Adapting Vision-Language Models for Evaluating World Models

Mariya Hendriksen [1] **Tabish Rashid** [1] **David Bignell** [1] **Raluca Georgescu** [1] **Abdelhak Lemkhenter** [1]
Katja Hofmann [1] **Sam Devlin** [*1] **Sarah Parisot** [*1]

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. The resulting unified evaluator matches the performance of task-specific baselines using a single checkpoint. Alignment with human judgments is additionally explored in an accompanying study, 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 (Ha & Schmidhuber, 2018; Hafner et al., 2025; Alonso et al., 2024). 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 (Kanervisto et al., 2025; Guo et al., 2025; Gao et al., 2025; Chen et al., 2025), embodied AI (Du et al.; Yang et al., 2024), and autonomous driving (Russell et al., 2025; Hu et al., 2023a; Ni et al., 2025). As their capabilities grow, a persistent challenge remains: evaluation.

Rollouts from world models are visually rich, temporally grounded, and semantically structured, requiring evaluation protocols that assess both (i) timestamp-level alignment with control inputs (Yang et al., 2024) and (ii) consistency of entities over time (Kanervisto et al., 2025). Existing metrics fall short: distributional metrics target static images (Salimans et al., 2016; Heusel et al., 2017), video metrics lack semantic grounding (Unterthiner et al., 2018), and multimodal metrics ignore action conditioning (Jayasumana et al., 2024). While human evaluation (Agarwal et al., 2025; Analysis, 2024) remains reliable, it is costly; recent T2V benchmarks (Liu et al., 2024b; Huang et al., 2024; Liao et al., 2024) emphasize open-ended generation and neglect fine-grained temporal control.

Vision-Language Models (VLMs) generalize well across multimodal tasks (Li et al., 2023; Driess et al., 2023; Chen et al., 2023; Wang et al., 2024; Abdin et al., 2024; Liu et al., 2024a; Deitke et al., 2024; McKinzie et al., 2024), and are increasingly used to evaluate generative models (Lee et al., 2024; Mañas et al., 2024; Lin et al., 2024; Chen et al., 2024). We extend this direction by exploring VLMs as fine-grained evaluators of world model rollouts—capturing semantic consistency, temporal coherence, and control alignment. Virtual environments provide ideal conditions for this, exposing timestamped actions and object states. This setting demands precise temporal grounding, action sensitivity, and multi-frame reasoning under limited supervision. Off-the-shelf VLMs struggle in this regime (see Section 4, *Zero-Shot Evaluation*).

We propose a structured protocol targeting two axes: *action alignment* and *character consistency*, formalized as two recognition tasks—Action Recognition (AR) and Character Recognition (CR)—evaluated across binary, multiple-choice, and open-ended formats. To support this, we introduce UNIVERSE (UNIfied Vision-language Evalua-

[*]Equal contribution [1]Microsoft Research, Cambridge, United Kingdom. Correspondence to: Mariya Hendriksen <mariya.hendriksen@gmail.com>.

ICML 2025 Workshop on Assessing World Models. Copyright 2025 by the author(s).

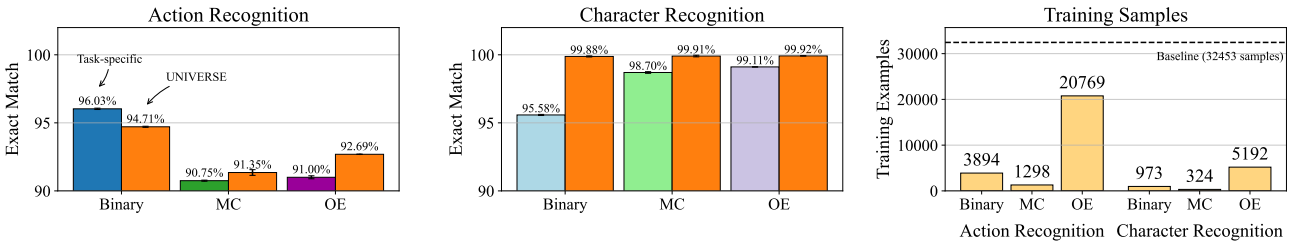


Figure 1: Performance and efficiency of UNIVERSE (orange) vs. task-specific baselines (various colors). **Left, Center:** Accuracy on Action and Character Recognition across binary, multiple-choice, and open-ended formats. **Right:** Sample efficiency—UNIVERSE achieves strong performance with far fewer training samples per epoch. A single checkpoint matches task-specific models, enabled by mixed supervision, efficient frame sampling, and lightweight fine-tuning.

tor for Rollouts in Simulated Environments), a method for adapting VLMs to structured rollout evaluation. UNIVERSE results from a systematic study of adaptation strategies, analyzing supervision regime, frame sampling, context length, and training budget. The final recipe combines mixed supervision, efficient frame selection, and lightweight fine-tuning. We validate UNIVERSE on WHAM (Kanervisto et al., 2025), finding strong agreement with human judgments and demonstrating its effectiveness as an automated evaluator.

2. Related Work

Challenges in Evaluating World Models. World models learn predictive representations of environment dynamics (Ha & Schmidhuber, 2018), and have become central to domains such as game engines (Kanervisto et al., 2025; Guo et al., 2025), embodied AI (Du et al.), and autonomous driving (Russell et al., 2025). Recent models like Dreamer (Hafner et al., 2025), MuZero (Schrittwieser et al., 2020), IRIS (Micheli et al., 2023), and DIAMOND (Alonso et al., 2024) have improved rollout fidelity and control. However, evaluation remains limited. Most approaches rely on downstream metrics such as game score or task success (Bellemare et al., 2013; Kaiser et al., 2020), which provide indirect signals of rollout quality. While Genie (Bruce et al., 2024) decouples agent training, it emphasizes perceptual quality over semantic fidelity. Structured protocols such as Cosmos (Agarwal et al., 2025) combine distributional metrics, 3D consistency, and human ratings, but remain simulator-specific and labor-intensive. Human-in-the-loop pipelines like the Video Generation Arena (Analysis, 2024) are similarly difficult to scale.

Evaluation Metrics and Protocols for Visual Generation. Early evaluations relied on full-reference metrics like PSNR and SSIM (Wang et al., 2004), which capture low-level fidelity but are sensitive to spatial shifts and lack semantic grounding. Distributional metrics such as IS (Salimans et al., 2016), FID (Heusel et al., 2017), KID (Binkowski et al., 2018), and PPL (Karras et al., 2019) focus on perceptual realism, but are limited to static images. For video, FVD (Unterthiner et al., 2018) incorporates motion features via I3D (Carreira & Zisserman, 2017), yet lacks causal or semantic structure. CLIP-based metrics (Hessel et al., 2021; Wu et al., 2021; Jayasumana et al., 2024) improve semantic alignment, but operate at the frame level and do not account for control inputs. Structured evaluation protocols using VLMs have emerged: VQA Accuracy (Mañas et al., 2024), VQAScore (Lin et al., 2024), and Prometheus (Lee et al., 2024) introduce query-based evaluations but remain limited to single-frame settings. Recent T2V benchmarks—EvalCrafter (Liu et al., 2024b), VBench (Huang et al., 2024), and DEVIL (Liao et al., 2024)—broaden coverage to text alignment, motion, and consistency, yet lack timestamp-level grounding and action-conditioned evaluation.

