DREAMGEN: Unlocking Generalization in Robot Learning through Video World Models

Joel Jang[1,2,*] **Seonghyeon Ye**[1,3,*] **Zongyu Lin**[1,4,*] **Jiannan Xiang**[1,5,*]
Johan Bjorck[1] **Yu Fang**[1] **Fengyuan Hu**[1] **Spencer Huang**[1] **Kaushil Kundalia**[1] **Lin Yen-Chen**[1]
Loic Magne[1] **Ajay Mandlekar**[1] **Avnish Narayan**[1] **You Liang Tan**[1] **Guanzhi Wang**[1,6]
Jing Wang[1,7] **Qi Wang**[1] **Yinzhen Xu**[1] **Xiaohui Zeng**[1] **Kaiyuan Zheng**[2] **Ruijie Zheng**[1,8]
Ming-Yu Liu[1] **Luke Zettlemoyer**[2] **Dieter Fox**[1,2] **Jan Kautz**[1]
Scott Reed[1,†] **Yuke Zhu**[1,9,†] **Linxi Fan**[1,†]

[1]NVIDIA [2]University of Washington [3]KAIST [4]UCLA [5]UCSD
[6]CalTech [7]NTU [8]University of Maryland [9]UT Austin

https://research.nvidia.com/labs/gear/dreamgen

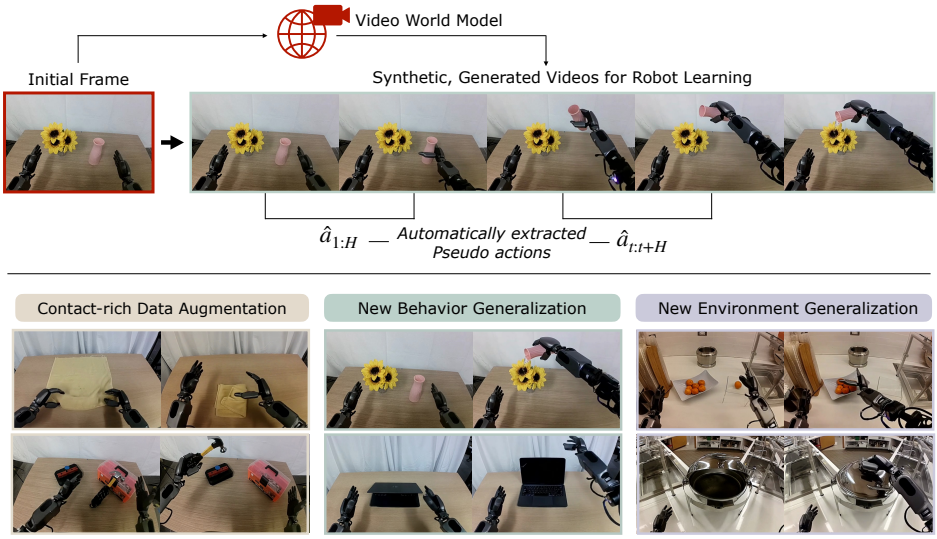


Figure 1: **Generalization through DREAMGEN**. We enable 2D visuomotor robot policies to generalize to **new environments** with **new behaviors**, while *only* collecting teleoperation data for a **single** behavior type (pick&place) in a **single** environment by utilizing video world models as synthetic data generators.

Abstract: We introduce DREAMGEN, a simple yet highly effective 4-stage pipeline for training robot policies that generalize across behaviors and environments through *neural trajectories*—synthetic robot data generated from video world models. DREAMGEN leverages state-of-the-art image-to-video generative models, adapting them to the target robot embodiment to produce photorealistic synthetic videos of familiar or novel tasks in diverse environments. Since these models generate only videos, we recover pseudo-action sequences using either a latent action model or an inverse-dynamics model (IDM). Despite its simplicity, DREAMGEN unlocks strong behavior and environment generalization: a humanoid robot can perform 22 new behaviors in both seen and unseen environments, while requiring teleoperation data from only a single pick-and-place

[*]Equal contribution.
[†]Equal advising.

