

[109] Bingchen Zhao, Haoqin Tu, Chen Wei, Jieru Mei, and Cihang Xie. Tuning layernorm in attention: Towards efficient multi-modal llm finetuning. In *ICLR*, 2024.

[110] Sharon Zhou, Mitchell L. Gordon, Ranjay Krishna, Austin Narcomey, Li Fei-Fei, and Michael S. Bernstein. HYPE: A benchmark for human eye perceptual evaluation of generative models. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3444–3456, 2019.

[111] Yiwei Zhou, Xiaobo Xia, Zhiwei Lin, Bo Han, and Tongliang Liu. Few-shot adversarial prompt learning on vision-language models. *Advances in Neural Information Processing Systems*, 37:3122–3156, 2024.

ADAPTING VISION-LANGUAGE MODELS FOR EVALUATING WORLD MODELS

SUPPLEMENTARY MATERIALS

TABLE OF CONTENTS

A Broader Impact

As world models become integral to simulation, planning, and decision-making in interactive environments, evaluation remains a key bottleneck for both research progress and safe deployment. We address this challenge by introducing a unified, sample-efficient framework for evaluating world model rollouts using adapted VLMs, designed for fine-grained, temporally grounded, and semantically coherent assessment.

This capability has direct implications for high-impact domains such as neural game engines [21, 33, 35, 56], embodied AI [31, 106], and autonomous driving [47, 77, 84], where world models simulate environment dynamics and support downstream control and generalization. In such contexts, precise and interpretable evaluation is critical not only for benchmarking, but also for diagnosing failure modes and ensuring alignment with intended behaviors.

By reducing dependence on human annotation and task-specific fine-tuning, UNIVERSE offers a scalable alternative that lowers the computational and environmental costs of rollout evaluation. However, reliance on automated evaluators introduces risks: adapted VLMs may inherit biases from pretraining, struggle under distributional shift, or yield unreliable judgments in edge cases. These risks are amplified in safety-critical settings, where miscalibrated evaluations can propagate downstream errors.

We therefore advocate for cautious deployment, accompanied by human oversight, rigorous validation, and transparent reporting. While UNIVERSE advances the automation of world model evaluation, it must be situated within evaluation pipelines that foreground robustness, interpretability, and accountability.

B Reproducibility Statement

To support reproducibility and facilitate future research, we provide detailed instructions for reproducing all main experiments. Detailed descriptions of model architectures, training procedures, and dataset construction are provided in Section 4 and Appendix E. A high-level overview of the overall implementation framework is included in Appendix C. All experiments have been repeated for three runs. Plots and tables with quantitative results show the standard deviation across these runs.

Use of Existing Assets. We experiment with a range of open-weight VLMs, including three PaliGemma variants (version 1 (3B) [10] and version 2 (3B and 10B) [91]), VideoLLaMA3 (2B, 7B) [14], and CLIP [82] with the following vision encoder configurations: ViT-B/32, ViT-B/16, ViT-L/14, and ViT-L/14 with 336×336 resolution. UNIVERSE is built on top of PaliGemma v2 (3B), using publicly released checkpoints for initialization. Further architectural and implementation details are provided in Appendix E.1. For our software stack, we use Matplotlib [51] for plotting, NumPy [43] for data handling, openCV [15], FFmpeg [93] and PIL [96] for video and image processing, and NLTK [13] for text processing. Parameter-efficient fine-tuning is implemented using the PEFT library [71]. We log our experiments using Weights and Biases [11].

Compute Resources. All experiments were conducted using NVIDIA A100 GPUs (40GB memory) on an internal compute cluster. Each model was trained and/or evaluated using 8 GPUs. The compute breakdown is as follows: zero-shot evaluation experiments consumed approximately 136 GPU-days; baseline fine-tuning experiments required around 864 GPU-days; analysis experiments contributed the bulk of usage, totaling 2,554 GPU-days. Human evaluation experiments—including rollout generation and response annotation using UNIVERSE—incurred an additional 1.125 GPU-days. Additional compute was required for preliminary experiments, and failed runs not included in the final paper. These development activities accounted for an estimated 1,599 GPU-days. In total, all experiments amounted to approximately 5,153 GPU-days, equivalent to 14.12 GPU-years.

C UNIVERSE: Implementation Overview

This section outlines the implementation of UNIVERSE in Python, presented as high-level pseudocode. The system is structured around two main stages:

(i) *Adaptation:* fine-tuning a VLM on task-specific question-answer (QA) supervision derived from ground truth;

(ii) *Evaluation:* using the adapted model to assess new rollouts via structured, prompt-based recognition tasks.

Adaptation Pipeline.

The adaptation stage can be implemented as two modules: `AdaptationDatasetBuilder` and `VLMAdapter`.

`AdaptationDatasetBuilder`. This class constructs an adaptation dataset from raw ground truth data, initialized via `load_ground_truth_data` (see Section 3 and Appendix D). The core method, `build`, takes four arguments: `alpha_task`, which specifies the task mixture ratio; `beta_format`, which controls the distribution over QA prompt formats; `context_length`, which determines the number of frames per QA instance; and `sampling_strategy`, which defines how frames are sampled from rollouts. The builder first applies `stratified_sample` to select a subset of annotated samples that match the specified configuration. For each sample, it invokes `_sample_visual_context` to extract the relevant frames, and constructs a triplet consisting of `frames`, `question`, and `answer`.

`VLMAdapter`. This class applies an adaptation strategy to a base VLM, passed via the `base_vlm` argument. Given an adaptation dataset `adaptation_data`, a tuning strategy specified by the `strategy` parameter, and a fixed number of training steps `num_steps`, the adapter trains the model by iteratively sampling a batch, computing the loss via `compute_loss`, and applying updates with `update_model`.

```
class AdaptationDatasetBuilder:
    def __init__(self, raw_data_path):
        self.samples = load_ground_truth_data(raw_data_path)

    def build(self, alpha_task, beta_format, context_length, sampling_strategy):
        formatted = stratified_sample(
            samples=self.samples,
            task_proportions=alpha_task,
            format_proportions=beta_format
        )
        dataset = []
        for sample in formatted:
            visual_ctx = self._sample_visual_context(
                sample["frames"], context_length, sampling_strategy
            )
            dataset.append({
                "frames": visual_ctx,
                "question": sample["question"],
                "answer": sample["answer"]
            })
        return dataset


class VLMAdapter:
    def __init__(self, base_vlm):
        self.base_vlm = base_vlm

    def adapt(self, adaptation_data, strategy, num_steps):
        configure_adaptation(self.base_vlm, strategy)
        for step in range(num_steps):
            batch = sample_from(adaptation_data)
```

```
            loss = compute_loss(self.base_vlm, batch)
            update_model(self.base_vlm, loss)
        return self.base_vlm
```

Evaluation Pipeline.

The evaluation stage can be implemented via two additional modules: `RolloutsGenerator` and `Universe`.

`RolloutsGenerator`. This component autoregressively samples rollout trajectories from a world model (texttt world_model). Given an initial observation `o_initial` and an action sequence `a_seq`, the `rollout` method generates a sequence of predicted observations by maintaining lists of past observations (`o_lt`) and actions (`a_lt`). At each timestep, it calls `predict_next_observation` to obtain the next predicted frame, appends it to the rollout sequence `o_seq`, and continues until `timestamps` is reached. This process produces a full trajectory simulating environment dynamics.

`Universe`. This module serves as the inference engine of our framework. It wraps an adapted VLM passed via `adapted_vlm`. Given a generated rollout and an evaluation specification, the method `evaluate_rollout` constructs a prompt using `generate_question`, parameterized by a recognition `target` and `complexity` level. It then calls `evaluate`, which queries the VLM with the resulting `rollout` and `question`, returning the model's answer.

```
class RolloutsGenerator:
    def __init__(self):
        self.world_model = WorldModel(...)

    def predict_next_observation(self, o_lt, a_lt):
        return self.world_model(o_lt, a_lt)

    def rollout(self, o_initial, a_seq, timestamps):
        o_seq = [o_initial]
        o_lt, a_lt = [o_initial], []
        for t in range(timestamps):
            a_lt.append(a_seq[t])
            o_t = self.predict_next_observation(o_lt, a_lt)
            o_seq.append(o_t)
            o_lt.append(o_t)
        return o_seq


class Universe:
    def __init__(self, adapted_vlm):
        self.vlm = adapted_vlm

    def evaluate(self, rollout, question):
        return self.vlm(rollout, question)

    def evaluate_rollout(self, rollout, target, complexity):
        question = generate_question(rollout, target, complexity)
        return self.evaluate(rollout, question)
```

D Dataset

This section details the construction and release of the dataset used to adapt VLMs for fine-grained evaluation of world model rollouts. We curate a realistic, human-centered dataset derived from actual gameplay in a complex multi-agent environment. Designed to provide temporally grounded and semantically structured supervision, the dataset aligns with the downstream evaluation setting and supports adaptation to both action and character recognition tasks across all QA formats. We describe the data construction pipeline, QA generation process, and release format below.

D.1 Construction Process

The ground truth dataset for adapting the evaluator (see Section 3) was developed in collaboration with *Ninja Theory* using human gameplay recordings from *Bleeding Edge*, a 4v4 multiplayer combat game. Data use was governed by a formal agreement with the studio, and collection adhered to the game's End User License Agreement (EULA). All protocols were approved by our Institutional Review Board (IRB), and personally identifiable information (PII) was removed prior to analysis.

Each gameplay session is represented as a tuple $s = (v, c, m)$, where v is a high-resolution MP4 video (60 FPS), c is the synchronized controller action log, and m contains structured metadata (e.g., player roles, agent identities, action categories, and map context). The full set of gameplay sessions is denoted by $\mathcal{S} = \{(v_i, c_i, m_i)\}_{i=1}^{|\mathcal{S}|}$.

The dataset construction pipeline proceeds in three stages:

(i) *Preprocessing*. We begin by filtering out corrupted applying or inactive sessions and synchronizes the video, controller logs, and metadata streams using internal game timestamps: $\mathcal{S}_{\text{valid}} = \texttt{Preprocessing}(\mathcal{S})$. Each valid session is segmented into non-overlapping clips of fixed length $L = 14$ frames, each paired with controller input and shared metadata; formally, for a session $s = (v, c, m) \in \mathcal{S}_{\text{valid}}$, the segmentation produces $\texttt{Segment}(v, c, m, L) = \{(f^{(1:L)}, c^{(1:L)}, m)\}$, where $f^{(1:L)}$ denotes the sequence of frames, $c^{(1:L)}$ the aligned controller inputs, and m the associated metadata. The complete set of extracted clips across all valid sessions is defined as $\mathcal{V} = \bigcup_{s \in \mathcal{S}_{\text{valid}}} \texttt{Segment}(s, L)$, where each element $v \in \mathcal{V}$ is a triplet $(f^{(1:L)}, c^{(1:L)}, m)$ consisting of video frames, corresponding controller inputs, and metadata.