Vision-Language Model Adaptation. VLMs have shown strong performance across multimodal tasks including captioning, retrieval, and instruction following (Li et al., 2023; Driess et al., 2023; Chen et al., 2023; Wang et al., 2024; Abdin et al., 2024; Liu et al., 2024a; Deitke et al., 2024; McKinzie et al., 2024). Adaptation strategies fall into two categories: prompt-level and weight-level. Prompt-level methods include prompt tuning (Miyai et al., 2023; Zhou et al., 2024), in-context learning (Brown et al., 2020; Alayrac et al., 2022), and retrieval-augmented generation (Lewis et al., 2020; Hu et al., 2023b), but these approaches struggle with temporal alignment and structured rollouts. Weight-level adaptation enables stronger domain alignment. Full fine-tuning is effective but costly; partial (Ye et al., 2023) and parameter-efficient strategies offer scalable alternatives. Low-rank methods such as LoRA (Hu et al., 2022), DoRA (Liu et al.), and QLoRA (Dettmers et al., 2023) enable efficient updates, while adapter-based methods insert lightweight modules into frozen networks (Luo et al., 2023; Zhao et al., 2024). Few-shot multimodal learning has also emerged as a lightweight alternative (Tsimpoukelli et al., 2021; Jin et al., 2022; Najdenkoska et al., 2023), though it remains underexplored for structured, temporally grounded

tasks.

3. Methodology

We study the problem of evaluating rollouts generated by *world models*— generative models trained to predict future observations o_t from past observations and actions, $(o_{<t}, a_{<t}) \rightarrow o_t$, where $o_t \in \mathcal{O}$ is typically an RGB frame. These rollouts are temporally grounded and causally structured, requiring fine-grained, timestamp-level evaluation.

We propose UNIVERSE, an adapted Vision-Language Model (VLM) that serves as a structured evaluator for world model outputs. It operates as a function $E : (V, Q) \rightarrow \hat{A}$, where $V = (o_{t_1}, \ldots, o_{t_k})$ is a sampled frame sequence, $Q \in \mathcal{L}$ is a natural language question, and $\hat{A} \in \mathcal{L}$ is the predicted answer.

Evaluation Protocol. We define two recognition tasks: (i) *Action Recognition (AR)*, which checks whether generated sequences reflect agent actions at each timestep; and (ii) *Character Recognition (CR)*, which evaluates whether entities maintain consistent identity over time. Each task is cast as a visual QA problem: given a sequence of frames and a natural language prompt (binary, multiple-choice, or open-ended), the model generates a textual response. Predictions are scored using Exact Match (EM) and ROUGE-F_1 to capture both literal and semantic alignment (Appendix E.1).

Dataset Construction. Effective VLM adaptation for rollout evaluation requires data that (i) captures realistic human behavior in interactive environments, and (ii) aligns with prior work in simulated settings to support comparability and reproducibility. Since WHAM, the world model used in our evaluation—is trained on *Bleeding Edge*, it is essential that the adaptation dataset matches its distribution. To meet these criteria, we partnered with Ninja Theory to curate a dataset from internal and public *Bleeding Edge* gameplay, focusing on the *Skygarden* map used in WHAM (Kanervisto et al., 2025). The dataset offers high visual and behavioral diversity (Pearce et al., 2025), includes a publicly available evaluation split, and aligns closely with prior work (Kanervisto et al., 2025; Pearce et al., 2025; Tot et al., 2025; Sharma et al., 2024; Devlin et al., 2021), 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) *QA Construction:* Generate six QA pairs per clip (binary, multiple-choice, and open-ended) spanning both AR and CR tasks. The final dataset includes 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 (Beyer et al., 2024; Steiner et al., 2024), comprising 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, featuring a 2B-parameter Gemma 2 decoder pretrained on 2T tokens.

Input frames are resized to 224×224 and tokenized into 256 patches each. Each model input sequence $S = \{S_{\mathcal{I}}, S_{\mathcal{T}}^{\text{PREF}}, S_{\mathcal{T}}^{\text{SUFF}}\}$ includes visual tokens $S_{\mathcal{I}}$ from k frames, a textual prefix $S_{\mathcal{T}}^{\text{PREF}}$ with the task-language cue and question, and a suffix $S_{\mathcal{T}}^{\text{SUFF}}$ containing the expected answer (training only). This format enables the decoder to attend jointly over visual and textual context. Architecture and prompt details are in Appendix E.2.

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 three axes of adaptation for pretrained VLMs: *fine-tuning configuration*, *frame sampling*, and *supervision composition*.

Fine-Tuning Configurations. We compare five strategies varying in trainable parameter count: (i) *Zero-shot prompting:* No tuning. (ii) *Full fine-tuning:* All parameters $\theta_V \cup \theta_P \cup \theta_L$. (iii) *Dual- and single-component:* Tune two or one module(s), e.g., $\theta_P \cup \theta_L$ or θ_P only. (iv) *LoRA tuning:* Apply low-rank adapters to attention and MLP layers (Hu et al., 2022) with $r \in \{8, 16, 32, 48, 64\}$, $\alpha = 8$.

Frame Sampling Policy. We sweep over input lengths $k \in [1, 8]$ and compare: (i) *First-n*: Select the first k frames; (ii) *Uniform-n*: Sample k frames uniformly across the clip.

Supervision Composition. We tune a hierarchical QA mixture across tasks (AR/CR) and formats (binary, MC, open-ended) via a three-stage grid search: (i) Task weighting: $\alpha_{\texttt{AR}}, \alpha_{\texttt{CR}}$; (ii) Format weighting: $\beta_{\texttt{OE}}, \beta_{\texttt{binary}}, \beta_{\texttt{MC}}$; (iii) Ratio optimization for open-ended generalization and format balance.

