OPENWORLD: SCALING WORLD MODEL PRETRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

World models can simulate how physical scenes respond to actions, but how their performance scales with model size, data, and compute remains poorly understood. Most existing approaches to action-conditioned video modeling finetune fixed-size pretrained text-to-video models, limiting systematic scaling studies. We curate an action-video dataset comprising 4 million video clips (120 billion image tokens) spanning 10,000 hours of human, robot, and navigation data. We then introduce a fully open recipe, OpenWorld, that enables scaling models from 600M to 7.5B parameters. In a dedicated isoFLOP sweep of this architecture family, we find that held-out diffusion loss exhibits a strongly model-heavy compute frontier, with $N_{\text{opt}} \propto C^{0.94}$. Separately, a constant-learning-rate training-loss frontier fitted on models up to 1.02B parameters predicts the excluded 7.5B run's smoothed training loss to within 2.5% near its predicted optimal compute. Rollout evaluations show improved in-domain video fidelity and behavior-level trajectory matching with model size; the 7.5B checkpoint achieves the best Bridge V2 finetuning results. Standard paired video metrics can nevertheless be unreliable proxies for action following. We demonstrate real-time human interaction simulation, multi-view simulation across robot morphologies, and policy ranking using a robot leaderboard. Qualitative rollouts from OpenWorld-7.5B further show plausible gravity-driven motion, contact-rich interaction, and deformable-object dynamics.

1 INTRODUCTION

World models learn dynamics from data and could emulate complex physical interactions (Yang et al., 2023), with applications in gaming (Bruce et al., 2024), self-driving (Hu et al., 2023), and robotics (Quevedo et al., 2025). Action-conditioned video generation may scale with data and model size similarly to text-to-video generation (Ho et al., 2022; Villegas et al., 2022; Blattmann et al., 2023b; Wan et al., 2025). Growing egocentric human and robot datasets (Hoque et al., 2025) motivate studying these scaling behaviors and the capabilities they enable.

Most existing approaches to action-conditioned video world modeling finetune fixed-size pretrained text-to-video models on relatively small action datasets (Agarwal et al., 2025; Guo et al., 2026; Zhang et al., 2025), assuming action-free video (e.g., YouTube) is more abundant than action-rich video (Bruce et al., 2024). Growing in-the-wild egocentric human and robot datasets may challenge this assumption and approach pretraining scale (Hoque et al., 2025; Luo et al., 2025; Kareer et al., 2025). Starting from fixed-size checkpoints limits model-size choices and makes scaling effects difficult to separate from those of the finetuning recipe. We therefore ask how to allocate compute between model capacity and training-token exposure, whether smaller-model scaling trends extrapolate to larger runs, and which metrics reliably measure progress.

We introduce OpenWorld to study action-conditioned video world model pretraining on a heterogeneous corpus (Figure 1). The corpus contains 4 million clips covering egocentric human hand movements, diverse robot morphologies, and embodied navigation, with wrist-mounted, first-person, and third-person views and actions including human poses, end-effector controls, robot states, and navigation controls. We hypothesize that pretraining on diverse interactions helps the model learn a general mapping from actions to changes in the visual environment. We use latent video diffusion with a Diffusion Transformer backbone (Peebles & Xie, 2023) and a unified continuous action interface, studying training stability as model size grows from 600M to 7.5B parameters.

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.

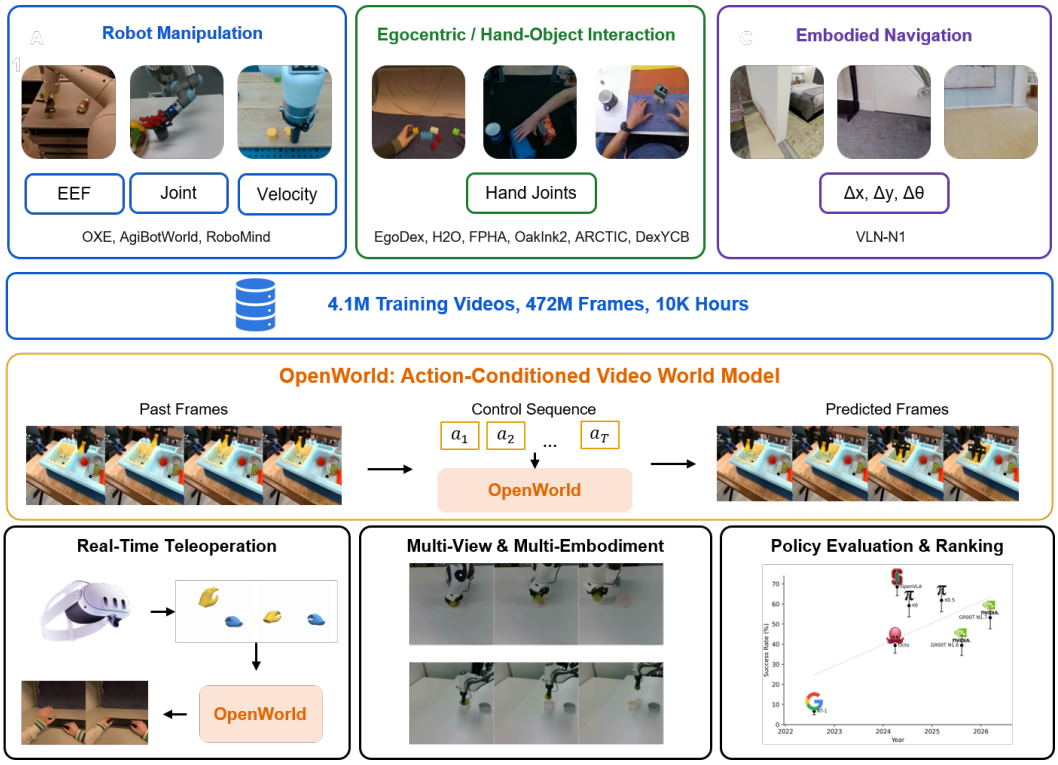

Figure 1: **Overview of OpenWorld.** OpenWorld is pretrained on a large heterogeneous action-video corpus spanning robot manipulation, egocentric hand-object interaction, and embodied navigation. Conditioned on past frames and a control sequence, the model predicts future frames and enables applications such as real-time teleoperation, multi-view/multi-embodiment simulation, and policy evaluation.

Our experiments reveal scaling trends in action-conditioned video world models. In our primary isoFLOP sweep, held-out diffusion loss yields a strongly model-heavy frontier, with $N_{\text{opt}} \propto C^{0.94}$, indicating that most additional compute in our setting is best allocated to model capacity rather than additional token exposure. Separately, a constant-learning-rate training-loss frontier fitted on models up to 1.02B parameters extrapolates to the excluded 7.5B run: near the predicted optimal compute, the 7.5B model's smoothed training loss is within 2.5% of the extrapolated frontier. A first-epoch analysis of a non-overlapping stream yields a consistent held-out-loss exponent of 0.89, providing a robustness check under a different sampling regime (Appendix C.3). Rollout evaluations show improved in-domain video fidelity and VLM-judged out-of-distribution trajectory matching across the 600M, 2.4B, and 7.5B checkpoints; the 7.5B model also achieves the best Bridge V2 finetuning results. At the same time, our experiments show standard paired video metrics can be too flat or noisy to precisely determine the scaling frontier, motivating the use of behavior-level evaluation for action following.

Our main contributions are:

- **Fully open data, models, and code.** We provide resources for pretraining action-conditioned video world models. The dataset contains 4 million clips spanning 10 thousand hours and 120 billion image tokens, with interactions involving 27 robot embodiments and humans. The models have 600M, 2.4B, and 7.5B parameters.
- **A systematic study of scaling.** We examine the effects of model size, data, and compute on video world models. We find a strongly model-heavy held-out-loss frontier, $N_{\text{opt}} \propto C^{0.89\text{–}0.94}$, and show that a separate training-loss frontier fitted on smaller models predicts the smoothed training loss of the excluded 7.5B run to within 2.5% near its predicted optimal compute.

- **An evaluation study of action-conditioned rollouts.** We compare in-domain video fidelity, downstream finetuning, and VLM-judged out-of-distribution trajectory matching across model sizes, and show that standard paired video metrics can be unreliable proxies for action following.
- **A suite of downstream applications.** We demonstrate real-time human interaction simulation, multi-view robot simulation, policy evaluation in the cloud, and qualitative simulations of gravity-driven motion, contact, and deformable-object dynamics.

2 Background and Related Work

Dynamics Models of the Physical World. World models predict visual dynamics under actions (Ha & Schmidhuber, 2018; Hafner et al., 2018) to support planning (Allen & Koomen, 1983; Du et al., 2023), behavior evaluation (Quevedo et al., 2025; Guo et al., 2026; Tseng et al., 2025), and learning from imagined experience (Hafner et al., 2019; Yang et al., 2023; Sharma et al., 2026). Early latent-space models learned control from predictive states instead of raw pixels (Ha & Schmidhuber, 2018; Hafner et al., 2018; 2019; 2023), later supporting real-world robot locomotion, manipulation, and navigation (Wu et al., 2023). JEPA-style models predict future representations from pixels (Assran et al., 2023; Bardes et al., 2024; Assran et al., 2025); semantically structured features simplify dynamics modeling but typically do not directly yield photorealistic rollouts. Other models use geometry or explicit memory to predict unseen viewpoints or maintain persistent 3D maps during rollout (Koh et al., 2021; Zhou et al., 2025a). We study how action-conditioned video world models scale with model size, data, and compute, including models with billions of parameters.

Video Generation as World Models. Text-to-video generation improves with large curated datasets, bigger models, and stronger architectures (Blattmann et al., 2023a; Kondratyuk et al., 2023; Wan et al., 2025). Work in embodied AI uses future-observation prediction for pretraining and adapts text-to-video models to robotics and interactive environments through action conditioning (Agarwal et al., 2025; Bruce et al., 2024; Hu et al., 2023), often using lightweight adapters or frame-level conditioning (Guo et al., 2026; Yang et al., 2023; Rigter et al., 2024; Zhu et al., 2025b). Pretrained video models capture temporal consistency from large action-free corpora, but their fixed sizes limit scaling studies of action-conditioned world models. Other methods pretrain on action-free videos or learn latent actions before adapting to action-rich domains (Zhu et al., 2025a; Gao et al., 2025; Bruce et al., 2024). Growing action-rich datasets (Hoque et al., 2025; Luo et al., 2025; Kareer et al., 2025) motivate direct pretraining; DreamDojo (Gao et al., 2026) trains a human-pose-conditioned video model and adapts it to robotics. We study scaling by co-training on human and robot videos.

Scaling Foundation Model Pretraining. Language-model scaling studies characterize power-law improvements as model size, data, and compute increase (Kaplan et al., 2020; Hoffmann et al., 2022), changes in these tradeoffs under data constraints (Muennighoff et al., 2023), and properties that emerge at larger scales (Wei et al., 2022). Image and video diffusion Transformers show related scaling patterns linking model size, data, compute, loss, and training hyperparameters (Liang et al., 2024; Yin et al., 2025). The prevalence of physical interaction and the growth of egocentric data collection (Hoque et al., 2025; Luo et al., 2025) suggest that action-rich video could support much larger pretraining datasets. Yet physically grounded world-model scaling remains largely unexplored: prior studies focus on language or action-free generation, while most embodied work finetunes fixed-size text-to-video models. Our study examines scaling laws for action-conditioned video world models and evaluates their capabilities across model sizes.

3 Scaling Experiments

We study three questions: **(Q1)** how to allocate compute between model size and training tokens; **(Q2)** whether a scaling law fitted on smaller models predicts a substantially larger run's training loss; and **(Q3)** which metrics provide a stable scaling signal. We use held-out diffusion-loss sweeps for compute allocation, a separate training-loss fit for extrapolation to 7.5B, and rollout and trajectory-level metrics to assess downstream behavior. Appendix C.3 reports a first-epoch robustness analysis using a non-overlapping stream.

Table 1: Model families used in the scaling study. Sizes give the model count and parameter range. Tokens/step $= B \times T \times S$, with S=256 spatial latent tokens per frame. Sweep A uses a weighted sampler. The 7.5B run is excluded from all fits.

Family	Sizes	B	T	Tokens/step	Sampling
Sweep A	16 (9.8M–1.02B)	256	24	1,572,864	Repeated
Sweep B	10 (27M–1.2B)	128	10	327,680	First epoch
7.5B	Holdout	256	24	1,572,864	Repeated

3.1 Pretraining Data and Action Representation

We first describe the pretraining data and shared conditioning interface.

