SCALING IN GAMES: CONTINUED PRE-TRAINING FOR EMBODIED AGENTS IN DIVERSE VIRTUAL WORLDS

Anonymous authors
Paper under double-blind review

ABSTRACT

Scaling data, diversity and model capacity has improved embodied learning in physical environments, where diverse experiences remain grounded in shared physical laws. Whether these gains extend to virtual worlds with different visual styles, dynamics, controls, and objectives remains unclear. We investigate this question through a controlled study of continued pre-training (CPT) across three axes: data volume, diversity, and model size. Our 120 training configurations span four model sizes from 0.8B to 27B, six data budgets from 50 to 1,600 hours, and five mixtures ranging from Minecraft-only data to diverse multi-world corpora. We assess performance and generalization using three-level validation losses and gameplay benchmarks from in-corpus worlds to unseen worlds. Our results show: (1) more data improves in-corpus performance, diversity improves transfer to related worlds with limited gains in distant worlds, and model size improves data efficiency but generalizes worse to distant worlds under single-world training; (2) we observe an early *interface phase* with broad cross-world benefits and a later *world phase* with gains narrowed to in-corpus and related worlds, and fit an empirical scaling law relating validation loss to data volume, diversity and model size; and (3) our models are competitive with Minecraft specialists and frontier models in 3D worlds, and perform best in real-time settings. However, the limits of imitation and persistent gaps in distant worlds suggest that building generalist embodied agents requires richer training stages beyond continued pre-training.

1 INTRODUCTION

Embodied agents are expected to perceive, reason, and act across changing environments. Recent scaling studies in robotics suggest that broader experience can improve such generality (Lin et al., 2025; Wang et al., 2026; Abe et al., 2026). Real-world scenes share one coherent physical reality, i.e., gravity, contact, and object permanence remain consistent, and the physical effect of an embodiment's controls changes relatively little across tasks. This shared structure makes experience collected in one setting potentially reusable in another.

In Figure 1, virtual worlds remove much of this consistency. Different worlds may vary simultaneously in entirely different visuals, rules, physics, action spaces and goals (Wang et al., 2025b; Tan et al., 2025; Zhang et al., 2026b). Humans adapt to the heterogeneity quickly: they easily figure out unfamiliar rules, adjust to different controls, and apply past experience to new visuals and mechanics. A generalist virtual embodied agent should exhibit the same ability. Interactive video games provide a diverse and controlled subset of such worlds, making them a particularly demanding test of embodied generality (Abi Raad et al., 2024; SIMA Team, 2025; Zhang et al., 2026a).

Recent work studies this broader goal by training models to perceive visual input from games and output native keyboard–mouse controls across different virtual game worlds (Baker et al., 2022; Magne et al., 2026; Yue et al., 2026). Across these systems, ***continued pre-training*** (CPT) on gameplay trajectories has become a common stage for generalist embodied intelligence (Abi Raad et al., 2024; Wang et al., 2025b; SIMA Team, 2025; Tan et al., 2025). However, virtual embodied continued pre-training remains largely a recipe: existing studies change the data volume, diversity and models together, leaving the effect of each factor unresolved. Most importantly, it remains unclear whether heterogeneous experiences help to build embodied agents that can generalize across different worlds, or if it just causes interference that ruins what they learned in any individual worlds.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

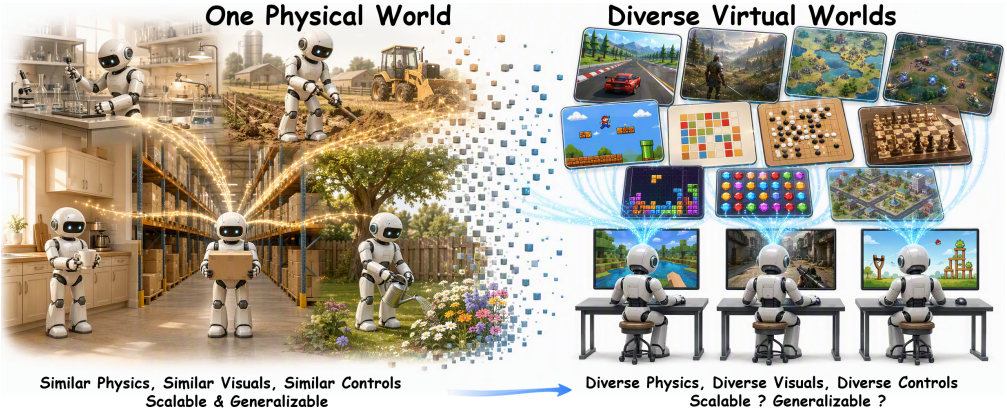

Figure 1: **One physical world versus diverse virtual worlds.** *Left:* Agents learn from varied physical scenes sharing common dynamics, where they can scale and generalize with broader experience (Wang et al., 2026). *Right:* Agents play diverse games with varying dynamics. We want to investigate the question that whether continued pre-training scales and generalizes in virtual worlds.

We address this question with a controlled grid study spanning four model sizes from 0.8B to 27B, six data budgets from 50 to 1,600 hours, and five data mixture settings from Minecraft-only to broad multi-game corpora. The resulting **120 configurations** are evaluated through three-level validation loss and benchmark ladders from in-corpus worlds, such as Minecraft (Zheng et al., 2025) and Doom (Wydmuch et al., 2019), to unseen near and distant worlds, such as OmniGameArena (Lin et al., 2026), and GameWorld (Ouyang et al., 2026). Our study reveals how data volume, mixture diversity, and model capacity affect embodied agents in these three levels of worlds, and whether heterogeneous experience becomes complementary as it scales.

Our contributions are mainly three-fold:

- **A controlled study of scaling CPT in virtual worlds.** We quantify how data volume, mixture diversity, and model size affect CPT of generalist embodied agents in virtual worlds. More relevant data improves performance on in-corpus worlds where model performance multiplies in Minecraft. Diversity improves transfer to related worlds but provides limited gains in distant worlds. Larger models improve data efficiency and achieve larger early gains, while generalizing worse under single-game training.
- **An empirical two-phase learning dynamics and scaling law.** We observe an empirical early *interface phase*: model learns basic self-control and action primitives, which are shared and improved across worlds; and a *world phase* then follows: further gains are obtained in trained and unseen related worlds. We fit an empirical scaling law to relate validation loss to data volume, mixture diversity and model size, achieving $R^2 = 0.982$ and $R^2 = 0.995$ on Minecraft and in-corpus loss, respectively.
- **Three-level evaluation of model performance and generalization.** We evaluate models in a three-level setting covering in-corpus, related, and distant worlds. This anchors our models among game and general foundation models, providing the community a reliable reference. Our models are competitive with Minecraft specialists and frontier models in 3D worlds, and outperform frontier models in real-time settings by up to $5\times$.

2 Related Work

Virtual Embodied Models. Early models were trained as environment-specific specialists. Deep reinforcement learning achieved superhuman performance in Atari, Go, StarCraft II, and Dota 2 while relying on a fixed environment, objective, and action interface (Mnih et al., 2015; Silver et al., 2016; Vinyals et al., 2019; OpenAI et al., 2019). Recent work instead focuses on large foundation models. VPT learns a Minecraft control prior from internet video, while NitroGen and OpenP2P extend pixel-to-action learning across broader game corpus (Baker et al., 2022; Magne et al., 2026;

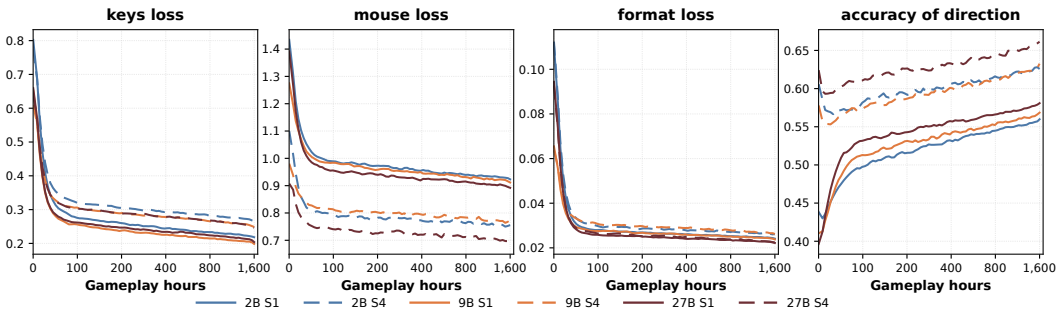

Figure 2: **Adaptation to the keyboard–mouse interface during CPT.** Smoothed training losses for key, mouse, and format tokens, and eight-direction mouse prediction accuracy for three model capacity trained on mixtures S1 and S4.

Yue et al., 2026). LLM and VLM based agents further reuse pre-trained language and visual knowledge for instruction following, planning and control (Abi Raad et al., 2024; Chen et al., 2025; Wang et al., 2025b; Tan et al., 2025; SIMA Team, 2025). Across these approaches, CPT has become a common mechanism for turning a general backbone into an interactive model for virtual embodied agents. To our knowledge, we conduct the largest controlled CPT study of virtual embodied models.

Virtual Worlds as Test Environments. Games provide a controllable subset of virtual worlds for evaluating perception, reasoning, and action under diverse rules and dynamics (Ouyang et al., 2026; Zhang et al., 2026a;b). Minecraft and Doom suites measure capabilities within a familiar in-corpus world (Wydmuch et al., 2019; Zheng et al., 2025); OmniGameArena tests transfer to unseen 3D worlds (Lin et al., 2026); and GameWorld extends evaluation to visually and mechanically distinct 2D, 2.5D, and 3D environments (Ouyang et al., 2026). Training in one kind of virtual worlds may transfer to others, but there is no previous study to investigate which factor is critical to the transfer.

Scalability of Continued Pre-training. Classical scaling laws relate model performance to capacity, data, and compute, while studies of domain-adaptive and repeated pre-training show that source composition can alter both adaptation and retention (Kaplan et al., 2020; Hoffmann et al., 2022; Que et al., 2024; Gu et al., 2024; Wu et al., 2026). Related evidence appears in computer-use and embodied learning: GUI studies find that task composition can matter as much as sample count (Liu et al., 2026; Zhang et al., 2025; Raoof et al., 2026), and robotic imitation studies report diversity scaling, negative transfer, and capacity-dependent interference (Lin et al., 2025; Wang et al., 2026; Abe et al., 2026). Virtual worlds sharpen this problem because visual style, dynamics, controls, and objectives change across sources. Existing research on CPT in virtual worlds alters model size, training data, and duration simultaneously. No prior work systematically isolates these three axes while measuring held-out loss and transfer capabilities at varying distances from training.

3 Preliminaries and Experiment Framework

3.1 Data construction

Gameplay trajectories. Following the setting in previous works (Wang et al., 2025b; Tan et al., 2025; SIMA Team, 2025; Lin et al., 2026), we formulate gameplay as visual decision-making process. At step t, the model takes prompt context c, the current observation o_t, and interaction history $\mathcal{H}_t$ as the input x_t and produces a reasoning trace r_t and then an executable action a_t:

$$r_t \sim p_\theta(\cdot \mid x_t), \qquad a_t \sim p_\theta(\cdot \mid x_t, r_t), \qquad x_t = (c, o_t, \mathcal{H}_t). \tag{1}$$

Reasoning and action have distinct roles. The possibly empty r_t is internal textual reasoning. The mandatory a_t is structured text chunks that are parsed into low-level device commands, which further changes the environment. Each chunk covers 200 ms (Tan et al., 2025; Lin et al., 2026):

$$a_t = \texttt{<|action_start|>}\ \Delta x_t\ \Delta y_t\ \Delta z_t; K_t^{(1)}; \ldots; K_t^{(6)}\ \texttt{<|action_end|>}, \tag{2}$$

where $(\Delta x_t, \Delta y_t) \in [-1000, 1000]^2$ are discretized relative mouse moves, $\Delta z_t \in [-5, 5]$ is the scroll delta, and each $K_t^{(j)}$ contains 0-4 keys or mouse buttons held for 33 ms. The model receives

observations at 5 Hz and controls at 30 Hz. History is modeled as interleaved image–action turns. Each trajectory has 20 turns. During training, preceding think and action are visible to later turns.

Three scaling axes. Our study evaluates a $4 \times 6 \times 5 = 120$ configuration grid across four Qwen3.5 backbones $\mathcal{M} = \{0.8\text{B}, 2\text{B}, 9\text{B}, 27\text{B}\}$ (Qwen Team, 2026)), six data budgets $\mathcal{T}$ hours $= \{50, 100, 200, 400, 800, 1600\}$, and five data diversity mixture $\mathcal{S} = \{\text{S1}, \ldots, \text{S5}\}$. This yields total 3.5×10^{11} training tokens and 2.2×10^{22} FLOPs via the "$6ND$ rule" (Hoffmann et al., 2022).

Data diversity mixtures. Our training data consists of two components: (1) 85% gameplay trajectories sampled from VPT, D2E, and OpenP2P (Baker et al., 2022; Choi et al., 2025; Yue et al., 2026), and (2) 15% general data covering vision–language, embodied and egocentric settings (Li et al., 2025a; 2026; Tong et al., 2026; Zhang et al., 2026c; Yang et al., 2026). We construct five data mixtures ($S1$-$S5$) with increasing mixture diversity by setting the Minecraft trajectory proportion in the gameplay data to 100%, 51.4%, 27.1%, 2.8%, and 0%, respectively. The remaining trajectories are sampled from a diverse pool of other games. We also augment a small fraction of gameplay turns with retrospective reasoning generated by GPT-5.6-Luna (OpenAI, 2026) based on the recorded observation and action, to preserve the backbone's native reasoning capabilities without altering action targets. See Appendices A.1 and A.2 for further details about scaling on mixture diversity.

3.2 Training and Evaluation

Training objective and validation sets. All models are continued pre-trained using a causal next-token prediction loss over assistant responses conditioned on interleaved histories. All parameters, including those of the ViT, are updated using AdamW (Loshchilov & Hutter, 2019) with a Warmup–Stable–Decay schedule (Hu et al., 2024). The resulting models are evaluated on three levels of validation data: Minecraft-only data, data from in-corpus games covered during training, and data from unseen games not covered in training. We also record key, mouse, and format losses. Apart from data volume, data diversity, and the choice of base model, all experimental settings are held fixed across training configurations. See Appendix A.4 for further details.

Three-level gameplay evaluation. We evaluate gameplay competence across three categories of virtual worlds that are increasing distant from the training distribution. (1) **In-corpus worlds**: we use the MCU benchmark (Zheng et al., 2025) to evaluate performance on multiple tasks in Minecraft, and ViZDoom (Wydmuch et al., 2019) to evaluate performance on single shooting tasks in DOOM. Both games are covered in our training corpus. (2) **Unseen related worlds**: we evaluate our model in OmniGameArena (OGA) (Lin et al., 2026), which consists of seven unseen 3D worlds. Although their scenes and tasks are absent from training, action primitives such as navigation, aiming, jumping, shooting, and interaction have semantic counterparts in the training corpus. (3) **Unseen distant world**: we evaluate our model in GameWorld (GW) (Ouyang et al., 2026), which primarily covers 2D and 2.5D games, alongside a small portion of 3D games. These games differ substantially from the training data in visual styles, physics, controls and objectives.

4 Scaling and Generalization in Heterogeneous Virtual Worlds

We investigate how three scaling axes: data volume, data diversity, and model size, affect performance and generalization across all 120 configurations. Figure 2 illustrates the action component losses. Figure 3 presents the three-level training and validation losses alongside our scaling law predictions. Figure 4 details benchmark evaluations. The next three subsections concentrate on each axis and the last subsection derives an empirical scaling law that unifies them.