UNIVERSE: UNIfied Vision-language Evaluator for Rollouts in Simulated Environments. We consolidate these findings into UNIVERSE, a compact and scalable method for adapting VLMs to structured rollout evaluation. UNIVERSE balances efficiency and generalization using a single partially tuned model across all QA formats and tasks. It integrates three core components:

(I) *Partial fine-tuning:* Only the projection head (θ_P)

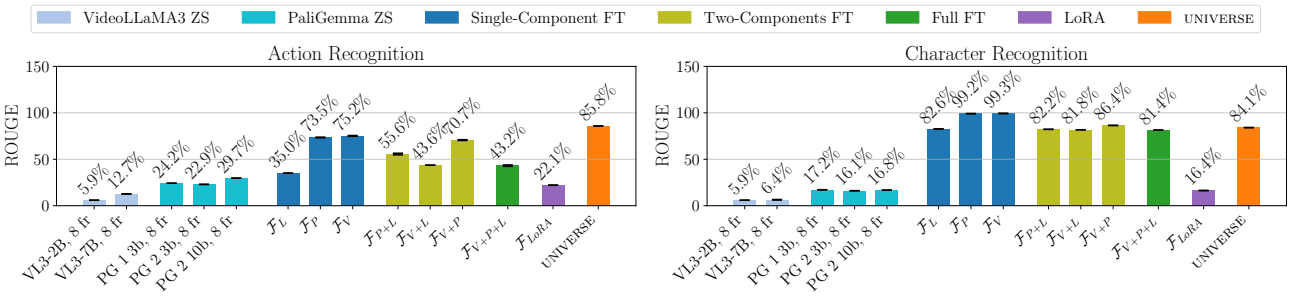


Figure 2: Comparison of UNIVERSE and baseline models on Action and Character Recognition. **Left**: UNIVERSE outperforms all baselines on AR. **Right**: On CR, it ranks third, behind models with either full vision encoder tuning or task-specific training with greater supervision. Trained under a unified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

is updated—just 0.07% of parameters—achieving second-best performance overall, behind vision encoder tuning (∼11%).

(II) *Efficient frame sampling:* Inputs include $k = 8$ frames sampled uniformly from a 14-frame clip, balancing temporal coverage and token efficiency.

(III) *Mixed supervision:* We prioritize AR ($\alpha_{\text{AR}} = 0.8$) and open-ended QA ($\beta_{\text{OE}} = 0.8$), with binary (0.15) and MC (0.05) for balance and stability.

4. Experiments

Baselines. We compare UNIVERSE to two baseline classes: (i) *Zero-shot VLMs:* Seven off-the-shelf models, including VideoLLaMA3 (2B, 7B) (Boqiang Zhang, 2025) and PaliGemma v1 (3B) and v2 (3B, 10B) (Beyer et al., 2024; Steiner et al., 2024), evaluated without adaptation using 8-frame inputs.[1] (ii) *Fine-tuned PaliGemma 2:* Variants adapted via full, partial, and parameter-efficient tuning, selected based on a zero-shot performance sweep (Appendix G.1). The adaptation space includes 8 baselines: (i) *Single-component tuning:* vision encoder ($\mathcal{F}_{\text{V}}$), projector ($\mathcal{F}_{\text{P}}$), or language head ($\mathcal{F}_{\text{L}}$); (ii) *Two-component tuning:* $\mathcal{F}_{\text{V+P}}$, $\mathcal{F}_{\text{V+L}}$, and $\mathcal{F}_{\text{P+L}}$; (iii) *Full-model tuning:* all components ($\mathcal{F}_{\text{V+P+L}}$); (iv) *LoRA-based tuning:* parameter-efficient adaptation with rank $r = 8$ (Appendix G.3). All models are trained on 8-frame clips for a single epoch.

Results. Figure 2 (left, center) shows performance on Action Recognition (AR) and Character Recognition (CR). Zero-shot models perform poorly: VideoLLaMA3 scores below 12.7% on AR and 6.4% on CR; PaliGemma variants reach up to 29.7% (AR) and 17.2% (CR), confirming that general-purpose VLMs lack the temporal grounding and domain-specific understanding needed for structured rollout evaluation. In contrast, UNIVERSE outperforms all models on AR and ranks third on CR. The top CR baselines either fine-tune the full vision encoder (∼400M parameters) or use 5× more CR supervision, with each model trained in isolation for a single task and format. UNIVERSE, by comparison, fine-tunes only the 2.66M-parameter projector (0.07% of the model) using a unified setup across both tasks, all prompt formats, and reduced supervision. These results highlight the efficiency and generality of our adaptation strategy for temporally grounded evaluation.

[1] CLIPScore-based results (Appendix G.2) underperformed and were limited to candidate sets, reinforcing the need for adaptation.

5. Conclusion

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

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.

Impact Statement

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 (Kanervisto et al., 2025; Guo et al., 2025; Gao et al., 2025; Chen et al., 2025), embodied AI (Du et al.; Yang et al., 2024), and autonomous driving (Russell et al., 2025; Hu et al., 2023a; Ni et al., 2025), 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.

Acknowledgements

We thank the Game Intelligence team at Microsoft Research, including Chentian Jiang, Linda Wen, Lukas Schäfer, Sergio Valcarcel Macua, Shanzheng Tan, Tim Pearce, and Yuhan Cao for helpful feedback and discussions. We thank Ninja Theory for the collaboration on unlocking the Bleeding Edge data for this research. We thank Oleg Losinets and the whole Microsoft Research Grand Central Resources team for infrastructure support. We are grateful to all participants who joined our user study. We are grateful to Doug Burger and the Microsoft Research Redmond GPU council for GPU resources.

References

Abdin, M. I., Jacobs, S. A., Awan, A. A., Aneja, J., Awadallah, A., Awadalla, H., Bach, N., Bahree, A., Bakhtiari, A., Behl, H. S., Benhaim, A., Bilenko, M., Bjorck, J., Bubeck, S., Cai, M., Mendes, C. C. T., Chen, W., Chaudhary, V., Chopra, P., Giorno, A. D., de Rosa, G., Dixon, M., Eldan, R., Iter, D., Garg, A., Goswami, A., Gunasekar, S., Haider, E., Hao, J., Hewett, R. J., Huynh, J., Javaheripi, M., Jin, X., Kauffmann, P., Karampatziakis, N., Kim, D., Khademi, M., Kurilenko, L., Lee, J. R., Lee, Y. T., Li, Y., Liang, C., Liu, W., Lin, E., Lin, Z., Madan, P., Mitra, A., Modi, H., Nguyen, A., Norick, B., Patra, B., Perez-Becker, D., Portet, T., Pryzant, R., Qin, H., Radmilac, M., Rosset, C., Roy, S., Ruwase, O., Saarikivi, O., Saied, A., Salim, A., Santacroce, M., Shah, S., Shang, N., Sharma, H., Song, X., Tanaka, M., Wang, X., Ward, R., Wang, G., Witte, P., Wyatt, M., Xu, C., Xu, J., Yadav, S., Yang, F., Yang, Z., Yu, D., Zhang, C., Zhang, C., Zhang, J., Zhang, L. L., Zhang, Y., Zhang, Y., Zhang, Y., and Zhou, X. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.