(ii) *Description Generation*. Next, for each sequence of frames $f^{(1:L)} \in \mathcal{V}$, we use the associated control log $c^{(1:L)}$ to extract action information and the metadata m to obtain character-related attributes. These are combined to generate a structured natural language description via $d = \texttt{Describe}(c^{(1:L)}, m)$. This yields a set of paired video–text examples: $\mathcal{Z} = \{(f^{(1:L)}, d) \mid f^{(1:L)} \in \mathcal{V}\}$.

(iii) *Question-Answer Pair Construction*. Finally, we generate six QA pairs per clip, spanning two predefined tasks (AR and CR), each instantiated in three question formats: binary, multiple-choice, and open-ended. To enable this, we define task-specific answer spaces using $\texttt{GetAnswerSpace}(\mathcal{Z})$, which returns $\mathcal{Y}_{\text{AR}}$ for action categories and $\mathcal{Y}_{\text{CR}}$ for character identities, based on all video–text pairs in $\mathcal{Z}$. For each clip, we extract the task-specific ground-truth answer from the corresponding description as $y = \texttt{ExtractLabel}(d, t)$, where $t \in \{\text{AR}, \text{CR}\}$. Each QA format is constructed as follows: (i) *Binary*: Two binary question-answer pairs are generated per instance using `FormatBinaryPrompt`. The positive question Q^{pos} is constructed using the correct label $y \in \mathcal{Y}^{(t)}$ and paired with the positive answer A^{pos}. The negative question Q^{neg} is constructed using an incorrect label $\tilde{y} \sim \texttt{SampleDistractor}(\mathcal{Y}^{(t)} \setminus \{y\})$ and paired with the negative answer A^{neg}. (ii) *Multiple-Choice*: A question Q is generated using the full set of candidate options, formatted via $\texttt{FormatOptions}(\mathcal{Y}_t)$. The question is constructed with $\texttt{FormatMCPrompt}(t, O)$ and paired with the correct answer $y \in \mathcal{Y}_t$. (iii) *Open-Ended*: A free-form question Q is generated using $\texttt{FormatOEPrompt}(t)$, prompting the model to produce the correct label $y \in \mathcal{Y}_t$ without access to predefined answer choices.

The final dataset is represented as $\mathcal{D} = \{(f_i^{(1:L)}, QA_i)\}_{i=1}^{|\mathcal{D}|}$, where each $f^{(1:L)}$ is a video clip and $QA = \{(Q_j, A_j)\}_{j=1}^{6}$ is the associated set of question–answer pairs, covering all combinations of three question formats (binary, multiple-choice, open-ended) and two tasks (Action Recognition and Character Recognition). A detailed data pipeline is provided in Algorithm 1.

Algorithm 1 Dataset Construction Process

Procedure `DatasetCreation`($\mathcal{S}, L$)**:**
- $\mathcal{S}_{\text{valid}} \leftarrow$ `Preprocessing`($\mathcal{S}$)
- $\mathcal{V} \leftarrow \emptyset$
- **for** $(v, c, m) \in \mathcal{S}_{valid}$ **do**
  - $\mathcal{V}_s \leftarrow$ `Segment`(v, c, m, L)
  - $\mathcal{V} \leftarrow \mathcal{V} \cup \mathcal{V}_s$
- $\mathcal{Z} \leftarrow \emptyset$
- **for** $(f^{(1:L)}, c^{(1:L)}, m) \in \mathcal{V}$ **do**
  - $d \leftarrow$ `Describe`($m, c^{(1:L)}$)
  - $\mathcal{Z} \leftarrow \mathcal{Z} \cup \{(f^{(1:L)}, d)\}$
- $\mathcal{D} \leftarrow \emptyset$
- $\mathcal{Y}_{\text{AR}}, \mathcal{Y}_{\text{CR}} \leftarrow$ `GetAnswerSpace`($\mathcal{Z}$)
- **for** $(f^{(1:L)}, d) \in \mathcal{V}$ **do**
  - $\mathcal{QA} \leftarrow$ `GenerateQAPairs`($d, \mathcal{Y}_{\text{AR}}, \mathcal{Y}_{\text{CR}}$)
  - **for** $(Q, A) \in \mathcal{QA}$ **do**
    - $\mathcal{D} \leftarrow \mathcal{D} \cup \{(f^{(1:L)}, Q, A)\}$
- **return** $\mathcal{D}$

Procedure `GenerateQAPairs`($d, \mathcal{Y}_{AR}, \mathcal{Y}_{CR}$)**:**
- $\mathcal{QA} \leftarrow \emptyset$
- **for** $t \in \{AR, CR\}$ **do**
  - $y \leftarrow$ `ExtractLabel`(d, t)
  - $QA_{\text{bin}}^{pos}, QA_{\text{bin}}^{neg} \leftarrow$ `CreateBinaryQA`(t, y)
  - $\mathcal{QA} \leftarrow \mathcal{QA} \cup \{QA_{\text{bin}}^{\text{pos}}, QA_{\text{bin}}^{\text{neg}}\}$
  - $QA_{\text{mc}} \leftarrow$ `CreateMCQA`($t, y, \mathcal{Y}_t$)
  - $\mathcal{QA} \leftarrow \mathcal{QA} \cup QA_{\text{mc}}$
  - $QA_{\text{oe}} \leftarrow$ `CreateOpenEndedQA`(t, y)
  - $\mathcal{QA} \leftarrow \mathcal{QA} \cup QA_{\text{oe}}$
- **return** $\mathcal{Q}$

Procedure `CreateBinaryQA`(t, y)**:**
- $\tilde{y} \leftarrow$ `SampleDistractor`($\mathcal{Y}_t \setminus \{y\}$)
- $Q^{pos} \leftarrow$ `FormatBinaryPrompt`(t, y)
- $Q^{neg} \leftarrow$ `FormatBinaryPrompt`($t, \tilde{y}$)
- **return** $\{(Q^{pos}, A^{pos}),\ (Q_{\tilde{y}}, A^{neg})\}$

Procedure `CreateMCQA`($t, y, \mathcal{Y}_t$)**:**
- $O \leftarrow$ `FormatOptions`($\mathcal{Y}_t$)
- $Q \leftarrow$ `FormatMCPrompt`(t, O)
- **return** Q, y

Procedure `CreateOpenEndedQA`(t, y)**:**
- $Q \leftarrow$ `FormatOEPrompt`(t)
- **return** Q, y

D.2 Release Details

To support reproducibility and further research, we release a subset of our evaluation data. This includes sampled human gameplay segments, aligned action vectors and environment states, natural language descriptions, and QA annotations spanning binary, multiple-choice, and open-ended formats. The dataset is included in the supplementary ZIP file and will be publicly released following the publication of the paper.

File Layout. The data is organized as follows:

- `human-gameplay-segments/`: directory of `.npz` files, each containing image frames along with frame-aligned actions and states;
- `annotations.jsonl`: line-delimited JSON file containing natural language descriptions, QA prompts, and ground truth answers.

Structure. Each dataset instance corresponds to a short human gameplay segment stored as a NumPy archive (`.npz`), containing:

(i) `images` $\in \mathbb{R}^{14\times3\times180\times300}$: a sequence of 14 RGB frames in channel-first (CHW) format;

(ii) `actions` $\in \mathbb{R}^{14\times16}$: frame-aligned control vectors;

(iii) `states` $\in \mathbb{R}^{14\times56}$: frame-aligned environment states.

Annotation Format. Annotations are provided in `annotations.jsonl`, a line-delimited JSON file where each entry corresponds to a single gameplay segment. Each entry includes structured prompts and ground truth answers spanning all tasks and formats.

Specifically, each annotation entry includes:

- `filename`: Unique identifier of the associated `.npz` file containing visual observations (frames), action vectors, and states.
- `description`: Natural language summary of the video segment.
- `ar_binary_pos_q`, `ar_binary_pos_a`: Affirmative binary question and corresponding answer, evaluating recognition of the correct action.
- `ar_binary_neg_q`, `ar_binary_neg_a`: Negative binary question and corresponding answer, targeting rejection of an incorrect action.
- `ar_mc`: Multiple-choice question prompting the model to select the correct action from a list of candidate classes.
- `ar_oe`: Open-ended question prompting free-form generation of the observed action.
- `ar_answer`: Ground truth action label corresponding to both `ar_mc` and `ar_oe`.
- `cr_binary_pos_q`, `cr_binary_pos_a`: Affirmative binary question and corresponding answer for identifying the correct character.
- `cr_binary_neg_q`, `cr_binary_neg_a`: Negative binary question and corresponding answer targeting an incorrect character identity.
- `cr_mc`: Multiple-choice question prompting identification of the correct character from a candidate set.
- `cr_oe`: Open-ended question prompting free-form naming of the character.
- `cr_answer`: Ground truth character label shared across both `cr_mc` and `cr_oe`.

E Experimental Details

In this section, we provide a detailed description of the dataset preparation process, model architecture, prompt templates, training procedure. Additionally, we provide an overview of all results presented in the main paper in numerical table form, an report additional experimental results leveraging alternate fine-tuning solutions.

E.1 Model

This section provides extended details on the architecture, pretraining configuration, and input formatting of the vision-language models used in our experiments. Our primary backbone is PaliGemma [10, 91].

Table 1: Detailed architecture of the PaliGemma model, comprising a SigLIP-So400m vision tower, a multimodal projection head, and a Gemma-based language decoder. All transformer layers follow standard design and include residual connections around attention and MLP blocks.

Component	**Configuration**
	Vision Tower: SigLIP-So400m
Patch Embedding	Conv2d(in=3, out=1152, kernel=14, stride=14)
Position Embedding	Embedding(num_embeddings=256, emb_dim=1152)
Encoder	27 × Transformer Encoder Layers
Self-Attention	—
Query / Key / Value projection	Linear(1152 → 1152, bias=True)
Layer Normalization	LayerNorm((1152,), eps=1e-6)
MLP Block	—
Activation Function	GELU-Tanh
Feedforward layer (up)	Linear(1152 → 4304, bias=True)
Feedforward layer (down)	Linear(4304 → 1152, bias=True)
Layer Normalization	LayerNorm((1152,), eps=1e-6)
Post-Encoder Layer Norm	LayerNorm((1152,), eps=1e-6)
	Multimodal Projection Head
Linear Projection	Linear(1152 → 2304, bias=True)
	Language Model: Gemma
Token Embedding	Embedding(vocab=257216, dim=2304)
Decoder Stack	26 × Transformer Decoder Layers
Self-Attention	—
Query projection	Linear(2304 → 2048, bias=False)
Key projection	Linear(2304 → 1024, bias=False)
Value projection	Linear(2304 → 1024, bias=False)
Output projection	Linear(2048 → 2304, bias=False)
MLP Block	—
Gating projection	Linear(2304 → 9216, bias=True)
Down projection	Linear(2304 → 9216, bias=True)
Up projection	Linear(9216 → 2304, bias=True)
Activation Function	GELU-Tanh
Normalization Layers	—
Input Norm	RMSNorm(2304, eps=1e-6)
Post-Attn Norm	RMSNorm(2304, eps=1e-6)
Pre-FFN Norm	RMSNorm(2304, eps=1e-6)
Post-FFN Norm	RMSNorm(2304, eps=1e-6)
Rotary Embeddings	GemmaRotaryEmbedding
LM Head	Linear(2304 → 257216, bias=False)

E.1.1 Overview

PaliGemma is a VLM that processes both images and text as input and autoregressively generates natural language output. It follows the training paradigm of PaLI-3 [24], combining a ViT-based vision encoder [29] with a decoder-only Transformer language model. The model is publicly available [101]. The architecture is fully modular, comprising three parameterized components: (i) *Vision encoder* ($\mathcal{M}_V$): based on SigLIP [108], specifically the "shape optimized" So400m [3]. (ii) *Multimodal projection head* ($\mathcal{M}_P$): a single linear layer for projecting visual features into the language decoder's embedding space. (iii) *Language decoder* ($\mathcal{M}_L$): a Transformer-based autoregressive model from the Gemma family [73, 83]. Below, we discuss the architecture in more details, the general layer-level overview is also provided in Table 1.