4.1 Data Scaling

Early training brings broad cross-world gains. Figure 2 shows the training loss of key, mouse, and format tokens. These losses drop rapidly during the initial tens of hours of training data before tapering into a more gradual decrease. Mouse direction prediction accuracy continues to improve. After exposure to 50 hours of training data, validation losses drop below those of the base models on all three validation sets (Figure 3), even for games absent from the training mixture. We refer to this early stage as the *interface phase*. This cross world improvement indicates that the model gradually acquires shared formats, representations and action patterns that can transfer across games.

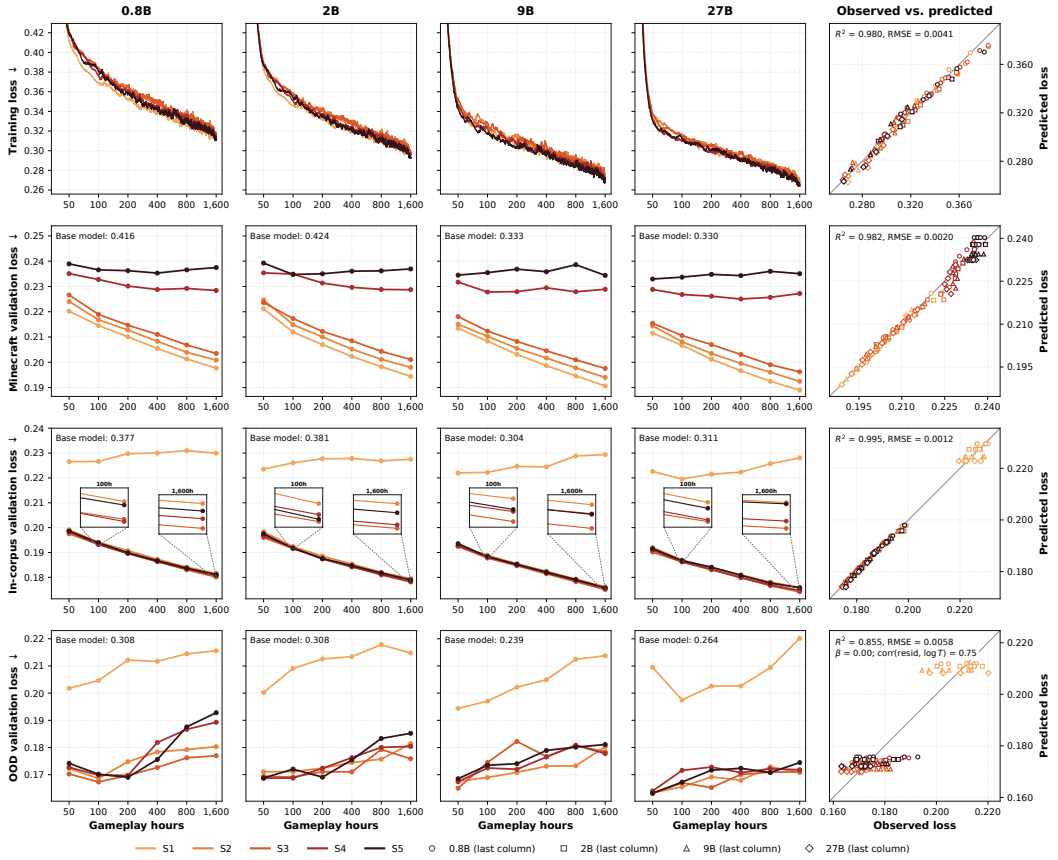

Figure 3: **Loss scaling across data volume, model size and data diversity.** Rows report the smoothed training loss and three-level validation loss. Columns 1–4 denote the 0.8B-27B backbones; colors denote the training mixtures S1–S5; markers denote the six checkpoints from 50 to 1,600 hours. The base model loss is annotated in each validation panel. All plots in a given row shares a common y-axis scale. Column 5 plots the loss predicted by the fitted scaling law (Section 4.4) against the observed loss.

Later gains depend more on the worlds seen during training. After the interface phase, further performance gains depend heavily on exposure to relevant game worlds. We term this stage the *world phase*. From 50 to 1,600 hours, the mean Minecraft validation loss decreases by about 10.4% across four model sizes under the MineCraft-heavy mixtures S1–S3, correlating with substantial gains in MCU performance (Figure 3). Conversely, scaling total data volume without sufficient Minecraft trajectories yields negligible further reduction in this specific validation loss. For the in-corpus validation set, diverse multi-game training consistently reduces loss across all model sizes, whereas Minecraft-only training causes the loss to plateau or even increase. Similarly, out-of-distribution (OOD) loss stagnates or rises, although it remains below base model levels. At the largest model size, GameWorld performance degrades across all five mixtures as data volume increases. Thus, more data improves performance on in-corpus and related environments, but fails to drive continued gains in distant worlds.

4.2 Diversity Scaling

The best training mixture depends on the evaluation set. Across all 24 configurations of model size and data volume, Minecraft validation loss is lowest with S1, whereas in-corpus loss is minimized with S3 or S4 (Figure 3). Increasing the Minecraft data proportion specifically improves Minecraft evaluation, whereas diverse mixtures improve general in-corpus performance. For in-corpus validation loss, differences among S2-S5 are negligible compared to their gap from S1.

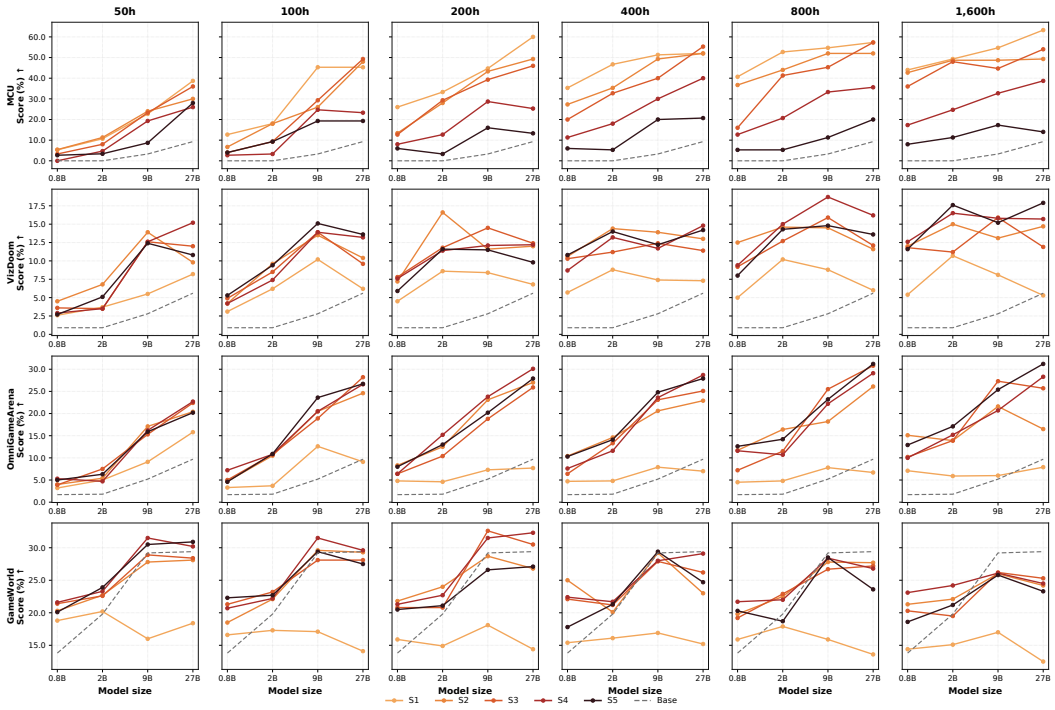

Figure 4: **Benchmark performance across data budget, model size and data diversity.** Rows report MCU, ViZDoom, OmniGameArena, and GameWorld performance. Columns denote the six checkpoints from 50 to 1,600 hours. Colors denote S1-S5 and the base-model reference, while each curve spans the 0.8B-27B backbones. All plots in a given row shares a common y-axis scale.

Remarkably, the zero-Minecraft S5 mixture still achieves a lower Minecraft validation loss than the base models. Comparisons at similar amounts of Minecraft data also show benefits from other games. Under a limited data budget, our results suggest matching the training distribution to the target evaluation domain rather than blindly expanding game diversity. Among all diversity mixtures, S1 has the highest in-corpus and OOD validation losses, along with the lowest ViZDoom, OGA, and GameWorld scores. Despite Minecraft encompassing diverse mechanics like combat and navigation, this within-game diversity fails to effectively generalize to other distinct virtual worlds.

Diversity mainly improves transfer to related worlds. At 1,600 hours, the mean OGA score across S2–S5 is roughly 2.8 times that of S1, with all 16 mixed configurations outperforming S1 at matched model scales. ViZDoom performance similarly improves despite DOOM data forming only a small fraction of the training mixture. This indicates that basic navigation and aiming abilities can easily transfer across 3D games. A smaller model trained on diverse games can outperform a larger model trained mostly on Minecraft. On GameWorld, models trained with S1 degrade as training data volume increases. Although multi-game training mitigates this decline, the models still fail to consistently surpass their base models. Therefore, broader training diversity enhances capabilities in related environments but yields inconsistent gains in distant worlds.

4.3 Model Scaling

Larger models use data more efficiently and achieve larger early gains. At limited data budgets, larger models generally show greater improvements on MCU and OGA. They achieve comparable or superior performance with up to a 32× less gameplay data. Large models trained on 50-hour data outperform small models trained on 1,600-hour data. These findings suggest that larger models extract more value from limited experience, requiring substantially less data to reach given performance level. However, further scaling yields uneven returns: Minecraft loss and ViZDoom performance exhibit diminishing gains beyond 9B, OGA does not improve uniformly across all mixtures, and CPT adds little to the existing GameWorld performance of larger backbones.

Single-game training can amplify cross-world decline in larger models. During the world phase, model size interacts with the training mixture. Under mixed-game training, OOD loss becomes more stable as models grow; Under single-game training (S1), the late-stage degradation in OOD performance is most severe for the largest model. In-corpus loss reflects a similar weakening of the larger models' advantage under S1. At 1,600 hours, 27B-S1 underperforms its base model on all three non-Minecraft benchmarks, suffering a maximum performance drop on GameWorld of roughly 17 points. Thus, greater data efficiency under single-game training can actively compromise cross-world generalization. Broadening the training data mitigates this loss, though generalization to distant worlds remains a challenge.

4.4 An Empirical Three-Factor Scaling Law on CPT Loss

Following D-CPT (Que et al., 2024), we formulate an empirical scaling law relating validation loss to data volume, diversity and model size. Specifically, the Chinchilla scaling law $L = E + A/N^{\alpha} + B/T^{\beta}$ (Hoffmann et al., 2022) is extended by introducing a ratio ρ into the data term to account for gameplay diversity. Moreover, we observe that even without Minecraft data, the S5 mixture yields a finite Minecraft validation loss well below that of the base model. To capture this cross-mixture transfer, we add a term that remains finite even at a zero data share:

$$L(N, T, \rho) = E + \frac{A}{N^{\alpha}} + \frac{B\rho^{\eta}}{T^{\beta}} + \frac{C}{(\rho + \epsilon)^{\gamma}}, \tag{3}$$

where N is the parameter count, T is gameplay hours, and E, A, α, B, η, β, C, γ, $\epsilon > 0$ are fitted parameters. Let r denote the proportion of Minecraft trajectories in the training mixture. We set $\rho = r$ when predicting Minecraft validation loss, and $\rho = 1 - r$ for in-corpus loss. Finally, because each mixture is evaluated on its respective training distribution, the training loss itself is fitted using the standard form $L = E_S + A/N^{\alpha} + B/T^{\beta}$.

The empirical scaling laws yield strong fits for training, Minecraft, and in-corpus losses, achieving R^2 values of 0.980, 0.982, and 0.995, with RMSE of 0.0041, 0.0020, and 0.0012, respectively. The fits also extrapolate: refitted without the 27B models, the 1,600-hour checkpoints or the 50-hour checkpoints, Eq. (3) predicts the held-out validation losses with RMSE between 0.0016 and 0.0027. OOD loss, in contrast, does not follow a scaling law. Its gain over the base model is already reached at 50 hours, after which it rises slightly with further training in all 20 curves. Fitting Eq. (3) to it therefore gives $\beta \approx 0$, and the remaining R^2 of 0.855 only reflects static differences between mixtures. Coupled with the marginal CPT gains on GameWorld in Figure 4, this suggests that CPT improves learning within the corpus but does not keep improving on distant worlds. Fitting details, the derivation of Eq. (3), comparisons with alternative models and uncertainty analyses are given in Appendix C.6.

5 Comparing with Frontier Models

This section compares our experimentally derived models (labeled as GameScaling in Tables 1 and 2) against current frontier models. By doing so, we establish a comprehensive performance reference across four distinct virtual benchmarks.

Comparison with Minecraft game models. Table 1 shows the performance of our model on MCU (Zheng et al., 2025). Existing baselines fall into two categories: environment-specific models built exclusively for Minecraft, from controllers trained on videos (Baker et al., 2022; Lifshitz et al., 2023; Cai et al., 2025) to instruction-tuned backbones (Li et al., 2025b; Wang et al., 2025a; Zhou et al., 2026); and general game models trained across diverse games environments using a unified keyboard–mouse interface, either with (Wang et al., 2025b) or without (Magne et al., 2026; Yue et al., 2026) a language backbone. Based on training data, our S1 models align with the former group and S3 models with the latter; critically, neither relies on task-specific supervision.

We highlight three key findings. First, generative backbone capability outweighs Minecraft-specific design: our 27B-S1 model outperforms the best specialized model by 13 points, and even the 0.8B-S1 surpasses 7B specialists. Second, a language foundation is more critical than sheer data scale: behavior-cloning models trained on tens of thousands of hours score near zero on instruction-conditioned tasks, while our 0.8B-S3 completes 36% of tasks. Third, task categories exhibit distinct

Table 1: **Comparison with baselines on MCU.** We report average steps and success rate (ASR, %) across 30 comprehensive tasks. Baselines are quoted from GameWAM. Within each model group, the red and blue indicate the best and second, respectively. "–" denotes an unavailable result.