Agarwal, N., Ali, A., Bala, M., Balaji, Y., Barker, E., Cai, T., Chattopadhyay, P., Chen, Y., Cui, Y., Ding, Y., Dworakowski, D., Fan, J., Fenzi, M., Ferroni, F., Fidler, S., Fox, D., Ge, S., Ge, Y., Gu, J., Gururani, S., He, E., Huang, J., Huffman, J. S., Jannaty, P., Jin, J., Kim, S. W., Klár, G., Lam, G., Lan, S., Leal-Taixé, L., Li, A., Li, Z., Lin, C., Lin, T., Ling, H., Liu, M., Liu, X., Luo, A., Ma, Q., Mao, H., Mo, K., Mousavian, A., Nah, S., Niverty, S., Page, D., Paschalidou, D., Patel, Z., Pavao, L., Ramezanali, M., Reda, F., Ren, X., Sabavat, V. R. N., Schmerling, E., Shi, S., Stefaniak, B., Tang, S., Tchapmi, L., Tredak, P., Tseng, W., Varghese, J., Wang, H., Wang, H., Wang, H., Wang, T., Wei, F., Wei, X., Wu, J. Z., Xu, J., Yang, W., Yen-Chen, L., Zeng, X., Zeng, Y., Zhang, J., Zhang, Q., Zhang, Y., Zhao, Q., and Zólkowski, A. Cosmos world foundation model platform for physical AI. *arXiv preprint arXiv:2501.03575*, 2025.

Alabdulmohsin, I. M., Zhai, X., Kolesnikov, A., and Beyer, L. Getting vit in shape: Scaling laws for compute-optimal model design. *Advances in Neural Information Processing Systems*, 36:16406–16425, 2023.

Alayrac, J., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., Ring, R., Rutherford, E., Cabi, S., Han, T., Gong, Z., Samangooei, S., Monteiro, M., Menick, J. L., Borgeaud, S., Brock, A., Nematzadeh, A., Sharifzadeh, S., Binkowski, M., Barreira, R., Vinyals, O., Zisserman, A., and Simonyan, K. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

Alonso, E., Jelley, A., Micheli, V., Kanervisto, A., Storkey, A. J., Pearce, T., and Fleuret, F. Diffusion for world modeling: Visual details matter in atari. *Advances in Neural Information Processing Systems*, 37:58757–58791, 2024.

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

Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. The arcade learning environment: An evaluation platform for general agents. *Journal of artificial intelligence research*, 47:253–279, 2013.

Beyer, L., Steiner, A., Pinto, A. S., Kolesnikov, A., Wang, X., Salz, D., Neumann, M., Alabdulmohsin, I., Tschannen, M., Bugliarello, E., Unterthiner, T., Keysers, D., Koppula, S., Liu, F., Grycner, A., Gritsenko, A. A., Houlsby, N., Kumar, M., Rong, K., Eisenschlos, J., Kabra, R., Bauer, M., Bosnjak, M., Chen, X., Minderer, M., Voigtlaender, P., Bica, I., Balazevic, I., Puigcerver, J., Papalampidi, P., Hénaff, O. J., Xiong, X., Soricut, R., Harmsen, J., and Zhai, X. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024.

Binkowski, M., Sutherland, D. J., Arbel, M., and Gretton, A. Demystifying MMD gans. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

Bird, S. and Loper, E. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pp. 214–217, Barcelona, Spain, July 2004. Association for Computational Linguistics.

Boqiang Zhang, Kehan Li, Z. C. Z. H. Y. Y. G. C. S. L. Y. J. H. Z. X. L. P. J. W. Z. F. W. L. B. D. Z. Videollama 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*, 2025. URL https://arxiv.org/abs/2501.13106.

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

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Bruce, J., Dennis, M. D., Edwards, A., Parker-Holder, J., Shi, Y., Hughes, E., Lai, M., Mavalankar, A., Steigerwald, R., Apps, C., Aytar, Y., Bechtle, S., Behbahani, F. M. P., Chan, S. C. Y., Heess, N., Gonzalez, L., Osindero, S., Ozair, S., Reed, S. E., Zhang, J., Zolna, K., Clune, J., de Freitas, N., Singh, S., and Rocktäschel, T. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

Carreira, J. and Zisserman, A. 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*, pp. 4724–4733. IEEE Computer Society, 2017.

Changpinyo, S., Kukliansky, D., Szpektor, I., Chen, X., Ding, N., and Soricut, R. All you may need for VQA are image captions. *arXiv preprint arXiv:2205.01883*, 2022.

Chen, D., Chen, R., Zhang, S., Wang, Y., Liu, Y., Zhou, H., Zhang, Q., Wan, Y., Zhou, P., and Sun, L. Mllm-as-a-judge: Assessing multimodal llm-as-a-judge with vision-language benchmark. In *Forty-first International Conference on Machine Learning*, 2024.

Chen, J., Zhao, Y., Huang, Y., Cui, L., Dong, L., Lv, T., Chen, Q., and Wei, F. Model as a game: On numerical and spatial consistency for generative games. *arXiv preprint arXiv:2503.21172*, 2025.

Chen, X., Wang, X., Changpinyo, S., Piergiovanni, A. J., Padlewski, P., Salz, D., Goodman, S., Grycner, A., Mustafa, B., Beyer, L., Kolesnikov, A., Puigcerver, J., Ding, N., Rong, K., Akbari, H., Mishra, G., Xue, L., Thapliyal, A. V., Bradbury, J., and Kuo, W. Pali: A jointly-scaled multilingual language-image model. *arXiv preprint arXiv:2209.06794*, 2022.

Chen, X., Djolonga, J., Padlewski, P., Mustafa, B., Changpinyo, S., Wu, J., Ruiz, C. R., Goodman, S., Wang, X., Tay, Y., Shakeri, S., Dehghani, M., Salz, D., Lucic, M., Tschannen, M., Nagrani, A., Hu, H., Joshi, M., Pang, B., Montgomery, C., Pietrzyk, P., Ritter, M., Piergiovanni, A. J., Minderer, M., Pavetic, F., Waters, A., Li, G., Alabdulmohsin, I., Beyer, L., Amelot, J., Lee, K., Steiner, A. P., Li, Y., Keysers, D., Arnab, A., Xu, Y., Rong, K., Kolesnikov, A., Seyedhosseini, M., Angelova, A., Zhai, X., Houlsby, N., and Soricut, R. Pali-x: On scaling up a multilingual vision and language model. *arXiv preprint arXiv:2305.18565*, 2023.

Deitke, M., Clark, C., Lee, S., Tripathi, R., Yang, Y., Park, J. S., Salehi, M., Muennighoff, N., Lo, K., Soldaini, L., Lu, J., Anderson, T., Bransom, E., Ehsani, K., Ngo, H., Chen, Y., Patel, A., Yatskar, M., Callison-Burch, C., Head, A., Hendrix, R., Bastani, F., VanderBilt, E., Lambert, N., Chou, Y., Chheda, A., Sparks, J., Skjonsberg, S., Schmitz, M., Sarnat, A., Bischoff, B., Walsh, P., Newell, C., Wolters, P., Gupta, T., Zeng, K., Borchardt, J., Groeneveld, D., Dumas, J., Nam, C., Lebrecht, S., Wittlif, C.,

Schoenick, C., Michel, O., Krishna, R., Weihs, L., Smith, N. A., Hajishirzi, H., Girshick, R. B., Farhadi, A., and Kembhavi, A. Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models. *arXiv preprint arXiv:2409.17146*, 2024.