Diverse Action-Rich Video Data. Existing datasets for video world models generally focus on a particular domain such as gaming (Bruce et al., 2024), self-driving (Hu et al., 2023), navigation (Bar et al., 2025), or robotics (Quevedo et al., 2025; Guo et al., 2026). We hypothesize that pretraining on diverse video-action data could teach a general mapping from actions to visual changes in the environment. To test this hypothesis, we curate a large mixture of action-rich videos spanning three interaction domains: robot manipulation (Open X-Embodiment Collaboration, 2023; Bu et al., 2025; Wu et al., 2025), human egocentric and hand-object manipulation (Hoque et al., 2025; Kwon et al., 2021; Garcia-Hernando et al., 2018; Zhan et al., 2024; Fan et al., 2023; Chao et al., 2021), and embodied navigation (Wei et al., 2025). The mixture covers varied viewpoints (wrist-mounted, first-person, third-person) and action spaces (human poses, end-effector controls, robot states, navigation controls). The combined dataset contains 4 million video clips (120 billion image tokens) spanning 10,000 hours of interaction. Sweep A and deployment runs use 24-frame windows; Sweep B uses 10-frame windows. Frames are center-cropped and resized to 256×256. Sweep A and deployment pretraining use subset-balanced sampling to prevent the largest datasets from dominating training; Sweep B uses a non-overlapping stream. Appendix A describes the component datasets and corpus construction.

Ensuring Data Quality. For world modeling, videos should clearly show the effects of their paired actions. We manually remove videos and camera views that do not reliably show the acting hand, robot, or interaction effects, including views recorded too far from the interaction.

Representing Diverse Action Formats. Existing work either unifies action spaces (Yang et al., 2023; Gao et al., 2026) or learns embodiment-specific action encoders or controllers (Chen et al., 2026; Garrido et al., 2026). We hypothesize that high-dimensional continuous action vectors let a world model learn how diverse actions change the visual environment. For pretraining, we represent action and state conditioning as a 256-dimensional vector, apply per-dataset normalization, and zero-pad action vectors with fewer than 256 dimensions. This shared interface accommodates hand poses, end-effector controls, robot states, and navigation controls, and can incorporate new formats during downstream finetuning.

Multi-View Videos. Several datasets in our mixture provide multiple camera views, including egocentric, wrist, top, and third-person observations. We treat each camera stream as an independent training example, hypothesizing that larger models can learn to align views conditioned on the same action stream. This keeps pretraining compatible with single-view datasets while exposing the model to diverse viewpoints.

3.2 Scaling Study Setup

All scaling runs use the same action-conditioned latent-video Diffusion Transformer (DiT) family (Peebles & Xie, 2023) with factorized spatial and causal temporal attention; full architecture and training-stability details are in Appendix B.1.

Model families. Our primary scaling family, **Sweep A**, covers 16 DiT sizes from approximately 10M to 1.02B trainable parameters at effective batch size B=256 and T=24 frames per clip (Table 1). It uses the standard weighted sampler, so clips may be revisited as training progresses. Runs share the optimizer, data mixture, base learning rate and warmup, batching configuration, and sampler; model sizes and scheduled run lengths vary as listed in Appendix C.2. A **7.5B** run with the same effective batch size and clip length is excluded from all fits; its AgiBot preprocessing configuration differs (Appendix C.4). All parameter counts are actual trainable counts from the training configurations, not architectural approximations. We report the separate first-epoch Sweep B analysis, including its full configuration and results, in Appendix C.3.

Compute accounting. Let N be the number of trainable parameters and D the number of latent tokens consumed ($D = \texttt{step} \times B \times T \times S$). We use

$$C \approx 4ND \tag{1}$$

as the training-compute proxy. Relative to the standard dense-transformer rule $C \approx 6ND$, roughly one third of our parameters belong to layer-specific adaptive layer-normalization and conditioning pathways that do not incur the full per-token dense cost, giving the first-order correction $6ND \times \frac{2}{3} \approx 4ND$. Since all models in a sweep share one architecture family, replacing $4ND$ by $6ND$ rescales every compute budget by 1.5. The fitted exponents and selected model sizes at corresponding budgets are unchanged; the numerical compute coordinates, including $C^\star$, rescale by the same factor.

Losses. The training objective is the denoising mean-squared error of the latent video diffusion model (Appendix B.1). Our primary isoFLOP analysis uses the held-out loss $\mathcal{L}_{\text{eval}}$, evaluated under a separate protocol for each sweep (Appendix D.1). At each compute budget, we use this loss to identify the best-performing model size; fitting these optima across budgets gives the compute-optimal scaling law $N_{\text{opt}} \propto C^a$. For scaling-law fitting, the online training loss $\mathcal{L}_{\text{train}}$ is used only for the separate 7.5B extrapolation, where a dense loss trajectory is required. The two analyses are fitted independently.

Annealed (cooled-down) endpoints. Because the sweeps train at constant learning rate, intermediate checkpoints overestimate the loss attainable at their compute budget (Hoffmann et al., 2022; Hägele et al., 2024). Following the warmup-stable-decay recipe (Hu et al., 2024; Hägele et al., 2024), we therefore anneal the learning rate for each measurement point: for a target step $T^\star$ we restore the latest checkpoint at or before $0.9\,T^\star$, cosine-decay the learning rate by $10\times$ over the final 10% of steps, and save the endpoint at exactly $T^\star$, chosen as the nearest optimizer step to each isoFLOP budget. Each endpoint checkpoint is then evaluated on the held-out set. The held-out-loss isoFLOP analyses use 67 Sweep A and 49 Sweep B annealed endpoints. The separate 7.5B extrapolation uses constant-learning-rate training histories.

3.3 A Strongly Model-Heavy IsoFLOP Frontier

For Sweep A we select eight isoFLOP budgets spaced $1.5\times$ apart in $[1.25, 21.4] \times 10^{19}$, restricted to the range where the loss-versus-size slice contains an interior minimum bracketed by at least one model on each side. For each compute budget we define N_{opt} as the model size that minimizes the fitted diffusion-loss slice (a quadratic in $\log_{10} N$), with $D_{\text{opt}} = C/(4N_{\text{opt}})$, and we fit power laws $N_{\text{opt}} \propto C^a$, $D_{\text{opt}} \propto C^b$ (Figure 2). Additional metadata, FLOP accounting, and the first-epoch robustness analysis are provided in Appendix C.

The frontier is strongly model-heavy. The fitted Sweep A frontier is $N_{\text{opt}} \propto C^{0.94\pm0.02}$ and $D_{\text{opt}} \propto C^{0.06\pm0.02}$. Within the compute range studied, the fitted frontier therefore allocates most additional compute to model capacity rather than token exposure — substantially more than the approximately balanced allocation reported for compute-optimal language-model training (Hoffmann et al., 2022). The optimal token budget remains near corpus scale: D_{opt} stays close to 10^{11} latent tokens ($\approx 0.8\times$ the corpus) over a $17\times$ compute range. A separate first-epoch analysis of a non-overlapping stream yields a similar model-size exponent of 0.89 ± 0.03 (Appendix C.3). Because the sweeps differ in batch size, clip length, and sampler, this is a robustness check rather than a matched ablation of data repetition.

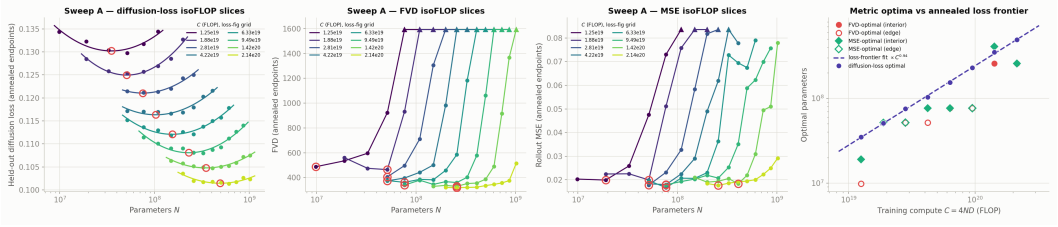

Figure 2: **Primary Sweep A isoFLOP analysis on annealed endpoints.** From left to right: held-out diffusion-loss, FVD, and rollout-MSE slices at the same eight budgets, followed by their per-budget optima. Red circles mark slice minima and triangles mark values clipped at the plot limit. Diffusion loss yields sharp interior minima whose optima (purple) lie tightly on the fitted power law, $N_{\text{opt}} \propto C^{0.94 \pm 0.02}$. FVD-optimal sizes follow the same broad trend with scatter; MSE-optimal sizes can be up to $\sim 3\times$ smaller than diffusion-loss optima. Both rollout metrics have flatter minima and less stable optimal-size estimates across budgets.

The Sweep A training-loss frontier extrapolates to 7.5B. A separate constant-learning-rate training-loss frontier fitted on the smaller Sweep A runs predicts that the excluded 7.5B model becomes optimal at $C^\star = 3.83 \times 10^{21}$ FLOPs (Figure 3). Its EMA-smoothed training loss comes within 2.48% of the frontier at 3.12×10^{21} FLOPs; at $C^\star$ the gap is 2.72%. This compute budget is $17.9\times$ the largest fitted budget, 2.14×10^{20} FLOPs. Both sides use constant-learning-rate training histories, distinct from the annealed held-out-loss analysis above. At larger compute, the 7.5B loss lies farther above the frontier. Its AgiBot preprocessing configuration differs from Sweep A's, so this comparison is not a fully matched data ablation; fitting and data-version details are in Appendix C.4.

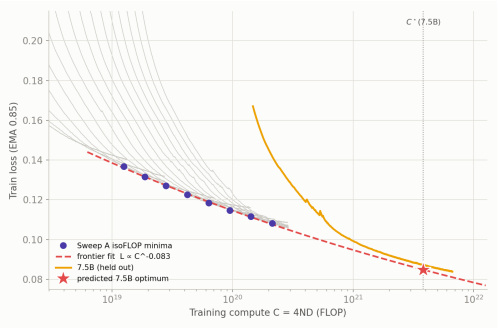

Figure 3: **Excluded 7.5B run versus the Sweep A training-loss frontier.** Gray curves show the Sweep A trajectories; purple points mark the isoFLOP minima; the dashed red line is the fitted frontier; the orange curve is the 7.5B trajectory; and the dotted vertical line marks the predicted $C^\star$.

3.4 Which Metrics Yield a Stable Scaling Frontier?

The usefulness of a scaling law depends on whether its evaluation metric yields a stable optimum at fixed compute. We observe a clear separation among the metrics considered here: diffusion loss produces sharp and consistent isoFLOP minima; rollout video metrics follow the same broad direction but have flatter and noisier slices; and trajectory-level VLM evaluation captures behavior-level differences among the deployment-scale checkpoints.

Rollout evaluation setup. Given a ground-truth clip and its action sequence, we initialize the model with the first frame, roll out autoregressively under the ground-truth actions, and compare the generated video to the paired real execution (Quevedo et al., 2025; Zhou et al., 2025b). On fixed-horizon held-out pretraining rollouts, we report paired frame-level metrics (MSE, PSNR (Hore & Ziou, 2010), SSIM (Hore & Ziou, 2010), LPIPS (Zhang et al., 2018)) and FVD (Unterthiner et al., 2018). On Bridge V2 finetuning rollouts, we report only paired frame-level metrics. The test set mixes horizons (13/24/49 frames), so a single FVD score would be confounded by rollout length. For OOD AutoEval rollouts, we use trajectory-level VLM matching as the primary behavioral metric. Further details on evaluation data, metrics, and the VLM evaluator are provided in Appendix D.

Rollout metrics provide directional corroboration. When we repeat the isoFLOP analysis on the Sweep A annealed endpoints using rollout video metrics (Figure 2), the model sizes preferred by

Table 2: **In-distribution rollout fidelity** across deployment checkpoints.

Model	MSE↓	PSNR↑	SSIM↑	LPIPS↓	FVD↓
600M	0.0123	19.10	0.744	0.230	115.536
2.4B	0.0098	20.07	0.765	0.178	102.179
7.5B	**0.0085**	**20.69**	**0.794**	**0.147**	**69.873**

Table 3: **Bridge V2 finetuning** across pretrained model sizes.