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

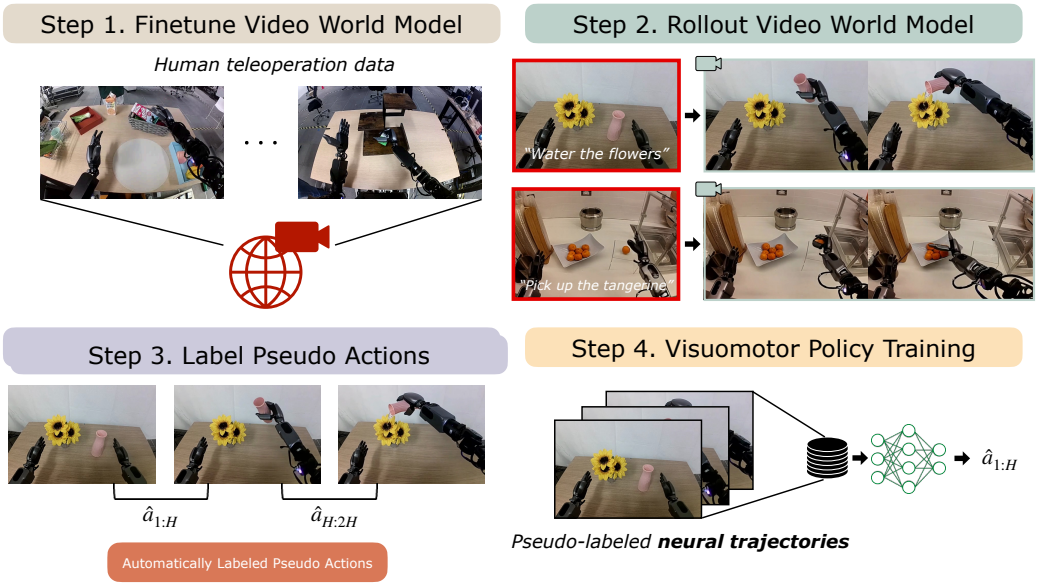


Figure 2: **DreamGen Overview.** We begin by fine-tuning a video world model on teleoperated robot trajectories. Given an initial frame and a language instruction, the model generates video rollouts depicting the intended behavior. As these videos lack action annotations, we infer pseudo-actions using either a latent action model or an inverse dynamics model, forming what we call *neural trajectories*. Finally, we train visuomotor robot policies on these neural trajectories.

task in one environment. To evaluate the pipeline systematically, we introduce DreamGenBench, a video generation benchmark that shows a strong correlation between benchmark performance and downstream policy success. Our work establishes a promising new axis for scaling robot learning well beyond manual data collection.

1 Introduction

Robot foundation models trained on large-scale human teleoperation data have shown strong potential for general-purpose robotic systems to perform dexterous real-world tasks [1, 2, 3, 4, 5, 6]. However, this paradigm relies heavily on collecting teleoperation data manually for every new task and environment, which remains costly and labor-intensive. Synthetic data generation in simulation offers an appealing alternative, but it often requires significant manual engineering and suffers from sim2real gap when deploying visuomotor policies on physical robots. To address these challenges, we propose DreamGen, a new synthetic data pipeline that leverages video world models to create realistic training data at scale with minimal manual labor or engineering.

DreamGen follows a simple 4-step recipe (Figure 2) for applying state-of-the-art video generative models [7, 8, 9, 10, 11, 12], also known as *video world models*, to generate synthetic training data. This pipeline is designed to be general-purpose across different robots, environments, and tasks. (1) We fine-tune video world models on a target robot to capture the dynamics and kinematics of the specific embodiment; (2) we prompt the model with pairs of initial frames and language instructions to generate large volumes of robot videos, capturing both familiar behaviors from fine-tuning and novel ones in unseen settings; (3) we then extract pseudo-actions using either a latent action model [13] or an inverse dynamics model (IDM)[14]; (4) finally, we use the resulting video-action sequence pairs, dubbed *neural trajectories*, for training downstream visuomotor policies. While prior work has focused on using video world models as real-time planners [15, 16, 17, 18,

19], DREAMGEN instead treats them as *synthetic data generators*, unlocking their strong priors for physical reasoning, naturalistic motion, and language grounding.