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115, 2023.

Devlin, S., Georgescu, R., Momennejad, I., Rzepecki, J., Zuniga, E., Costello, G., Leroy, G., Shaw, A., and Hofmann, K. Navigation turing test (ntt): Learning to evaluate human-like navigation. In *International Conference on Machine Learning*, pp. 2644–2653. PMLR, 2021.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. 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.

Driess, D., Xia, F., Sajjadi, M. S. M., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q., Yu, T., Huang, W., Chebotar, Y., Sermanet, P., Duckworth, D., Levine, S., Vanhoucke, V., Hausman, K., Toussaint, M., Greff, K., Zeng, A., Mordatch, I., and Florence, P. PaLM-E: An embodied multimodal language model. In *International Conference on Machine Learning*, pp. 8469–8488. PMLR, 2023.

Du, Y., Yang, S., Florence, P., Xia, F., Wahid, A., Ichter, B., Sermanet, P., Yu, T., Abbeel, P., Tenenbaum, J. B., Kaelbling, L. P., Zeng, A., and Tompson, J. Video language planning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*.

Esser, P., Rombach, R., and Ommer, B. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12873–12883, 2021.

Gao, S., Zhou, S., Du, Y., Zhang, J., and Gan, C. Adaworld: Learning adaptable world models with latent actions. *arXiv preprint arXiv:2503.18938*, 2025.

Guo, J., Ye, Y., He, T., Wu, H., Jiang, Y., Pearce, T., and Bian, J. Mineworld: a real-time and open-source interactive world model on minecraft. *arXiv preprint arXiv:2504.08388*, 2025.

Ha, D. and Schmidhuber, J. Recurrent world models facilitate policy evolution. *Advances in Neural Information Processing Systems*, 31:2451–2463, 2018.

Hafner, D., Pasukonis, J., Ba, J., and Lillicrap, T. Mastering diverse control tasks through world models. *Nature*, pp. 1–7, 2025.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

Hendrycks, D. and Gimpel, K. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Hessel, J., Holtzman, A., Forbes, M., Le Bras, R., and Choi, Y. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7514–7528, 2021.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 6626–6637, 2017.

Hu, A., Russell, L., Yeo, H., Murez, Z., Fedoseev, G., Kendall, A., Shotton, J., and Corrado, G. Gaia-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023a.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. 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.

Hu, Z., Iscen, A., Sun, C., Wang, Z., Chang, K.-W., Sun, Y., Schmid, C., Ross, D. A., and Fathi, A. 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*, pp. 23369–23379, 2023b.

Huang, Z., He, Y., Yu, J., Zhang, F., Si, C., Jiang, Y., Zhang, Y., Wu, T., Jin, Q., Chanpaisit, N., et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21807–21818, 2024.

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

Jayasumana, S., Ramalingam, S., Veit, A., Glasner, D., Chakrabarti, A., and Kumar, S. 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*, pp. 9307–9315. IEEE, 2024.

Jin, W., Cheng, Y., Shen, Y., Chen, W., and Ren, X. 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)*, pp. 2763–2775, 2022.

Kaiser, L., Babaeizadeh, M., Milos, P., Osinski, B., Campbell, R. H., Czechowski, K., Erhan, D., Finn, C., Kozakowski, P., Levine, S., Mohiuddin, A., Sepassi, R., Tucker, G., and Michalewski, H. Model based reinforcement learning for atari. *ICLR*, 1:2, 2020.

Kanervisto, A., Scheller, C., and Hautamäki, V. Action space shaping in deep reinforcement learning. In *2020 IEEE conference on games (CoG)*, pp. 479–486. IEEE, 2020.

Kanervisto, A., Bignell, D., Wen, L. Y., Grayson, M., Georgescu, R., Macua, S. V., Tan, S. Z., Rashid, T., Pearce, T., Cao, Y., Lemkhenter, A., Jiang, C., Costello, G., Gupta, G., Tot, M., Ishida, S., Gupta, T., Arora, U., White, R. W., Devlin, S., Morrison, C., and Hofmann, K. World and human action models towards gameplay ideation. *Nature*, 638(8051):656–663, 2025.

Karras, T., Laine, S., and Aila, T. 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*, pp. 4401–4410. Computer Vision Foundation / IEEE, 2019.

Kudo, T. and Richardson, J. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Blanco, E. and Lu, W. (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, pp. 66–71. Association for Computational Linguistics, 2018.

Lee, S., Kim, S., Park, S., Kim, G., and Seo, M. Prometheus-vision: Vision-language model as a judge for fine-grained evaluation. In *Findings of the Association for Computational Linguistics ACL 2024*, pp. 11286–11315, 2024.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020.

Li, J., Li, D., Savarese, S., and Hoi, S. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PMLR, 2023.

Liao, M., Ye, Q., Zuo, W., Wan, F., Wang, T., Zhao, Y., Wang, J., Zhang, X., et al. Evaluation of text-to-video generation models: A dynamics perspective. *Advances in Neural Information Processing Systems*, 37:109790–109816, 2024.

Lin, Z., Pathak, D., Li, B., Li, J., Xia, X., Neubig, G., Zhang, P., and Ramanan, D. Evaluating text-to-visual generation with image-to-text generation. In *European Conference on Computer Vision*, pp. 366–384. Springer, 2024.

Liu, H., Li, C., Li, Y., and Lee, Y. J. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26296–26306, 2024a.

Liu, S., Wang, C., Yin, H., Molchanov, P., Wang, Y. F., Cheng, K., and Chen, M. Dora: Weight-decomposed low-rank adaptation. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*.

Liu, Y., Cun, X., Liu, X., Wang, X., Zhang, Y., Chen, H., Liu, Y., Zeng, T., Chan, R., and Shan, Y. Evalcrafter: Benchmarking and evaluating large video generation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22139–22149, 2024b.

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

Loshchilov, I. and Hutter, F. Sgdr: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2022.

Luo, G., Zhou, Y., Ren, T., Chen, S., Sun, X., and Ji, R. Cheap and quick: Efficient vision-language instruction tuning for large language models. *Advances in neural information processing systems (NeurIPS)*, 2023.

Mañas, O., Krojer, B., and Agrawal, A. Improving automatic vqa evaluation using large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 4171–4179, 2024.

Mangrulkar, S., Gugger, S., Debut, L., Belkada, Y., Paul, S., and Bossan, B. Peft: State-of-the-art parameter-efficient fine-tuning methods, 2022. URL https://github.com/huggingface/peft.

McKinzie, B., Gan, Z., Fauconnier, J., Dodge, S., Zhang, B., Dufter, P., Shah, D., Du, X., Peng, F., Weers, F., Belyi, A., Zhang, H., Singh, K., Kang, D., Hè, H., Schwarzer, M., Gunter, T., Kong, X., Zhang, A., Wang, J., Wang, C., Du, N., Lei, T., Wiseman, S., Lee, M., Wang, Z., Pang, R., Grasch, P., Toshev, A., and Yang, Y. MM1: Methods, analysis and insights from multimodal LLM pre-training. In *European Conference on Computer Vision*, pp. 304–323. Springer, 2024.