Model	MSE↓	PSNR↑	SSIM↑	LPIPS↓
600M	0.0141	18.519	0.813	0.120
2.4B	0.0135	18.691	0.811	0.114
7.5B	**0.0112**	**19.538**	**0.821**	**0.103**

Table 4: **Corpus-fraction comparison** at 2.4B on in-distribution rollouts.

Data	MSE↓	PSNR↑	SSIM↑	LPIPS↓	FVD↓
10%	0.0102	19.92	0.7577	**0.1666**	118.251
100%	**0.0101**	**19.96**	**0.7610**	0.1686	**107.153**

Table 5: **Bridge V2 adaptation at 2.4B**: pretraining versus training from scratch.

Initialization	MSE↓	PSNR↑	SSIM↑	LPIPS↓
Scratch	0.0141	18.502	0.797	0.127
Pretrained	**0.0135**	**18.691**	**0.811**	**0.114**

FVD and rollout MSE generally increase with compute. However, their isoFLOP slices are nearly flat around the minima, and the sweep's model-size spacing limits the precision of the estimated optima. We therefore fit the allocation law on diffusion loss and use rollout video metrics as directional cross-checks rather than as estimators of a precise scaling exponent.

Evaluation across model sizes. To connect objective-level scaling to downstream performance, we evaluate three deployment-scale checkpoints: 600M, 2.4B, and 7.5B. Larger deployment checkpoints obtain better in-distribution rollout fidelity across all reported metrics (Table 2), and the 7.5B checkpoint achieves the best result on every reported metric after Bridge V2 finetuning (Table 3). On Bridge V2, the 600M-to-2.4B change improves MSE, PSNR, and LPIPS, but slightly lowers SSIM. However, the correspondence between loss, visual fidelity, and behavioral capability is not one-to-one. Paired frame-level metrics measure similarity to the paired ground-truth frames, but they can miss task-relevant differences in gripper motion, object contact, and outcome. This mismatch is clearest out of distribution. On AutoEval (Zhou et al., 2025b) rollouts with unseen viewpoints, scenes, and lighting, the 600M model matches or beats larger models on MSE, PSNR (Hore & Ziou, 2010), and SSIM (Hore & Ziou, 2010), despite much lower trajectory-match rates (Appendix D.2). LPIPS (Zhang et al., 2018) weakly favors the 7.5B model, and FVD (Unterthiner et al., 2018) improves substantially for the 7.5B model, indicating better distribution-level video quality. But FVD is still not a paired behavioral matching metric: it does not directly determine whether the generated rollout matches the specific ground-truth execution under the same action sequence. A VLM judge that checks gripper motion, object interaction, and task outcome (Figure 4) shows the task-relevant trend: match rates increase across the three deployment-scale checkpoints while remaining low in absolute terms.

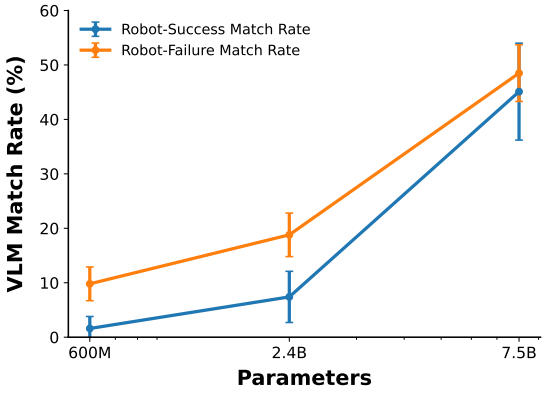

Figure 4: **OOD trajectory matching on AutoEval** (Zhou et al., 2025b), judged by GPT-5.2. Match rates increase with model size for both successful and failed ground-truth trajectories.

In the corpus-fraction comparison (Table 4), FVD is 118.251 with 10% of the corpus and 107.153 with 100%, while paired frame-level metrics remain close. We report this comparison separately from the deployment-checkpoint comparison in Table 2. At fixed 2.4B model size, broad pretraining followed by Bridge V2 finetuning outperforms training from scratch across all reported metrics (Table 5).

Why is the frontier so model-heavy? One possible interpretation is that token count overstates the marginal interaction diversity of the current corpus: neighboring frames, clips from the same scene, and trajectories collected in the same laboratory share substantial structure. Under this interpretation, additional token exposure saturates earlier than additional model capacity. Our observations are consistent with this explanation, but the current experiments do not separately identify the effects of token quantity, unique interaction diversity, and repetition.

Takeaway. Within the compute range studied, the diffusion objective supports a strongly model-heavy allocation. Diffusion loss provides the clearest basis for estimating this frontier, while rollout video metrics and trajectory-level evaluation serve as complementary diagnostics. The result is specific to the current corpus and objective and should not be interpreted as evidence that additional interaction diversity is unimportant for downstream behavior.

4 Applications of OpenWorld

We demonstrate interactive human and robot simulation, policy evaluation, and qualitative examples of physical dynamics.

4.1 Real-Time Simulation of Human Interaction

Interactive world models have been demonstrated in games (Bruce et al., 2024). We use OpenWorld, pretrained on human hand data, in a real-time pipeline that streams Meta Quest 3 hand poses as live controls (Appendix E). Figure 6 shows novel hand actions, gravity-driven motion, and contact-rich manipulation, suggesting applications in VR gaming and humanoid-policy training and evaluation.

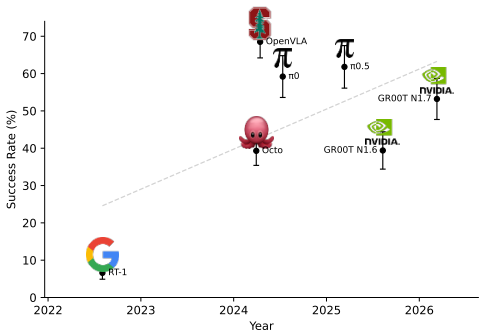

Figure 5: **Policy ranking in OpenWorld.** Points show macro-average success rates by year; icons identify each policy. The dashed line is an unweighted linear fit across the seven policies. Error bars summarize variation across 12 task groups (Appendix D.4).

4.2 Diverse Simulation across Embodiments and Viewpoints

Simulating Diverse Morphologies. Our pretraining data spans 27 robot embodiments and humans. On held-out action sequences in in-distribution settings, OpenWorld generates rollouts consistent with each morphology and action space (Figure 16).

Simulating from Diverse Viewpoints. Many robotics datasets provide multiple synchronized camera views of the same interaction. Although we train on separate camera streams, OpenWorld can generate aligned rollouts across views in some cases (Figure 17, top). It also produces inconsistent states across views (Figure 17, bottom), such as a toaster oven door being half open in one view but fully open in another.

4.3 Ranking Policies in OpenWorld

OpenWorld provides a simulated setting for evaluating policies with a VLM judge, using simulated task performance as a proxy for real-world performance. We evaluate policies (Kim et al., 2024; Bjorck et al., 2025; Team et al., 2024; Brohan et al., 2022; Black et al., 2024; Intelligence et al., 2025) trained or finetuned on Bridge (Walke et al., 2023) (Appendix D.4). OpenVLA has the highest mean success rate (Figure 5); newer π and GR00T variants have higher mean success rates than their respective predecessors.

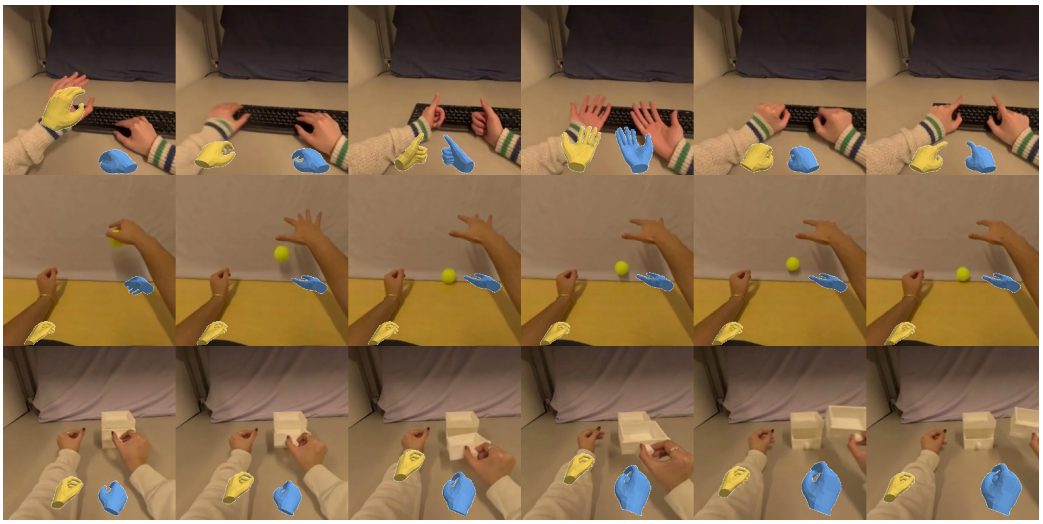

Figure 6: **Real-time human interaction simulation in OpenWorld.** OpenWorld generates video in real time, conditioned on hand poses captured by a Meta Quest 3. The rollouts show novel hand actions (top), plausible gravity-driven motion (middle), and contact-driven object manipulation (bottom).

4.4 Simulating Physical Phenomena

OpenWorld-7.5B generates rollouts with qualitatively plausible gravity-driven motion, contact, and deformable-object dynamics. Figure 18 shows hand-to-hand ball tossing (top), a toaster oven door falling after a gripper opens it (bottom left), and jacket folding (bottom right).

5 Conclusion

We present OpenWorld, a family of pretrained action-conditioned video world models with 600M to 7.5B parameters and a fully open training recipe. We train on 4 million clips spanning human interaction, robot manipulation, and navigation. Dedicated scaling sweeps show that held-out diffusion loss favors allocating most additional compute to model capacity. A separate training-loss frontier fitted to constant-learning-rate runs predicts the smoothed training loss of the excluded 7.5B run. Larger deployment checkpoints improve rollout fidelity and out-of-distribution trajectory matching; the 7.5B model achieves the best Bridge V2 finetuning results. OpenWorld enables real-time human interaction simulation, multi-view simulation across robot morphologies, and simulation-based policy evaluation; OpenWorld-7.5B rollouts show plausible effects of gravity, contact, and deformable-object motion.

Limitations. OpenWorld remains an early step toward scalable action-conditioned world models. The 7.5B model still underfits our corpus; progress likely requires larger models, longer runs, and stronger infrastructure for distributed training, failure recovery, and experiment monitoring. Our data scaling study is limited by available open-source datasets and the lack of a large-scale data processing pipeline for video filtering, action-video alignment checks, view selection, deduplication, and quality rebalancing. These limitations may affect compute-optimal allocation and downstream behavior. Adapting to new embodiments or environments may still require finetuning. Evaluation remains imperfect: paired video metrics can be unreliable proxies for action following, and rollouts can contain multi-view inconsistencies, missing occluded objects, or long-horizon physical errors.

AI USE STATEMENT

We used generative AI models as automated evaluators of world-model rollouts. GPT-5.2 (OpenAI, 2025) assessed whether generated AutoEval trajectories matched ground-truth executions in gripper motion, object interaction, state changes, and task outcome (Section 3; Appendix D.3). We compared these judgments with human annotations on sampled trajectories (Table 11); the evaluator captures the direction of the scaling trend but can overestimate absolute match rates. GPT-5.5 (OpenAI, 2026) judged task success in the policy-ranking experiments described in Appendix D.4. These model judgments serve as evaluation proxies with the limitations discussed in Appendix D.2. Generative AI tools also assisted with language polishing of the manuscript, organizing LaTeX formatting, debugging code in the training pipeline, and implementing portions of the evaluation pipeline. The authors are responsible for the final content of this work, including AI-assisted writing, code, and evaluation artifacts.

ETHICS STATEMENT

This work investigates the scaling of action-conditioned video world models for research in simulation and embodied intelligence. We do not identify specific ethical concerns arising from this study beyond those generally associated with generative modeling and robotics research. As discussed in our limitations, generated rollouts can contain physical inaccuracies, and simulation-based evaluation should be complemented by real-world validation.

REPRODUCIBILITY STATEMENT

We document the experimental procedures and configurations to facilitate reproducibility. Section 3.1 and Appendix A describe the pretraining data sources, curation, and sampling procedures. Appendix B details the model architecture, training hyperparameters, hardware, and action normalization. Section 3.2 and Appendix C specify the scaling experiment configurations, compute accounting, and fitting procedures. Appendix D describes the evaluation data, metrics, human annotation protocol, VLM evaluator prompts, and uncertainty estimation. Additional implementation details for real-time interaction and downstream adaptation are provided in Appendices E and G, respectively.