First, we investigate DREAMGEN for generating additional training data for tasks where teleoperation data is already available, both in simulation and the real world. In simulation, we apply DREAMGEN to the RoboCasa benchmark [20], scaling synthetic data up to 333× relative to the original human demonstrations. This yields log-linear improvements in policy performance as the number of neural trajectories increases (Figure 4). In the real world, we validate our approach on 9 diverse tasks on Fourier GR1, Franka Emika, and SO-100 robots, demonstrating the flexibility of our pipeline across embodiments and challenging dexterous tasks that are difficult to simulate, such as folding towels, wiping liquids, using hammers, and scooping M&Ms. DREAMGEN show consistent gains on success rate across all robots: from 37% to 46.4% on average of 4 GR1 humanoid tasks, 23% to 37% on average of 3 Franka tasks, and from 21% to 45.5% on average of 2 SO-100 tasks, all using just 10 to 13 real-world trajectories per task.

Next, we highlight two key generalization capabilities unlocked by DREAMGEN: **behavior generalization** and **environment generalization**. For behavior generalization, we enable the GR1 humanoid to perform 22 novel behaviors, such as pouring, opening/closing articulated objects, and manipulating a variety of tools. Note that the original teleoperation dataset only includes pick-and-place and no other verbs. For environment generalization, we prompt video world models (fine-tuned on just a single environment) with initial frames from 10 new environments. This allows us to train visuomotor policies that generalize to novel behaviors and settings using only teleoperation data from a single task in a single environment. These represent true zero-to-one improvements – GR00T N1 trained on pick-and-place alone achieves 0% success rates on most novel behavior and environment experiments, while DREAMGEN enables 43.2% success rates on new behaviors in seen environments and 28.5% in completely unseen environments. These empirical results point towards a new paradigm for scalable robot learning without extensive manual demonstrations.

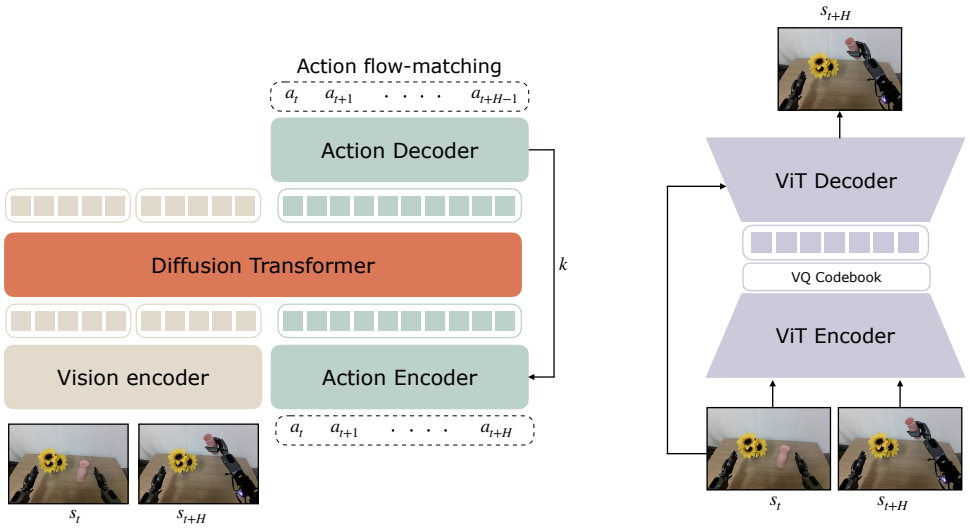
Lastly, we introduce DREAMGENBENCH (Appendix B), a new video generation benchmark designed to evaluate how well different video world models adapt to novel robot embodiments. We assess whether 8 models, 4 zero-shot and 4 fine-tuned, can generate robot videos that involve manipulating unseen objects, performing unseen behaviors, and operating in unseen environments, all while abiding by the laws of physics. Empirically, we find that models with higher scores also yield stronger downstream robot policy performance. DREAMGENBENCH provides a diagnostic and low-cost way to connect video world models to robotics, without requiring a physical robot in the loop. We hope this offers an accessible pathway for video model researchers to contribute to robot learning.