Mesnard, T., Hardin, C., Dadashi, R., Bhupatiraju, S., Pathak, S., Sifre, L., Rivière, M., Kale, M. S., Love, J., Tafti, P., Hussenot, L., Chowdhery, A., Roberts, A., Barua, A., Botev, A., Castro-Ros, A., Slone, A., Héliou, A., Tacchetti, A., Bulanova, A., Paterson, A., Tsai, B., Shahriari, B., Lan, C. L., Choquette-Choo, C. A., Crepy, C., Cer, D., Ippolito, D., Reid, D., Buchatskaya, E., Ni, E., Noland, E., Yan, G., Tucker, G., Muraru, G., Rozhdestvenskiy, G., Michalewski, H., Tenney, I., Grishchenko, I., Austin, J., Keeling, J., Labanowski, J., Lespiau, J., Stanway, J., Brennan, J., Chen, J., Ferret, J., Chiu, J., and et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

Micheli, V., Alonso, E., and Fleuret, F. Transformers are sample-efficient world models. In *The Eleventh International Conference on Learning Representations*, 2023.

Miyai, A., Yu, Q., Irie, G., and Aizawa, K. Locoop: Few-shot out-of-distribution detection via prompt learning. *Advances in Neural Information Processing Systems*, 36: 76298–76310, 2023.

Najdenkoska, I., Zhen, X., and Worring, M. 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.

Ni, J., Guo, Y., Liu, Y., Chen, R., Lu, L., and Wu, Z. Maskgwm: A generalizable driving world model with video mask reconstruction. *arXiv preprint arXiv:2502.11663*, 2025.

Pearce, T., Rashid, T., Bignell, D., Georgescu, R., Devlin, S., and Hofmann, K. Scaling laws for pre-training agents and world models. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025.

Piergiovanni, A., Kuo, W., and Angelova, A. Pre-training image-language transformers for open-vocabulary tasks. *arXiv preprint arXiv:2209.04372*, 2022.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision. *Proceedings of the International Conference on Machine Learning (ICML)*, 2021.

Rivière, M., Pathak, S., Sessa, P. G., Hardin, C., Bhupatiraju, S., Hussenot, L., Mesnard, T., Shahriari, B., Ramé, A., Ferret, J., Liu, P., Tafti, P., Friesen, A., Casbon, M., Ramos, S., Kumar, R., Lan, C. L., Jerome, S., Tsitsulin, A., Vieillard, N., Stanczyk, P., Girgin, S., Momchev, N., Hoffman, M., Thakoor, S., Grill, J., Neyshabur, B., Bachem, O., Walton, A., Severyn, A., Parrish, A., Ahmad, A., Hutchison, A., Abdagic, A., Carl, A., Shen, A., Brock, A., Coenen, A., Laforge, A., Paterson, A., Bastian, B., Piot, B., Wu, B., Royal, B., Chen, C., Kumar, C., Perry, C., Welty, C., Choquette-Choo, C. A., Sinopalnikov, D., Weinberger, D., Vijaykumar, D., Rogozinska, D., Herbison, D., Bandy, E., Wang, E., Noland, E., Moreira, E., Senter, E., Eltyshev, E., Visin, F., Rasskin, G., Wei, G., Cameron, G., Martins, G., Hashemi, H., Klimczak-Plucinska, H., Batra, H., Dhand, H., Nardini, I., Mein, J., Zhou, J., Svensson, J., Stanway, J., Chan, J., Zhou, J. P., Carrasqueira, J., Iljazi, J., Becker, J., Fernandez, J., van Amersfoort, J., Gordon, J., Lipschultz, J., Newlan, J., Ji, J., Mohamed, K., Badola, K., Black, K., Millican, K., McDonell, K., Nguyen, K., Sodhia, K., Greene, K., Sjösund, L. L., Usui, L., Sifre, L., Heuermann, L., Lago, L., and McNealus, L. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Russell, L., Hu, A., Bertoni, L., Fedoseev, G., Shotton, J., Arani, E., and Corrado, G. Gaia-2: A controllable multi-view generative world model for autonomous driving. *arXiv preprint arXiv:2503.20523*, 2025.

Salimans, T., Goodfellow, I. J., Zaremba, W., Cheung, V., Radford, A., and Chen, X. Improved techniques for training gans. In Lee, D. D., Sugiyama, M., von Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 2226–2234, 2016.

Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T., Lillicrap, T. P., and Silver, D. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

Sharma, P., Ding, N., Goodman, S., and Soricut, R. 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), pp. 2556–2565, 2018.

Sharma, S., Davidson, G., Khetarpal, K., Kanervisto, A., Arora, U., Hofmann, K., and Momennejad, I. Toward human-ai alignment in large-scale multi-player games. In *ACL 2024 Worldplay Workshop*, 2024.

Srinivasan, K., Raman, K., Chen, J., Bendersky, M., and Najork, M. 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*, pp. 2443–2449, 2021.

Steiner, A., Pinto, A. S., Tschannen, M., Keysers, D., Wang, X., Bitton, Y., Gritsenko, A. A., Minderer, M., Sherbondy, A., Long, S., Qin, S., Ingle, R. R., Bugliarello, E., Kazemzadeh, S., Mesnard, T., Alabdulmohsin, I., Beyer, L., and Zhai, X. Paligemma 2: A family of versatile vlms for transfer. *arXiv preprint arXiv:2412.03555*, 2024.

Tomar, S. Converting video formats with ffmpeg. *Linux Journal*, 2006(146):10, 2006.

Tot, M., Ishida, S., Lemkhenter, A., Bignell, D., Choudhury, P., Lovett, C., França, L., de Mendonça, M. R. F., Gupta, T., Gehring, D., et al. Adapting a world model for trajectory following in a 3d game. In *ICLR 2025 Workshop on World Models*, 2025.

Tsimpoukelli, M., Menick, J. L., Cabi, S., Eslami, S., Vinyals, O., and Hill, F. Multimodal few-shot learning with frozen language models. *Advances in Neural Information Processing Systems*, 34:200–212, 2021.

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

Unterthiner, T., van Steenkiste, S., Kurach, K., Marinier, R., Michalski, M., and Gelly, S. Towards accurate generative models of video: A new metric and challenges. *arXiv preprint arXiv:1812.01717*, 2018.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Wang, P., Bai, S., Tan, S., Wang, S., Fan, Z., Bai, J., Chen, K., Liu, X., Wang, J., Ge, W., Fan, Y., Dang, K., Du, M., Ren, X., Men, R., Liu, D., Zhou, C., Zhou, J., and Lin, J. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Wu, C., Huang, L., Zhang, Q., Li, B., Ji, L., Yang, F., Sapiro, G., and Duan, N. Godiva: Generating open-domain videos from natural descriptions. *arXiv preprint arXiv:2104.14806*, 2021.