REFERENCES

Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

James F Allen and Johannes A Koomen. Planning using a temporal world model. In *Proceedings of the Eighth international joint conference on Artificial intelligence-Volume 2*, pp. 741–747, 1983.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. *arXiv preprint arXiv:2301.08243*, 2023.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew J. Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 15791–15801, 2025.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *arXiv preprint arXiv:2404.08471*, 2024.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023a.

Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. *arXiv preprint arXiv:2304.08818*, 2023b.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xu Huang, Shu Jiang, et al. AgiBot World Colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025. URL https://arxiv.org/abs/2503.06669.

Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S. Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, Jan Kautz, and Dieter Fox. DexYCB: A benchmark for capturing hand grasping of objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. URL https://arxiv.org/abs/2104.04631.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024.

Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis, 2023. URL https://arxiv.org/abs/2310.00426.

Yixiang Chen, Peiyan Li, Jiabing Yang, Keji He, Xiangnan Wu, Yuan Xu, Kai Wang, Jing Liu, Nianfeng Liu, Yan Huang, et al. Bridgev2w: Bridging video generation models to embodied world models via embodiment masks. *arXiv preprint arXiv:2602.03793*, 2026.

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *Advances in neural information processing systems*, 36:9156–9172, 2023.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis, 2024. URL https://arxiv.org/abs/2403.03206.

Zicong Fan, Omid Taheri, Dimitrios Tzionas, Muhammed Kocabas, Manuel Kaufmann, Michael J. Black, and Otmar Hilliges. ARCTIC: A dataset for dexterous bimanual hand-object manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. URL https://arxiv.org/abs/2204.13662.

Shenyuan Gao, Siyuan Zhou, Yilun Du, Jun Zhang, and Chuang Gan. Adaworld: Learning adaptable world models with latent actions. In *International Conference on Machine Learning*, 2025.

Shenyuan Gao, William Liang, Kaiyuan Zheng, Ayaan Malik, Seonghyeon Ye, Sihyun Yu, Wei-Cheng Tseng, Yuzhu Dong, Kaichun Mo, Chen-Hsuan Lin, et al. Dreamdojo: A generalist robot world model from large-scale human videos. *arXiv preprint arXiv:2602.06949*, 2026.

Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim. First-person hand action benchmark with RGB-D videos and 3D hand pose annotations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 409–419, 2018. URL https://arxiv.org/abs/1704.02463.

Quentin Garrido, Tushar Nagarajan, Basile Terver, Nicolas Ballas, Yann LeCun, and Michael Rabbat. Learning latent action world models in the wild. *arXiv preprint arXiv:2601.05230*, 2026.

Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. CTRL-WORLD: A controllable generative world model for robot manipulation. In *International Conference on Learning Representations*, 2026.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. *arXiv preprint arXiv:1811.04551*, 2018.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.

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

Alexander Hägele, Elie Bakouch, Atli Kosson, Loubna Ben Allal, Leandro Von Werra, and Martin Jaggi. Scaling laws and compute-optimal training beyond fixed training durations. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, DDL Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 10, 2022.

Ryan Hoque, Peide Huang, David J Yoon, Mouli Sivapurapu, and Jian Zhang. Egodex: Learning dexterous manipulation from large-scale egocentric video. *arXiv preprint arXiv:2505.11709*, 2025.

Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th international conference on pattern recognition*, pp. 2366–2369. IEEE, 2010.

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

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, et al. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*, 2024.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. In *Advances in Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=mSiN7i0BYH.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Simar Kareer, Dhruv Patel, Ryan Punamiya, Pranay Mathur, Shuo Cheng, Chen Wang, Judy Hoffman, and Danfei Xu. Egomimic: Scaling imitation learning via egocentric video. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 13226–13233. IEEE, 2025.

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

Jing Yu Koh, Honglak Lee, Yinfei Yang, Jason Baldridge, and Ruslan Salakhutdinov. Pathdreamer: A world model for indoor navigation. *arXiv preprint arXiv:2105.08756*, 2021.

Dan Kondratyuk, Lijun Yu, Xiuye Gu, Jose Lezama, Jonathan Huang, Rachel Hornung, Hartwig Adam, Hassan Akbari, Yair Alon, Vighnesh Birodkar, et al. Videopoet: A large language model for zero-shot video generation. *arXiv preprint arXiv:2312.14125*, 2023.

Taein Kwon, Bugra Tekin, Jan Stühmer, Federica Bogo, and Marc Pollefeys. H2O: Two hands manipulating objects for first person interaction recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10138–10148, 2021. URL https://arxiv.org/abs/2104.11181.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space, 2025. URL https://arxiv.org/abs/2506.15742.

Zhengyang Liang, Hao He, Ceyuan Yang, and Bo Dai. Scaling laws for diffusion transformers. *arXiv preprint arXiv:2410.08184*, 2024.

Hao Luo, Yicheng Feng, Wanpeng Zhang, Sipeng Zheng, Ye Wang, Haoqi Yuan, Jiazheng Liu, Chaoyi Xu, Qin Jin, and Zongqing Lu. Being-h0: vision-language-action pretraining from large-scale human videos. *arXiv preprint arXiv:2507.15597*, 2025.

Niklas Muennighoff, Alexander Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin A Raffel. Scaling data-constrained language models. *Advances in Neural Information Processing Systems*, 36:50358–50376, 2023.

Mang Ning, Mingxiao Li, Jianlin Su, Albert Ali Salah, and Itir Onal Ertugrul. Elucidating the exposure bias in diffusion models, 2024. URL https://arxiv.org/abs/2308.15321.

Open X-Embodiment Collaboration. Open X-Embodiment: Robotic learning datasets and RT-X models, 2023. URL https://arxiv.org/abs/2310.08864.

OpenAI. Introducing GPT-5.2. https://openai.com/index/introducing-gpt-5-2/, 2025. Product release, December 11, 2025. Accessed: May 5, 2026.

OpenAI. Introducing GPT-5.5. https://openai.com/index/introducing-gpt-5-5/, 2026. Product release, April 23, 2026. Accessed: May 5, 2026.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Julian Quevedo, Ansh Kumar Sharma, Yixiang Sun, Varad Suryavanshi, Percy Liang, and Sherry Yang. Worldgym: World model as an environment for policy evaluation. *arXiv preprint arXiv:2506.00613*, 2025.

Marc Rigter, Tarun Gupta, Agrin Hilmkil, and Chao Ma. AVID: Adapting video diffusion models to world models. *arXiv preprint arXiv:2410.12822*, 2024.

Ansh Kumar Sharma, Yixiang Sun, Ninghao Lu, Yunzhe Zhang, Jiarao Liu, and Sherry Yang. World-gymnast: Training robots with reinforcement learning in a world model. *arXiv preprint arXiv:2602.02454*, 2026.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

Wei-Cheng Tseng, Jinwei Gu, Qinsheng Zhang, Hanzi Mao, Ming-Yu Liu, Florian Shkurti, and Lin Yen-Chen. Scalable policy evaluation with video world models. *arXiv preprint arXiv:2511.11520*, 2025.

Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable length video generation from open domain textual description. *arXiv preprint arXiv:2210.02399*, 2022.

Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning*, pp. 1723–1736. PMLR, 2023.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.

Meng Wei, Chenyang Wan, Jiaqi Peng, Xiqian Yu, Yuqiang Yang, Delin Feng, Wenzhe Cai, Chenming Zhu, Tai Wang, Jiangmiao Pang, and Xihui Liu. Ground slow, move fast: A dual-system foundation model for generalizable vision-and-language navigation. *arXiv preprint arXiv:2512.08186*, 2025. URL https://arxiv.org/abs/2512.08186.

Kun Wu, Chengkai Hou, Jiaming Liu, Zhengping Che, Xiaozhu Ju, Zhuqin Yang, Meng Li, Yinuo Zhao, Zhiyuan Xu, Guang Yang, et al. RoboMIND: Benchmark on multi-embodiment intelligence normative data for robot manipulation. In *Robotics: Science and Systems*, 2025. URL https://arxiv.org/abs/2412.13877.

Philipp Wu, Alejandro Escontrela, Danijar Hafner, Pieter Abbeel, and Ken Goldberg. Daydreamer: World models for physical robot learning. In *Proceedings of the 6th Conference on Robot Learning*, volume 205 of *PMLR*, pp. 2226–2240, 2023.

Ge Yang, Edward Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tuning large neural networks via zero-shot hyperparameter transfer. *Advances in Neural Information Processing Systems*, 34:17084–17097, 2021.

Mengjiao Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 2023.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T Freeman. Improved distribution matching distillation for fast image synthesis. *Advances in neural information processing systems*, 37:47455–47487, 2024a.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024b. URL https://openaccess.thecvf.com/content/CVPR2024/html/Yin_One-step_Diffusion_with_Distribution_Matching_Distillation_CVPR_2024_paper.html.

Yuanyang Yin, Yaqi Zhao, Mingwu Zheng, Ke Lin, Jiarong Ou, Rui Chen, Victor Shea-Jay Huang, Jiahao Wang, Xin Tao, Pengfei Wan, et al. Towards precise scaling laws for video diffusion transformers. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 18155–18165, 2025.

Xinyu Zhan, Lixin Yang, Yifei Zhao, Kangrui Mao, Hanlin Xu, Zenan Lin, Kailin Li, and Cewu Lu. OAKINK2: A dataset of bimanual hands-object manipulation in complex task completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. URL https://arxiv.org/abs/2403.19417.

Jiahan Zhang, Muqing Jiang, Nanru Dai, Taiming Lu, Arda Uzunoglu, Shunchi Zhang, Yana Wei, Jiahao Wang, Vishal M Patel, Paul Pu Liang, et al. World-in-world: World models in a closed-loop world. *arXiv preprint arXiv:2510.18135*, 2025.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

Siyuan Zhou, Yilun Du, Yuncong Yang, Lei Han, Peihao Chen, Dit-Yan Yeung, and Chuang Gan. Learning 3d persistent embodied world models. *arXiv preprint arXiv:2505.05495*, 2025a.

Zhiyuan Zhou, Pranav Atreya, You Liang Tan, Karl Pertsch, and Sergey Levine. Autoeval: Autonomous evaluation of generalist robot manipulation policies in the real world. *arXiv preprint arXiv:2503.24278*, 2025b.

Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. *arXiv preprint arXiv:2504.02792*, 2025a.

Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chi-Lam Cheang, and Tao Kong. IRASim: A fine-grained world model for robot manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025b.

APPENDIX

A DETAILS OF THE PRETRAINING DATASET CURATION

Table 6 gives details of the component datasets in the pretraining corpus for OpenWorld. Different camera views are counted as separate episodes, and long trajectories are divided into shorter overlapping episodes.

Table 6: Training data composition. Different camera views of the same scene are counted as separate episodes, and long trajectories are divided into shorter overlapping episodes. Hours are computed from frame counts using each dataset's original recording frequency. Frame counts are measured from the processed data and rounded for display.

Category	Dataset	Episodes	Hours	Frames
Robot Manip.	OXE (54 sub-datasets)	2,106,776	5,392	141.7M
	AgiBotWorld Alpha	846,471	1,378	148.8M
	RoboMind	252,748	1,098	39.5M
	SVLA	889	2	179K
	Subtotal	*3,206,884*	*7,870*	*330.2M*
Human	EgoDex	533,268	838	90.5M
	FPHA	405	<1	37K
	OakInk2	20,040	37	4.0M
	ARCTIC	4,852	9	975K
	DexYCB	7,920	5	576K
	H2O	3,565	3	337K
	Subtotal	*570,050*	*893*	*96.4M*
Navigation	InternVLN-N1	321,892	1,279	46.0M
Total		**4,098,826**	**10,042**	**472.6M**

B Details of Model Training

B.1 Model Architecture and Training Stability

DiT Backbone. OpenWorld extends a standard latent diffusion model with a Diffusion Transformer (DiT) backbone (Peebles & Xie, 2023) to predict future video frames conditioned on actions, following prior work (Chen et al., 2024; Quevedo et al., 2025). We use a causal temporal attention mask. The model takes observed frames and action sequences as inputs and predicts future video frames in the latent space of a Stable Diffusion VAE (Esser et al., 2024). We experimented with other pretrained image autoencoders (Labs et al., 2025), but did not observe a significant benefit for video modeling. Architecture and pretraining hyperparameters for the deployment checkpoints are reported in Table 7; the scaling sweeps use the configurations in Appendix C.