2 DREAMGEN

In the next subsections, we describe in detail the 4 different steps (shown in Figure 2) of DREAMGEN, creating and utilizing neural trajectories to train visuomotor robot policies.

2.1 Video World Model Fine-tuning

In the initial phase, we fine-tune video world models on human-teleoperated robot trajectories. This adaptation enables the model to learn the robot's physical constraints and movement capabilities. To mitigate forgetting prior internet video knowledge, we use Low-Rank Adaptation (LoRA) [21] by default for the different video world model fine-tuning we conduct. When fine-tuning these models, we look at two metrics, *instruction following* and *physics following*, to determine whether the video world model has been optimally adapted to the target robot domain (details provided in Section B). For the majority of our downstream robot experiments, we utilize WAN2.1 [9] as our base video world model. In cases where there are multiple viewpoints in the training dataset (RoboCasa [20] and DROID [22]), we concatenate the viewpoints into a 2×2 grid (with one grid with black pixels)



(a) Inverse Dynamics Model (IDM) [23]

(b) LAPA [13]

Figure 3: **Extracting Pseudo Actions.** (a) shows the architecture of our IDM model and (b) shows the architecture of our latent action model.

and fine-tune the video world models.[1] We also observe that the optimal amount of fine-tuning required for each video world model and fine-tuning data pair differs.[2]

2.2 Video World Model Rollout

After fine-tuning the video world models on the target robot embodiment, we generate synthetic robot videos using various initial frames and language instructions. For simulation experiments, we collect new initial frames from the simulator, randomizing the locations of the target objects or environments for each task. For real-world experiments, we manually take new initial frames while randomizing the location of the target object. For environment generalization experiments, we also take initial frames of new environments, while we restrict ourselves to training the video world model collected from a single environment (pictures shown in Appendix D). Lastly, we manually come up with novel behavior prompts for the behavior generalization experiments, and also include all of the candidates in our video benchmark in Section B.[3]

2.3 Pseudo Action Labeling

Figure 3 shows the (a) architecture we use to train the IDM model and the (b) architecture that we use to train the latent action model (LAPA), both used to extract pseudo action labels for the generated videos.

IDM Actions. For the inverse dynamics model (IDM) architecture, we use diffusion transformers with SigLIP-2 vision encoder and train with a flow matching objective. IDM is conditioned on two image frames and is trained to predict action chunks between the image frames (Figure 3). We do not explicitly use any language or proprioception as input, since we want the IDM model to only capture the dynamics of the robot. For the IDM training data, we use the same dataset used to train

[1] Examples are shown in Appendix E.

[2] We provide the hyperparameters (learning rate, number of epochs, etc.) used for all of the experimental setups in Appendix F.

[3] Even though collecting new initial frames requires some manual work, it significantly alleviates the need for collecting new teleoperation data. Furthermore, we hope to utilize image-to-image diffusion techniques to alleviate this burden, where we can start off with a single initial frame, and randomize new initial frames by impainting the object locations, type of objects, as well as the environment for future work.

the video world models for each setup, unless explicitly stated otherwise. After training, we employ a sliding window approach for pseudo-labeling: the IDM predicts H actions, $\hat{a}_t$ to $\hat{a}_{t+H}$. Next, it slides one window and predicts another H actions, $\hat{a}_{t+1}$ to $\hat{a}_{t+1+H}$, and so forth. More details are provided in Appendix C.

Latent Actions. For latent actions, we use the LAPA latent action model [13], which has a transformer encoder-decoder architecture and is trained on diverse robot and human videos. The latent action model is trained with a VQ-VAE objective so that the latent actions can capture the visual delta information between two frames in a video. To obtain the latent actions from the generated videos, we condition the latent action model on the current frame and the future frame (1 second ahead) of the trajectory. We use the pre-quantized continuous embedding as the latent action following GR00T N1 [5]. The exact training data mixture used to train the latent action model is provided in Table 3. One benefit of latent actions is that it does not require actually having ground-truth actions for the target robot embodiment when training latent action models.