Model	Size	Embodied Tasks		Combat Tasks		GUI Tasks		Average	
		Steps ↓	ASR ↑	Steps ↓	ASR ↑	Steps ↓	ASR ↑	Steps ↓	ASR ↑
Environment-Specific Model: Trained on MineCraft trajectories									
VPT (Baker et al., 2022)	248M	377	$10.1^{\pm 3.6}$	396	$3.6^{\pm 7.7}$	398	$0.7^{\pm 0.1}$	390	4.8
STEVE-1 (Lifshitz et al., 2023)	248M	384	$8.4^{\pm 3.0}$	395	$4.9^{\pm 1.8}$	391	0.0	390	4.4
ROCKET-1 (Cai et al., 2025)	72B	392	$19.2^{\pm 6.1}$	320	$29.8^{\pm 9.0}$	–	–	356	24.5
JARVIS-VLA (Li et al., 2025b)	7B	305	$31.0^{\pm 12.7}$	352	$18.3^{\pm 5.2}$	339	$25.3^{\pm 5.7}$	332	24.9
OpenHA (Wang et al., 2025a)	7B	287	$37.0^{\pm 15.9}$	316	$40.0^{\pm 19.6}$	314	$33.3^{\pm 13.3}$	306	36.8
MAIN-VLA (Zhou et al., 2026)	7B	263	$38.5^{\pm 11.8}$	248	$49.3^{\pm 13.5}$	291	$36.7^{\pm 8.1}$	267	41.5
GameScaling-0.8B-S1	0.8B	255	$74.0^{\pm 34.7}$	374	$50.0^{\pm 33.8}$	590	$8.0^{\pm 13.3}$	406	44.0
GameScaling-2B-S1	2B	263	$80.0^{\pm 23.7}$	335	$52.0^{\pm 32.5}$	553	$16.0^{\pm 12.0}$	384	49.3
GameWAM (Guo et al., 2026)	6B	138	$70.0^{\pm 32.2}$	203	$39.0^{\pm 25.9}$	155	$43.0^{\pm 32.9}$	165	50.7
GameScaling-9B-S1	9B	319	$66.0^{\pm 28.4}$	307	$58.0^{\pm 37.4}$	483	$40.0^{\pm 35.8}$	370	54.7
GameScaling-27B-S1	27B	272	$76.0^{\pm 21.5}$	320	$64.0^{\pm 33.2}$	422	$50.0^{\pm 39.2}$	338	63.3
General Game Foundation Model: Trained across diverse games									
NitroGen (Magne et al., 2026)	500M	581	$0.5^{\pm 0.1}$	510	$2.1^{\pm 0.3}$	594	$0.3^{\pm 0.0}$	562	1.0
Open-P2P (Yue et al., 2026)	1.2B	489	$1.5^{\pm 0.6}$	466	$5.3^{\pm 1.5}$	586	$0.4^{\pm 0.1}$	514	2.4
GameScaling-0.8B-S3	0.8B	345	$68.0^{\pm 31.2}$	420	$38.0^{\pm 31.6}$	596	$2.0^{\pm 6.0}$	454	36.0
GameTARS-Dense (Wang et al., 2025b)	7B	373	$50.4^{\pm 20.7}$	372	$38.1^{\pm 24.6}$	406	$39.1^{\pm 27.5}$	384	42.5
GameScaling-9B-S3	9B	329	$66.0^{\pm 28.4}$	425	$36.0^{\pm 28.0}$	512	$32.0^{\pm 29.9}$	422	44.7
GameScaling-2B-S3	2B	260	$76.0^{\pm 19.6}$	355	$52.0^{\pm 25.6}$	563	$16.0^{\pm 12.0}$	393	48.0
GameScaling-27B-S3	27B	326	$62.0^{\pm 22.7}$	339	$58.0^{\pm 30.3}$	460	$42.0^{\pm 22.7}$	375	54.0
GameTARS-MoE (Wang et al., 2025b)	20B-active MoE	188	$72.0^{\pm 21.5}$	271	$66.1^{\pm 25.2}$	288	$55.4^{\pm 17.6}$	249	64.5

Table 2: **Performance across four benchmarks under two interaction settings.** For proprietary models whose size label follows the approximate estimates (Li, 2026). TP and RT denotes the time-paused and real-time settings. The best is marked in red and the second in blue.

Model	Size	MCU		ViZDoom		OmniGameArena		GameWorld		Average	
		TP	RT	TP	RT	TP	RT	TP	RT	TP	RT
Proprietary Frontier Models											
Gemini-3.8-Flash	∼T	88.7	7.3	27.8	2.8	45.2	18.6	60.7	48.2	55.6	19.2
GPT-5.6-Sol	∼T	82.0	8.7	26.7	2.7	45.6	14.5	53.9	44.3	52.1	17.6
GPT-5.6-Luna	∼T	54.0	5.3	24.1	2.9	26.7	13.2	44.9	41.6	37.4	15.8
Claude-Opus-5.0	∼T	56.7	8.7	20.8	2.0	44.4	16.4	52.8	42.0	43.7	17.3
Open-Source Frontier Models											
GLM-5.3-Flash	320B	47.3	7.3	18.9	1.9	23.5	12.3	34.0	29.6	30.9	12.8
DeepSeek-V4-Flash-Vision	305B	36.0	7.3	13.0	3.3	27.1	11.6	41.8	28.5	29.5	12.7
Open-Source Qwen Models and GameScaling Models											
Qwen3.5-0.8B	0.8B	0.0	0.0	0.9	1.0	1.7	1.8	13.8	15.1	4.1	4.5
GameScaling-0.8B-S3	0.8B	36.0	37.3	11.9	11.8	10.2	9.8	20.3	22.5	19.6	20.4
Qwen3.5-2B	2B	0.0	0.0	0.9	1.5	1.8	1.7	19.8	16.3	5.6	4.9
GameScaling-2B-S3	2B	48.0	44.0	11.2	10.4	13.9	12.2	19.5	20.0	23.2	21.7
Qwen3.5-9B	9B	3.3	3.3	2.8	3.2	5.2	4.1	29.2	27.7	10.1	9.6
GameScaling-9B-S3	9B	44.7	36.7	15.9	16.4	27.3	20.2	26.2	27.8	28.5	25.3
Qwen3.5-27B	27B	9.3	4.0	5.6	2.9	9.7	7.6	29.3	25.2	13.5	9.9
GameScaling-27B-S3	27B	54.0	16.7	14.7	13.2	25.7	17.8	25.3	24.8	29.9	18.1

scaling behaviors. Success rates on embodied tasks remain stable across our model sizes and lead all baselines, whereas GUI task success increases sixfold from 0.8B to 27B. This suggests that while coarse movement is acquired early in the interface phase; precise cursor control necessitates substantial model capacity.

Cross-benchmark reference. Table 2 compares proprietary frontier models (OpenAI, 2026; Google DeepMind, 2026; Anthropic, 2026), open-source frontier models (Qwen Team, 2026; Z.ai, 2026; DeepSeek-AI, 2026), and our S3 models using a unified keyboard–mouse interface. We evaluate under two settings: 1. time-paused interaction, which isolates pure reasoning competence, and 2.

real-time interaction, which assesses competence and inference latency simultaneously. Under time-paused interaction, performance strongly depends on generalization distance. In 3D worlds, CPT amplifies the 27B backbone's score by roughly $6\times$ on MCU and $2.6\times$ on OGA; notably, our 9B and 27B models match the performance of $\sim$300B open models.

In the distant GameWorld benchmark, frontier models score twice as high as ours, and CPT only yields improvements at 0.8B and 2B scales, highlighting distant-world transfer as the primary bottleneck for generality. Under real-time interaction, this performance hierarchy reverses. On ViZDoom and MCU, every frontier model suffers an approximate 90% score reduction because delayed decisions are executed in an environment state that has already evolved. Our 0.8B-9B models retain 88% of their time-paused performance, outscoring the best frontier models several times over. On OGA, which features exploration-heavy tasks, this real-time penalty for frontier models is roughly halved. Ultimately, the real-time penalty scales with the tempo of the world, and a model two orders of magnitude smaller is the better player when the world does not pause.

6 Discussion and Conclusion

6.1 Scalability and Generalization Depend on Virtual Worlds

The observed two-phase learning dynamics clarify why scaling reliably enhances in-domain gameplay but struggles to achieve broad transfer. Early training yields cross-worlds improvements, reflecting the acquisition of universal controls and action patterns. Subsequent gains, however, depend more on the training distribution. By 1,600 hours, diverse mixed training produces a mean OGA score about $2.8\times$ higher than Minecraft-only training, averaged across model sizes (Section 4.2). These benefits fail to generalize consistently to distant virtual worlds. Even a feature-rich environment like Minecraft, which encompasses combat, exploration, and navigation, yields limited transfer when used in isolation. Consequently, scaling model capacity or training duration must be paired with strategic curation of the training corpus. With a limited data budget, the training mixture should closely mirror the target evaluation environments. Finally, while empirical scaling laws accurately estimate in-corpus losses, a strong in-domain fit does not guarantee robust play in distant worlds.

6.2 Games Require Both Ability and Speed

Frontier models have advanced on gaming benchmarks (Figure 6), and our GameScaling models prove highly competitive with both Minecraft specialists and frontier models in 3D environments (Section 5). Crucially, their relative performance hinges heavily on whether the environment pauses during inference. Under real-time evaluation, GameScaling-9B-S3 achieves an average score of 25.3 across the four benchmarks, outperforming the best-tested frontier model (19.2) (Table 2). In dynamic unpaused worlds, a delayed action often becomes obsolete by the time it is executed. Real-time competence therefore depends on both gameplay capability and response latency. Further, these metrics reflect real-world deployment realities: GameScaling uses vLLM API endpoints on our H200 cluster, whereas frontier models incur network latency via provider APIs. Future comparisons must transparently report task scores, response times, and deployment infrastructure to accurately assess a model's true utility during active gameplay.

6.3 Toward a General Game Model

While CPT establishes a strong foundation for generalist game agents, performance gains in related environments do not necessarily translate to distant worlds. For instance, compared to its base model, the 27B-S3 checkpoint improves the time-paused OGA score by 16.0 points but suffer a 4.0-points drop on GameWorld (Table 2). This discrepancy underscores the necessity of distant-world evaluation to accurately assess model generality. CPT relies on behavior cloning from recorded actions, lacking direct environmental feedback regarding task success or failure. The observed generalization gap may stem from this imitation-based objective, insufficient game coverage, or a combination of both. Future work should investigate whether scaling world diversity, integrating online feedback during actual play, or combining both paradigms enhances cross-world transfer (Zhang et al., 2026a). This discrepancy underscores the absolute necessity of distant-world evaluation to accurately assess model generality.

AI USE STATEMENT

In this work, GPT-5.6-Luna was used to generate sparse retrospective reasoning annotations from recorded gameplay video and the corresponding recorded action. Generative AI tools were also used to assist with language editing, document organization, LaTeX drafting, and plotting code. They were not used to generate experimental measurements or to determine the reported results. The authors reviewed AI-assisted annotations, text, code, citations, and claims, and take responsibility for the final content of the paper and its supplementary materials.

REPRODUCIBILITY STATEMENT

We describe data construction and world mixtures in Appendix A.1, and provide the training grid, training schedule, budget definitions, and optimization settings in Appendices A.3 and A.4. Evaluation protocols and model deployment are detailed in Appendix B. After the review process concludes, we will open-source the complete training code and publicly release the GameScaling model checkpoints to support reproduction of our results and further research on continued pre-training in virtual worlds.

ETHICS STATEMENT

Our gameplay training data are sampled from the publicly released VPT, D2E, and OpenP2P datasets (Baker et al., 2022; Choi et al., 2025; Yue et al., 2026) under their respective licenses. These datasets were cleaned by their providers to protect player privacy. We sample existing gameplay trajectories and add retrospective reasoning annotations, without collecting new gameplay recordings or personal data. We identify no additional ethical concerns arising from these procedures.

REFERENCES

Haruki Abe, Takayuki Osa, Yusuke Mukuta, and Tatsuya Harada. Cross-embodiment offline reinforcement learning for heterogeneous robot datasets. In *International Conference on Learning Representations*, 2026.

Maria Abi Raad, Arun Ahuja, Catarina Barros, Frederic Besse, Andrew Bolt, Adrian Bolton, et al. Scaling instructable agents across many simulated worlds. *arXiv preprint arXiv:2404.10179*, 2024.

Anthropic. System card: Claude Opus 5. System card, 2026. URL https://www.anthropic.com/claude-opus-5-system-card.

Bowen Baker, Ilge Akkaya, Peter Zhokhov, Joost Huizinga, Jie Tang, Adrien Ecoffet, et al. Video pretraining (VPT): Learning to act by watching unlabeled online videos. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24639–24654, 2022.

Shaofei Cai, Zihao Wang, Kewei Lian, Zhancun Mu, Xiaojian Ma, Anji Liu, et al. ROCKET-1: Mastering open-world interaction with visual-temporal context prompting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12122–12131, 2025. doi: 10.1109/CVPR52734.2025.01132.

Peng Chen, Pi Bu, Yingyao Wang, Xinyi Wang, Ziming Wang, Jie Guo, Yingxiu Zhao, Qi Zhu, Jun Song, Siran Yang, Jiamang Wang, and Bo Zheng. Combatvla: An efficient vision-language-action model for combat tasks in 3d action role-playing games, 2025. URL https://arxiv.org/abs/2503.09527.

Suhwan Choi, Jaeyoon Jung, Haebin Seong, Minchan Kim, Minyeong Kim, Yongjun Cho, et al. D2E: Scaling vision-action pretraining on desktop data for transfer to embodied ai. *arXiv preprint arXiv:2510.05684*, 2025.

DeepSeek-AI. DeepSeek-V4-Flash-Vision-Exp, 2026. URL https://huggingface.co/deepseek-ai/DeepSeek-V4-Flash-Vision-Exp. Model card; builds on the DeepSeek-V4-Flash architecture with visual modules.

Google DeepMind. Gemini 3.8 Flash model card. Model card, 2026. URL https://deepmind.google/models/model-cards/gemini-3-8-flash.

Jiawei Gu, Zacc Yang, Chuanghao Ding, Rui Zhao, and Fei Tan. CMR scaling law: Predicting critical mixture ratios for continual pre-training of language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 16143–16162. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.903.

Yuncheng Guo, Zhanqiu Zhang, Yiwen Guo, and Weijia Li. GameWAM: A world action model for video games. *arXiv preprint arXiv:2608.26200*, 2026.

Mark O. Hill. Diversity and evenness: A unifying notation and its consequences. *Ecology*, 54(2): 427–432, 1973.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, et al. An empirical analysis of compute-optimal large language model training. In *Advances in Neural Information Processing Systems*, volume 35, pp. 30016–30030, 2022.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, et al. MiniCPM: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*, 2024.

Lou Jost. Entropy and diversity. *Oikos*, 113(2):363–375, 2006.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, et al. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Ang Li, Charles L. Wang, Deqing Fu, Kaiyu Yue, Zikui Cai, Wang Bill Zhu, et al. Zebra-CoT: A dataset for interleaved vision-language reasoning. In *International Conference on Learning Representations*, 2026.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, et al. LLaVA-OneVision: Easy visual task transfer. *Transactions on Machine Learning Research*, 2025a.

Bojie Li. Incompressible knowledge probes: Estimating black-box llm parameter counts via factual capacity. *arXiv preprint arXiv:2604.24827*, 2026.

Muyao Li, Zihao Wang, Kaichen He, Xiaojian Ma, and Yitao Liang. JARVIS-VLA: Post-training large-scale vision language models to play visual games with keyboards and mouse. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 17878–17899. Association for Computational Linguistics, 2025b. doi: 10.18653/v1/2025.findings-acl.920.

Shalev Lifshitz, Keiran Paster, Harris Chan, Jimmy Ba, and Sheila McIlraith. STEVE-1: A generative model for text-to-behavior in minecraft. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Fanqi Lin, Yingdong Hu, Pingyue Sheng, Chuan Wen, Jiacheng You, and Yang Gao. Data scaling laws in imitation learning for robotic manipulation. In *International Conference on Learning Representations*, 2025.

Mingxian Lin, Shengju Qian, Yuqi Liu, Yi-Hua Huang, Yiyu Wang, Wei Huang, et al. OmniGameArena: A unified UE5 benchmark for VLM game agents with improvement dynamics. *arXiv preprint arXiv:2606.09826*, 2026.

Zhaoyang Liu, JingJing Xie, Zichen Ding, Zehao Li, Bowen Yang, Zhenyu Wu, et al. ScaleCUA: Scaling open-source computer use agents with cross-platform data. In *International Conference on Learning Representations*, 2026.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Loïc Magne, Anas Awadalla, Guanzhi Wang, Yinzhen Xu, Joshua Belofsky, Fengyuan Hu, et al. NitroGen: An open foundation model for generalist gaming agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21511–21521, 2026.

Markov AI. Gaming-500-Hours: Native gameplay screen recordings with input events. Hugging Face dataset, 2026. URL https://huggingface.co/datasets/markov-ai/gaming-500-hours. 494.7 hours of gameplay across 168 games.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015. doi: 10.1038/nature14236.