Table 2: Component-wise parameter overview of the PaliGemma model.

Component	Model / Variant	Details	# Params
Vision Encoder	SigLIP-So400m	Input resolutions: 224px^2, 448px^2, 896px^2	400M
Multimodal Projection	—	Connects vision and language components	2.66M
Language Model	PG 1	Gemma 1 2B, pre-trained on 6T tokens	3B
	PG 2	Gemma 2 2B, pre-trained on 2T tokens	3B
	PG 3	Gemma 2 9B, pre-trained on 8T tokens	9.7B

Vision Encoder: SigLIP-So400m. The visual backbone $\mathcal{M}_V$ is a ViT-style encoder pretrained using a Sigmoid contrastive loss (SigLIP). It processes input images by dividing them into non-overlapping 14×14 patches. Each patch is linearly projected into a 1152-dimensional embedding via a convolutional stem. To encode spatial structure, learned positional embeddings are added before the representation is passed through a stack of 27 SigLIP encoder layers. Each encoder layer contains multi-head self-attention with projection layers for queries, keys, and values, followed by an MLP block with GELU-Tanh activations. All transformer blocks use LayerNorm and residual connections. The vision tower supports multiple input resolutions (224, 448, 896), though our experiments fix resolution at 224px^2 for consistency and efficiency.

Multimodal Projection Head. The projection head $\mathcal{M}_P$ is a lightweight linear mapping from the vision encoder's output dimension (1152) to the language decoder's input dimension (2304). It contains approximately 2.66M parameters and is initialized with zero-mean weights. This head enables alignment between visual and linguistic modalities and is important for bridging the representation gap between the vision and language components.

Language Decoder: Gemma. The language module $\mathcal{M}_L$ is a decoder-only Transformer with 26 layers and 2304-dimensional hidden states. Token embeddings are learned over a vocabulary of 257,216 tokens, encoded using the SentencePiece tokenizer [58]. Each Transformer block contains a self-attention mechanism with separate linear projections for queries, keys, and values. The MLP block follows a gated architecture, where the input is processed through parallel down projection and gating projection layers, modulated by a GELU-Tanh activation [44], combined via elementwise multiplication, and then passed through an up projection to return to the model's hidden dimension. RMSNorm is applied before and after both attention and MLP sublayers to stabilize training. Rotary positional embeddings are added to enable relative position encoding. Output tokens are produced via a tied language modeling head that projects back to the vocabulary space.

E.1.2 Configurations

Table 2 summarizes the architecture components and parameter counts of the PaliGemma configurations available for experimentation. While we focus on the PaliGemma 2 3b variant in our study, we include all publicly released configurations for completeness and to clarify how our selected model compares to other available options. All three variants share the same vision encoder and multimodal integration strategy, differing only in the language decoder. The first configuration, PaliGemma 1 3b, pairs the visual encoder with Gemma 1 (2B), pretrained on 6 trillion tokens, resulting in a total model size of approximately 3 billion parameters. The second configuration, PaliGemma 2 3b, replaces the decoder with Gemma 2 (2B), pretrained on 2 trillion tokens, and maintains a comparable total parameter count. The third and largest variant, PaliGemma 2 10b, uses Gemma 2 (9B) as the decoder, pretrained on 8 trillion tokens, yielding a total model size of approximately 9.7 billion parameters.

E.1.3 Prompt Format

To generate textual responses, we adopt a unified prompt format for the decoder. Each input sequence consists of image tokens $S_{\mathcal{I}}$, a textual prefix $S_{\mathcal{T}}^{\text{PREF}}$ containing the question, and a suffix $S_{\mathcal{T}}^{\text{SUFF}}$ containing the expected answer. The model autoregressively generates the answer tokens, and training loss is applied only to the suffix.

Let n denote the number of input frames and p the number of visual tokens (patch embeddings) per frame. In our setting, each frame is encoded as $p = 256$ visual tokens. The overall input schema is as follows:

$$
\begin{aligned}
S = \; & \underbrace{\texttt{<image>}_1^{(1)}, \ldots, \texttt{<image>}_p^{(1)},\ \ldots,\ \texttt{<image>}_1^{(n)}, \ldots, \texttt{<image>}_p^{(n)}}_{S_{\mathcal{I}}\text{: Visual tokens from } n \text{ frames, each represented as } p \text{ patches}} \\
& \underbrace{\texttt{<BOS>},\ \texttt{answer en},\ \texttt{<QUESTION>},\ \texttt{<SEP>}}_{S_{\mathcal{T}}^{\text{PREF}}\text{: Prefix (cue + question)}} \\
& \underbrace{\texttt{<ANSWER>},\ \texttt{<EOS>},\ \texttt{<PAD>}, \ldots, \texttt{<PAD>}}_{S_{\mathcal{T}}^{\text{SUFF}}\text{: Suffix (answer)}}
\end{aligned}
$$

Here, $S_{\mathcal{I}}$ contains visual tokens produced by the vision encoder $\mathcal{M}_V$, and projected into $\mathcal{M}_L$ space using $\mathcal{M}_P$. The prefix $S_{\mathcal{T}}^{\text{PREF}}$ starts with a special `<BOS>` token and includes a task-language cue (e.g., ''`answer en`''), the question, and a separator `<SEP>`. The suffix $S_{\mathcal{T}}^{\text{SUFF}}$ contains the target answer, terminated with `<EOS>` and padded with `<PAD>` tokens for batching.

E.1.4 Pretraining Data and Filtering

PaliGemma is pretrained on a mixture of large-scale vision-language datasets, including WebLI [23], CC3M-35L [87], VQ^{2}A-CC3M-35L [19], OpenImages [80], and WIT [90]. Data quality and safety are maintained through pornographic content filtering, text safety and toxicity filtering, and privacy-preserving measures.

E.2 Evaluation Metrics

In this section, we provide additional details on metrics used for quantitative evaluation. We employ two complementary metrics: *Exact Match (EM)* and *ROUGE-F$_1$ (ROUGE)*, which together capture both syntactic precision and semantic alignment.

Exact Match Accuracy (**EM**) measures whether the generated answer is identical to the expected answer, providing a high-precision signal for correctness. Formally, it is defined as:

$$EM = \mathbb{1}(\hat{A} = A) \tag{2}$$

where $\hat{A}$ is the model's prediction and A is the corresponding ground-truth answer. This metric is especially informative for binary and multiple-choice formats where the output space is well-defined.

ROUGE F$_1$ (ROUGE) captures token-level semantic overlap between generated and reference responses by computing the harmonic mean of precision and recall. This allows us to account for partially correct or paraphrased answers. For binary questions, we compute the metric on the bigram level, while for multiple-choice and open-ended formats, we use trigram-level evaluation.

Formally, let G and R denote the sets of n-grams in the generated and reference answers, respectively. Precision and recall are defined as:

$$P = \frac{|G \cap R|}{|G|}, \quad R = \frac{|G \cap R|}{|R|} \tag{3}$$

where $|G \cap R|$ counts overlapping n-grams. The ROUGE score is then computed as:

$$\text{ROUGE} = 2 \times \frac{P \times R}{P + R} \tag{4}$$

Together, these metrics provide a robust view of model performance: EM reflects exact correctness, while ROUGE provides a softer measure of semantic fidelity, particularly useful for evaluating open-ended generations.

E.3 Results

Hyperparameters. Table 3 summarizes the core training hyperparameters used across all adaptation experiments. We train all models on 8 NVIDIA A100 GPUs with a batch size of 1 per device and accumulate gradients over 4 steps, yielding an effective batch size of 32. Each epoch corresponds

Table 3: Summary of hyperparameters used in our experiments.

Hyperparameter	**Value**
Input resolution	224×224
Image frames per input	1–8
Number of epochs	1–10
Batch size (per device)	1
Gradient accumulation steps	4
Optimizer	AdamW [67]
Learning rate	5×10^{-5}, cosine annealing
Learning rate warmup	10%
Weight decay	1×10^{-6}
Gradient clipping	Global norm, threshold 1.0
VLM backbone	PaliGemma 2 (3B) [10]

to a full pass over the adaptation dataset, and no early stopping is applied. Models were trained for 1–10 epochs depending on task and setting. Optimization is performed using AdamW [67] with parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, a base learning rate of 5×10^{-5}, and weight decay of 1×10^{-6}. We use cosine learning rate annealing [68] with a linear warmup over the first 10% of training steps. To stabilize training, we apply gradient clipping with a global norm threshold of 1.0. All models use PaliGemma 2 (3B) [10] as the vision-language backbone unless otherwise noted. We vary the number of input frames between 1 and 8 depending on task, and all images are resized to a fixed resolution of 224×224. Training is conducted in `bfloat16` precision using data parallelism. Model selection is based on final validation accuracy.

Tabular Results Summary. The following tables summarize primary experimental findings across our study. Each entry corresponds to a core evaluation or analysis in the paper, organized by experimental section and aligned with the corresponding table description.

- *Zero-Shot Evaluation* (Section 4): Table 4 reports ROUGE-F_1 zero-shot performance of pretrained PaliGemma and VideoLLaMA3 models on Action and Character Recognition tasks. Models are evaluated in a zero-shot setting with 1 or 8 input frames, across binary, multiple-choice, and open-ended formats.

- *Fine-Tuned Baselines* (Section 4): Table 5 reports ROUGE-F_1 and Exact Match performance of PaliGemma 2 variants fine-tuned using full, partial, and parameter-efficient strategies. All models are trained on a single frame for one epoch, and evaluated across binary, multiple-choice, and open-ended formats.