We use the VAE from Stable Diffusion 3 (Esser et al., 2024) to independently encode 256×256 image frames into latent space. For the 7.5B model, we use a 32-layer transformer with a hidden dimension of 2560 and 16 attention heads. The deployment checkpoints in Table 7 are pretrained with 24-frame clips; their longer rollouts use a sliding window of the last 24 frames. The separate adaptation experiment in Appendix G uses 10-frame clips.

Table 7: Pretraining configurations for the 600M, 2.4B, and 7.5B deployment checkpoints. Scaling-sweep configurations are listed separately in Table 8.

Hyperparameter	**600M**	**2.4B**	**7.5B**
Total parameters	608,245,824	2,430,742,592	7,573,355,584
Image Resolution	256×256		
Number of Frames	24		
DiT Patch Size	2		
Input Channels	16		
Hidden Size	1024	2048	2560
Layers	16	16	32
Attention Heads	16		
MLP Ratio	4		
Optimizer	AdamW (weight decay = 0, $\beta_1 = 0.9$, $\beta_2 = 0.99$)		
Learning rate	1×10^{-4} with 10k-step warmup from 1×10^{-7}		
Batch size	64	64	256
Action dimension	256		
Training hardware	4×NVIDIA H200	4×NVIDIA H200	16×NVIDIA H200
Training steps	230k observed	260k observed	259k observed
Diffusion noise schedule	sigmoid		
Diffusion training timesteps	1000		
Sampling timesteps	10		
Prediction target	v		

Parameter Efficiency for Action Conditioning. We compare two approaches to conditioning the DiT on actions: adaptive layer normalization (AdaLN) (Peebles & Xie, 2023) and cross-attention in the DiT blocks. For AdaLN conditioning, we project each action vector to the DiT hidden dimension and add it to the timestep embedding. Unlike popular DiT models that share a single AdaLN module across layers (Wan et al., 2025; Chen et al., 2023), we find that layer-specific AdaLN modules increase conditioning capacity and help the model capture more diverse action-conditioned dynamics. At similar parameter counts, layer-specific AdaLN conditioning gives lower training loss than cross-attention, as shown in Figure 7 (left).

Scaling Width vs. Depth. We also study how scaling transformer width and depth affects training loss. In this comparison, increasing the hidden dimension reduces training loss more than increasing the number of transformer layers (Figure 7, right). We therefore prioritize hidden-dimension growth in the deployment-scale configurations. Table 7 reports the final deployment-scale configurations.

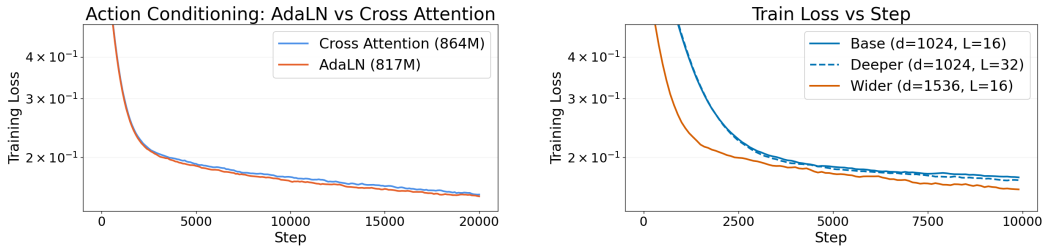

Figure 7: **Parameter efficiency of action conditioning and model scaling.** Left: AdaLN vs. cross-attention for action conditioning. Adaptive layer normalization (AdaLN) with 817M parameters achieves lower training loss than cross-attention with 864M parameters despite using 5.4% fewer parameters. Right: Scaling transformer width vs. depth. For a similar total parameter count, a wider model ($d = 1536$, $L = 16$) achieves lower loss than a deeper model ($d = 1024$, $L = 32$).

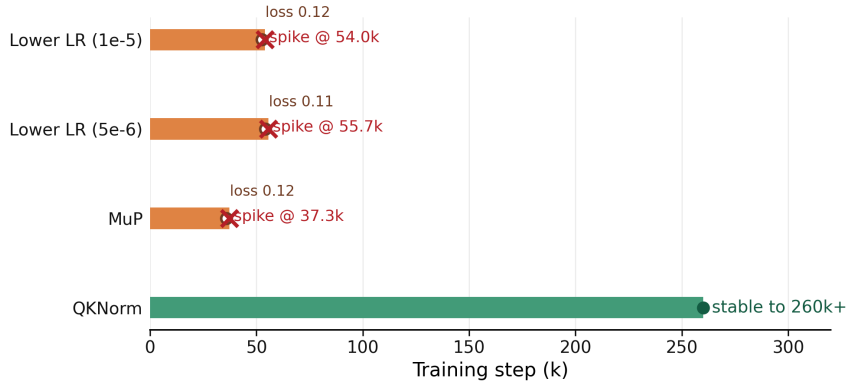

Figure 8: **Training stability** of a 2.4B model under different stabilization strategies. Lower learning rates and μP (Yang et al., 2021) do not eliminate loss spikes, whereas RMSNorm-based query-key normalization remains stable beyond 260k training steps.

Stabilizing Training during Scaling. As we scale the model from 600M to 7.5B parameters, training becomes increasingly unstable. While the 600M model can often train stably, the 2.4B and 7.5B models exhibit much more frequent and severe gradient and loss explosions. This suggests that optimization instability becomes a major bottleneck for scaling model size. We explored several techniques to improve stability, as shown in Figure 8, including gradient clipping, action normalization (details in Appendix B.2), additional data cleaning to remove problematic training data points associated with unusually large gradient magnitudes, and μP (Yang et al., 2021) hyperparameter transfer to avoid directly reusing small-model hyperparameters at larger scales. These interventions provided partial improvements, but did not reliably eliminate training instability at larger scales. Among all methods we tested, RMSNorm-based query-key normalization was the most effective, remaining stable beyond 260k steps. Following prior implementations in open-source video diffusion transformers such as Wan (Wan et al., 2025), we RMS-normalize the query and key projections before computing attention logits in self-attention layers. This substantially stabilizes optimization at larger model sizes.

B.2 Details of Action Normalization

Since action spaces across our pretraining mixture have substantially different scales and conventions, we normalize actions before feeding them to the model. For each dataset subset, we compute per-dimension action statistics on the training split. Let $a_t \in \mathbb{R}^{d_a}$ denote the action at time t. For each action dimension, we compute the empirical mean μ and standard deviation σ, with the variance clamped from below by 10^{-12} before taking the square root. We then apply z-score normalization:

$$\tilde{a}_t = \frac{a_t - \mu}{\sigma}.$$

To reduce the effect of rare outlier actions, we additionally estimate the 1st and 99th percentiles using reservoir sampling with 5×10^5 samples, and clip actions to this range before normalization:

$$a'_t = \text{clip}(a_t, q_{0.01}, q_{0.99}), \qquad \hat{a}_t = \frac{a'_t - \mu}{\sigma}.$$

The resulting action $\hat{a}_t$ is used as the action-conditioning input to the DiT. During finetuning and evaluation, we use the same action statistics as the corresponding training dataset to keep the conditioning scale consistent across training and rollout generation.

C SCALING SWEEP DETAILS

Following prior scaling-law studies (Kaplan et al., 2020; Hoffmann et al., 2022), we describe the model grid, compute convention, endpoint selection, and fitting protocol. All reported analysis budgets, endpoint targets, and compute axes use the same proxy, $C = 4ND$, calculated from actual trainable parameter counts.

C.1 FLOP ACCOUNTING

For all scaling analyses, an optimizer step consumes BTS latent image tokens, where B is the effective global batch size, T is the clip length, and $S = 256$ is the number of spatial latent tokens per frame. After s optimizer steps, $D = sBTS$ and $C = 4ND$, so the per-step compute proxy is $F_{\text{step}} = 4NBTS$. D counts token exposure, including repeated presentations, rather than unique tokens. This is an analytical comparison convention, not measured hardware compute; it does not explicitly account for activation-checkpoint recomputation, VAE encoding, optimizer updates, or other implementation overheads.

For the sweep architecture, the exact trainable parameter count is

$$N = (36L + 3)d^2 + (28L + 646)d + 64,$$

where L is the number of DiT blocks and d is the hidden width. We compute budgets from these full counts before rounding table labels. For example, $L = 16$ and $d = 1024$ give $N = 608{,}245{,}824$; with $B = 256$ and $T = 24$, the proxy is 3.827 PF per optimizer step. Table 8 uses this same convention for every model.

C.2 MODEL GRID AND BUDGET SCHEDULE

Table 8 gives the architecture grid and configured training limits. Sweep A uses $B = 256$, $T = 24$, and a repeated-data weighted sampler. Sweep B uses $B = 128$, $T = 10$, and non-overlapping clips. Although its configured limit is 500,000 steps, the first-epoch analysis retains only endpoints at steps $\leq 355{,}617$ (355,618 optimizer steps per epoch). At budget C, we round $C/(4NBTS)$ to the nearest optimizer step and require an available checkpoint within the analysis range. The training-loss frontier additionally requires at least 15,000 steps per model to exclude undertrained points.

C.3 FIRST-EPOCH ROBUSTNESS SWEEP

Sweep B contains 10 models from 27M to 1.2B parameters and uses $B = 128$, $T = 10$ and non-overlapping clips. We restrict this analysis to 49 annealed endpoints within the first epoch, excluding the L9, $d = 288$ endpoint at step 376,157 because it exceeds the 355,618-step epoch length. Applying the held-out-loss isoFLOP procedure at seven budgets in $[0.6, 6.83] \times 10^{19}$ yields $N_{\text{opt}} \propto C^{0.89 \pm 0.03}$ and $D_{\text{opt}} \propto C^{0.11 \pm 0.03}$. All seven slices retain interior minima. The model-heavy trend therefore persists under this endpoint restriction. Batch size, clip length, and sampler also differ from Sweep A, so this is a robustness check rather than a matched ablation of data repetition.

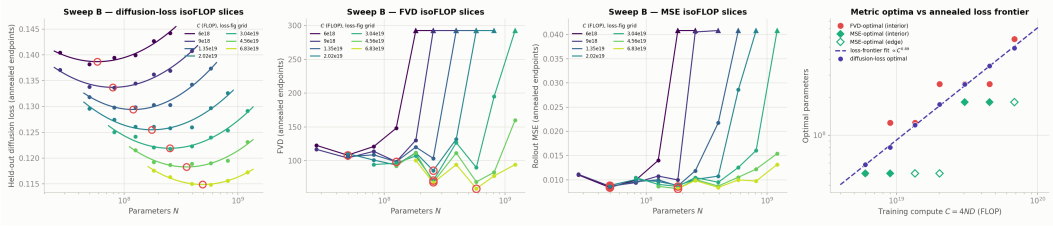

Figure 9: **First-epoch Sweep B isoFLOP analysis.** From left to right: held-out diffusion-loss, FVD, and rollout-MSE slices at the seven annealed budgets, followed by their per-budget optima. The diffusion-loss optima yield $N_{\text{opt}} \propto C^{0.89 \pm 0.03}$ using 49 endpoints within the first epoch. As in the primary sweep, the rollout metrics provide a flatter and noisier directional cross-check.

Table 8: Scaling-sweep run grid. Parameter counts are rounded from actual trainable counts. PF/step is $4NBTS/10^{15}$; maximum compute is this proxy multiplied by the configured maximum number of steps. Sweep B analysis is restricted to the first epoch, rather than its configured 500,000-step limit.