OpenAI. GPT-5.6 preview system card, June 2026. URL https://deploymentsafety.openai.com/gpt-5-6-preview. Official system card, published 26 June 2026; accessed 2026-07-31.

OpenAI, Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

Mingyu Ouyang, Siyuan Hu, Kevin Qinghong Lin, Hwee Tou Ng, and Mike Zheng Shou. GameWorld: Towards standardized and verifiable evaluation of multimodal game agents. *arXiv preprint arXiv:2604.07429*, 2026.

Haoran Que, Jiaheng Liu, Ge Zhang, Chenchen Zhang, Xingwei Qu, Yinghao Ma, et al. D-CPT Law: Domain-specific continual pre-training scaling law for large language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 90318–90354, 2024. doi: 10.52202/079017-2867.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Negin Raoof, Richard Zhuang, Marianna Nezhurina, Etash Guha, Atula Tejaswi, Ryan Marten, et al. OpenThoughts-Agent: Data recipes for agentic models. *arXiv preprint arXiv:2606.24855*, 2026.

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, et al. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016. doi: 10.1038/nature16961.

SIMA Team. SIMA 2: A generalist embodied agent for virtual worlds. *arXiv preprint arXiv:2512.04797*, 2025.

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

Jingqi Tong, Jixin Tang, Hangcheng Li, Yurong Mou, Ming Zhang, Jun Zhao, et al. Game-RL: Synthesizing multimodal verifiable game data to boost VLMs' general reasoning. In *International Conference on Learning Representations*, 2026.

Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019. doi: 10.1038/s41586-019-1724-z.

Ye Wang, Sipeng Zheng, Hao Luo, Wanpeng Zhang, Haoqi Yuan, Chaoyi Xu, et al. Rethinking visual-language-action model scaling: Alignment, mixture, and regularization. *arXiv preprint arXiv:2602.09722*, 2026.

Zihao Wang, Muyao Li, Kaichen He, Xiangyu Wang, Zhancun Mu, Anji Liu, et al. OpenHA: A series of open-source hierarchical agentic models in minecraft. *arXiv preprint arXiv:2509.13347*, 2025a.

Zihao Wang, Xujing Li, Yining Ye, Junjie Fang, Haoming Wang, Longxiang Liu, et al. Game-TARS: Pretrained foundation models for scalable generalist multimodal game agents. *arXiv preprint arXiv:2510.23691*, 2025b.

Boqian Wu, Qiao Xiao, Patrik Okanovic, Tomasz Sternal, Maurice van Keulen, Mykola Pechenizkiy, et al. When data is scarce: Scaling sparse language models with repeated training. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Marek Wydmuch, Michał Kempka, and Wojciech Jaśkowski. ViZDoom Competitions: Playing Doom from Pixels. *IEEE Transactions on Games*, 11(3):248–259, 2019. doi: 10.1109/TG.2018.2877047. The 2022 IEEE Transactions on Games Outstanding Paper Award.

Shusheng Yang, Jihan Yang, Pinzhi Huang, Ellis Brown, Zihao Yang, Yue Yu, et al. Cambrian-S: Towards spatial supersensing in video. In *International Conference on Learning Representations*, 2026.

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

Z.ai. GLM-5.3-Flash: Frontier intelligence, flash cost, 2026. URL `https://z.ai/blog/glm-5.3-flash`. Technical blog and model card; the model follows the GLM-5 technical report.

Junlei Zhang, Zichen Ding, Chang Ma, Zijie Chen, Qiushi Sun, Zhenzhong Lan, et al. Breaking the data barrier – building GUI agents through task generalization. *arXiv preprint arXiv:2504.10127*, 2025.

Kuan Zhang, Dongchen Liu, Qiyue Zhao, Jinkun Hou, Xinran Zhang, Qinlei Xie, et al. GameVerse: Can vision-language models learn from video-based reflection? *arXiv preprint arXiv:2603.06656*, 2026a.

Kuan Zhang, Dongchen Liu, Qiyue Zhao, Tianyu Xin, Yue Su, Haisheng Wang, et al. Towards generalist game players: An investigation of foundation models in the game multiverse. *arXiv preprint arXiv:2605.09965*, 2026b.

Wenqi Zhang, Mengna Wang, Gangao Liu, Huixin Xu, Yiwei Jiang, Yongliang Shen, et al. Embodied-Reasoner: Synergizing visual search, reasoning, and action for embodied interactive tasks. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 41178–41207. Association for Computational Linguistics, 2026c.

Xinyue Zheng, Haowei Lin, Kaichen He, Zihao Wang, Qiang Fu, Haobo Fu, et al. MCU: An evaluation framework for open-ended game agents. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 78221–78259. PMLR, 2025.

Zheyuan Zhou, Liang Du, Zixun Sun, Xiaoyu Zhou, Ruimin Ye, Qihao Chen, et al. MAIN-VLA: Modeling abstraction of intention and environment for vision-language-action models. *arXiv preprint arXiv:2602.02212*, 2026.

A Training and Dataset Details

A.1 Dataset Construction and Mixture Accounting

All configurations allocate 85% of training tokens to gameplay trajectories and 15% to a fixed vision–language anchor. The gameplay pool combines VPT, D2E, and OpenP2P (Baker et al., 2022; Choi et al., 2025; Yue et al., 2026). The vision–language anchor comprises LLaVA-OneVision, Zebra-CoT, GameQA, Embodied-Reasoner, and Cambrian-S data (Li et al., 2025a; 2026; Tong et al., 2026; Zhang et al., 2026c; Yang et al., 2026). The general vision–language data does not include game content from the 85% gameplay component. This prevents the fixed general vision–language data from providing additional exposure to these game worlds when we vary the diversity of the gameplay component. Its composition and token share remain unchanged across S1–S5. We vary data diversity by changing the balance between Minecraft and a fixed pool of other game worlds. Across S2–S5, the non-Minecraft worlds remain the same, and their relative proportions within that pool are approximately unchanged. This shifts training from Minecraft-dominated data toward a more balanced mix of 79 worlds, without progressively adding new non-Minecraft games. Holding the pool fixed helps separate the effect of world proportions from the effect of the number of worlds.

Running the 120-condition grid with one training seed required approximately 45,000 H200 GPU-hours. Given this cost, we leave repeated training with different seeds and experiments that vary the number of worlds to future work.

We normalize source-specific mouse coordinates, sample observations at 5 Hz, and construct 20-turn windows with a 10-turn stride. Each window supervises 4.0 seconds of actions and spans 3.8 seconds between its first and last observations. We use the latter interval to measure gameplay duration. Actions are converted to the unified six-substep format, and idle-dominated windows are downsampled. For the heterogeneous pool, we sample complete recordings and cap each source stream to limit dominance by long streams. S2–S4 use nested subsets of this capped pool, while S5 follows the same procedure after removing Minecraft. So the measured Minecraft shares among gameplay segments are 100%, 51.4%, 27.1%, and 2.8%, respectively. S5 removes these residual Minecraft trajectories, reducing the share to 0%.

Fixed training-game inventory. S2–S5 draw from the same inventory of 124 non-Minecraft source streams; the mixtures change only how many segments are drawn from those streams. S2–S4 additionally contain Minecraft, whereas S5 removes both D2E and OpenP2P Minecraft streams. S1 contains only VPT Minecraft. The 124 shared streams correspond to 79 canonical non-Minecraft labels; adding Minecraft gives the 80 labels listed in Table 4. Thus, "124" refers to the fixed source-stream inventory, not to 124 title-disjoint commercial games. Two very small residual metadata labels (*Roblox: Debug* and *Other / Unlabeled*) are retained in the accounting table rather than silently reassigned.

Mixture diversity. Because the world inventory is fixed, S1–S5 cannot be ordered by the number of worlds: S2, S3, and S4 contain the same 80 canonical labels. What changes is how gameplay experience is distributed over them. Let p_w be the share of gameplay segments from canonical world w in a mixture. We measure mixture diversity by the effective number of worlds,

$$D = \exp\Big(-\sum_w p_w \log p_w \Big), \tag{4}$$

the Hill number of order one (Hill, 1973; Jost, 2006). D equals the number of equally weighted worlds that would give the same Shannon entropy. It combines how many worlds are present with how evenly experience is spread across them, so it is 1 for a single world and 80 for 80 equally weighted worlds. Table 3 reports D together with the inverse Simpson index $1/\sum_w p_w^2$, which weights dominant worlds more heavily, and the largest single-world share. Both indices order S1–S4 identically. Only the step from S1 to S2 enlarges the world set; from S2 to S4 the set is unchanged, and diversity grows because Minecraft's dominant share is reassigned to the heterogeneous worlds. S5 has nearly the same diversity as S4 and differs from it mainly by excluding the residual 2.8% of Minecraft, so it serves as a source-free control rather than a further diversity step. At a fixed budget, raising D also raises the hours of each heterogeneous world and lowers the Minecraft hours, because S2–S4 draw nested subsets from the same capped pool.

Table 3: **Mixture diversity of S1–S5.** Shares are computed over gameplay segments after cross-source aliases are merged into canonical world labels. D is the effective number of worlds (Hill number of order one); the inverse Simpson index emphasizes dominant worlds.

Mixture	Worlds	Minecraft share	Largest share	$D = \exp H$	Inverse Simpson
S1	1	100%	100%	1.0	1.0
S2	80	51.4%	51.4%	14.5	3.7
S3	80	27.1%	27.1%	35.2	11.9
S4	80	2.8%	4.2%	60.1	52.6
S5	79	0%	4.3%	58.7	51.0

We use the six game categories shown in the dataset table: AVG (adventure/survival), STG (shooting), RPG (role-playing), RAC (racing), SIM (simulation), and OTH (platform, sports, MOBA, and other games). The category assignment remain fixed across S1–S5.

Validation sets. We use three fixed held-out sets to evaluate action loss. The Minecraft set consists of trajectories from the VPT test split. The in-corpus set is game-balanced and draws from

Table 4: **Fixed game inventory represented across S1–S5.** The table merges source aliases into canonical labels. n is the number of distinct D2E/OpenP2P source-stream labels represented by that canonical label; the non-Minecraft entries sum to 124. VPT Minecraft is accounted for separately.

Canonical game	Category	n	Canonical game	Category	n
Barony	AVG	1	GTA: San Andreas	RPG	1
Core Keeper	AVG	1	MapleStory Worlds Southperry	RPG	1
Dinkum	AVG	1	Monster Hunter Wilds	RPG	1
Grounded	AVG	1	Saints Row 4	RPG	1
Minecraft	AVG	2	Saints Row: The Third	RPG	1
Ogu Forest	AVG	1	Skul	RPG	1
PEAK	AVG	1	Elefant Hovercraft	RAC	5
Raft	AVG	1	Need for Speed	RAC	1
Roblox: A Dusty Trip	AVG	4	Need for Speed: Carbon	RAC	1
Roblox: Natural Disaster Survival	AVG	5	Need for Speed: Most Wanted	RAC	1
Sea of Thieves	AVG	1	Need for Speed: Underground	RAC	1
Spider-Man 2	AVG	1	Need for Speed: Underground 2	RAC	3
Apex Legends	STG	1	DRL Simulator	SIM	2
Battlefield 6 Open Beta	STG	1	Euro Truck Simulator 2	SIM	4
Brotato	STG	1	FPV Kamikaze Drone	SIM	3
Call of Duty: Black Ops Zombies	STG	1	House Flipper	SIM	1
Call of Duty Mobile	STG	1	House Flipper 2	SIM	1
Call of Duty: Black Ops 2	STG	4	Liftoff	SIM	2
Call of Duty: Black Ops 3	STG	2	Medieval Dynasty	SIM	1
Counter-Strike 2	STG	1	PowerWash Simulator	SIM	2
DOOM	STG	7	Satisfactory	SIM	1
DOOM Eternal	STG	2	Slime Rancher	SIM	1
Elefant Test FPS	STG	4	Stardew Valley	SIM	1
Fortnite	STG	1	Roblox: Debug (source label)	OTH	1
Helldivers 2	STG	1	Eternal Return	OTH	1
Left 4 Dead 2	STG	2	Goat Simulator	OTH	1
PUBG	STG	1	Goat Simulator 3	OTH	1
Quake	STG	2	League of Legends	OTH	2
Rainbow Six	STG	1	Other / Unlabeled	OTH	1
Ready or Not	STG	1	Roblox: Be a Shark	OTH	1
Resident Evil 5	STG	1	Roblox: Be a Snake	OTH	1
Resident Evil: Revelations 2	STG	1	Roblox: Be a Tornado	OTH	1
Roblox: Hypershot	STG	1	Roblox: Blade Ball	OTH	1
Roblox: Murderers vs Sheriffs	STG	2	Roblox: Death Ball	OTH	3
Roblox: Rivals	STG	2	Roblox: Eat the World	OTH	1
VALORANT	STG	1	Roblox: Evade	OTH	2
Warhammer: Vermintide	STG	1	Roblox: Slap Battle	OTH	2
Warhammer: Vermintide 2	STG	1	Rocket League	OTH	1
Cyberpunk 2077	RPG	1	Super Bunny Man	OTH	1
GTA V	RPG	2	Vampire Survivors	OTH	1

D2E and OpenP2P recordings held out from the S4–S5 construction, together with a size-matched Minecraft subset from the VPT test split. The out-of-distribution (OOD) set contains only game titles absent from every S1–S5 mixture. All three sets use the same observation and action representation as training. Recordings and adjacent overlapping windows are kept separate across training and validation to prevent leakage.

OOD source and construction. The OOD set is drawn from the processed Gaming500 (Markov AI, 2026) recording pool, which is not used to construct any S1–S5 training mixture. We first remove exact-title overlaps with VPT, D2E, and OpenP2P, as well as obvious aliases, games from the same retained franchise, and emulator labels that resolve to a training title. We then retain keyboard–mouse recordings only. A recording is rejected when at least 5% of its foreground events belong to a browser, desktop, launcher, or other non-game application; when at least half of its mouse events

Table 5: **Composition of the OOD validation set.** All clips are drawn from Gaming500 recordings and come from titles absent from S1–S5. "Sessions" denotes the number of independent source recordings. Each game contributes exactly 160 clips.

#	Game	Category	Space	Sessions	Clips
1	Palworld	RPG	3D	4	160
2	Assassin's Creed Syndicate	AVG	3D	4	160
3	007 First Light	STG	3D	4	160
4	Age of Empires II: Definitive Edition	SIM	2.5D	4	160
5	Forza Horizon 5	RAC	3D	4	160
6	Asphalt 9: Legends	RAC	3D	4	160
7	PRAGMATA	AVG	3D	4	160
8	Candy Crush Saga	OTH	2D	4	160
9	Dota 2	OTH	3D	4	160
10	The Finals	STG	3D	4	160
11	Farming Simulator 18	SIM	3D	4	160
12	Wolfenstein: The Old Blood	STG	3D	4	160
13	Hogwarts Legacy	RPG	3D	4	160
14	Just Cause 2	AVG	3D	2	160
15	ARC Raiders	STG	3D	2	160
16	2XKO	OTH	2.5D	4	160
17	Terraria	AVG	2D	3	160
18	Pokémon Z (fan game)	RPG	2D	2	160
19	Township	SIM	2D	3	160
20	South of Midnight	AVG	3D	4	160
21	Ori and the Will of the Wisps	OTH	2.5D	2	160
22	Free Fire MAX	STG	3D	4	160
23	Beach Buggy Racing	RAC	3D	4	160
24	Gear.Club Unlimited 2	RAC	3D	2	160
25	Pacify	AVG	3D	2	160
26	ONE PIECE Bounty Rush	OTH	3D	2	160
27	Schedule I	SIM	3D	2	160
28	Dead as Disco	OTH	3D	2	160
		Total	**21 3D / 3 2.5D / 4 2D**	**92**	**4,480**

exhibit cursor recentering; or when its keyboard–mouse activity is too sparse (fewer than 60 input events per minute, fewer than three keyboard or click events per minute, or more than 60% idle windows). We additionally require at least 60% of a title's clean recordings to satisfy the keyboard–mouse criteria. These filters remove gamepad sessions whose labels are all zero, cursor-locked sessions whose mouse deltas are uninformative, and recordings dominated by menus or unrelated applications.