- *Analysis: Supervision and Temporal Context* (Section 5): Table 6 examines early-stage learning dynamics on Character Recognition (CR), with evaluation at sub-epoch intervals. Table 7 reports AR performance as a function of training budget, scaling the number of epochs with a single input frame. Table 8 extends this analysis to jointly vary training epochs and the number of input frames, disentangling the effects of temporal context and supervision on AR.

- *Analysis: Temporal Sampling Strategies* (Section 5): Table 9 compares first-n and uniform-n frame sampling strategies for Action Recognition, evaluating model performance across varying temporal context lengths ($n \in \{1, \ldots, 8\}$).

- *Analysis: Optimizing Data Mix for Unified Multi-Task Evaluation* (Section 5): This analysis spans three tables. Table 10 explores task-level trade-offs when jointly training on Action and Character Recognition by varying $\alpha_{\texttt{AR}}$ vs. $\alpha_{\texttt{CR}}$, with format distribution held uniform. Table 11 fixes $\alpha_{\texttt{AR}} = 0.8$ and searches over format-level ratios (β), revealing the impact of increased open-ended (OE) supervision. Table 12 further investigates this high-OE regime, balancing the remaining budget between binary and multiple-choice for optimal performance.

Table 4: Zero-shot ROUGE-F_1-based evaluation of PaliGemma (PG) and VideoLLaMA3 (VL3) models on Action and Character Recognition tasks using 1 and 8 input frames. "MC" denotes multiple-choice and "OE" open-ended formats.

		Action Recognition			Character Recognition		
Fr	**Model**	**Binary**	**MC**	**OE**	**Binary**	**MC**	**OE**
1	PG 1 3B	50.43 ± 0.13	8.12 ± 0.02	10.83 ± 0.01	50.73 ± 0.38	0.46 ± 0.06	0.00 ± 0.00
	PG 2 3B	44.69 ± 0.03	9.30 ± 0.17	12.64 ± 0.01	48.58 ± 0.07	0.28 ± 0.06	0.01 ± 0.00
	PG 2 10B	50.04 ± 0.03	26.98 ± 0.00	12.35 ± 0.21	50.08 ± 0.07	8.33 ± 0.50	0.00 ± 0.00
	VL3-2B	3.24 ± 0.00	18.52 ± 0.06	6.27 ± 0.05	8.76 ± 0.04	3.44 ± 0.08	0.50 ± 0.01
	VL3-7B	45.02 ± 0.28	15.53 ± 0.05	6.54 ± 0.04	39.09 ± 0.73	6.21 ± 0.05	0.51 ± 0.02
8	PG 1 3B	51.67 ± 0.02	10.68 ± 0.00	10.32 ± 0.00	51.39 ± 0.07	0.25 ± 0.00	0.00 ± 0.00
	PG 2 3B	47.61 ± 0.19	6.73 ± 0.04	14.52 ± 0.00	48.37 ± 0.18	0.03 ± 0.00	0.01 ± 0.00
	PG 2 10B	50.02 ± 0.06	26.93 ± 0.01	12.12 ± 0.00	50.09 ± 0.06	0.22 ± 0.00	0.00 ± 0.00
	VL3-2B	13.92 ± 0.13	3.47 ± 0.02	0.32 ± 0.01	13.92 ± 0.13	3.46 ± 0.04	0.32 ± 0.01
	VL3-7B	15.05 ± 0.21	16.67 ± 0.35	6.35 ± 0.06	12.76 ± 0.52	5.88 ± 0.01	0.54 ± 0.01

Table 5: Performance of fine-tuned PaliGemma 2 variants on Action and Character Recognition tasks. We compare full, partial, and parameter-efficient tuning strategies. "MC" denotes multiple-choice and "OE" open-ended formats.

	Binary		Multiple-choice		Open-ended	
Model	**EM**	**ROUGE**	**EM**	**ROUGE**	**EM**	**ROUGE**
			Action Recognition			
$\mathcal{F}_L$	50.00 ± 0.00	50.00 ± 0.00	13.13 ± 0.00	27.57 ± 0.00	13.13 ± 0.00	27.57 ± 0.00
$\mathcal{F}_P$	83.97 ± 0.02	83.97 ± 0.02	61.43 ± 0.58	68.05 ± 0.70	61.68 ± 0.35	68.46 ± 0.19
$\mathcal{F}_V$	83.70 ± 0.97	83.70 ± 0.97	63.40 ± 0.45	69.87 ± 0.44	66.03 ± 0.10	71.92 ± 0.08
$\mathcal{F}_{P+L}$	74.47 ± 1.64	74.47 ± 1.64	13.13 ± 0.00	27.57 ± 0.00	55.74 ± 0.70	64.83 ± 0.29
$\mathcal{F}_{V+L}$	75.80 ± 0.16	75.80 ± 0.16	13.13 ± 0.00	27.57 ± 0.00	13.13 ± 0.00	27.57 ± 0.00
$\mathcal{F}_{V+P}$	73.46 ± 0.85	73.46 ± 0.85	61.21 ± 0.23	67.57 ± 0.21	64.70 ± 0.02	70.93 ± 0.01
$\mathcal{F}_{all}$	74.35 ± 1.37	74.35 ± 1.37	13.13 ± 0.00	27.57 ± 0.00	13.13 ± 0.00	27.57 ± 0.00
$\mathcal{F}_{LoRA}$	44.66 ± 0.21	44.66 ± 0.21	0.02 ± 0.01	9.21 ± 0.01	0.00 ± 0.00	12.49 ± 0.00
			Character Recognition			
$\mathcal{F}_L$	50.00 ± 0.00	50.00 ± 0.00	98.92 ± 0.00	98.92 ± 0.00	98.98 ± 0.00	98.99 ± 0.01
$\mathcal{F}_P$	99.09 ± 0.11	99.09 ± 0.11	99.22 ± 0.33	99.22 ± 0.33	99.15 ± 0.07	99.15 ± 0.07
$\mathcal{F}_V$	99.31 ± 0.01	99.31 ± 0.01	99.14 ± 0.42	99.14 ± 0.42	99.61 ± 0.12	99.61 ± 0.12
$\mathcal{F}_{P+L}$	50.00 ± 0.00	50.00 ± 0.00	98.28 ± 0.00	98.30 ± 0.02	98.39 ± 0.00	98.39 ± 0.00
$\mathcal{F}_{V+L}$	50.00 ± 0.00	50.00 ± 0.00	96.88 ± 0.00	96.88 ± 0.00	98.45 ± 0.00	98.45 ± 0.00
$\mathcal{F}_{V+P}$	60.32 ± 0.02	60.32 ± 0.02	99.22 ± 0.00	99.22 ± 0.00	99.79 ± 0.00	99.79 ± 0.00
$\mathcal{F}_{all}$	50.00 ± 0.00	50.00 ± 0.00	97.67 ± 0.06	97.67 ± 0.06	96.55 ± 0.01	96.55 ± 0.01
$\mathcal{F}_{LoRA}$	48.76 ± 0.00	48.76 ± 0.00	0.00 ± 0.00	0.32 ± 0.00	0.00 ± 0.00	0.01 ± 0.01

F Human Annotation Study

This section provides full details of our human annotation study, including rollout generation, annotation procedures, inter-annotator agreement, and evaluation metrics. The goal is to validate the adapted VLM's fine-grained predictions on generated video rollouts.

F.1 Study Design

Task Overview. Human annotators were presented with short video clips generated by a world model, each paired with a natural language question and an answer generated by the VLM. They were asked to judge whether the model's answer accurately described what was shown in the video. Each QA pair was rated using one of four categories: *Correct* (score = 1), *Partially Correct* (0.5), *Incorrect* (0), or *Unclear / Cannot Tell* (excluded from accuracy computation).

Table 6: *Supervision and Temporal Context:* Training budget analysis for Character Recognition. Models are fine-tuned for sub-epoch durations and evaluated across binary, multiple-choice (MC), and open-ended (OE) formats.

	Binary		Multiple-choice		Open-ended	
Ep	**EM**	**ROUGE**	**EM**	**ROUGE**	**EM**	**ROUGE**
0.005	50.84 ± 1.70	50.84 ± 1.70	14.27 ± 0.00	14.27 ± 0.00	13.16 ± 0.00	13.27 ± 0.00
0.01	54.92 ± 0.84	54.92 ± 0.84	17.85 ± 0.00	17.85 ± 0.00	16.78 ± 0.14	16.88 ± 0.01
0.02	57.91 ± 3.19	57.91 ± 3.19	28.64 ± 0.00	28.64 ± 0.00	26.29 ± 2.96	28.38 ± 0.01
0.03	57.45 ± 0.58	57.45 ± 0.58	64.19 ± 0.00	64.19 ± 0.00	40.76 ± 0.01	40.98 ± 0.00
0.06	65.88 ± 3.71	65.88 ± 3.71	93.96 ± 0.00	93.96 ± 0.00	88.89 ± 0.00	88.95 ± 0.01
0.10	87.47 ± 4.11	87.47 ± 4.11	96.54 ± 0.00	96.54 ± 0.00	97.01 ± 0.38	97.02 ± 0.40
0.125	91.63 ± 7.71	91.63 ± 7.71	97.08 ± 0.00	97.08 ± 0.00	97.28 ± 0.00	97.30 ± 0.00
0.20	97.96 ± 0.45	97.96 ± 0.45	97.89 ± 0.28	97.89 ± 0.28	98.12 ± 0.00	98.14 ± 0.02
0.25	97.75 ± 0.74	97.75 ± 0.74	98.08 ± 0.00	98.08 ± 0.00	98.12 ± 0.00	98.15 ± 0.00
0.33	98.42 ± 0.20	98.42 ± 0.20	98.19 ± 0.00	98.19 ± 0.00	98.30 ± 0.00	98.35 ± 0.00
0.50	98.74 ± 0.06	98.74 ± 0.06	98.45 ± 0.00	98.45 ± 0.00	98.51 ± 0.00	98.54 ± 0.00
0.67	99.11 ± 0.10	99.11 ± 0.10	98.99 ± 0.08	98.99 ± 0.08	99.09 ± 0.00	99.09 ± 0.00
0.75	99.03 ± 0.04	99.03 ± 0.04	99.15 ± 0.00	99.15 ± 0.00	99.21 ± 0.00	99.22 ± 0.01
1	99.09 ± 0.11	99.09 ± 0.11	99.22 ± 0.33	99.22 ± 0.33	99.15 ± 0.07	99.15 ± 0.07

Table 7: *Supervision and Temporal Context:* Training budget analysis for Action Recognition, with models fine-tuned for up to 10 epochs. Evaluated using across binary, multiple-choice (MC), and open-ended (OE) formats.