2.4 Policy Training on Neural Trajectories

Lastly, we train visuomotor robot policies on neural trajectories generated by DREAMGEN by conditioning on language instruction and image observations. We condition state information with zero values, since neural trajectories do not contain state information.[4] More specifically, given o_t, the image observation, and i_t, the task instruction, we train the policies to generate $\hat{a}_{t:t+H}$, which can be either latent actions or IDM-labeled actions from the previous subsection. Since neural trajectories are independent of the underlying robot policy architecture, we showcase the effectiveness of DREAMGEN for generating synthetic training data for 3 different visuomotor policy models, Diffusion Policy [24], π_0 [2], and GR00T N1 [5].

We propose two scenarios of training with neural trajectories: co-training with real-world trajectories, and solely training on the neural trajectories labeled with IDM actions. When we co-train neural trajectories with real trajectories, we co-train with a sampling ratio of 1:1. For GR00T N1, we treat the two types of trajectories as separate embodiments by using separate action encoder and decoder. For behavior and environment generalization experiments, we only use neural trajectories for policy training.

3 Experiments

In this section, we demonstrate three key applications of DREAMGEN: (1) Augmenting training data for existing tasks, (2) Enabling generalization to novel behaviors, and (3) Enabling generalization to novel environments.

3.1 Training Data Augmentation

For simulation experiments, we evaluate our pipeline on the RoboCasa benchmark [20], using the same training and evaluation protocol as outlined in the original work. For real-world experiments, we evaluate on 9 real-world tasks across three embodiments: the GR1 humanoid robot, the Franka arm robot, and the low-cost SO-100 robot arm.

Simulation experiments Figure 4 shows the downstream robot policy results as we scale the total number of neural trajectories in three different scenarios of ground-truth data: low data (720), mid-data (2.4k), and high-data (7.2k) on RoboCasa. Each scenario determines how *strong* our IDM model can become, since the more ground-truth data we have about a given robot, the more useful dynamics the model can learn. In this particular setup, we train our video world model on 1,200

[4] From preliminary experiments, we observed that having zero state does not harm the performance. We leave training the IDM to predict state information for future work.

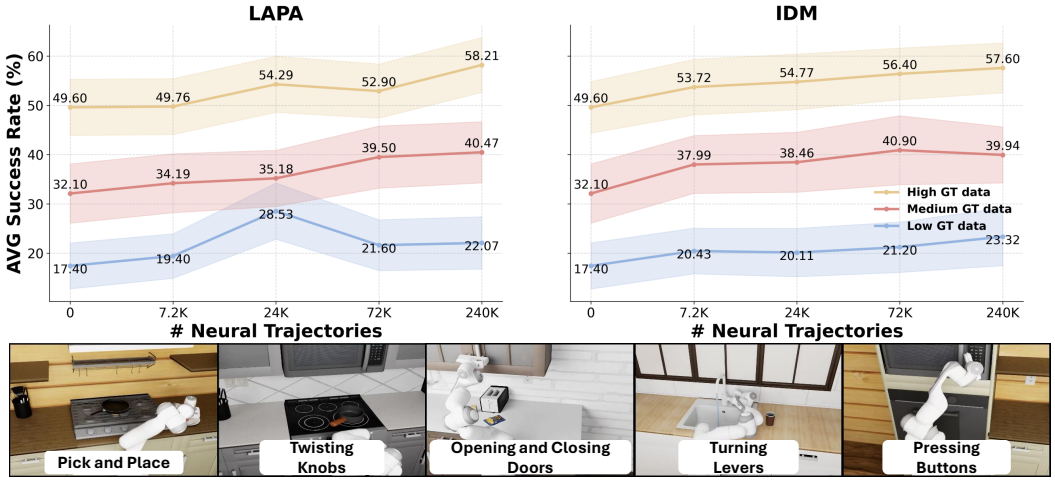


Figure 4: **Scaling # of Neural Trajectories in RoboCasa**. We vary the sizes of neural trajectories (x-axis) and ground-truth trajectories (low, mid, high) and report results with both latent and IDM actions as pseudo action labels. We report the average success rate (%) across 24 tasks. The results at $x = 0$ correspond to the baseline **only** trained on ground-truth videos.