The resulting set contains 28 games and 4,480 clips. We sample exactly 160 non-idle clips per game, so every title contributes $1/28 = 3.57\%$ of the set, and distribute those clips as evenly as possible over two to four independent recording sessions (92 sessions in total). The set is therefore balanced by game rather than by the amount of available source video duration. Using the rendered interaction space as the taxonomy, it comprises 21 3D games (3,360 clips; 75.0%), three 2.5D games (480 clips; 10.7%), and four 2D games (640 clips; 14.3%). We count isometric scenes rendered with 3D geometry, such as *Dota 2*, as 3D; 2.5D denotes side-view or planar interaction with mixed 2D/3D presentation. Under the same game-category taxonomy used for the training inventory, the OOD set contains six AVG, five STG, three RPG, four RAC, four SIM, and six OTH titles. Table 5 gives the complete composition.

A.2 Sparse Retrospective Reasoning Annotations and Action Template

Qwen3.5 natively places deliberation within a `<think>` span before producing its final output. We preserve this structure across all CPT configurations because the same behavior frequently occurs

during online gameplay evaluation. Retaining the reasoning channel keeps the training response format aligned with the reason-then-act format used at evaluation.

Annotation procedure. For selected gameplay windows, GPT-5.6-Luna receives the recorded visual context together with the action that the player subsequently executed. It is instructed to adopt the player's perspective and reconstruct a short retrospective explanation of the apparent observation, intent, and chosen action. The annotation describes the recorded behavior rather than proposing or relabeling an action, so the original keyboard–mouse supervision remains unchanged. We inject these rationales sparsely among ordinary gameplay turns and apply the same procedure to S1–S5, preventing reasoning density from becoming an unintended diversity variable. Table 6 reports the resulting density for each mixture. A 20-frame training sample carries 1.08–1.19 non-empty rationales on average, 74–78% of samples contain at least one, and only 5.4–6.0% of assistant turns include reasoning. Each rationale averages 64.5–65.4 tokens under the Qwen3.5 tokenizer (pooled median 65; 10th–90th percentile 56–74), or 70–78 reasoning tokens per sample. Because rationale count, coverage, and length differ only marginally across S1–S5, reasoning supervision does not act as an additional variable alongside data diversity.

Table 6: **Sparse reasoning density in the S1–S5 training mixtures.** Counts are from a full scan of the training parquets. A rationale is a non-empty `<think>` span attached to an assistant turn. Token lengths are measured with the Qwen3.5 tokenizer on 13k–19k sampled rationales per mixture.

Mixture	Samples	Samples w/ rationale	Rationales / sample	Reasoning turns	Tokens / rationale	Reasoning tokens / sample
S1	1.50M	74.0%	1.08	5.40%	65.0	70.2
S2	1.50M	75.8%	1.14	5.69%	64.5	73.4
S3	1.50M	76.7%	1.17	5.83%	65.3	76.1
S4	1.50M	77.6%	1.19	5.96%	65.4	78.0
S5	1.50M	77.6%	1.19	5.97%	64.6	77.1

Training format.

```
User: <image>
Assistant: <think>
[Sparse reasoning, when available]
</think>
<|action_start|> Δx Δy Δz ; k1 ; ... ; k6
<|action_end|>
```

When no retrospective annotation is attached, the chat template supplies an empty `<think>` span, keeping the response grammar identical across examples. At evaluation, the model is free to generate a non-empty reasoning segment before the action packet. Figure 5 shows five consecutive turns from an S4 training window, two with rationales and three without.

A.3 Full Accounting of the 120-Condition Factorial Grid

Table 7 outlines the 20 primary Warmup–Stable–Decay (WSD) trunks (Hu et al., 2024). For every evaluated checkpoint, the reported gameplay-hour budget $T \in \{50, 100, 200, 400, 800, 1600\}$ includes all three phases: warmup, stable training, and cosine decay. Thus, the 50-hour model has a total budget of 50 hours, including its decay stage. For each model–mixture pair, the shorter-budget models branch from earlier trunk checkpoints and complete their cosine decay when the cumulative budget reaches T. The 1,600-hour condition uses the trunk's own final decay endpoint. This reuse produces six completed checkpoints per model–mixture pair and 120 conditions in total.

A.4 Training Hyperparameters and Optimization Stability

All CPT runs use bf16 AdamW (Loshchilov & Hutter, 2019) ($\beta_1 = 0.9$, $\beta_2 = 0.95$, weight decay 0.01) with gradient clipping at 1.0 and a fixed target batch budget of 2,359,296 tokens. The language-model peak learning rate is 1×10^{-5} at every parameter scale. The vision encoder is unfrozen and

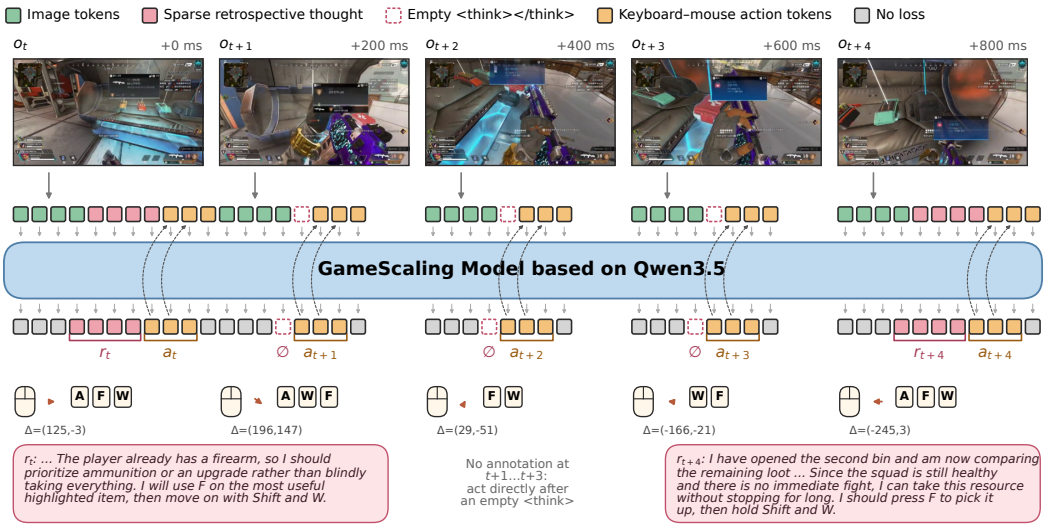

Figure 5: **A training window with sparse reasoning.** Five consecutive 200 ms turns from an Apex Legends recording in S4. Each turn begins with the current frame. Only o_t and o_{t+4} carry a GPT-5.6-Luna rationale; the other three turns emit an empty `<think>` span and act directly. Every action packet contains a mouse displacement $(\Delta x, \Delta y)$ and the keys held during six 33 ms sub-steps; key icons show the distinct keys in each packet. Image tokens receive no loss, while thought and action tokens are trained autoregressively. Rationales are abridged excerpts of the stored annotations.

Table 7: **Factorial 120-condition experimental matrix.** Twenty primary WSD trunks, each producing five annealed branch checkpoints and one final trunk endpoint at total budgets $T \in \{50\text{h}, 100\text{h}, 200\text{h}, 400\text{h}, 800\text{h}, 1600\text{h}\}$. Each budget includes warmup, stable training, and decay.

Data Mixture ID	Composition Breakdown	Model Scales	Trunks	Evaluated Checkpoints
S1 (Pure Single-Source)	100% Minecraft	{0.8B, 2B, 9B, 27B}	4	24
S2 (Equal Mixture)	51.4% Minecraft	{0.8B, 2B, 9B, 27B}	4	24
S3 (Heterogeneous-Dominant)	27.1% Minecraft	{0.8B, 2B, 9B, 27B}	4	24
S4 (Broad Multi-Game)	2.8% Minecraft	{0.8B, 2B, 9B, 27B}	4	24
S5 (Minecraft-Excluded)	0% Minecraft	{0.8B, 2B, 9B, 27B}	4	24
Total Grid			**20 Trunks**	**120 Conditions**

uses a separate rate of 7×10^{-6}. The WSD schedule (Hu et al., 2024) allocates 3% of training to linear warmup, 87.3% to the stable phase, and the final 9.7% to cosine decay with a minimum learning-rate ratio of 0.1. All three phases are included in the reported training budget, as defined in Appendix A.3. Every condition uses 854×480 frames, a 20-frame gameplay context, a fixed 15% vision–language token share, and seed 3407.

B Evaluation Details

B.1 Benchmark Evaluation Environments

Level 1: Source-World Benchmarks (Minecraft and ViZDoom). MCU (Zheng et al., 2025) uses 30 long-horizon Minecraft tasks spanning navigation, survival, GUI interaction, combat, and crafting. ViZDoom (Wydmuch et al., 2019) evaluates first-person control on four skill-1 DOOM maps; the scoring protocol is given in Appendix B.3. Both environments are represented in the gameplay corpus, but at opposite densities: Minecraft spans 0–100% of the gameplay budget across S1–S5, whereas footage of the classic DOOM titles on whose engine ViZDoom runs never exceeds one to two percent of any mixture; later DOOM releases in the corpus (e.g., DOOM Eternal) share the first-person shooting skill but differ in art style and rules. ViZDoom therefore probes a rare world and a frequent skill: DOOM itself is minimally exposed, but the first-person aiming and firing

it tests is common to the games that dominate the heterogeneous pool, which is why the main text treats the pool, not DOOM, as its source.

Level 2: Related-World Benchmark (OmniGameArena - OGA). Consists of 7 diverse photo-realistic Unreal Engine 5 games. Scenarios cover third-person exploration, tactical combat, target shooting, and obstacle platforming, running on native UE5 execution environments.

Level 3: Distant OOD Benchmark (GameWorld - GW). Comprises 170 tasks across 34 environments, mostly 2D and 2.5D, including classic arcade platformers, puzzle solvers (2048, Sokoban), and word-guessing games, whose visual styles, physics, and control logic depart from the training worlds.

Action-interface adaptation. OGA natively consumes a packet similar to the `<|action_start|> ... <|action_end|>` format described in Section 3; we set its mouse-displacement range and the number of keys per substep to match ours, and change nothing else. MCU, ViZDoom, and GameWorld do not. For each of these three benchmarks we add a thin adapter that parses the packet into the harness's native keyboard–mouse events: the six per-substep key sets are replayed as a 6×33 ms key timeline, and the mouse displacement is mapped to the environment's native camera or cursor control. The adapter changes only the action interface. Task definitions, observation resolution, episode length, step caps, success criteria, scoring, and every other evaluation parameter remain those of the original benchmark, and the same adapter is used for every model we evaluate in Table 2, including the frontier models, which receive the packet grammar through the same prompt and are parsed by the same code.

B.2 Evaluation Protocol

Episodes and averaging. Every reported score is a mean over tasks of the mean over episodes. MCU runs each of its 30 tasks 5 times and report the mean and standard deviation over 5 runs in Table 1; OGA runs each of its 7 games 10 times; ViZDoom runs each of its 4 maps 10 times; GameWorld runs each of its 170 tasks once. The same counts apply to every model and every checkpoint in the grid, and to every model in Table 2. The average score on the four benchmarks is reported in Table 2.

Provenance of the two comparison tables. The prior-model rows of Table 1 are quoted from their source papers. Their backbones, action interfaces, training data, and harnesses are therefore not aligned with ours, and the table should be read as a reference to the published state of Minecraft agents rather than as a controlled comparison. Table 2 is different: we evaluated every model in it ourselves, under our action interface, with the adapters above, the same prompts, the same observation resolution and history, and the same episode counts. Only the model changes between rows.

Episode budgets and termination. Each benchmark keeps its native budget. MCU allows a fixed 600 step budget in the time-paused setting and 3 minutes of game time per task in the real time setting. ViZDoom and GameWorld use a fixed step cap. OGA ends an episode when either its game-time limit or its step cap is reached, whichever comes first. On MCU, ViZDoom, and OGA an episode stops immediately when the task fails or the agent dies, and the episode is scored as it stands. On GameWorld, failure or death restarts the game, and the agent keeps playing until the step cap is reached; the score is the progress accumulated within the cap. Under real-time interaction, the difference is that the world advances while the model reasons.

B.3 ViZDoom Scoring Protocol

Maps and per-map score. We use four maps at `doom_skill` 1: `battle1` and `battle2` from Direct Future Prediction, and the standard `defend_the_center` and `defend_the_line` scenarios. The two families carry different native units. On the battle maps we record the frag count, following the convention of the work that introduced them. On the defend maps we record the cumulative engine reward, which awards $+1$ per kill and -1 on death. The battle WADs also expose a per-kill reward through ACS global 0, and we verified over 3,358 episodes that it equals the frag

count on every one, so the two conventions coincide there and a paper reporting "average episode reward" on these maps reports the same quantity.

Aggregate score. The reported ViZDoom number is the unweighted mean of the four per-map scores after each is divided by 100:

$$S_{\text{ViZDoom}} = \frac{1}{4} \sum_{m \in \mathcal{M}} \frac{\bar{s}_m}{100}, \qquad \mathcal{M} = \{\texttt{battle1}, \texttt{battle2}, \texttt{dtc}, \texttt{dtl}\},$$

where $\bar{s}_m$ averages 10 episodes on map m.

B.4 Serving and Deployment Configurations

All GameScaling checkpoints are deployed in bf16 on an NVIDIA H200 cluster and evaluated through vLLM API endpoints. The context length is sized to the observation history, `limit-mm-per-prompt` equals the history length plus one, sampling uses temperature 1.0, and generation stops at `<|action_end|>`. Under time-paused interaction, every MCU, ViZDoom, OGA, and GameWorld evaluation job reserves eight H200 GPUs and serves the model through vLLM. Under real-time interaction, where inference latency enters the score, the 0.8B and 2B models use tensor-parallel size 1 on one H200 per vLLM instance, while the 9B and 27B models use tensor-parallel size 4 on four H200 GPUs per instance. Table 8 summarizes these protocol-dependent layouts.

Table 8: **Serving layouts for the GameScaling models.** All time-paused benchmark jobs reserve eight H200 GPUs. Real-time vLLM instances use TP = 1 for 0.8B/2B and TP = 4 for 9B/27B so that the reported score includes the latency of the stated deployment.

Protocol	Benchmarks	Model sizes	vLLM deployment	Timing semantics
Time-paused	MCU, ViZDoom, OGA, GameWorld	0.8B, 2B, 9B, 27B	8 H200 GPUs per job	Environment frozen during inference
Real-time	MCU, ViZDoom, OGA, GameWorld	0.8B, 2B	TP = 1, one H200 per instance	Environment advances during inference
Real-time	MCU, ViZDoom, OGA, GameWorld	9B, 27B	TP = 4, four H200 GPUs per instance	Environment advances during inference

Time-paused interaction. Because the environment is frozen while the model reasons, decision latency does not enter the score and requests may be scheduled for throughput. Every time-paused run on MCU, ViZDoom, OGA, and GameWorld uses an eight-H200 job with vLLM serving. The environment backend remains benchmark-specific: Minecraft, ViZDoom, and Chromium execute outside the model server, while OGA uses UE5 instances on GPU. For OGA, each UE5 instance retains a dedicated GPU because packing multiple instances on one GPU starves real-time rendering during the 6×33 ms action chunk and can depress the score without producing an execution error.