	Binary		Multiple-Choice		Open-Ended	
Ep	**EM**	**ROUGE**	**EM**	**ROUGE**	**EM**	**ROUGE**
1	83.97 ± 0.02	83.97 ± 0.02	61.43 ± 0.58	68.05 ± 0.70	61.68 ± 0.35	68.46 ± 0.19
2	84.92 ± 0.23	84.92 ± 0.23	64.90 ± 0.15	71.17 ± 0.01	64.05 ± 0.01	70.36 ± 0.01
4	85.37 ± 0.30	85.37 ± 0.30	64.58 ± 0.69	70.89 ± 0.55	64.79 ± 0.31	70.88 ± 0.27
8	85.18 ± 0.20	85.18 ± 0.20	63.53 ± 0.35	69.95 ± 0.37	63.43 ± 0.28	69.91 ± 0.18
10	85.11 ± 0.41	85.11 ± 0.41	62.88 ± 1.35	69.34 ± 1.20	66.82 ± 3.75	72.34 ± 3.04

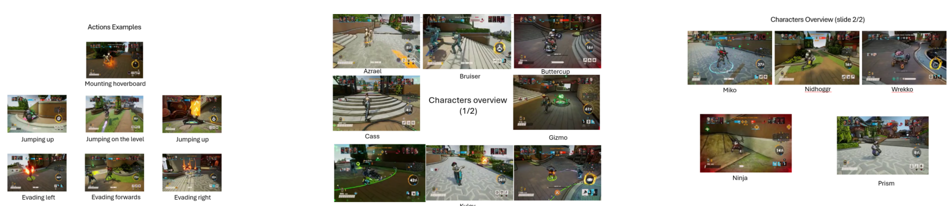

Figure 9: Reference slides shown to annotators during the human annotation study, illustrating the two recognition targets: *actions* (left) and *characters* (center and right). The slides include 20 exemplar videos (7 actions, 13 characters) to support consistent evaluation of VLM-generated responses.

Annotation Setup and Interface. Annotations were collected using a custom PowerPoint-based interface (see Figure 10). Each slide presented a short video, a question, and a generated answer. Annotators selected a rating from a predefined rubric. The full annotation guidelines – including action and character definitions and rating instructions – were embedded in the annotation deck for reference. For completeness, we also provide them in Table 13 and Figure 9. The annotation study was carried out by a subset of the authors with prior experience in the environment. Judging correctness required non-trivial familiarity with the visual dynamics and task ontology, making expert annotation necessary. All annotators were compensated above local minimum wage rates. Each QA pair was independently rated by two primary annotators. In cases of disagreement or if either annotator marked the example as *Unclear*, a third, more experienced adjudicator reviewed the pair and assigned a final rating.

Table 8: *Supervision and Temporal Context:* Training budget and temporal context analysis for Action Recognition. Models are fine-tuned for up to 10 epochs and evaluated with up to 8 input frames.

		Binary		**Multiple-choice**		**Open-ended**	
Ep	**Fr**	**EM**	**ROUGE**	**EM**	**ROUGE**	**EM**	**ROUGE**
1	1	83.97 ± 0.02	83.97 ± 0.02	61.43 ± 0.58	68.05 ± 0.70	61.68 ± 0.35	68.46 ± 0.19
	2	84.42 ± 0.06	84.42 ± 0.06	65.53 ± 0.27	72.03 ± 0.06	65.38 ± 0.06	71.74 ± 0.23
	4	90.97 ± 0.10	90.97 ± 0.10	83.11 ± 0.08	87.13 ± 0.04	82.26 ± 0.14	87.02 ± 0.06
	8	93.85 ± 0.28	93.85 ± 0.28	88.89 ± 0.14	93.40 ± 0.76	87.80 ± 0.20	92.23 ± 0.16
2	1	85.10 ± 0.02	85.10 ± 0.02	64.93 ± 0.06	71.09 ± 0.11	64.05 ± 0.01	70.36 ± 0.01
	2	86.53 ± 0.45	86.40 ± 0.26	69.20 ± 0.88	75.02 ± 0.67	68.83 ± 0.47	72.45 ± 2.76
	4	92.26 ± 0.34	92.26 ± 0.34	84.19 ± 0.10	88.46 ± 0.11	83.34 ± 0.15	87.84 ± 0.06
	8	95.05 ± 0.15	95.05 ± 0.15	89.27 ± 0.14	93.30 ± 0.22	89.42 ± 0.22	93.25 ± 0.15
4	1	85.37 ± 0.30	85.37 ± 0.30	64.58 ± 0.69	70.89 ± 0.55	64.79 ± 0.31	70.88 ± 0.27
	2	86.89 ± 0.06	86.89 ± 0.06	69.89 ± 0.83	75.49 ± 0.69	70.04 ± 0.08	75.74 ± 0.06
	4	92.58 ± 0.18	92.58 ± 0.18	85.13 ± 0.19	89.28 ± 0.17	84.61 ± 0.07	88.81 ± 0.04
	8	95.29 ± 0.07	95.29 ± 0.07	90.64 ± 0.00	94.09 ± 0.04	90.18 ± 0.16	93.81 ± 0.11
8	1	85.04 ± 0.00	85.04 ± 0.00	63.53 ± 0.35	69.95 ± 0.37	63.62 ± 0.00	70.03 ± 0.00
	2	87.27 ± 0.40	87.27 ± 0.40	69.84 ± 0.66	75.44 ± 0.45	70.11 ± 0.65	75.75 ± 0.52
	4	92.97 ± 0.49	92.97 ± 0.49	85.32 ± 0.08	89.27 ± 0.08	84.93 ± 0.21	89.05 ± 0.11
	8	95.48 ± 0.21	95.48 ± 0.21	90.71 ± 0.14	94.15 ± 0.13	91.02 ± 0.28	93.96 ± 0.71
10	1	85.40 ± 0.00	85.40 ± 0.00	62.88 ± 1.35	69.34 ± 1.20	66.82 ± 3.75	72.34 ± 3.04
	2	87.17 ± 0.22	87.17 ± 0.22	70.18 ± 0.00	75.64 ± 0.00	69.59 ± 0.20	75.37 ± 0.07
	4	92.96 ± 0.37	92.96 ± 0.37	85.02 ± 0.58	89.05 ± 0.49	84.71 ± 0.08	88.94 ± 0.06
	8	96.03 ± 0.05	96.03 ± 0.05	90.75 ± 0.04	94.22 ± 0.05	91.00 ± 0.11	94.33 ± 0.09

Table 9: Comparison of frame sampling strategies for Action Recognition. We evaluate first-n vs. uniform-n sampling across varying temporal context lengths ($n \in \{1, \ldots, 8\}$).

		Binary		**Multiple-choice**		**Open-ended**	
	Fr	**EM**	**ROUGE**	**EM**	**ROUGE**	**EM**	**ROUGE**
First-N	1	83.97 ± 0.02	83.97 ± 0.02	61.43 ± 0.58	68.05 ± 0.70	61.68 ± 0.35	68.46 ± 0.19
	2	84.42 ± 0.06	84.42 ± 0.06	65.53 ± 0.27	72.03 ± 0.06	65.38 ± 0.06	71.74 ± 0.23
	3	87.93 ± 0.28	87.93 ± 0.28	75.73 ± 0.18	81.07 ± 0.06	74.68 ± 0.16	80.21 ± 0.08
	4	90.97 ± 0.10	90.97 ± 0.10	83.11 ± 0.08	87.13 ± 0.04	82.26 ± 0.14	87.02 ± 0.06
	5	92.00 ± 0.30	92.00 ± 0.30	85.46 ± 0.34	89.84 ± 0.18	85.10 ± 0.16	89.47 ± 0.10
	6	92.95 ± 0.30	92.95 ± 0.30	86.86 ± 0.08	91.13 ± 0.06	86.59 ± 0.39	90.82 ± 0.30
	7	93.31 ± 0.03	93.31 ± 0.03	87.95 ± 0.08	92.06 ± 0.06	87.58 ± 0.17	91.82 ± 0.08
	8	93.85 ± 0.28	93.85 ± 0.28	88.89 ± 0.14	93.40 ± 0.76	87.80 ± 0.20	92.23 ± 0.16
Uniform-N	1	83.97 ± 0.02	83.97 ± 0.02	61.43 ± 0.58	68.05 ± 0.70	61.68 ± 0.35	68.46 ± 0.19
	2	90.47 ± 0.62	90.47 ± 0.62	83.93 ± 0.04	88.36 ± 0.08	82.68 ± 0.19	87.33 ± 0.01
	3	93.59 ± 0.07	93.59 ± 0.07	88.90 ± 0.11	92.85 ± 0.10	88.49 ± 0.24	92.57 ± 0.04
	4	93.57 ± 0.39	93.57 ± 0.39	89.94 ± 0.04	93.65 ± 0.01	89.56 ± 0.42	93.49 ± 0.28
	5	94.25 ± 0.04	94.25 ± 0.04	89.99 ± 0.18	93.70 ± 0.13	89.72 ± 0.10	93.63 ± 0.23
	6	94.01 ± 0.57	94.01 ± 0.57	90.03 ± 0.04	93.73 ± 0.06	90.09 ± 0.18	93.88 ± 0.16
	7	93.96 ± 0.16	93.96 ± 0.16	90.34 ± 0.23	94.00 ± 0.10	89.94 ± 0.11	93.73 ± 0.10
	8	94.62 ± 0.48	94.62 ± 0.48	90.72 ± 0.12	94.30 ± 0.10	90.30 ± 0.04	94.01 ± 0.02

Selected World Models. For our study, we select two autoregressive world models of different scales, both based on the WHAM architecture [56], a publicly available world model. These models are trained to model sequences of visual frames and controller actions, without any textual supervision. Each world model is a decoder-only transformer [81, 98] trained to autoregressively predict discrete tokens representing visual observations and actions. Visual frames are first encoded using a VQGAN [32], while joystick actions are tokenized using a learned discretization scheme based on action bucketization [55]. The model is trained to predict the next token in the sequence, conditioned on prior visual and action tokens. Specifically, we focus on two versions of WHAM with differing model capacities and training environments:

Table 10: *Optimizing Data Mix for Unified Multi-Task Evaluation:* Performance tradeoffs under varying task-level allocation ratios for Action (α_{AR}) vs. Character Recognition (α_{CR}), with a fixed format distribution ($\beta = 1/3$ per format). Evaluated across binary, multiple-choice (MC), and open-ended (OE) formats.