Sweep	Run	L	d	Params (M)	Max $4ND$	PF/ step	Max steps
A	R01	4	256	9.8	1.34×10^{19}	0.062	216,565
A	R02	5	320	19.0	2.73×10^{19}	0.119	228,412
A	R03	6	384	32.6	2.77×10^{19}	0.205	134,857
A	R04	7	448	51.6	6.99×10^{19}	0.324	215,407
A	R05	8	512	76.7	1.41×10^{20}	0.483	291,790
A	R06	9	576	109.0	1.42×10^{20}	0.686	206,698
A	R07	10	640	149.3	1.42×10^{20}	0.939	151,726
A	R08	11	704	198.4	2.86×10^{20}	1.248	229,283
A	R09	12	768	257.3	2.87×10^{20}	1.619	177,446
A	R10	13	832	326.9	2.88×10^{20}	2.057	140,128
A	R11	14	896	408.0	2.89×10^{20}	2.567	112,583
A	R12	15	960	501.5	2.90×10^{20}	3.155	91,809
A	R13	16	1024	608.2	2.90×10^{20}	3.827	75,848
A	R14	17	1088	729.2	2.91×10^{20}	4.588	63,382
A	R15	18	1152	865.3	2.91×10^{20}	5.444	53,505
A	R16	19	1216	1017.3	2.92×10^{20}	6.400	45,578
B	R01	9	288	27.4	1.79×10^{19}	0.036	500,000
B	R02	11	352	49.8	3.26×10^{19}	0.065	500,000
B	R03	13	416	81.9	5.37×10^{19}	0.107	500,000
B	R04	15	480	125.6	8.23×10^{19}	0.165	500,000
B	R05	17	544	182.6	1.20×10^{20}	0.239	500,000
B	R06	19	608	254.7	1.67×10^{20}	0.334	500,000
B	R07	22	704	394.9	2.59×10^{20}	0.518	500,000
B	R08	25	800	579.0	3.79×10^{20}	0.759	500,000
B	R09	28	896	812.9	5.33×10^{20}	1.066	500,000
B	R10	32	1024	1212.7	7.95×10^{20}	1.589	500,000

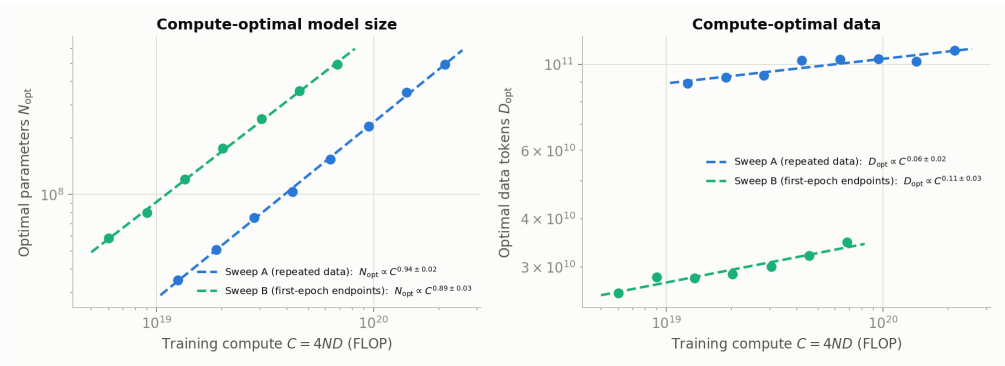

Figure 10: **Appendix comparison of compute-optimal allocation laws.** Left: $N_{\text{opt}}(C)$; right: $D_{\text{opt}}(C)$. Sweep A uses the repeated-data weighted sampler; Sweep B uses first-epoch endpoints from non-overlapping clips. Both are strongly model-heavy. Comparisons of their absolute allocation levels should account for differences in clip length, batch size, and sampling regime. Both use 256 spatial latent tokens per frame.

C.4 Fitting Protocol and Training-Loss Extrapolation

The main text estimates the allocation exponents from the minima of the held-out diffusion-loss isoFLOP slices, using the interior-minimum criterion of Section 3.3 to select budgets. We fit an unweighted quadratic in $\log_{10} N$ to each loss slice, then fit power laws to the resulting optimal allocations by ordinary least squares in log–log space. Reported uncertainties are twice the standard error of the log–log slope.

Training-loss extrapolation and data versions. For the separate 7.5B comparison, we merge training-history fragments by optimizer step, retain the most recently created fragment at overlapping steps, and apply an EMA with decay 0.85. At each of the eight Sweep A budgets, we fit a quadratic in $\log_{10} N$ using models with at least 15,000 logged steps at that budget. Power laws through the resulting loss minima and optimal sizes give the loss frontier and $C^\star$. The 7.5B curve is shown only through step 140,000 and is excluded from every fit. At step 65,399 ($C = 3.12 \times 10^{21}$, about 18.5% below $C^\star$), its smoothed loss is 0.088416 versus a predicted loss of 0.086280; at $C^\star = 3.83 \times 10^{21}$, the interpolated model loss and predicted frontier loss are 0.087143 and 0.084832, respectively.

Sweep A uses the AgiBotWorld preprocessing variant `no-fisheye` and the plotted 7.5B trajectory uses `final_cleaned`. Shared settings include a global batch size of 256, 24 frames, a learning rate of 10^{-4}, and a 10,000-step warmup. This comparison is therefore an extrapolation check across these data configurations, rather than a controlled ablation of data preprocessing.

D Details of Evaluation Setup

D.1 Details of Evaluation Data

We evaluate OpenWorld in four settings: scaling-sweep evaluation, in-distribution rollout evaluation, targeted finetuning evaluation, and out-of-distribution evaluation.

Scaling-sweep evaluation. The held-out diffusion-loss evaluation and the video-rollout evaluation use separate protocols. For each annealed checkpoint, diffusion loss is averaged over 640 minibatches, configured with 16 clips per batch on one worker. Sweep A uses 24-frame clips and Sweep B uses 10-frame clips. These are evaluation batch settings; the training batch sizes are given in Table 1.

For the FVD and rollout-MSE curves in Figures 2 and 9, each Sweep A endpoint uses 100 paired OXE rollouts, while each Sweep B endpoint uses 2,100 paired rollouts. These rollout sets are separate from the deployment-checkpoint evaluation below.

In-distribution rollout evaluation. For in-distribution rollout evaluation of OpenWorld, we use a separate held-out set of 900 trajectories from OXE, AgiBotWorld, and EgoDex. Given the first frame and the ground-truth action sequence, the model rolls out future frames autoregressively, and we compare the generated rollout against the paired ground-truth execution using visual metrics.

Targeted finetuning evaluation. For targeted finetuning, we finetune pretrained OpenWorld checkpoints on Bridge V2 (Walke et al., 2023) and evaluate on a held-out split of 100 Bridge trajectories. We report paired frame-level metrics on this split.

Out-of-distribution evaluation. For OOD evaluation, we use AutoEval (Zhou et al., 2025b). We evaluate on 479 paired AutoEval trajectories across four tasks: closing the drawer, opening the drawer, putting the eggplant in the blue sink, and putting the eggplant in the yellow basket.

D.2 Details of Evaluation Metrics

For the in-distribution evaluation, we use similarity-based metrics, including pixel-wise mean squared error (MSE), PSNR (Hore & Ziou, 2010), SSIM (Hore & Ziou, 2010), LPIPS (Zhang et al., 2018), and FVD (Unterthiner et al., 2018). For targeted finetuning on Bridge V2 (Walke et al., 2023), we report only the paired frame-level metrics. The Bridge test set has mixed rollout horizons (13/24/49 frames), so a single FVD score would be confounded by rollout length. For OOD AutoEval evaluation, we

find that paired frame-level metrics such as MSE, PSNR, SSIM, and LPIPS can be unreliable proxies for task-relevant trajectory matching. FVD provides a useful distribution-level video-quality signal, but it does not directly determine whether a generated rollout matches the specific ground-truth execution under the same action sequence. We therefore use human and VLM-based trajectory matching as the primary OOD behavioral evaluation. We report results separately for successful and failed robot trajectories. Here, success and failure refer to the outcome of the real robot execution in the ground-truth video, not the generated rollout. Human annotators and the VLM evaluator judge whether the generated rollout matches the corresponding ground-truth execution, including reproducing a failed execution when the ground-truth trajectory ends in failure.

In this section, we first describe our human evaluation setup, which we use as the reference for task-relevant trajectory matching. We then provide both quantitative and qualitative evidence that frame-level visual metrics can disagree with this human reference. Finally, we introduce VLM-based trajectory matching as a scalable proxy for human evaluation and validate the VLM evaluator against human annotations.

Human Evaluation. We conduct a small-scale human evaluation of generated rollouts. This serves two purposes: first, it directly tests whether model scaling improves behaviorally faithful rollout generation in zero-shot OOD settings; second, it provides a human reference for assessing whether automatic metrics reflect trajectory matching. We therefore manually annotate a subset of AutoEval rollouts. For each model size, we sample 120 trajectories, balanced across four AutoEval (Zhou et al., 2025b) tasks and robot success/failure splits: 15 ground-truth success cases and 15 ground-truth failure cases per task. Each trajectory is labeled as *match* or *mismatch* according to whether the generated rollout matches the corresponding ground-truth execution in task-relevant behavior. Table 9 reports the resulting human match rates. The 600M and 2.4B models almost never produce rollouts that humans judge as behaviorally matching the paired ground-truth execution because of their poor generation quality. In contrast, the 7.5B model reaches a substantially higher human match rate on both successful and failed robot trajectories. These results show that the 7.5B model produces more behaviorally faithful AutoEval rollouts than the two smaller models and provide a human reference for evaluating whether automatic metrics capture this improvement.

Table 9: **Human-rated trajectory matching on sampled AutoEval rollouts.** We report the percentage of generated rollouts labeled as *match* by human annotators, separately for successful and failed robot trajectories.

Model	GT-Success Match (%)	GT-Failure Match (%)
600M	0.00	1.67
2.4B	0.00	1.67
7.5B	25.00	45.00

Limitations of Visual Metrics. To test whether standard visual metrics capture task-relevant trajectory matching on AutoEval, we compare MSE, PSNR (Hore & Ziou, 2010), SSIM (Hore & Ziou, 2010), LPIPS (Zhang et al., 2018), and FVD (Unterthiner et al., 2018) against the human reference described above. Table 10 reports these visual metrics across model sizes. These results highlight a distinction between paired frame-level metrics and FVD. First, frame-level metrics can be misleading for trajectory matching: MSE, PSNR, and SSIM slightly favor the 600M model, even though human evaluation shows that the 7.5B model is much more behaviorally faithful. These metrics are sensitive to low-level alignment and appearance, but do not directly measure whether the robot follows the same interaction pattern as the ground-truth execution. Second, FVD behaves differently: it improves substantially for the 7.5B model, suggesting better distribution-level video quality and temporal realism. However, FVD is still not a paired behavioral matching metric. It does not directly determine whether the generated rollout matches the specific ground-truth execution under the same action sequence, including gripper motion, object contact, object state change, and final outcome. We therefore treat FVD as an auxiliary video-quality diagnostic and use trajectory-level human/VLM matching as the primary OOD behavioral metric.

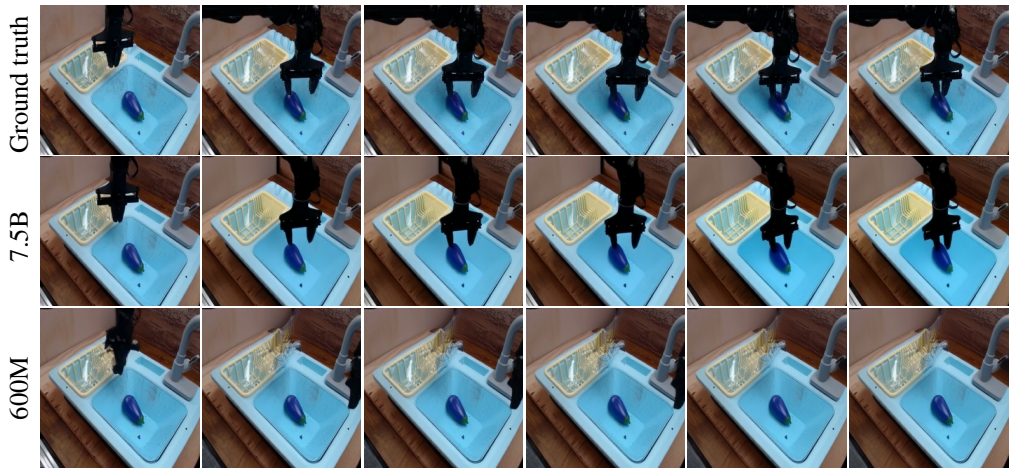

Figure 11: **Qualitative AutoEval comparison illustrating a limitation of paired frame-level metrics.** We show the ground-truth trajectory and predicted rollouts from the 7.5B and 600M models given the same initial frame and action sequence. Although the 600M model obtains comparable or slightly better MSE, PSNR, and SSIM on AutoEval in Table 10, its rollout is less behaviorally faithful in this example. The 7.5B model better preserves the task-relevant trajectory, while the 600M model fails to maintain consistent robot-object interactions.