Real-time interaction. Under real-time interaction the measured latency of each decision is part of the score: the environment keeps advancing and the model keeps idle until the new one arrives. Actually, once model latency significantly exceeds 200 ms (such as 0.5s), both executing the previous action chunk and staying idle result in a severe performance degradation by more than $5\times$. Due to high latency, our 27B model also suffers an obvious performance drop. Each live worker therefore uses a dedicated vLLM endpoint rather than sharing a request queue. The 0.8B and 2B models are served on one H200 with TP = 1; the 9B and 27B models are served on four H200 GPUs with TP = 4. Splitting the 0.8B model across four GPUs raised its per-step latency from about 130 ms to 178–190 ms.

Frontier models. Proprietary and hosted open-source models are evaluated through their provider APIs with the same adapter and prompt. All six models run at the default reasoning effort documented by their providers, which we do not override: `medium` for GPT-5.6-Sol, GPT-5.6-Luna, and Gemini-3.8-Flash; adaptive thinking at `high` effort for Claude-Opus-5.0; `high` for DeepSeek-V4-Flash-Vision; and `max` for GLM-5.3-Flash, whose thinking cannot be disabled. The output cap is 2,048 tokens on MCU, ViZDoom and OGA and 4,096 on GameWorld, the same for every model and both protocols. Models are run strictly one at a time with 5 concurrent workers under time-paused interaction and 2 under real-time interaction; their real-time scores therefore include provider-side latency, which is the latency a deployed agent would experience.

C SUPPLEMENTARY EXPERIMENTS

C.1 FRONTIER PROGRESS BY RELEASE DATE

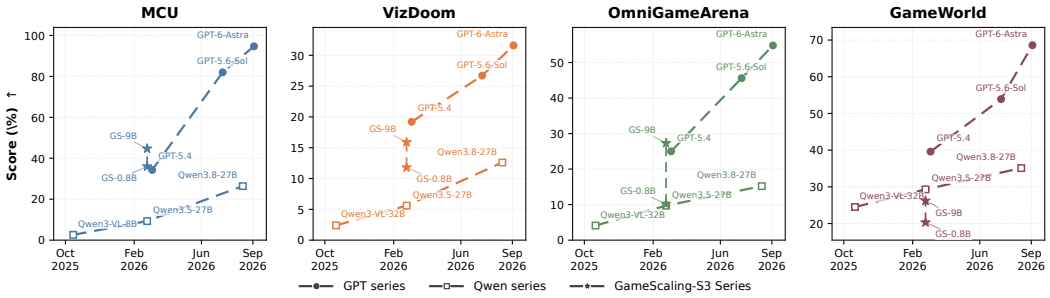

Figure 6: **Frontier progress on the four benchmarks against release date.** Circles: GPT series; squares: Qwen series; stars: GameScaling S3 models at 1,600 h, all under time-paused interaction.

Progress in general models can bring larger gains than game-specific CPT. Figure 6 shows substantial improvements in gameplay across successive GPT and Qwen models under time-paused evaluation. Over the release range shown, gains in frontier-model performance can exceed those from our game-specific CPT on a fixed backbone. This difference is particularly clear on GameWorld, where CPT provides limited gains while newer frontier models improve markedly. These results suggest that general model capability is a major contributor to gameplay performance. Game-specific CPT develops useful skills, but improving the backbone itself remains an important route to stronger game agents.

C.2 LOSS SCALING AGAINST MODEL SIZE

Figure 7 replots the losses of Figure 3 with model size on the horizontal axis and one column per duration. Across the six budgets, larger models generally have lower Minecraft and mixed-training in-corpus losses. OOD loss behaves differently: mixed training limits its later increase in larger models, while single-game training can weaken their advantage.

C.3 BENCHMARK SCALING AGAINST GAMEPLAY HOURS

Figure 8 replots the 100 scores of Figure 4 with gameplay hours on the horizontal axis and one column per model size, so the duration dynamics of Section 4.1 can be read directly: the monotone gains of 0.8B and 2B under every mixture, the flattening of 9B and 27B after 100–400 h, the decline of S1 at 9B and 27B on OGA and ViZDoom, and the decline of every 27B mixture on GameWorld.

C.4 TEST-TIME SCALING ON MCU

The standard MCU protocol caps every rollout at 600 steps, and all MCU numbers in the paper use this cap. Figure 9 asks how much of the remaining failure is a step budget rather than a competence limit. We rerun GameScaling-0.8B-S1 and GameScaling-9B-S1 at 1,600 h, together with their untrained Qwen3.5 backbones, with the cap raised to 3,000 steps and record, for each threshold, the fraction of the 150 rollouts that have succeeded within it. Success rises roughly log-linearly with the allowed steps for both models: from about 42% and 51% at the standard 600 steps to 59.3% and 69.3% at 3,000, with no sign of a plateau. The two curves are nearly parallel, so the gain from more interaction steps is about the same for a 0.8B and a 9B model, and the size gap of about 10 points is preserved at every budget. The backbones do not scale this way. Qwen3.5-0.8B completes no task at any budget, and Qwen3.5-9B rises only from about 2% at 600 steps to 4.0% at 3,000. More steps help only a model that can already act in the world: CPT raises both the level and the test-time slope, and within the range we tested a GameScaling model given more time keeps converting it into completed tasks, independent of its size.

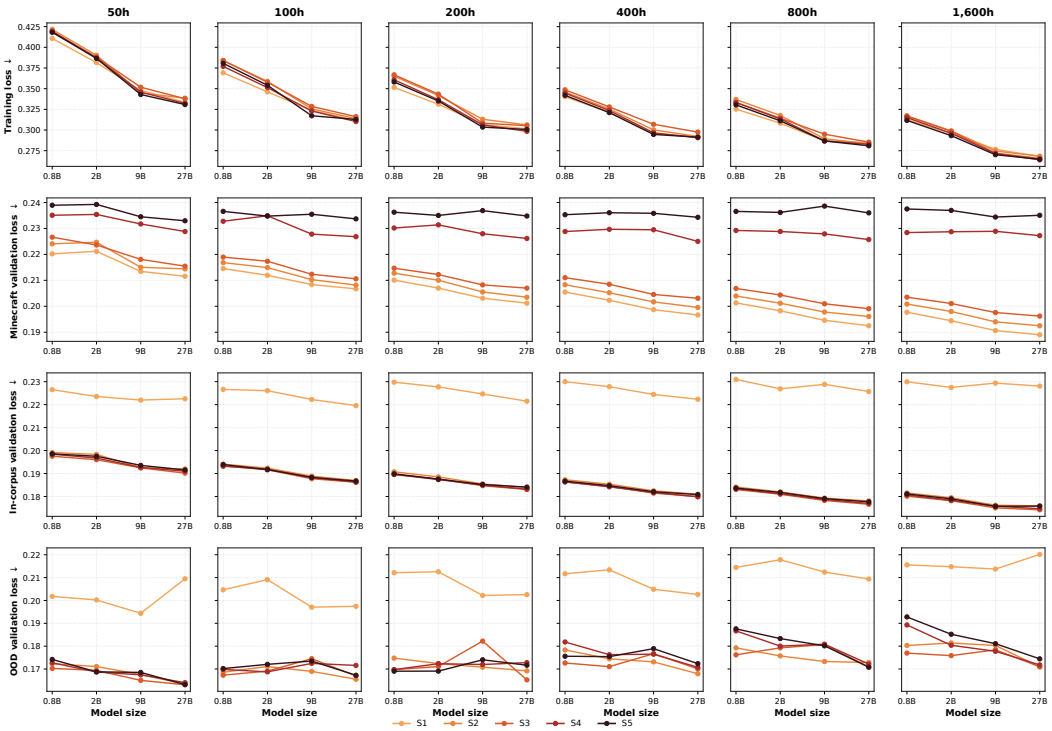

Figure 7: **Loss scaling against model size.** Rows: training loss at the checkpoint, and held-out validation loss on Minecraft, on the game-balanced in-corpus set, and on the out-of-distribution set; columns: 50, 100, 200, 400, 800, and 1,600 gameplay hours; colors: mixtures S1–S5.

C.5 Inference Latency and Output Length

The real-time columns of Table 2 compare API-served frontier models with GameScaling models on one or four H200 GPUs, and this setup does not favor GameScaling on hardware. The providers serve models of hundreds of billions to trillions of parameters on production clusters with far more compute than the single H200 behind GameScaling-0.8B and 2B or the four behind 9B and 27B. Even so, Table 9 shows frontier models taking 4.0–23.7 s per step, against 0.13–0.69 s for GameScaling; GameScaling-0.8B and 2B decide within the 200 ms action chunk. The reason is output length: at their default reasoning efforts (Appendix B.4), counting hidden reasoning, visible `<think>` content, and the action, frontier models generate 123–1,027 tokens per step, 3 to 29 times the 35–39 of GameScaling, which learns the sparse reasoning ability by continued pre-training on the gameplay trajectories with sparse reasoning. Frontier models are slow in real time because they are large and reason at length before every action. Even the fastest of them takes about 20 times the 200 ms action chunk per step. GameScaling acts within a second on one to four GPUs, which makes it the practical choice when both competence and latency count.

C.6 More Experiments on Scaling Law

This section details the fits behind Section 4.4. We first describe the fitting procedure and check which losses decrease with training at all. We then derive Eq. (3) from the Chinchilla law, compare it with a categorical-mixture model, and report hold-out predictions and bootstrap uncertainty.

Fitting procedure. Following Chinchilla (Hoffmann et al., 2022), we minimize the Huber loss of the log residual $\log \hat{L} - \log L$ with $\delta = 10^{-3}$, so that errors are relative and single outliers do not dominate. The optimizer is L-BFGS-B, started from 192 initializations for Eq. (3) and 324 for the two-factor law, and the best optimum is kept. E, A, B and C are optimized in log space; the remaining parameters are bounded by $\alpha, \beta \in [0, 3]$, $\eta \in [-3, 3]$, $\gamma \in [0, 5]$ and $\epsilon \in [10^{-4}, 1]$. N is measured in billions of parameters and T in gameplay hours. Unless a subset is stated, validation-

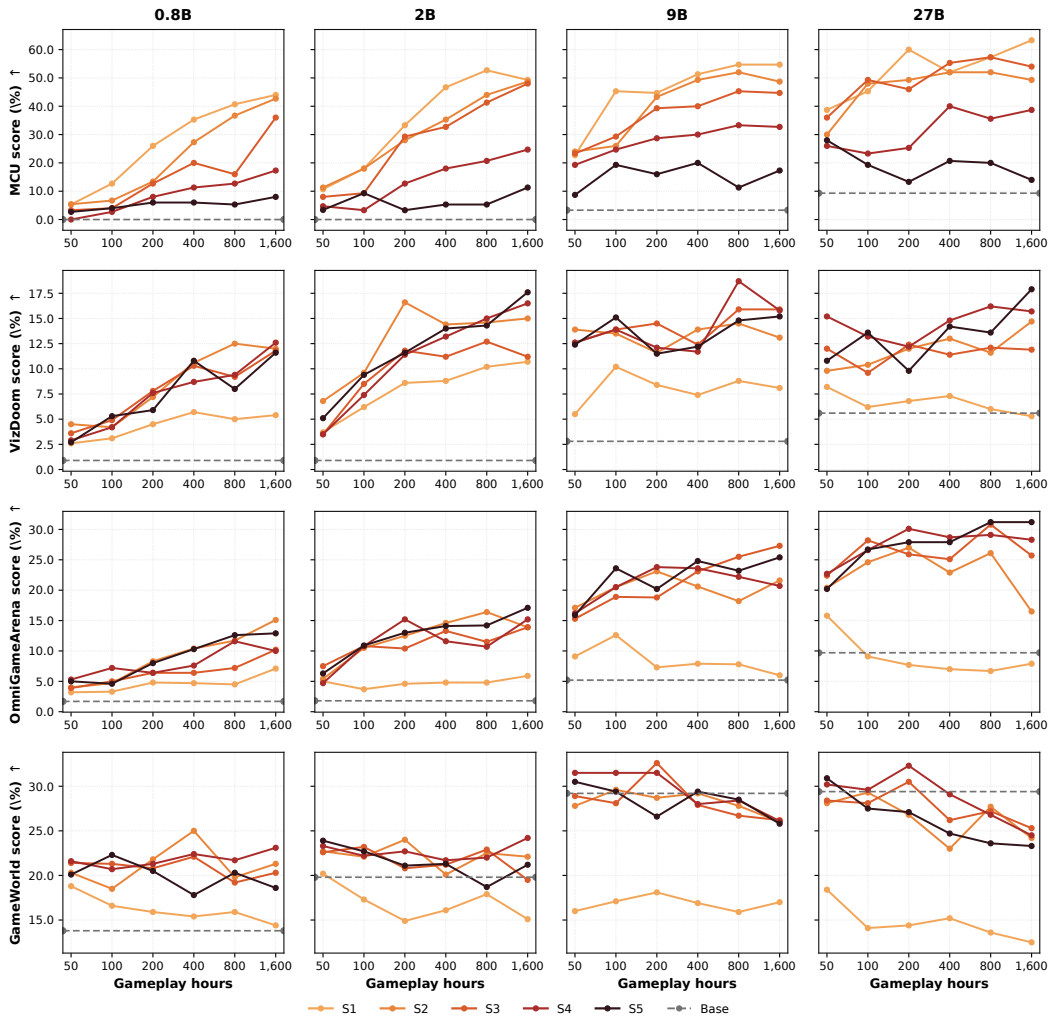

Figure 8: **Benchmark scaling against gameplay hours under time-paused interaction.** Rows: MCU, ViZDoom, OGA, and GameWorld (all in %); columns: 0.8B, 2B, 9B, and 27B backbones; points span 50 to 1,600 gameplay hours; colors: mixtures S1–S5; dashed gray: the untrained Qwen3.5 backbone.

loss laws are fitted to all 120 conditions of Figure 3, namely four sizes, five mixtures and six budgets from 50 to 1,600 h. Each 50 h validation point comes from its own annealing branch, like every other budget. The training loss is the 100-step EMA of the WSD trunk and is fitted at the five stable-phase budgets from 100 to 1,600 h, giving 100 conditions. At 50 h the trunk has just left its 3% warmup, and its loss still falls faster than any power law in T.

Besides RMSE and R^2 over all conditions, we report the within-mixture R^2_{w}, which is the share of variance explained after the five mixture means are removed. Between-mixture differences account for 79%, 88% and 84% of the total variance of the Minecraft, in-corpus and OOD losses. Any model with a mixture-dependent level therefore already reaches a high R^2, and R^2_{w} measures what the N and T terms add on top of it. Two further points matter when reading the exponents. First, the objective is the Huber loss rather than R^2, so a parameter value that maximizes R^2 along a profile need not equal the estimate. Second, since ρ takes five values, we draw only two qualitative conclusions from the ρ-dependent part of Eq. (3), namely that a world with $\rho = 0$ has a finite floor and that $\eta > 0$, and we do not interpret γ or ϵ on their own.