		Binary		**Multiple-choice**		**Open-ended**	
α_{AR}	α_{CR}	**EM**	**ROUGE**	**EM**	**ROUGE**	**EM**	**ROUGE**
				Action Recognition			
0.20	0.80	84.13 ± 1.66	84.13 ± 1.66	26.10 ± 0.43	39.04 ± 0.10	27.03 ± 0.21	39.60 ± 0.13
0.40	0.60	88.11 ± 1.44	88.11 ± 1.44	28.13 ± 1.15	40.93 ± 0.57	29.17 ± 0.40	41.66 ± 0.51
0.50	0.50	88.59 ± 1.41	88.59 ± 1.41	29.10 ± 0.65	41.20 ± 0.16	29.66 ± 0.54	41.17 ± 0.22
0.60	0.40	90.80 ± 0.04	90.80 ± 0.04	30.44 ± 0.03	42.54 ± 0.01	30.55 ± 0.49	42.32 ± 0.60
0.80	0.20	91.23 ± 0.91	91.23 ± 0.91	84.06 ± 1.26	89.42 ± 1.00	30.88 ± 0.63	42.85 ± 0.42
				Character Recognition			
0.20	0.80	98.57 ± 0.47	98.57 ± 0.47	98.95 ± 0.16	98.95 ± 0.16	98.94 ± 0.22	98.97 ± 0.21
0.40	0.60	98.51 ± 0.53	98.51 ± 0.53	98.77 ± 0.16	98.77 ± 0.16	98.98 ± 0.06	98.98 ± 0.06
0.50	0.50	96.33 ± 1.81	96.33 ± 1.81	98.03 ± 0.06	98.03 ± 0.06	98.23 ± 0.06	98.23 ± 0.06
0.60	0.40	93.22 ± 3.38	93.22 ± 3.38	96.91 ± 1.81	96.94 ± 1.77	97.93 ± 0.02	97.93 ± 0.02
0.80	0.20	80.53 ± 0.49	80.53 ± 0.49	89.08 ± 0.49	89.08 ± 0.49	89.02 ± 0.25	89.18 ± 0.39

Table 11: *Optimizing Data Mix for Unified Multi-Task Evaluation:* Performance on Action and Character Recognition under varying format-level sampling ratios (β) for Binary, Multiple-choice (MC), and Open-ended (OE) questions. We fix $\alpha_{\mathtt{AR}} = 0.8$ and train all models on the first 8 frames.

				Binary		**Multiple-Choice**		**Open-Ended**	
Ep	$\beta_{\mathtt{binary}}$	$\beta_{\mathtt{MC}}$	$\beta_{\mathtt{OE}}$	**EM**	**ROUGE**	**EM**	**ROUGE**	**EM**	**ROUGE**
					Action Recognition				
1	0.4	0.2	0.4	92.32 ± 0.37	92.32 ± 0.37	84.32 ± 0.89	89.54 ± 0.64	31.61 ± 0.23	43.51 ± 0.09
	0.2	0.4	0.4	90.80 ± 0.71	90.80 ± 0.71	86.60 ± 0.38	91.27 ± 0.42	32.06 ± 0.18	43.58 ± 0.54
	0.0	0.4	0.6	49.98 ± 0.04	49.98 ± 0.04	86.65 ± 0.99	91.26 ± 0.87	85.51 ± 1.78	90.13 ± 1.53
	0.0	0.2	0.8	50.11 ± 0.06	50.11 ± 0.06	86.58 ± 0.47	91.42 ± 0.21	87.45 ± 0.09	91.78 ± 0.09
2	0.4	0.2	0.4	93.14 ± 0.48	93.14 ± 0.48	86.77 ± 0.00	91.28 ± 0.00	32.96 ± 0.00	44.00 ± 0.00
	0.2	0.4	0.4	92.89 ± 0.16	92.89 ± 0.16	87.83 ± 0.00	92.13 ± 0.00	33.37 ± 0.00	44.13 ± 0.00
	0.0	0.4	0.6	41.22 ± 0.00	41.30 ± 0.01	89.17 ± 0.07	93.12 ± 0.02	88.55 ± 0.00	93.65 ± 0.00
	0.0	0.2	0.8	49.98 ± 0.03	49.99 ± 0.02	88.68 ± 0.00	92.71 ± 0.00	88.59 ± 0.00	92.56 ± 0.00
4	0.4	0.2	0.4	94.33 ± 0.34	94.33 ± 0.34	92.67 ± 0.07	93.27 ± 0.71	33.94 ± 0.01	43.96 ± 0.00
	0.2	0.4	0.4	94.19 ± 0.13	94.19 ± 0.13	93.04 ± 0.01	93.57 ± 0.74	33.78 ± 0.00	44.73 ± 0.00
	0.0	0.4	0.6	50.19 ± 0.27	50.19 ± 0.27	89.78 ± 0.11	93.52 ± 0.03	88.95 ± 0.05	92.75 ± 0.06
	0.0	0.2	0.8	49.98 ± 0.02	49.98 ± 0.02	89.25 ± 0.00	93.13 ± 0.00	89.41 ± 0.00	93.16 ± 0.00
					Character Recognition				
1	0.4	0.2	0.4	86.42 ± 0.25	86.42 ± 0.25	94.77 ± 0.14	94.77 ± 0.14	94.76 ± 0.08	94.63 ± 0.21
	0.2	0.4	0.4	77.57 ± 0.01	77.57 ± 0.01	94.93 ± 0.03	94.93 ± 0.03	94.51 ± 0.57	94.15 ± 0.00
	0.0	0.4	0.6	50.37 ± 0.52	50.37 ± 0.52	96.56 ± 0.28	96.56 ± 0.28	96.81 ± 0.39	96.82 ± 0.37
	0.0	0.2	0.8	50.51 ± 0.03	50.51 ± 0.03	96.88 ± 0.27	96.88 ± 0.27	97.39 ± 0.18	97.39 ± 0.18
2	0.4	0.2	0.4	89.95 ± 0.46	89.95 ± 0.46	87.35 ± 0.00	87.36 ± 0.00	88.64 ± 0.00	88.64 ± 0.00
	0.2	0.4	0.4	91.28 ± 0.20	91.28 ± 0.20	93.90 ± 0.00	93.93 ± 0.04	93.06 ± 0.00	93.09 ± 0.00
	0.0	0.4	0.6	47.37 ± 0.02	47.48 ± 0.04	97.69 ± 0.00	97.70 ± 0.00	98.07 ± 0.00	98.07 ± 0.00
	0.0	0.2	0.8	50.07 ± 0.04	50.07 ± 0.04	97.75 ± 0.00	97.79 ± 0.00	98.07 ± 0.00	98.07 ± 0.00
4	0.4	0.2	0.4	97.71 ± 0.05	97.71 ± 0.05	97.70 ± 0.00	97.70 ± 0.00	97.88 ± 0.01	97.91 ± 0.03
	0.2	0.4	0.4	96.55 ± 0.06	96.55 ± 0.06	98.51 ± 0.00	98.51 ± 0.00	98.46 ± 0.00	98.46 ± 0.00
	0.0	0.4	0.6	51.25 ± 1.56	51.25 ± 1.56	98.21 ± 0.13	98.21 ± 0.13	98.48 ± 0.06	98.49 ± 0.06
	0.0	0.2	0.8	50.93 ± 0.10	50.93 ± 0.10	98.79 ± 0.00	98.79 ± 0.00	99.08 ± 0.00	99.09 ± 0.00

- *WHAM 140M*: A 140M-parameter model trained for 100K steps on gameplay from a single environment (Environment A / Skygarden) at 128×128 resolution.
- *WHAM 1.6B*: A 1.6B-parameter model trained for 200K steps on gameplay from seven environments (Environments A–G, including Skygarden) at 300×180 resolution.

Rollouts Generation. Rollout generation follows a consistent protocol for both world models: at inference time, the model is conditioned on 1 second of ground-truth gameplay (visual and action

Table 12: *Optimizing Data Mix for Unified Multi-Task Evaluation:* Performance on Action and Character Recognition under high open-ended (OE) supervision, with $\beta_{\texttt{OE}} = 0.8$ and remaining budget split between Binary and Multiple-choice (MC).

				Binary		Multiple-Choice		Open-Ended	
Ep	$\beta_{\texttt{BN}}$	$\beta_{\texttt{MC}}$	$\beta_{\texttt{OE}}$	**EM**	**ROUGE**	**EM**	**ROUGE**	**EM**	**ROUGE**
				Action Recognition					
1	0.15	0.05	0.80	88.85 ± 0.04	88.85 ± 0.04	81.93 ± 0.00	89.08 ± 0.00	86.72 ± 0.00	91.33 ± 0.00
	0.10	0.10	0.80	87.38 ± 0.59	87.38 ± 0.59	85.58 ± 0.00	90.50 ± 0.00	86.88 ± 0.00	91.25 ± 0.00
	0.05	0.15	0.80	85.67 ± 0.19	85.67 ± 0.19	86.38 ± 0.09	91.21 ± 0.06	86.84 ± 0.02	91.34 ± 0.03
2	0.15	0.05	0.80	92.45 ± 0.11	92.45 ± 0.11	87.52 ± 0.00	91.90 ± 0.00	87.97 ± 0.63	92.19 ± 0.41
	0.10	0.10	0.80	92.11 ± 0.18	92.11 ± 0.18	88.42 ± 0.00	92.54 ± 0.00	88.50 ± 0.00	92.54 ± 0.00
	0.05	0.15	0.80	91.98 ± 0.24	91.98 ± 0.24	88.72 ± 0.00	92.78 ± 0.00	88.56 ± 0.00	92.66 ± 0.00
4	0.15	0.05	0.80	92.98 ± 0.21	92.98 ± 0.21	88.93 ± 0.00	93.02 ± 0.00	89.64 ± 0.00	93.34 ± 0.00
	0.10	0.10	0.80	92.81 ± 0.11	92.81 ± 0.11	91.40 ± 2.81	93.88 ± 0.70	89.43 ± 0.00	93.20 ± 0.00
	0.05	0.15	0.80	91.52 ± 0.37	91.52 ± 0.37	89.80 ± 0.00	93.54 ± 0.01	89.81 ± 0.01	93.49 ± 0.06
				Character Recognition					
1	0.15	0.05	0.80	59.75 ± 0.04	59.75 ± 0.04	95.45 ± 0.00	95.45 ± 0.00	97.16 ± 0.00	97.16 ± 0.00
	0.10	0.10	0.80	56.31 ± 0.53	56.31 ± 0.53	94.55 ± 0.00	94.55 ± 0.00	95.96 ± 0.00	95.96 ± 0.00
	0.05	0.15	0.80	50.86 ± 0.08	50.86 ± 0.08	96.87 ± 0.02	96.87 ± 0.02	97.12 ± 0.01	97.12 ± 0.01
2	0.15	0.05	0.80	80.18 ± 0.09	80.18 ± 0.09	95.41 ± 0.00	95.41 ± 0.00	96.91 ± 0.00	96.91 ± 0.00
	0.10	0.10	0.80	70.20 ± 0.21	70.20 ± 0.21	98.02 ± 0.00	98.02 ± 0.00	98.15 ± 0.00	98.15 ± 0.00
	0.05	0.15	0.80	69.67 ± 0.36	69.67 ± 0.36	97.37 ± 0.00	97.37 ± 0.00	97.79 ± 0.00	97.79 ± 0.00
4	0.15	0.05	0.80	94.16 ± 0.12	94.16 ± 0.12	97.06 ± 0.00	97.07 ± 0.00	97.91 ± 0.00	97.91 ± 0.00
	0.10	0.10	0.80	86.67 ± 0.13	86.67 ± 0.13	98.50 ± 0.00	98.50 ± 0.00	98.77 ± 0.00	98.77 ± 0.00
	0.05	0.15	0.80	71.79 ± 0.01	71.79 ± 0.01	98.22 ± 0.00	98.22 ± 0.00	98.57 ± 0.00	98.57 ± 0.00

tokens), after which it generates 10 seconds of future gameplay conditioned only on a sequence of held-out controller actions. The generated rollout is then split into 14-frame chunks. This setup enables a comprehensive analysis of the UNIVERSE's evaluation capabilities across two axes: (i) *in-domain performance*: evaluating on Skygarden (Environment A), the environment used for fine-tuning; (ii) *generalization*: assessing performance on six unseen environments (Environments B–G). It also allows comparison across generation quality and model capacity. We generate 82 rollouts for each model-environment setting, resulting in 656 rollouts in total.