Table 10: **OOD automatic video metrics on AutoEval.** Paired frame-level metrics are mixed and do not reliably track task-relevant trajectory matching: MSE, PSNR, and SSIM slightly favor the 600M model despite its much lower human/VLM match rates, while LPIPS only weakly favors the 7.5B model. FVD better reflects the improved video quality of the 7.5B model, but it does not directly measure whether the generated rollout matches the specific ground-truth execution.

Model	MSE↓	PSNR↑	LPIPS↓	SSIM↑	FVD↓
600M	**0.0199**	**17.016**	0.181	**0.742**	285.811
2.4B	0.0204	16.902	0.188	0.730	280.249
7.5B	0.0201	16.978	**0.175**	0.724	**198.994**

Table 11: Human validation of VLM-based AutoEval judgments, reported separately for successful and failed robot trajectories. We compare VLM and human labels on the same sampled trajectories, treating human ratings as the reference. Each row is computed over 60 trajectories. Human Match denotes the percentage of trajectories rated as *match* by human annotators, while VLM Match denotes the percentage rated as *match* by the VLM. TP denotes human *match* and VLM *match*; TN denotes human *mismatch* and VLM *mismatch*; FP denotes human *mismatch* but VLM *match*; FN denotes human *match* but VLM *mismatch*. All values are percentages.

Model	GT Split	Human Match (%)	VLM Match (%)	TP (%)	TN (%)	FP (%)	FN (%)
600M	GT-Failure	1.7	5.0	1.7	95.0	3.3	0.0
600M	GT-Success	0.0	1.7	0.0	98.3	1.7	0.0
2.4B	GT-Failure	1.7	11.7	1.7	88.3	10.0	0.0
2.4B	GT-Success	0.0	11.7	0.0	88.3	11.7	0.0
7.5B	GT-Failure	45.0	51.7	33.3	36.7	18.3	11.7
7.5B	GT-Success	25.0	51.7	25.0	48.3	26.7	0.0

VLM-based trajectory matching. Human trajectory-level matching provides the most direct reference for AutoEval, but it is expensive to apply exhaustively across all trajectories and model

variants. We therefore use a VLM-based trajectory matching metric as a scalable automatic proxy for human evaluation. For each AutoEval trajectory, the VLM evaluator observes the generated rollout and the corresponding ground-truth execution under the same initial frame and action sequence, and outputs a binary *match*/*mismatch* label. The evaluator is instructed to focus on task-relevant behavior, including gripper motion, object–gripper interaction, object state changes, and final task outcome.

We define the VLM match rate as the mean of these binary labels:

$$\text{MatchRate} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\left[\text{VLM}(x_i, \hat{x}_i) = \textit{match}\right],$$

where x_i is the ground-truth execution and $\hat{x}_i$ is the corresponding generated rollout.

Validation of the VLM proxy against human annotations. We next evaluate how well the VLM proxy agrees with the human reference described above. The prompts used by the VLM evaluator are provided in Appendix D.3. Table 11 compares GPT-5.2 (OpenAI, 2025) judgments against human ratings on the same sampled trajectories. The results show that the VLM proxy captures the direction of the human-rated scaling trend across model sizes, although it tends to overestimate the absolute match rates. Overall, this validation supports using VLM-based AutoEval as a scalable proxy for behavioral matching, while interpreting absolute VLM match rates with caution.

D.3 Details of the VLM-based Trajectory Evaluator

Evaluator Prompt and Task Setup. We use GPT-5.2 (OpenAI, 2025) as the VLM evaluator for four AutoEval (Zhou et al., 2025b) tasks: (1) close the drawer, (2) open the drawer, (3) put the eggplant in the blue sink, and (4) put the eggplant in the yellow basket.

```
You are evaluating robot manipulation trajectories for closing
the drawer.

Two trajectories are provided:
Video 1:  Ground truth trajectory (real robot execution)
Video 2:  Predicted trajectory (a rollout from a learned world
model)

Your task is to determine whether the predicted trajectory
matches the ground truth.  Carefully compare the two
trajectories in terms of:

1.  Gripper motion:
- Whether the interaction pattern matches the ground truth
- If the ground truth involves contact with the drawer, the
predicted trajectory should also make similar contact
- If the ground truth does not involve contact, the predicted
trajectory should also avoid contact
- Pay attention to whether the gripper correctly engages (or
does not engage) with the drawer when required

2.  Final outcome:
- Whether the gripper state in the predicted trajectory matches
that in ground truth trajectory
- Whether the drawer state in the predicted trajectory matches
that in ground truth trajectory

Important:
- Small visual differences (lighting, texture, rendering
artifacts) should be ignored
- Focus only on action and interaction correctness

Output exactly one word:

MATCH
```

```
or
MISMATCH


You are evaluating robot manipulation trajectories for opening
the drawer.

Two trajectories are provided:
Video 1:  Ground truth trajectory (real robot execution)
Video 2:  Predicted trajectory (a rollout from a learned world
model)

Your task is to determine whether the predicted trajectory
matches the ground truth.  Carefully compare the two
trajectories in terms of:

1.  Gripper motion:
- Whether the interaction pattern matches the ground truth
- If the ground truth involves contact with the drawer, the
predicted trajectory should also make similar contact
- If the ground truth does not involve contact, the predicted
trajectory should also avoid contact
- Pay attention to whether the gripper correctly engages (or
does not engage) with the drawer when required

2.  Final outcome:
- Whether the gripper state in the predicted trajectory matches
that in ground truth trajectory
- Whether the drawer state in the predicted trajectory matches
that in ground truth trajectory

Important:
- Small visual differences (lighting, texture, rendering
artifacts) should be ignored
- Focus only on action and interaction correctness

Output exactly one word:

MATCH
or
MISMATCH


You are evaluating robot manipulation trajectories for putting
the eggplant in the blue sink.

Two trajectories are provided:
Video 1:  Ground truth trajectory (real robot execution)
Video 2:  Predicted trajectory (a rollout from a learned world
model)

Your task is to determine whether the predicted trajectory
matches the ground truth.  Carefully compare the two
trajectories in terms of:

1.  Gripper motion:
- Whether the interaction pattern matches the ground truth
- If the ground truth involves contact with the eggplant, the
predicted trajectory should also make similar contact
- If the ground truth does not involve contact, the predicted
trajectory should also avoid contact
```

```
- Pay attention to whether the gripper correctly engages (or
does not engage) with the eggplant when required

2.  Final outcome:
- Whether the gripper state in the predicted trajectory matches
that in ground truth trajectory
- Whether the eggplant state in the predicted trajectory
matches that in ground truth trajectory

Important:
- Small visual differences (lighting, texture, rendering
artifacts) should be ignored
- Focus only on action and interaction correctness

Output exactly one word:

MATCH
or
MISMATCH


You are evaluating robot manipulation trajectories for putting
the eggplant in the yellow basket.

Two trajectories are provided:
Video 1:  Ground truth trajectory (real robot execution)
Video 2:  Predicted trajectory (a rollout from a learned world
model)

Your task is to determine whether the predicted trajectory
matches the ground truth.  Carefully compare the two
trajectories in terms of:

1.  Gripper motion:
- Whether the interaction pattern matches the ground truth
- If the ground truth involves contact with the eggplant, the
predicted trajectory should also make similar contact
- If the ground truth does not involve contact, the predicted
trajectory should also avoid contact
- Pay attention to whether the gripper correctly engages (or
does not engage) with the eggplant when required

2.  Final outcome:
- Whether the gripper state in the predicted trajectory matches
that in ground truth trajectory
- Whether the eggplant state in the predicted trajectory
matches that in ground truth trajectory

Important:
- Small visual differences (lighting, texture, rendering
artifacts) should be ignored
- Focus only on action and interaction correctness

Output exactly one word:

MATCH
or
MISMATCH
```

Error bars. We estimate uncertainty using bootstrap resampling over trajectories. For each model and each split, we resample the binary VLM match labels with replacement 10,000 times and recompute the mean match rate for each bootstrap sample. Error bars in Fig. 4 show ± 1.96 times the bootstrap standard error. We use the same procedure for successful and failed robot trajectories.

D.4 POLICY RANKING IN OPENWORLD

Table 12: **Detailed Bridge evaluation results** comparing RT-1, Octo, OpenVLA, π_0, $\pi_{0.5}$, GR00T N1.6, and GR00T N1.7 across the 12 reported Bridge task groups. We report task success rates (%) and their macro-average with the descriptive error defined below. The two panels show complementary policy columns for the same task groups.

(a) RT-1, Octo, and OpenVLA

Task	RT-1	Octo	OpenVLA
Move Carrot Left of Object.	16.7	50.0	66.7
Move Carrot Right of Object.	11.1	33.3	66.7
Move Mushroom Left of Object.	10.0	30.0	50.0
Move Mushroom Right of Object.	10.0	60.0	80.0
Place Carrot on Cloth.	0.0	44.4	66.7
Place Spoon on Cloth.	0.0	20.0	50.0
Put Broccoli on Cloth.	0.0	11.1	44.4
Put Carrot in Pot.	11.1	33.3	77.8
Put Corn in Pot.	0.0	50.0	80.0
Put Eggplant in Pot.	10.0	50.0	70.0
Put Spoon on Cloth.	0.0	50.0	70.0
Put Mushroom in Pot.	10.0	40.0	100.0
Mean Success Rate	6.6±1.7	39.3±3.9	68.5±4.3

(b) π_0, $\pi_{0.5}$, GR00T N1.6, and GR00T N1.7

Task	π_0	$\pi_{0.5}$	GR00T N1.6	GR00T N1.7
Move Carrot Left of Object.	42.9	100.0	66.7	71.4
Move Carrot Right of Object.	66.7	44.4	33.3	62.5
Move Mushroom Left of Object.	30.0	90.0	30.0	50.0
Move Mushroom Right of Object.	80.0	70.0	30.0	60.0
Place Carrot on Cloth.	88.9	66.7	44.4	77.8
Place Spoon on Cloth.	50.0	70.0	10.0	50.0
Put Broccoli on Cloth.	55.6	55.6	33.3	11.1
Put Carrot in Pot.	66.7	44.4	55.6	55.6
Put Corn in Pot.	60.0	40.0	50.0	20.0
Put Eggplant in Pot.	90.0	70.0	60.0	70.0
Put Spoon on Cloth.	30.0	30.0	50.0	60.0
Put Mushroom in Pot.	50.0	60.0	10.0	50.0
Mean Success Rate	59.2±5.6	61.8±5.7	39.4±5.0	53.2±5.5

Each policy takes an initial frame and a language instruction as input, generates control actions, receives the resulting video from OpenWorld, and conditions on the latest generated video to continue the policy rollout. We also train a state-prediction model to support rollouts of policies that condition on states. Rollouts are evaluated using GPT-5.5 (OpenAI, 2026) as a VLM judge for task success.

We evaluate the policies on the 12 reported task groups in Table 12. The groups include natural-language instruction variants such as "left of the orange cloth" and "left of the blue spoon". The base variants of OpenVLA, Octo (base-1.5), and RT-1 were compatible with rollouts in the Bridge setup. Versions post-trained on Bridge were directly available for both GR00T variants. Both π policies required explicit finetuning on the Bridge dataset.

For each policy, let r_j be the displayed success rate for row j and $K = 12$. We compute the table's macro-average and the corresponding point in Figure 5 as $\bar{r} = K^{-1} \sum_j r_j$. The reported $\pm$ values and figure error bars are $\sqrt{K^{-1} \sum_j (r_j - \bar{r})^2} / \sqrt{K}$, using the population standard deviation of the displayed row rates. The bars summarize dispersion across the reported task groups. Each row receives equal weight, including the two separately reported spoon-on-cloth groups. The dashed line in Figure 5 is an unweighted least-squares fit of the seven plotted macro-average success rates against

their displayed year coordinates. It describes this set of policies; individual policies need not improve monotonically with year.

OpenVLA achieves the highest mean success rate in this evaluation. Within the π and GR00T families, the newer variants achieve higher mean success rates than their respective predecessors.

E Details of the Real-Time Human Interaction Simulation Demo

E.1 Self-Forcing Distillation

To support real-time simulation of human interaction, we further distill the action-conditioned world model following Self-Forcing (Huang et al., 2025) to reduce exposure bias (Ning et al., 2024) during autoregressive rollout. Starting from the initial frame, the model autoregressively generates future frames conditioned on the action sequence. We apply the distillation objective to these self-generated rollouts. We use Distribution Matching Distillation (DMD) (Yin et al., 2024b;a) for this stage. This aligns training with inference-time rollout and helps reduce compounding errors over long horizons.