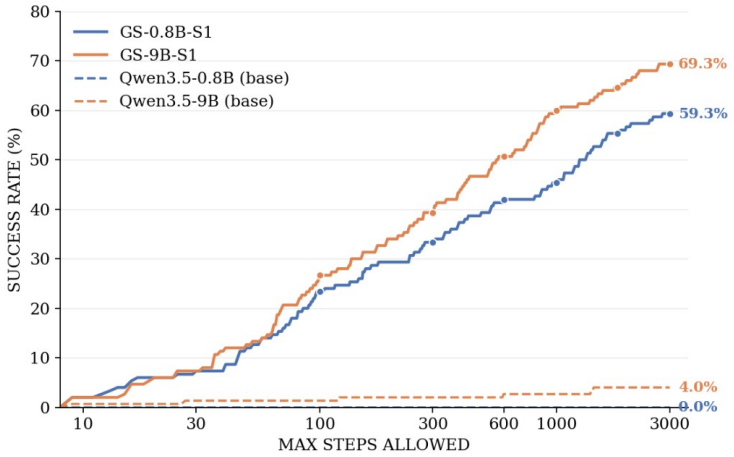

Figure 9: **Test-time scaling on MCU.** Cumulative success rate of GameScaling-0.8B-S1 and GameScaling-9B-S1 (1,600 h, solid) and their untrained Qwen3.5 backbones (dashed) on the 30 MCU tasks as a function of the maximum steps allowed, on a logarithmic axis. The standard protocol uses 600 steps.

Table 9: **Per-step inference latency and output length** of the Table 2 models, averaged over MCU, ViZDoom, OGA and GameWorld. Latency is the round trip of the model request, excluding screenshot capture and client-side image encoding; frontier models are called through the provider API, and GameScaling (S3, 1,600 h) runs under real-time interaction on the deployment of Table 8. Output length is the `completion_tokens` reported by the server, covering hidden reasoning, visible `<think>` content, and the action.

Model	Params	Latency (s/step)	Output tokens/step
Gemini-3.8-Flash	∼T	7.75	444
GPT-5.6-Sol	∼T	5.57	123
GPT-5.6-Luna	∼T	5.84	144
Claude-Opus-5.0	∼T	8.59	183
GLM-5.3-Flash	320B	23.66	1,027
DeepSeek-V4-Flash-Vision	305B	3.97	659
GameScaling-0.8B	0.8B	0.13	35
GameScaling-2B	2B	0.16	35
GameScaling-9B	9B	0.31	39
GameScaling-27B	27B	0.69	36

Which losses admit a scaling law. A power law in T presupposes that loss falls with hours, so before fitting we check this on the raw curves, one per model size and mixture, or 20 per loss. On the Minecraft loss, 19 of the 20 curves end lower at 1,600 h than at 50 h, and all curves without a clear decrease belong to S5, which contains no Minecraft. On the in-corpus loss, the four rising curves are exactly those of S1, which contains no pool games. A world's loss thus decreases with hours whenever the world is in the corpus, and the exceptions are the mixtures that lack it. This is the pattern that the share ρ and the floor term of Eq. (3) have to describe.

The OOD loss behaves differently in every cell. CPT lowers it from 0.24–0.31 for the backbone to about 0.17, but this gain is already realized at 50 h, and further hours raise the loss again. All 20 curves end 0.007–0.019 higher at 1,600 h than at 50 h, the rank correlation between hours and loss is positive in all 20 with a median of 0.91, the loss rises over 76 of the 100 doublings, and the per-curve slope ranges from $+0.001$ to $+0.004$ per doubling. The OOD loss therefore violates the premise of a data scaling law before any law is fitted. We do not model its dynamics, because a form that can follow a rising curve, such as one with an added $D \log T$ term, would describe drift rather than scaling. Instead, we fit Eq. (3) to the OOD loss only as a negative control, to test whether a

nine-parameter law manufactures a data term out of mixture differences alone. As shown below, it does not: β goes to zero, as the raw curves imply it should.

Fitted parameters. Table 10 lists the point estimates behind the last column of Figure 3; their uncertainty is analysed at the end of this section. The reading that survives this analysis is structural: the floor term $C/(\rho+\epsilon)^\gamma$ produces the gap between mixtures with and without a world. On the Minecraft loss, going from $\rho = 1$ to $\rho = 0$ raises the floor term by 0.08 and removes a data term worth 0.05 at 100 h. The net increase of 0.03 matches the observed gap of 0.032 between S5 and S1. On the in-corpus loss the corresponding numbers are 0.08, 0.04 and 0.04, against an observed gap of 0.040 between S1 and S2. Within mixtures, where the floor term is constant and only N and T act, the law still explains 91% of the variance of the Minecraft loss and 96% of the in-corpus loss, compared with 8% of the OOD loss. The individual exponents are less certain. α is of order 0.2 for both validation losses but has wide intervals, β trades off against the offset E along a ridge, and on the OOD loss β sits on its lower bound.

Table 10: **Parameters of the laws in Figure 3, column 5.** The validation losses follow Eq. (3) with the stated ρ and are fitted on 120 conditions. The training loss follows $L = E_S + A/N^\alpha + B/T^\beta$, with E_S between 0.005 and 0.011 across S1–S5, and is fitted on the 100 stable-phase conditions. R^2_{w} is the within-mixture coefficient of determination.

Loss	ρ	E	A	α	B	β	η	C	γ	ϵ	R^2	R^2_{w}	RMSE
Training	–	E_S	0.069	0.39	0.400	0.07	–	–	–	–	0.980	0.980	0.0041
Minecraft	r	0.047	0.019	0.15	0.106	0.16	0.24	0.0987	0.13	0.011	0.982	0.912	0.0020
In-corpus	$1-r$	0.136	0.014	0.17	0.078	0.14	0.10	0.0031	1.00	0.039	0.995	0.959	0.0012
OOD	$1-r$	0.112	0.036	0.03	0.006	0.00	3.00	0.0213	0.26	0.014	0.855	0.079	0.0058

From Chinchilla to Eq. (3). The Chinchilla law $L = E + A/N^\alpha + B/D^\beta$ assumes that a single distribution is both trained on and evaluated, so that D counts data from the scored distribution, and that loss falls monotonically with D. Our grid trains on mixtures and scores four distributions, which breaks the first assumption, and the OOD loss breaks the second. Like the Chinchilla and D-CPT laws, Eq. (3) is an empirical description rather than a law derived from a model of learning, and we make no claim beyond the grid. Its components are nevertheless not free design choices. The share ρ is neither fitted nor tuned but fixed by how the mixtures are built (Appendix A.1). The functional form is the published D-CPT law with its domain ratio replaced by ρ. Each term is forced by an observation that a simpler form cannot accommodate. We reached Eq. (3) by relaxing the Chinchilla assumptions one at a time on the Minecraft loss, each step guided by the residuals of the previous one (Table 11). The form is then tested against a categorical model of the same size and on held-out blocks, including held-out mixtures.

1. *Total hours, $D = T$.* On S1–S3 the fit reaches only $R^2 = 0.925$. The three mixtures share every value of T, yet their losses differ by up to 0.007, which the law cannot separate. Total hours are therefore not the data variable of an evaluated world.
2. *Minecraft hours, $D = rT$.* R^2 rises to 0.959, but the residuals are ordered by mixture, from $+0.0016$ for S1 to -0.0012 for S3. At equal Minecraft hours, the mixture with more pool hours has the lower Minecraft loss, so hours on other worlds also lower a world's loss, at a lower rate.
3. *Two data terms.* Adding a shared term in total hours, $L = E + A/N^\alpha + B/(rT)^\beta + C/T^\gamma$, removes the ordering and reaches $R^2 = 0.989$; the fitted $\gamma \approx 0.5$ means that the shared term saturates early. This form is undefined at $r = 0$, however, whereas S4 and S5 reach finite Minecraft losses of 0.229 and 0.236, far below the backbone. A world absent from training thus has a floor rather than an infinite loss, and the share must enter the law in a form that admits $\rho = 0$.
4. *Eq. (3).* The D-CPT form (Que et al., 2024), with D replaced by T and the domain ratio by ρ, covers all five mixtures with one parameter set and reaches $R^2 = 0.982$. Its RMSE is at most 0.0015 on S1–S3 and 0.003 on S4–S5, where only two mixtures anchor the floor. The shares are the measured Minecraft trajectory shares of 100, 51.4, 27.1, 2.8 and 0% from Appendix A.1, not the nominal VPT fractions.

5. *What the fitted form says.* η is positive, 0.24 for Minecraft and 0.10 for the pool, so the data term does not reduce to effective hours ρT. Forcing $\eta = -\beta$, which gives $B/(\rho T)^\beta$, lowers R^2 to 0.825 on the Minecraft loss and 0.945 on the in-corpus loss, and to 0.923 and 0.953 when the $\rho = 0$ mixture is excluded. Constraining $\eta \leq 0$ on the $\rho > 0$ conditions alone drives η to the bound and lowers the Minecraft R^2 from 0.979 to 0.934. The sign of η follows from the shape of the curves rather than from how the shares are labelled. Mixtures with small ρ are nearly flat in T while $\rho = 1$ decays, so the amplitude of the T term must grow with ρ. At 0.8B, for example, the S5 Minecraft loss moves only from 0.239 to 0.237 over 50–1,600 h, while the S1 loss falls from 0.220 to 0.198. This is step 2 seen from inside the final law: the floor term, not the data term, carries the mixture dependence.
6. *Second evaluation world.* With $\rho = 1 - r$ the same form describes the in-corpus loss, while the ranking of the restricted baselines inverts. $D = T$ fits S2–S5 well, with $R^2 = 0.991$, because the four diverse mixtures have almost identical in-corpus losses. $D = (1 - r)T$ is worse, at 0.955, because Minecraft hours also lower the loss on pool games. Only Eq. (3) admits S1, where $\rho = 0$. S1 remains the largest residual, with RMSE 0.0024 against at most 0.0007 for the other mixtures, because its in-corpus loss rises with hours, which no monotone law can reproduce.
7. *Training and OOD loss.* The training loss is scored on each mixture's own stream, so ρ is undefined and the law reduces to Chinchilla with a per-mixture offset E_S. On the OOD loss monotonicity already fails in the raw curves, so Eq. (3) is applied there only as a negative control. Replacing it with a form that fits a rising loss would no longer yield a scaling law.

Two alternatives were set aside. A weighted sum of two Chinchilla laws, one per validation slice with $D = rT$ and $D = (1 - r)T$, is easier to interpret but diverges at $r = 0$ and $r = 1$, so it covers only S2–S4. Treating the number of worlds as a variable cannot be tested on this grid, because S2–S5 share one fixed pool.

The Minecraft slice of the in-corpus set. The in-corpus set contains 30 pool games and Minecraft with 160 clips each, so Minecraft makes up $\pi = 1/31 = 3.2\%$ of it and $\rho = 1 - r$ is an approximation. To measure its effect, we write the measured loss as $L_{\text{in}} = \pi L_{\text{mc}} + (1 - \pi)L_{\text{pool}}$ and remove the Minecraft slice using the measured Minecraft loss. Relative to L_{in}, this shifts L_{pool} by $+0.0007$ for S1 and by -0.0007 to -0.0017 for S2–S5. The shift is largest for S4 and S5, whose Minecraft loss is highest, so the slice slightly narrows the gap between S1 and the diverse mixtures. Fitting Eq. (3) with $\rho = 1 - r$ to L_{pool} gives $R^2 = 0.995$ and the same per-mixture RMSEs. Combining the two laws as $\pi \hat{L}_{\text{mc}} + (1 - \pi)\hat{L}_{\text{pool}}$ predicts L_{in} with $R^2 = 0.995$ and RMSE 0.0013. Replacing ρ by the validation-weighted share $\pi r + (1 - \pi)(1 - r)$ fits worse, with $R^2 = 0.989$, so the Minecraft slice enters additively rather than through the share.

Table 11: **Restricted scaling-law baselines against the three-factor law.** Each baseline drops the share ρ and keeps only N and one data variable D (the two-factor Chinchilla law); "two-phase" adds a second data term in total hours. n is the number of fitted conditions.

Loss	Law	Conditions	n	R^2	RMSE
Minecraft	Chinchilla, $D = T$ (total hours)	S1–S3	72	0.925	0.0024
Minecraft	Chinchilla, $D = rT$ (Minecraft hours)	S1–S3	72	0.959	0.0018
Minecraft	Two-phase, $D = rT$ and T	S1–S3	72	0.989	0.0009
Minecraft	Eq. (3), $\rho = r$	S1–S5	120	0.982	0.0020
In-corpus	Chinchilla, $D = T$	S2–S5	96	0.991	0.0006
In-corpus	Chinchilla, $D = (1 - r)T$ (pool hours)	S2–S5	96	0.955	0.0014
In-corpus	Eq. (3), $\rho = 1 - r$	S1–S5	120	0.995	0.0012
OOD	Chinchilla, $D = T$	S2–S5	96	0.117	0.0055
OOD	Eq. (3), $\rho = 1 - r$	S1–S5	120	0.855	0.0058

Eq. (3) against a categorical-mixture model. With only five levels on the mixture axis, the natural alternative to a parametric ρ is to treat the mixture as a factor. Table 12 compares Eq. (3) with the *categorical offset* model $L = E_S + A/N^\alpha + B/T^\beta$, which gives every mixture its own offset and

has the same nine parameters as Eq. (3). Both are fitted with the same procedure on the same 120 conditions. We compare them by Gaussian AIC and BIC on the loss residuals and by blocked cross-validation, holding out one block of the grid at a time: the 27B models, the 1,600 h checkpoints, the 50 h checkpoints, or one mixture.

The comparison supports two conclusions. First, the mixture effect is not a mere offset. The categorical offset model has about twice the RMSE of Eq. (3) on both losses, 0.0044 against 0.0020 on Minecraft and 0.0035 against 0.0012 in-corpus. Its R^2_{w} drops to 0.56 and 0.64, and BIC prefers Eq. (3) by 190 and 260. What the offset misses is that the slope in T depends on the share. The Minecraft loss falls by 0.0045 per doubling of hours under S1, by 0.0013 under S4 and not at all under S5, and the in-corpus loss falls by 0.0035 per doubling under S2–S5 but rises under S1. Eq. (3) captures this interaction through η and the floor term.

Second, only Eq. (3) can predict a mixture it has not seen, because ρ enters as a covariate rather than a label. It predicts a held-out S2 or S3 with RMSE 0.0020–0.0021 on the Minecraft loss and 0.0008–0.0016 on the in-corpus loss, whereas the categorical model has no parameter for such a fold. With the same number of parameters, Eq. (3) therefore both fits better and generalizes across mixtures.

Table 12: **Eq. (3) against a categorical-mixture model of the same size.** Both models are fitted on the 120 conditions with the Huber-log objective, and k is the number of parameters. AIC and BIC use the Gaussian likelihood of the loss residuals, where lower is better. CV columns give the RMSE on the held-out block after refitting on the rest; "–" marks folds on which the model has no parameter for the unseen mixture.

Loss	Model	k	R^2	R^2_{w}	RMSE	AIC	BIC	CV 27B	CV 1,600 h	CV 50 h	CV S2	CV S3	CV S4
Minecraft	Eq. (3), $\rho = r$	9	0.982	0.912	0.0020	-1135	-1110	0.0026	0.0027	0.0025	0.0020	0.0021	0.0057
Minecraft	Categorical offset E_S	9	0.909	0.562	0.0044	-942	-917	0.0049	0.0071	0.0070	–	–	–
In-corpus	Eq. (3), $\rho = 1 - r$	9	0.995	0.959	0.0012	-1257	-1232	0.0016	0.0018	0.0022	0.0016	0.0008	0.0005
In-corpus	Categorical offset E_S	9	0.958	0.636	0.0035	-996	-971	0.0037	0.0051	0.0059	–	–	–

Negative control on the OOD loss. Since the OOD loss rises with hours in every cell, a law with a decreasing data term should find no such term, and both laws behave accordingly. The two-factor law fitted to S2–S5 reaches $R^2 = 0.117$ with $B \to 0$ and degenerates to a function of size only. Eq. (3) raises the nominal R^2 to 0.855, but only through the floor term, which absorbs the 0.035 gap between S1 and the diverse mixtures. Within mixtures it explains 8% of the variance, β is driven to its lower bound in 97% of the bootstrap replicates of Table 14, and the predicted loss is flat in T. The residuals therefore carry the rise seen in the raw curves. Averaged over sizes, the observed loss increases by 0.002–0.003 per doubling under S2–S5, from 0.0016 at 27B to 0.0031 at 0.8B, and by 0.003–0.004 under S1. $\beta \approx 0$ is thus exactly the outcome the control is designed to detect, not a failure to fit: the flexibility of Eq. (3) does not create a data-scaling term where the data contain none. Within 50–1,600 h, CPT hours do not scale the loss on worlds absent from the corpus.