Yang, S., Du, Y., Ghasemipour, S. K. S., Tompson, J., Kaelbling, L. P., Schuurmans, D., and Abbeel, P. Learning interactive real-world simulators. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*, 2024.

Ye, P., Huang, Y., Tu, C., Li, M., Chen, T., He, T., and Ouyang, W. Partial fine-tuning: A successor to full finetuning for vision transformers. *CoRR*, 2023.

Zhai, X., Mustafa, B., Kolesnikov, A., and Beyer, L. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11975–11986, 2023.

Zhao, B., Tu, H., Wei, C., Mei, J., and Xie, C. Tuning layernorm in attention: Towards efficient multi-modal llm finetuning. In *ICLR*, 2024.

Zhou, Y., Xia, X., Lin, Z., Han, B., and Liu, T. Few-shot adversarial prompt learning on vision-language models. *Advances in Neural Information Processing Systems*, 37: 3122–3156, 2024.

Appendix

Table of Contents

A. Reproducibility Statement 12

B. UNIVERSE: Implementation Overview 12

C. Dataset 13

D. Experimental Details 15
- D.1 Model 16
- D.2 Evaluation Metrics 15
- D.3 Results 19

E. Human Annotation Study 19
- E.1 Study Design 19
- E.2 Evaluation Metrics 22
- E.3 Results 22

F. Supplementary Experimental Results 23
- F.1 Zero-Shot Performance of PaliGemma Models 24
- F.2 CLIPScore Comparisons 24
- F.3 Low-Rank Adaptation Comparisons 25

A. Evaluating World Model Rollouts with UNIVERSE

We evaluate UNIVERSE as an automated evaluator of world model rollouts using the WHAM benchmark (Kanervisto et al., 2025), 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 3 summarizes graded accuracy across models and environments. We observe a big performance gap between rollouts from WHAM-140M and WHAM-1.6B. Despite being in-domain, WHAM-140M yields lower evaluation

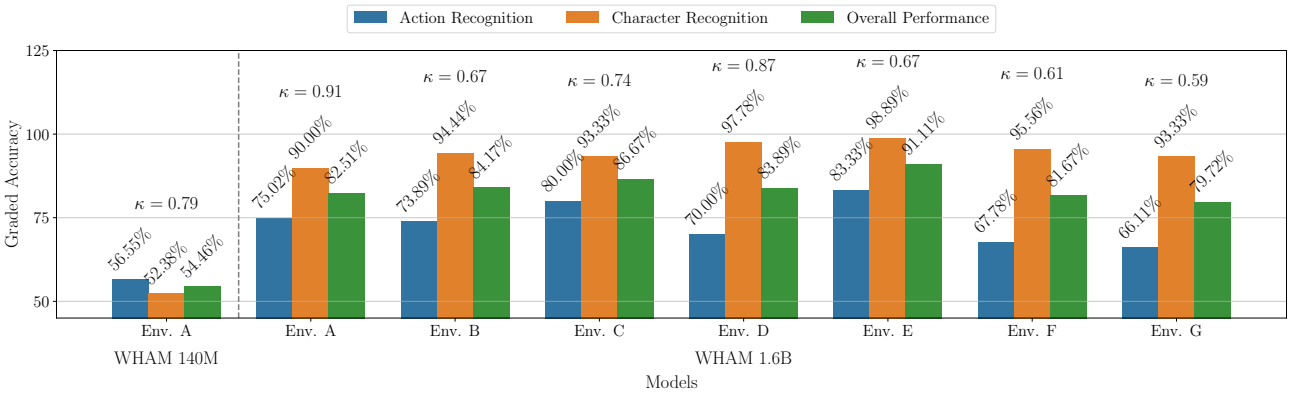


Figure 3: Graded accuracy of UNIVERSE across rollouts from WHAM-140M (Env. A) and WHAM-1.6B (Envs. A–G). Performance improves markedly with higher-fidelity rollouts from WHAM-1.6B, even in out-of-domain settings (B–G). Cohen's κ (above bars) reflects inter-rater agreement.

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.

B. Reproducibility Statement

To ensure reproducibility and support future research, we provide detailed instructions to reproduce all main experiments. Detailed descriptions of model architectures, training procedures, and dataset construction are provided in Section 4 and Appendix E. 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) and version 2 (3B and 10B) (Beyer et al., 2024; Steiner et al., 2024)), VideoLLaMA3 (2B, 7B) (Boqiang Zhang, 2025), and CLIP (Radford et al., 2021) 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.2. For our software stack, we use Matplotlib (Hunter, 2007) for plotting, NumPy (Harris et al., 2020) for data handling, openCV (Bradski, 2000), FFmpeg (Tomar, 2006) and PIL (Umesh, 2012) for video and image processing, and NLTK (Bird & Loper, 2004) for text processing. Parameter-efficient fine-tuning is implemented using the PEFT library (Mangrulkar et al., 2022).

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; 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 429 GPU-days. In total, all experiments amounted to approximately 1,430.12 GPU-days, equivalent to 3.92 GPU-years.

C. UNIVERSE: Implementation Details

This section presents a high-level implementation overview and pseudocode for our framework, UNIVERSE, which adapts VLMs for evaluating the semantic and temporal fidelity of world model rollouts. The framework consists of 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.

C.1. Adaptation Pipeline

The adaptation stage is composed of 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`.

C.2. Evaluation Pipeline

To support downstream evaluation, we introduce two additional modules: `RolloutsGenerator` and `Universe`.

`RolloutsGenerator`. This component autoregressively samples rollout trajectories from a world model (textttworld_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.

D. Dataset

This appendix 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.

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 $\texttt{FormatBinaryPrompt}$. 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

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. 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 $\mathbf{F}_1$ (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.2. 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 (Beyer et al., 2024; Steiner et al., 2024), which serves as the core of UNIVERSE

E.2.1. 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 (Chen et al., 2023), combining a ViT-based vision encoder (Dosovitskiy et al., 2021) with a decoder-only Transformer language model. The architecture is fully modular, comprising three parameterized components: (i) *Vision encoder* ($\mathcal{M}_V$): based on SigLIP (Zhai et al., 2023), specifically the "shape optimized" So400m (Alabdulmohsin et al., 2023). (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 (Mesnard et al., 2024; Rivière et al., 2024). Below, we discuss the architecture in more details, the general layer-level overview is also provided in Table 1.

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

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)

SentencePiece tokenizer (Kudo & Richardson, 2018). 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 (Hendrycks & Gimpel, 2016), 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.

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

E.2.2. 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.2.3. 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:

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

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.2.4. Pretraining Data and Filtering

PaliGemma is pretrained on a mixture of large-scale vision-language datasets, including WebLI (Chen et al., 2022), CC3M-35L (Sharma et al., 2018), VQ^{2}A-CC3M-35L (Changpinyo et al., 2022), OpenImages (Piergiovanni et al., 2022), and WIT (Srinivasan et al., 2021). Data quality and safety are maintained through pornographic content filtering, text safety and toxicity filtering, and privacy-preserving measures.