E.2 Aligning the Meta Quest to EgoDex Action Spaces for Teleoperation

For the EgoDex teleoperation interface, we maintain the full upper-body and hand skeleton as 68 skeletal transform keys plus the camera transform. During teleoperation, however, we do not attempt to drive this entire skeleton from Meta Quest tracking. Instead, we initialize the full EgoDex action state from the first frame and update only the hand joints that can be reliably mapped from Quest hand tracking. Meta Quest/OpenXR provides 26 tracked hand joints per hand; we use 24 of them (the wrist, thumb proximal/distal/tip, and the metacarpal/proximal/intermediate/distal/tip joints of the index, middle, ring, and little fingers) because these correspond exactly to the joints tracked in EgoDex. These 24 Quest joints are mapped to the corresponding EgoDex left/right hand keys, while all remaining EgoDex skeleton joints are held fixed at their initial values. We find that this hand-only update is sufficient to generate high-quality rollouts.

Figure 6 shows real-time rollouts driven by this interface.

F Additional Results

F.1 Additional Results on Cross-Morphology Simulation

We provide additional qualitative examples of rollouts from OpenWorld across different morphologies in Figure 12.

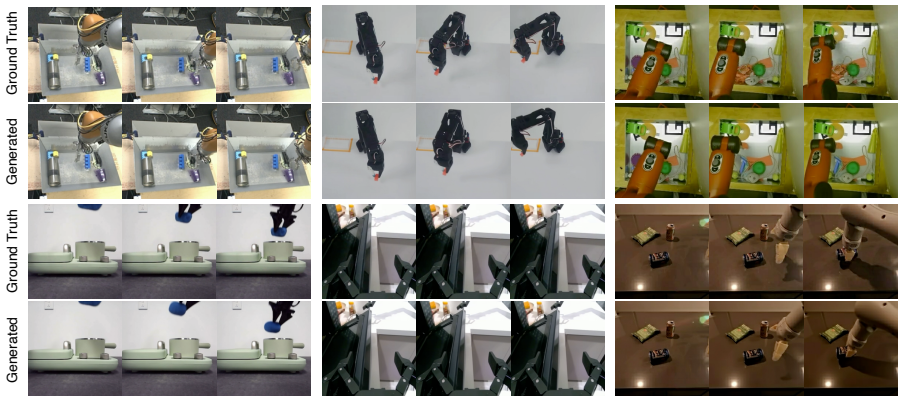

Figure 12: Qualitative examples of rollouts across additional morphologies in OpenWorld.

F.2 Additional Results on Multi-View Robot Simulation

In Figure 13, we show independently generated rollouts from different viewpoints in OpenWorld. In some failure cases, objects occluded in the initial frame are omitted from the rollout even when they become visible in the corresponding ground-truth sequence.



Figure 13: **Qualitative examples of multi-view rollouts in OpenWorld.** The bottom row shows a failure mode of single-view rollout: the green plate is occluded in the initial gripper-camera view and does not appear in the generated rollout, even when it becomes visible in the corresponding ground-truth sequence.

G Novel Downstream Domain Adaptation

G.1 Bimanual Manipulation Setup

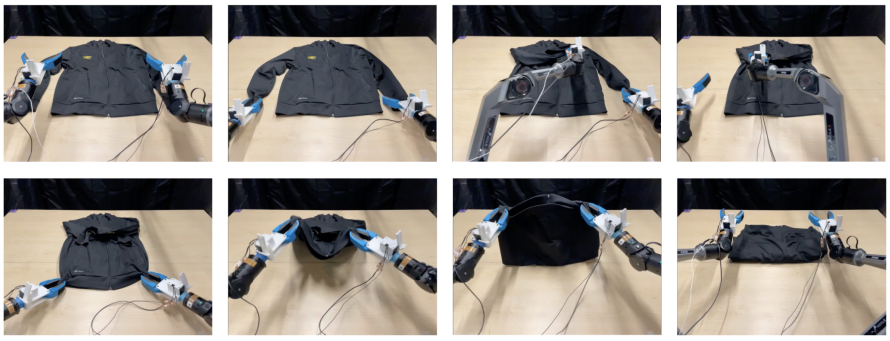

Figure 14: A successful trajectory for our custom jacket-folding task.

We set up a custom tabletop robot manipulation environment with a bimanual robot comprising two Piper 6-DoF AgileX arms and one egocentric RGB camera. We collected 214 expert jacket-folding trajectories through teleoperation (Figure 14), finetuned a $\pi_{0.5}$ (Intelligence et al., 2025) policy on them, and additionally collected 156 play trajectories. We recorded absolute end-effector poses at 10 Hz and represented the actions using 20 dimensions: Cartesian position deltas in the world frame, relative rotations encoded using a 6D representation, and a scalar gripper command for each arm.

For the adaptation comparison in this appendix, we also use a 4.85B model (4,847,579,200 trainable parameters; 32 layers, hidden dimension 2048, and 16 attention heads). All adaptation runs use 10-frame clips and an effective batch size of 16.

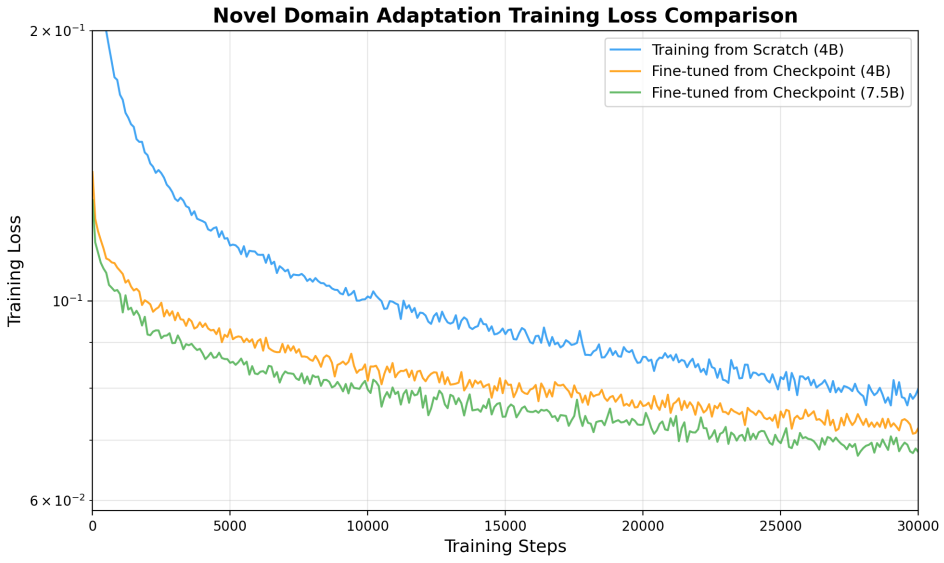

Figure 15: **Training loss for novel domain adaptation.** The pretrained 4.85B and 7.5B models reach lower training losses than the randomly initialized 4.85B model within the displayed training budget. In the legend, 4B denotes the additional 4.85B model.

The comparison between the pretrained and randomly initialized 4.85B models in Figure 15 supports effective knowledge transfer to this novel robot manipulation setting: pretraining yields a lower initial logged training loss and lower loss within the displayed adaptation budget.

H APPLICATION FIGURES

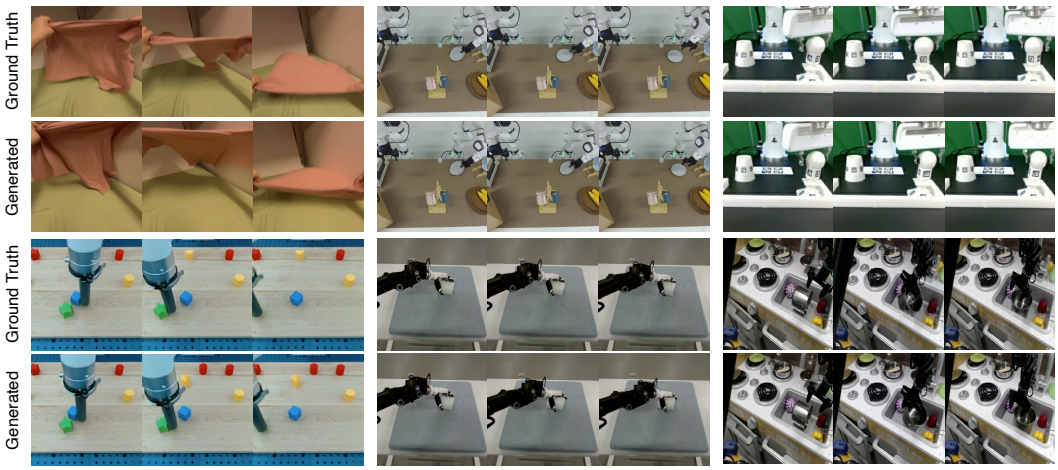

Figure 16: **Simulating diverse morphologies with OpenWorld.** We roll out held-out action sequences across 27 distinct robot embodiments and humans. The examples show future frames consistent with the morphology and action space of each embodiment. See additional results in Appendix F.1.

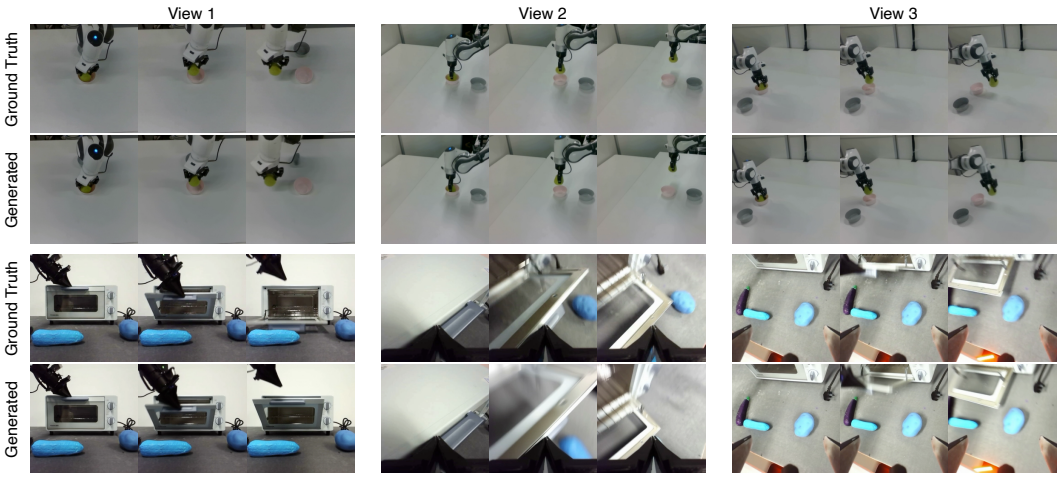

Figure 17: **Simulating multiple viewpoints in OpenWorld.** We roll out held-out action sequences from different camera viewpoints given the same initial scene. OpenWorld can follow actions across viewpoints, but independently generated views are not always mutually consistent. In the bottom row, each view is individually plausible, but view 1 shows a half-open toaster oven door, while the other views show it fully open. See additional multi-view rollouts in Appendix F.2.

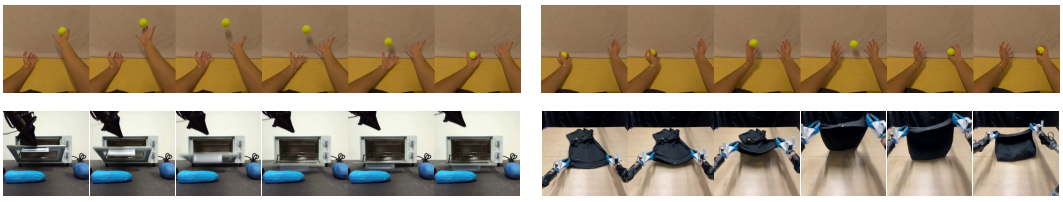

Figure 18: **Physical phenomena** simulated by OpenWorld-7.5B. Each panel shows generated videos conditioned on actions from humans or robots. The top row shows two egocentric real-time simulations of ball tossing, where a tennis ball follows a gravity-driven trajectory. The bottom-left panel shows a toaster oven door falling open under gravity and bouncing after impact. The bottom-right panel shows contact-rich robot manipulation of a deformable object: as the grippers lift and move the object, the generated object undergoes plausible shape changes and casts shadows on the table.