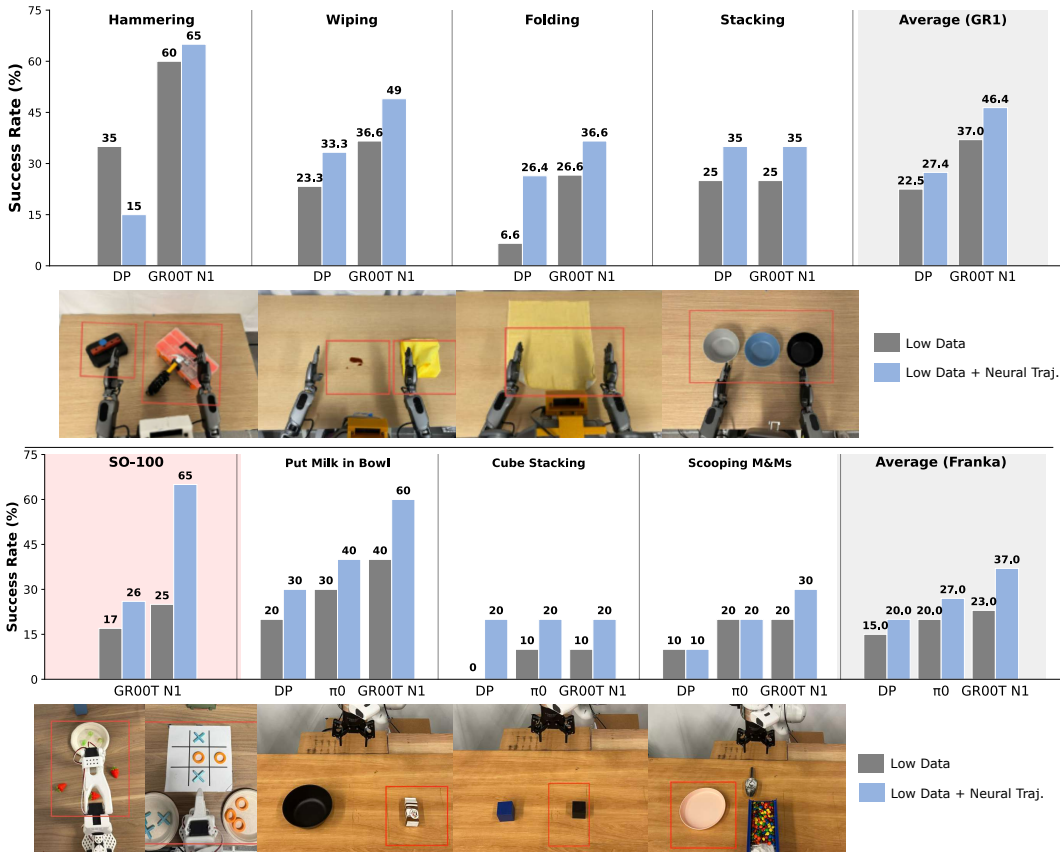


Figure 5: **Real-world Robot Evaluation Results**. The red rectangular box shows the range of object randomization during training and evaluation. *Low Data* denotes training 10% of available training data (only 10 trajectories per task except for GR1-folding, where we used 25 trajectories), and *Low Data + Neural Traj.* denotes co-training with neural trajectories.

original human demonstrations, whereas IDM and policy training are conducted in different data scenarios from the benchmark.[5]

First, we observe that co-training with neural trajectories yields a performance boost for both IDM and LAPA actions across all data regime scenarios. Since both approaches have similar effects, we use IDM as the default for the rest of the experiments, as IDM actions enable *solely* training on neural trajectories and evaluating the policy performance, and in all of our experimental set-up, we do have access to teleoperation data to train strong enough IDMs for each robot embodiment. [6] Second, we observe that there is a consistent log-linear slope between the total number of neural trajectories and the downstream robot policy performance. This hints towards a potential for a new paradigm in robot learning, as synthetic data generation through neural trajectories is significantly more scalable compared to the traditional method of manual teleoperation for imitation learning. Lastly, we show that *solely* training on neural trajectories with IDM actions enables us to reach a non-trivial performance (20.6% average success rate across 24 tasks), further highlighting the quality of neural trajectories (a detailed breakdown of results is provided in Appendix G).