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 (Loshchilov & Hutter, 2019)
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) (Beyer et al., 2024)

E.3. Main Quantitative 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 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 (Loshchilov & Hutter, 2019) 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 (Loshchilov & Hutter, 2022) 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) (Beyer et al., 2024) 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.

F. Human Evaluation Details

This appendix provides full details of our human evaluation protocol, 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).

Annotation Setup and Interface. Annotations were collected using a custom PowerPoint-based interface (see Figure 5).

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

Figure 4: Instructional slides shown to annotators, illustrating the two recognition targets: *actions* (left) and *characters* (center and right). The slides contain 20 reference videos in total (7 for actions, 13 for characters). These examples were used to help annotators consistently evaluate VLM-generated responses during the labeling task.

Table 6: Instructions provided to annotators prior to the evaluation task. The interface includes task context, rating criteria, general guidelines, and detailed descriptions of 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.

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 6 and Figure 4. The annotation study was carried out by a subset of the authors and close collaborators 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. When it comes to annotators expertise, the two primary annotators spent several hours familiarizing themselves with the game environment and became proficient in identifying characters, actions, and gameplay dynamics. The adjudicator further reviewed numerous generated rollouts.

Selected World Models. We evaluate the VLM on video rollouts generated by two autoregressive world models of different scales, both based on the WHAM architecture (Kanervisto et al., 2025), 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 (Radford et al., 2019; Vaswani et al., 2017) trained to autoregressively predict discrete tokens representing visual observations and actions. Visual frames are first encoded using a VQGAN (Esser et al., 2021), while joystick actions are tokenized using a learned discretization scheme based on action bucketization (Kanervisto et al., 2020). 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:

- *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 models: at inference time, the model is conditioned on 1 second of ground-truth gameplay (visual and action 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 for fine-grained evaluation. This setup enables a comprehensive analysis of the VLM'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.

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 three characters to reduce annotation ambiguity.

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. We then 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 uniformly 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.

F.2. Evaluation Metrics

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

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

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

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

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

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 7.

Sample Size Justification. We sample 30 rollouts per model–environment pair. With $\sigma \approx 0.2$ and 95% confidence, the CI width is CI Width $= z_{\frac{1-C}{2}} \cdot \frac{\sigma}{\sqrt{n}}$. This yields $\sim$7.1% CI for $n = 30$ (one environment), and 2.5% when aggregating over $n = 240$ rollouts.

Given the video clip and six corresponding question–answer (QA) pairs, evaluate each answer.
For each QA pair, assign a label based on how accurately the answer reflects the visual content of the video:

Correct (1) // Partially Correct (0.5) // Incorrect (0) // Unclear.

Refer to the annotation rubric and examples provided in the guidelines.

Question	Response	Score
1. Is the character mounting hoverboard?	Yes	1
2. What is the character doing? Choose from: evading backwards, evading forwards, evading left, evading right, jumping down, jumping on the level, jumping up, mounting hoverboard.	Mounting hoverboard	1
3. What is the character doing?	Mounting hoverboard	1
4. Is the shown character Gizmo?	Yes	1
5. What character is shown? Choose from: Nidhoggr, Makutu, Prism, Ninja, Kulev, Azrael, Miko, Keyboardguy, Gizmo, Bruiser, Buttercup, Wrekko, Cass.	Gizmo	1
6. What character is shown?	Gizmo	1

Figure 5: 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).

F.3. Results

Table 8 reports graded and strict accuracy across environments, recognition targets (Action and Character Recognition), and question formats (Binary, Multiple-Choice, Open-Ended).

Table 7: 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 8: Graded / Strict accuracy of the adapted evaluator on Action and Character Recognition tasks across environments and question formats, as evaluated by human annotators. We report accuracy for Binary, Multiple-Choice (MC), and Open-Ended (OE) formats, disaggregated by recognition task and model. All scores reflect 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

We observe a clear gap in performance between rollouts generated by the two world models. The evaluator 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 evaluator's expected input of 224×224. Despite resizing, the resulting frames often lack sharpness, making actions and characters harder to recognize. In contrast, the VLM 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 Action Recognition (AR) and 90.00% for Character Recognition (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 6 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 appendix 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 provide valuable insight into model selection, adaptation efficiency, and the limitations of standard evaluation proxies in our setting.

Table 9: 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 7 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 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 9 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.

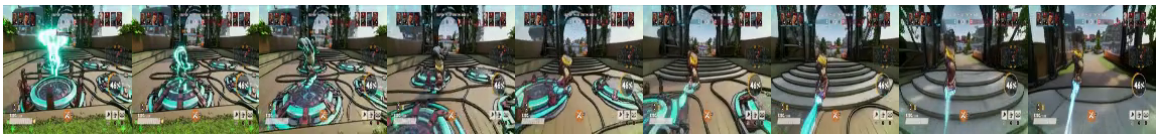
Table 10: Performance on Action and Character Recognition tasks after LoRA-based adaptation with varying ranks ($r \in \{8, 16, 32, 48, 64\}$). Adapters are applied to attention and MLP layers in both vision and language components.

	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

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 10 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.



Environment A, WHAM 140M

Environment A, WHAM 1.6B

Environment B, WHAM 1.6B

Environment C, WHAM 1.6B

Environment D, WHAM 1.6B

Environment E, WHAM 1.6B

Environment F, WHAM 1.6B

Environment G, WHAM 1.6B

Figure 6: 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.

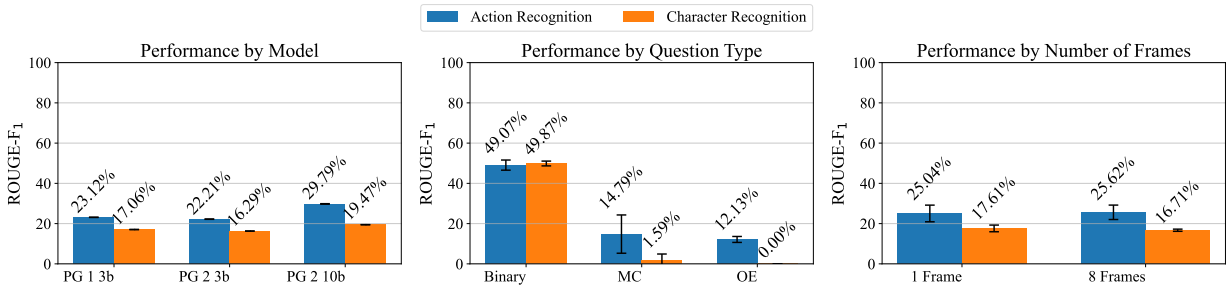


Figure 7: Zero-shot evaluation results for PaliGemma variants across tasks, prompt formats, and visual context sizes. Overall performance remains limited, indicating the need for task-specific adaptation.