Rollout Filtering. To ensure quality and clarity, we filtered out rollouts that: (i) had no visible agents, (ii) featured stationary agents, (iii) were taken from early uninformative environment segments, or (iv) had significant visual obstruction. We also excluded sequences containing more than four characters to reduce annotation ambiguity.

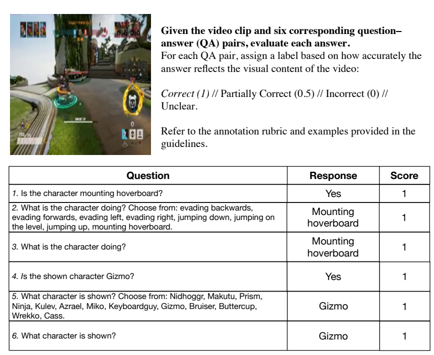

Figure 10: Annotation interface example. Each instance includes a video clip, task instructions, and a table with: *Question* (generated via evaluation protocol), *Response* (VLM output), and *Score* (human-assigned label).

UNIVERSE Response Generation. To obtain responses from UNIVERSE, we provide it with a video segment (resized to match the evaluator's input resolution) along with its corresponding question. We then sample five responses using greedy decoding and select the most frequent response as the final answer. In cases where all five responses are unique (i.e., no majority), one response is selected at random. The resulting dataset comprises rollouts from 8 model–environment pairs: rollouts generated by WHAM 140M on Environment A (Skygarden), and rollouts generated by WHAM 1.6B across seven distinct environments (Environments A–G). For each model–environment pair, we sample 30 rollouts. Each rollout is annotated with 6 question–answer (QA) pairs, along with a corresponding response from the adapted evaluator. Each of the resulting 1,440 QA instances was rated by 3 annotators, yielding 4,320 total human judgments.

Table 13: Annotation instructions provided to human raters as part of the study. The interface outlines task context, scoring criteria, general guidelines, and reference definitions for supported action categories.

1. Task Overview
You will be presented with: • A short video clip; • A natural language question about the video; • An answer generated by a vision-language model. Your task is to evaluate whether the model's answer accurately describes the events depicted in the video.
2. How to Rate Each Answer
Assign one of the following categories: • *Correct (1.0)*: Fully matches the event in the video; • *Partially Correct (0.5)*: Captures the general idea but contains a minor error; • *Incorrect (0.0)*: Wrong, hallucinated, or mismatched with the visual evidence; • *Unclear / Cannot Tell*: Not enough evidence to confidently decide.
3. General Guidelines
• Watch the full video before rating; • Base your decision solely on visible content; • Use provided action and character references; • If multiple plausible interpretations exist and the answer matches one, mark as *Correct*; • If unsure even after review, mark *Unclear / Cannot Tell*; • Optionally leave comments for ambiguous or interesting cases.
5. Action Label Definitions
• *Evading Backwards*: Moves backwards to avoid threat or reposition. • *Evading Forwards*: Moves forwards. • *Evading Left / Right*: Lateral movement left or right. • *Jumping Down*: Jumps from a higher to a lower platform or level. • *Jumping on the Level*: Jumps without elevation change. • *Jumping Up*: Jumps upward to reach a higher platform. • *Mounting Hoverboard*: Begins riding or is seen riding a hoverboard.

F.2 Evaluation Metrics

We report two accuracy-based metrics using the adjudicated labels:

Strict Accuracy.: The proportion of QA pairs labeled as *Correct*:

$$\mathrm{Acc}_{\text{Strict}} = \frac{N_{\text{Correct}}}{N_{\text{Answerable}}}, \tag{5}$$

Graded Accuracy.: Partial credit given to *Partially Correct* responses:

$$\mathrm{Acc}_{\text{Graded}} = \frac{N_{\text{Correct}} + 0.5 \times N_{\text{Partial}}}{N_{\text{Answerable}}}. \tag{6}$$

Only examples not marked *Unclear* by adjudication are included in $N_{\text{Answerable}}$.

Inter-Annotator Agreement. To quantify rating consistency, we compute Cohen's κ between the two primary annotators. The adjudicator's label is used only when disagreement occurs and is excluded from agreement computation. Results are shown in Table 14.

Sample Size Justification. We annotate 30 rollouts per model–environment pair. Assuming a standard deviation of $\sigma \approx 0.2$ and a 95% confidence level, the confidence interval (CI) width is given by CI Width $= z_{\frac{1-C}{2}} \cdot \frac{\sigma}{\sqrt{n}}$. This yields an estimated CI of ∼7.1% for individual model–environment pairs (n = 30), and ∼2.5% when aggregating across all eight pairs (n = 240), offering sufficient precision for comparative evaluation.

Table 14: Inter-annotator agreement and valid QA coverage across environments. We report Cohen's κ between the two primary annotators for each world model–map pair. The total number of valid examples excludes QA pairs marked as *Unclear* by at least one annotator.

Model	Env.	Valid QA Pairs	Cohen's κ
WHAM 140M	A	24	0.79
WHAM 1.6B	A	29	0.91
	B	28	0.67
	C	28	0.74
	D	29	0.87
	E	30	0.67
	F	29	0.61
	G	30	0.59

Table 15: Graded/strict accuracy of UNIVERSE on Action and Character Recognition tasks, evaluated by human annotators across different environments and question formats. We report results for Binary, Multiple-Choice (MC), and Open-Ended (OE) prompts, disaggregated by task and world model. All metrics are based on final adjudicated ratings.

		Action Recognition			Character Recognition		
Model	Env.	Binary	MC	OE	Binary	MC	OE
WHAM 140M	A	92.9 / 89.3	35.7 / 32.1	41.1 / 39.3	85.7 / 85.7	10.7 / 10.7	60.7 / 60.7
WHAM 1.6B	A	98.3 / 96.7	51.7 / 46.7	75.0 / 73.3	93.3 / 93.3	83.3 / 83.3	93.3 / 93.3
	B	96.7 / 96.7	60.0 / 60.0	65.0 / 60.0	99.9 / 99.9	90.0 / 90.0	93.3 / 93.3
	C	96.7 / 96.7	63.3 / 63.3	80.0 / 80.0	99.9 / 99.9	86.7 / 86.7	93.3 / 93.3
	D	93.3 / 93.3	43.3 / 43.3	73.3 / 73.3	96.7 / 96.7	96.7 / 96.7	99.9 / 99.9
	E	80.0 / 76.7	76.7 / 73.3	93.3 / 93.3	96.7 / 96.7	99.9 / 99.9	99.8 / 99.8
	F	71.7 / 70.0	56.7 / 56.7	75.0 / 70.0	96.7 / 96.7	93.3 / 93.3	96.7 / 96.7
	G	68.3 / 66.7	50.0 / 46.7	80.0 / 76.7	93.3 / 93.3	90.0 / 90.0	96.7 / 96.7

F.3 Results

Table 15 reports graded and strict accuracy across environments, recognition targets (Action and Character Recognition), and question formats (Binary, Multiple-Choice, Open-Ended). We observe a clear gap in performance between rollouts generated by the two world models. UNIVERSE struggles with outputs from WHAM 140M, achieving substantially lower accuracy compared to WHAM 1.6B. This is likely due to a mismatch in image resolution: WHAM 140M generates frames at 128×128 resolution, which must be upsampled to the UNIVERSE's expected input of 224×224. Despite resizing, the resulting frames often lack sharpness, making actions and characters harder to recognize. In contrast, UNIVERSE performs well on rollouts from WHAM 1.6B, even across diverse environments. On the in-domain setting (Environment A), the model achieves strong results—averaging 75.02% graded accuracy for AR and 90.00% for CR. When evaluating on the six unseen environments (Environments B–G), performance for AR drops slightly (from 75.02% to 73.52%), while CR remains stable or improves, suggesting strong generalization in character grounding and visual consistency tracking.

Qualitative Examples. Figure 11 illustrates the diversity of generated rollouts across environments. WHAM 1.6B captures greater visual variation and scene composition compared to WHAM 140M.

G Supplementary Experimental Results

This section presents additional experimental results that support the main findings but are omitted from the main paper for clarity and space. These include: (i) a zero-shot analysis of PaliGemma variants to motivate backbone selection, (ii) CLIPScore-based baselines to contextualize performance without adaptation, and (iii) a study of low-rank adaptation (LoRA) across different rank values. While these results are not central to the unified evaluation framework proposed in the main text, they