Hold-out predictions. Table 13 refits Eq. (3) on the grid with one block of conditions held out and predicts that block, covering all three axes of the grid. Along the size axis, a fit on 0.8B–9B predicts the 27B models with RMSE 0.0026 on the Minecraft loss and 0.0016 on the in-corpus loss. Along the data axis the law extrapolates in both directions. A fit on 50–800 h predicts the 1,600 h checkpoints with RMSE 0.0027 and 0.0018, and a fit on 100–1,600 h predicts the 50 h checkpoints with RMSE 0.0025 and 0.0022. These errors are 1.3–1.8 times the in-sample RMSE, and the mean residuals of these size and data blocks stay below 0.0014, so the extrapolated predictions carry no systematic bias. Along the diversity axis, the law interpolates to a held-out S2 or S3 with RMSE 0.0020–0.0021 on the Minecraft loss and 0.0008–0.0016 on the in-corpus loss. Leaving out S4 is harder for the Minecraft loss, with RMSE 0.0057, because S5 is then the only mixture near $\rho = 0$; the in-corpus loss of S4 is still predicted with RMSE 0.0005. Eq. (3) therefore interpolates between the observed shares and extrapolates about one doubling in model size and in either direction of the data budget.

Uncertainty of the fits. The grid has a single training seed, so the resampling below measures how much each fit depends on which conditions enter it, given the measured losses. It does not measure

Table 13: **Hold-out predictions of Eq. (3).** Each row refits the law on all conditions outside the held-out block and evaluates it on the block. The in-sample RMSE on all 120 conditions is 0.0020 for the Minecraft loss and 0.0012 for the in-corpus loss. Held-out S4 Minecraft losses span a narrow range, so R^2_{test} is not informative there and is omitted.

Axis	**Fit → predict**	**Loss**	n_{fit}	n_{test}	R^2_{test}	RMSE	Mean residual
Size	0.8B–9B → 27B	Minecraft	90	30	0.971	0.0026	+0.0013
		In-corpus	90	30	0.991	0.0016	+0.0008
Data	50–800 h → 1,600 h	Minecraft	100	20	0.978	0.0027	+0.0003
		In-corpus	100	20	0.992	0.0018	+0.0004
	100–1,600 h → 50 h	Minecraft	100	20	0.916	0.0025	+0.0007
		In-corpus	100	20	0.967	0.0022	+0.0009
Diversity	S1,S3–S5 → S2	Minecraft	96	24	0.945	0.0020	+0.0013
		In-corpus	96	24	0.939	0.0016	+0.0015
	S1,S2,S4,S5 → S3	Minecraft	96	24	0.929	0.0021	−0.0008
		In-corpus	96	24	0.986	0.0008	−0.0007
	S1–S3,S5 → S4	Minecraft	96	24	–	0.0057	−0.0021
		In-corpus	96	24	0.993	0.0005	−0.0002

the seed-to-seed variance of the losses themselves. We use two schemes with $B = 300$ replicates each and refit every replicate from the full-data estimate plus eight perturbed starts. The *stratified case bootstrap* resamples conditions with replacement within each mixture, so every replicate keeps all five mixtures. Without stratification a replicate can lose most of S5, the only mixture at $\rho = 0$, and γ and ϵ then diverge for reasons unrelated to the model. The *residual bootstrap* keeps every (N, T, ρ) and resamples the fitted residuals. It preserves the factorial design but assumes exchangeable residuals, which the systematic S1 in-corpus residual violates. Table 14 reports 95% percentile intervals from the case scheme. The residual scheme gives intervals of similar width for every parameter of the training, Minecraft and in-corpus laws. For hold-out errors, only the fitted conditions are resampled, and the held-out conditions stay fixed.

Table 14: **Uncertainty of the three-factor laws.** Each entry is the point estimate with a 95% percentile interval over $B = 300$ stratified case-bootstrap replicates. "RMSE on grid" evaluates each replicate law on the original conditions and thus measures how much the predicted surface moves. "Stable β" is the range of fixed β for which refitting the other parameters keeps R^2 within 0.001 of the unconstrained fit, together with the resulting range of E. Hold-out rows refit each law on the stated conditions and report RMSE and mean residual on the fixed held-out conditions; their point estimates are the fits of Tables 12 and 13.

	Training	**Minecraft**	**In-corpus**
α	0.39 [0.31, 0.53]	0.15 [0.02, 0.30]	0.17 [0.06, 0.30]
β	0.070 [0.064, 0.078]	0.16 [0.12, 0.27]	0.14 [0.08, 0.20]
η	–	0.24 [0.18, 0.35]	0.10 [0.03, 0.18]
γ	–	0.13 [0.10, 0.26]	1.00 [0.86, 1.82]
E	E_{S1} 0.005 [0.001, 0.009]	0.047 [0.013, 0.114]	0.136 [0.095, 0.149]
R^2	0.980 [0.973, 0.986]	0.982 [0.975, 0.988]	0.995 [0.993, 0.997]
RMSE on grid	0.0041 [0.0042, 0.0049]	0.0020 [0.0019, 0.0024]	0.0012 [0.0012, 0.0013]
Stable β (profile)	[0.07, 0.12], $E_{S1} \in [0.003, 0.11]$	[0.15, 0.50], $E \in [0.04, 0.15]$	[0.03, 0.30], $E \in [0.01, 0.15]$
Hold-out ≤9B → 27B: RMSE	0.0092 [0.0059, 0.0129]	0.0026 [0.0019, 0.0035]	0.0016 [0.0014, 0.0019]
mean residual	+0.0078 [+0.0019, +0.0117]	+0.0013 [+0.0004, +0.0022]	+0.0008 [+0.0003, +0.0012]
Hold-out ≤800 h → 1,600 h: RMSE	0.0092 [0.0065, 0.0141]	0.0027 [0.0023, 0.0039]	0.0018 [0.0017, 0.0023]
mean residual	−0.0081 [−0.0128, −0.0048]	+0.0003 [−0.0007, +0.0013]	+0.0004 [−0.0001, +0.0011]
Hold-out 100–1,600 h → 50 h: RMSE	–	0.0025 [0.0023, 0.0034]	0.0022 [0.0019, 0.0024]
mean residual	–	+0.0007 [−0.0002, +0.0014]	+0.0009 [+0.0006, +0.0011]

Four conclusions follow. First, the predicted surfaces are stable even where individual parameters are not. Evaluated on the original grid, the replicate laws change the RMSE by at most 0.0004 for the Minecraft law and 0.0001 for the in-corpus law, and the 5–95% bands in Figure 10 are narrower than the markers over most of the grid.

Second, four model sizes determine α only loosely. Its Minecraft interval spans 0.02 to 0.30 because E, A and α compensate each other, so we only state that capacity enters with an exponent of order 0.2 and do not compare α across losses.

Third, β and E lie on a ridge. Fixing β anywhere between 0.15 and 0.5 for Minecraft, or between 0.03 and 0.3 for the in-corpus loss, and refitting the rest keeps R^2 within 0.001 of the unconstrained fit, while E moves by 0.11–0.14. Along the Minecraft ridge R^2 even peaks at $\beta \approx 0.3$, above the Huber estimate of 0.16, as expected for a flat ridge under two different criteria. The asymptotic floor is therefore not a quantity this grid can determine, and we do not report it. For the training law the ridge is one-sided, with β between 0.07 and 0.12. Below 0.07 the offsets E_S reach their lower bound and R^2 drops steeply, so the estimate sits at the constrained end of the ridge, and we likewise do not interpret E_S as an asymptote.

Fourth, the hold-out errors of the two validation laws exceed their in-sample errors by a factor of 1.3–1.8, and their mean residuals lie within one in-sample RMSE of zero in both directions along the data axis. One doubling of size or hours is therefore a fair extrapolation range. The training law extrapolates worse, with RMSE 0.009 in both directions, and the signs of its residuals show why. The 1,600 h point is the end of the WSD decay, which lowers the trunk loss faster than any power law in T, and the 27B loss is higher than an α fitted on 0.8B–9B predicts. On the OOD loss, finally, β sits at its lower bound in 97% of case-bootstrap replicates while η and γ remain unconstrained, so the absence of a data term is a property of the data rather than of one optimizer run. Under the residual scheme the fraction drops to 51%, as expected, because resampling the residuals destroys exactly the monotone trend in T that they carry.

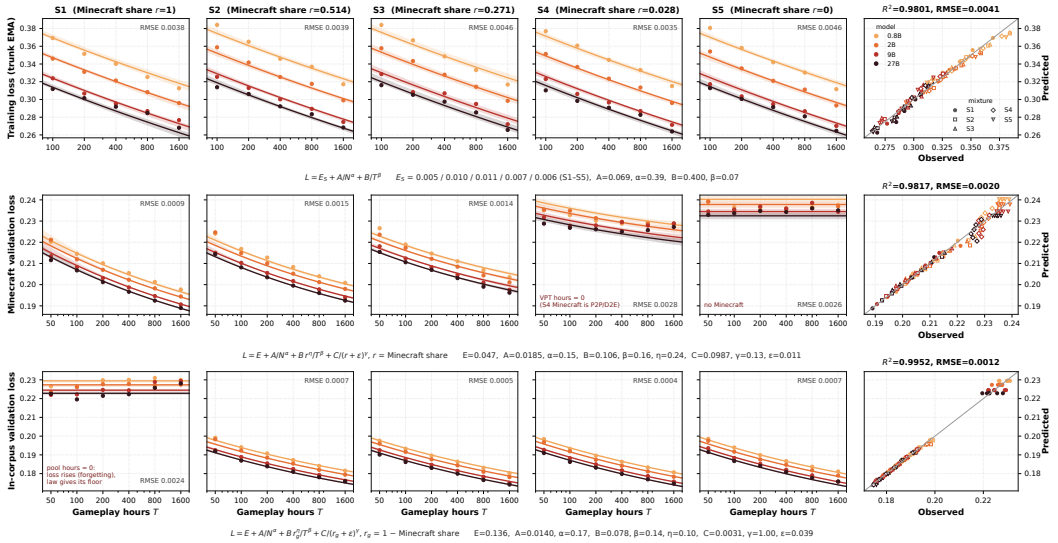

Figure 10: **Three-factor laws with prediction bands.** Rows are the training loss, the Minecraft validation loss and the in-corpus validation loss; columns are the mixtures S1–S5; colors are model sizes. The horizontal axis is gameplay hours, from 100 h for the training loss and from 50 h for the validation losses. Points are measurements, lines are the single fitted law of each row evaluated at that panel's (N, r), and shaded bands are the 5–95% range of the law over $B = 300$ stratified case-bootstrap replicates. The last column is predicted against observed loss; per-panel RMSE is printed.

D Comparison with Related Scaling Studies

GameScaling studies CPT across 120 conditions, combining four model sizes, six data budgets, and five world mixtures. We compare its training scale with language-model scaling studies, and its model range, data coverage, and experimental scope with GUI and robotics studies cited in Section 2.

D.1 LANGUAGE-MODEL SCALING STUDIES

D-CPT Law (Que et al., 2024) and CMR Scaling Law (Gu et al., 2024) study how data mixtures affect domain-adaptive CPT, while *When Data Is Scarce* (Wu et al., 2026) studies sparse models trained from scratch with repeated data. As shown in Table 15, GameScaling's reported 2.2×10^{22} FLOPs are about $4.7\times$ the estimated CMR budget and of the same order as D-CPT and the dense-equivalent estimate for *When Data Is Scarce*. Our controlled grid extends to 27B parameters and connects training scale with both held-out loss and performance in interactive worlds.

Table 15: **Training scale of language-model studies and GameScaling.** Budgets cover the experiments listed, excluding original backbone pre-training for CPT. Related-work budgets are estimates; GameScaling uses the reported totals in Section 3.

Study	Experiments covered	Tokens	FLOPs
D-CPT Law (Que et al., 2024)	0.5–4B; 6 domains; 9 mixtures (main grid)	$4.2 \times 10^{12\dagger}$	$5.3 \times 10^{22\dagger}$
CMR Scaling Law (Gu et al., 2024)	0.46–3.1B; 29 Finance and Academic Papers runs	$5.8 \times 10^{11\dagger}$	$4.7 \times 10^{21\dagger}$
When Data Is Scarce (Wu et al., 2026)	Up to 1.92B dense-equivalent; 500 fitting runs	$\sim 3.3 \times 10^{12\ddagger}$	$\sim 1.2 \times 10^{22\ddagger}$
GameScaling	0.8–27B; 120 conditions; 6 budgets, 5 mixtures	3.5×10^{11}	2.2×10^{22}

† Estimated from published training settings; D-CPT counts the nominal grid before possible reuse of general-only runs. ‡ Rough estimate from 500 fitting records with balanced model, data, and repetition settings; dense-equivalent FLOPs (ideal sparse: $\sim 3.7 \times 10^{21}$). Larger-model extrapolation tests are excluded. Estimates indicate scale, not an exact cost ranking.

D.2 GUI AND ROBOTICS SCALING STUDIES

Table 16 compares model sizes, data coverage, experimental variables, and training resources. GUI studies cover models up to 32B and large collections of interface examples or trajectories. The robotics studies listed here focus on data diversity, robot coverage, or training design at a fixed model size. GameScaling combines a broad model range with controlled changes in data volume and world mixture. Its approximately 45,000 H200 GPU-hours support a common protocol for measuring both scaling within training worlds and transfer to unseen related and distant worlds.

Table 16: **Scope of GUI, robotics, and virtual-world scaling studies.** GameScaling cost covers the full grid; other costs are per run. †: time estimate at unchanged recipe and throughput. Data use the original units; GameScaling hours include warmup, stable training, and decay.

Study	Models	Data scale	Main variables	Training resources
ScaleCUA (Liu et al., 2026)	3B, 7B, 32B	17.1M grounding; 471K understanding; 19K trajectories	Model size; data mixture	128 A100 (3B/7B); 128 H200 (32B)
GUIMid (Zhang et al., 2025)	7B	Up to 300K examples; $\sim$55K GUI samples	Task type; mixture; volume	8 GPUs; 1 epoch
OpenThoughts-Agent (Raoof et al., 2026)	8B, 32B	316–100K trajectories	Data quality; volume; $>$ 100 ablations	GH200 GPU-h: 160 (10K), $\sim$1,600† (100K); 8B SFT
Imitation data scaling (Lin et al., 2025)	$>$ 396M	$>$ 40K demonstrations; 4 tasks	Environments; objects; demos	8 A800; $\sim$64†–600 GPU-h
Rethinking VLA (Wang et al., 2026)	2B + 0.7B	182.5M transitions	Action format; schedule; regularization	8 A800; 40K+200K updates
Cross-Embodiment Offline RL (Abe et al., 2026)	URMA	16M transitions/variant; 16 robots	Data quality; robot grouping	100K updates; 5 seeds
GameScaling	0.8–27B	50–1,600 hours; 5 world mixtures	Model size; data volume; world mixture	45K H200 GPU-h; 120 conditions

VLA sizes list the VLM and action expert separately. GPU-hours across different device types are not directly comparable.