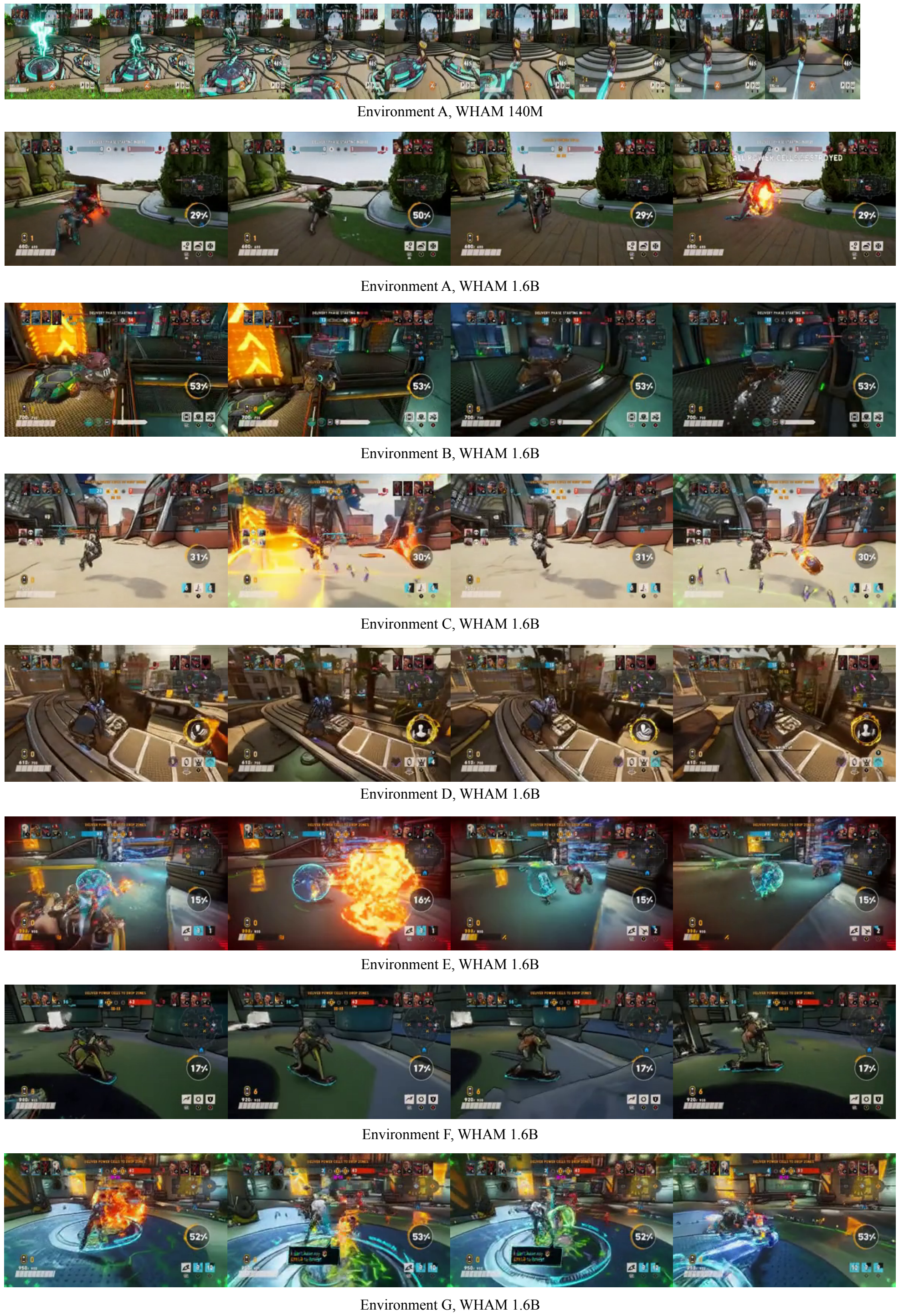

Figure 11: Representative frames from rollouts generated across seven environments. WHAM 140M (top row) was trained only on Skygarden (Environment A); WHAM 1.6B (rows 2-8) generalizes across seven environments.

provide valuable insight into model selection, adaptation efficiency, and the limitations of standard evaluation proxies in our setting.

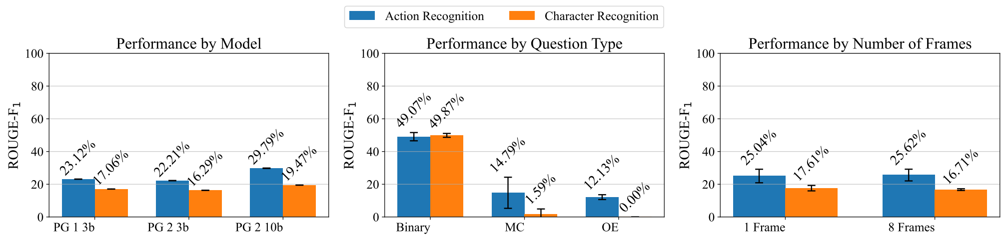

Figure 12: Zero-shot evaluation results for PaliGemma variants across tasks, prompt formats, and visual context sizes. Overall performance remains limited, indicating the need for task-specific adaptation.

Table 16: Zero-shot accuracy-based evaluation of CLIP models and baseline methods on Action and Character Recognition tasks using 1 and 8 input frames.

Fr	Model	Action Recognition	Character Recognition
1	CLIP ViT-B/32	24.04 ± 0.00	13.32 ± 0.00
	CLIP ViT-B/16	52.67 ± 0.00	16.47 ± 0.00
	CLIP ViT-L/14	24.60 ± 0.00	9.95 ± 0.00
	CLIP ViT-L/14-336	12.17 ± 0.00	8.85 ± 0.05
8	CLIP ViT-B/32	36.22 ± 0.00	14.41 ± 0.00
	CLIP ViT-B/16	57.36 ± 0.00	17.24 ± 0.00
	CLIP ViT-L/14	17.57 ± 0.00	10.10 ± 0.00
	CLIP ViT-L/14-336	23.12 ± 0.00	8.64 ± 0.00

G.1 Zero-Shot Performance of PaliGemma Models

In this section, we benchmark three pretrained configurations—PaliGemma 1 3b, PaliGemma 2 3b, and PaliGemma 2 10b—under our proposed protocol and motivate our choice of PaliGemma 2 3b as the default backbone for subsequent experiments. Each model receives a natural language prompt along with either 1 or 8 image frames as input and produces a textual response. This experiment probes both model capacity and the role of temporal visual context in zero-shot settings.

Results. Figure 12 reports ROUGE scores across task types, question formats, and visual context lengths. While zero-shot performance reveals some capacity for structured reasoning—particularly in the multiple-choice setting—it remains limited overall. Binary accuracy hovers near chance, and open-ended responses frequently lack specificity. Performance is strongest on action recognition (AR), likely reflecting pretrained models' familiarity with generic visual dynamics. In contrast, character recognition (CR) lags behind, underscoring a lack of grounding in domain-specific entities. Increasing the number of input frames modestly improves AR, but yields diminishing returns for CR. Among the evaluated configurations, PaliGemma 2 10b performs best in absolute terms. However, the margin over PaliGemma 2 3b is narrow, and PaliGemma 2 3b offers a substantially smaller footprint while using a newer Gemma 2 decoder architecture. We therefore adopt PaliGemma 2 3b as the default model for all subsequent adaptation experiments, balancing performance, compute efficiency, and architectural recency.

G.2 CLIPScore Comparisons

To further evaluate zero-shot recognition capabilities without adaptation, we apply CLIPScore to our rollout evaluation protocol. Specifically, we assess four pretrained CLIP variants – ViT-B/32, ViT-B/16, ViT-L/14, and ViT-L/14-336 – across both Action Recognition (AR) and Character Recognition (CR) tasks using 1-frame and 8-frame visual inputs. For each evaluation instance, we extract either 1 or 8 frames from the video segment and compute the cosine similarity between each image and a predefined set of textual labels (i.e., action verbs for AR, character names for CR). For single-frame settings, we select the label with the highest similarity score as the predicted class. In the multi-frame setting, we compute predictions for each frame independently and use a majority vote to produce the final prediction. We also report two reference baselines for context: a random classifier, which

Table 17: Performance on Action and Character Recognition tasks after LoRA-based adaptation with varying ranks ($r \in \{8, 16, 32, 48, 64\}$). Adapters are applied to attention and MLP layers in both vision and language components.

	Binary		Multiple-choice		Open-ended	
Rank	**EM**	**ROUGE**	**EM**	**ROUGE**	**EM**	**ROUGE**
			Action Recognition			
8	44.66 ± 0.21	44.66 ± 0.21	0.02 ± 0.00	9.21 ± 0.00	0.00 ± 0.00	12.49 ± 0.00
16	44.47 ± 0.43	44.47 ± 0.43	0.02 ± 0.00	9.21 ± 0.00	0.00 ± 0.00	12.49 ± 0.00
32	44.59 ± 0.03	44.59 ± 0.03	0.02 ± 0.00	9.21 ± 0.00	0.00 ± 0.00	12.49 ± 0.00
48	46.71 ± 3.20	46.71 ± 3.20	0.02 ± 0.00	9.21 ± 0.00	0.00 ± 0.00	12.49 ± 0.00
64	48.67 ± 0.13	48.67 ± 0.13	0.02 ± 0.00	9.21 ± 0.00	0.00 ± 0.00	12.49 ± 0.00
			Character Recognition			
8	48.76 ± 0.00	48.76 ± 0.00	0.00 ± 0.00	0.32 ± 0.00	0.00 ± 0.00	0.01 ± 0.01
16	48.62 ± 0.23	48.62 ± 0.23	0.00 ± 0.00	0.14 ± 0.00	0.00 ± 0.00	0.05 ± 0.00
32	48.98 ± 0.08	48.98 ± 0.08	0.00 ± 0.00	0.14 ± 0.00	0.00 ± 0.00	0.05 ± 0.00
48	48.91 ± 0.09	48.91 ± 0.09	0.00 ± 0.00	0.14 ± 0.00	0.00 ± 0.00	0.05 ± 0.00
64	48.72 ± 0.06	48.72 ± 0.06	0.00 ± 0.00	0.14 ± 0.00	0.00 ± 0.00	0.05 ± 0.00

achieves 12.5% on AR and 7.7% on CR, and a majority-class predictor, which yields 35.5% and 17.6% respectively. These are included only for calibration.

Results. Table 16 demonstrates the results. While CLIP ViT-B/16 performs relatively well on AR in both input settings, performance remains inconsistent across model scales and tasks. In particular, CR accuracy remains low, reflecting CLIP's limited grounding in domain-specific visual semantics and fine-grained identity resolution. Larger CLIP models such as ViT-L/14 do not consistently outperform smaller variants, and 8-frame inputs provide only marginal gains over single-frame inputs.

Overall, these results suggest that while CLIPScore offers a lightweight and scalable evaluation proxy, it lacks the temporal grounding and semantic specificity required for structured rollout evaluation. Performance falls short relative to our selected baselines, and the method is inherently constrained to predefined candidate sets—limiting its applicability to open-ended or compositional tasks. As such, we exclude CLIP-based scores from our primary comparisons and instead focus on adapted, generative VLM-based evaluators.

G.3 Low-Rank Adaptation Comparisons

This section presents an extended analysis of low-rank adaptation (LoRA) as a parameter-efficient strategy for adapting vision-language models to our protocol. We systematically vary the rank parameter r and measure its impact on Action and Character Recognition performance across all prompt formats. All experiments in this section are conducted using PaliGemma 2 (3B) as the backbone model, consistent with the main fine-tuning results. These experiments assess whether increasing rank provides meaningful gains, and inform our decision to report only the rank-8 setting in the main paper.

Results. Table 17 presents the performance of LoRA-based adaptation across a range of rank values ($r \in \{8, 16, 32, 48, 64\}$) for both Action Recognition (AR) and Character Recognition (CR) tasks, across all prompt formats. We report exact match (EM) and ROUGE-F_1 averaged over three runs. Increasing the rank beyond $r = 8$ yields no consistent improvements across tasks or formats. Performance on binary prompts remains close to random, while performance on multiple-choice and open-ended formats stays near zero across all ranks. These results suggest that LoRA, even with increased capacity, is insufficient for capturing the fine-grained temporal and semantic dependencies required by our evaluation protocol. Given the lack of benefit from increasing rank—and the added parameter cost—it is inefficient to scale LoRA rank beyond $r = 8$. Accordingly, all results reported in the main paper use $r = 8$, while extended comparisons with higher ranks are presented here for completeness.