Real-world Experiments For real-world experiments, we collect 100 trajectories per task for the four GR1 and three Franka tasks. For the two SO-100 tasks, we collect 40 and 50 trajectories for the strawberry pick-and-place and tic-tac-toe tasks, respectively. Details of the data collection and evaluation criteria for each of the 9 tasks are provided in the Appendix K, and details of the video world model training procedure for each task are provided in Appendix H. As default, we use only 10% of the collected trajectories for our main experiment to test data efficiency for GR1 and Franka tasks (only 10 real-world trajectories per task) and 25% of the collected trajectories for SO-100 tasks (10 and 13 trajectories per task).[7] We generate 300 neural trajectories for each GR1 task, 100 neural trajectories for each Franka task, and 40 and 50 neural trajectories for the two SO-100 tasks, respectively, to co-train with real-world trajectories with a 1:1 sampling ratio.

As shown in Figure 5, neural trajectories consistently improve performance for different visuomotor policies (Diffusion Policy, π_0, and GR00T N1) across all robot embodiments for dexterous tasks involving tool manipulation, manipulation with deformable objects, and pick-and-place. Importantly, these tasks present significant simulation challenges due to their complex physical interactions with tools and deformable materials, making synthetic data generation infeasible with current approaches in the literature. Empricially, we observe a higher performance gain for GR00T N1 compared to DP and π_0; we hypothesize that having separate action and decoder parameters for the IDM actions help with the fact that neural trajectories have 0's as state.

3.2 Unlocking Generalization

To demonstrate how DREAMGEN can unlock generalization in robot learning, we train our target video world model on 2,884 trajectories of the GR1 Humanoid performing diverse pick-and-place motions. Next, we prompt the model with (1) novel behaviors in seen environments and (2) seen and novel behaviors in novel environments, generating neural trajectories. The visualization of the evaluation configuration (how much randomization is done for the target object) is provided in Figure 11. We use GR00T N1 as the base policy for this section.

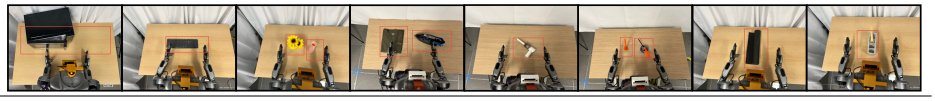
Behavior Generalization We investigate whether our pipeline enables robots to learn entirely new behaviors *solely* from neural trajectories without involving any human teleoperation. We define "new behaviors" as novel action *verbs* beyond adapting existing motions. Surprisingly, just given the initial frame and the language instruction, we observe that the video world model can generalize

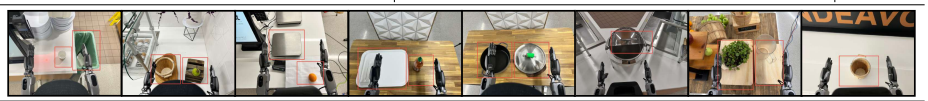
[5] RoboCasa Benchmark consists of three different viewpoints for visuomotor policy training: left, right, and wrist. We utilize GR00T N1 [5] as the base robot policy for this experiment.

[6] Enabling zero-shot generalization to novel behaviors and novel environments with robot embodiments with *zero* ground-truth data still remains an open research question.

[7] We also provide the evaluation results of models trained on "High Data" (100% of training data) in Appendix I.

Table 1: **Success Rate (%) Across New Behaviors (14 tasks) and Environments (13 tasks).**

	Seen Environments, Novel Behaviors														
Model	Open Microwave	Open Macbook	Close Lunchbox	Hit Tambourine	Hit Keyboard	Grab button	Pour Water	Water flowers	Light Candle	Use Vacuum	Iron shirt	Take Spoon Out	Unroll mat	Move Mouse	Average
GR00T N1	0	0	0	5	0	45	40	50	10	0	0	7	0	0	11.2
w/ DREAMGEN	**23**	**45**	**10**	**15**	**90**	**75**	**55**	**95**	**15**	**55**	**20**	**17**	**55**	**35**	**43.2**
Examples															

	Novel Environments, Seen Behaviors						Novel Environments, Novel Behaviors							
Model	Pick up Tangerine	Box sandwich	Weigh the Orange	Put cup in trash	Put pear in basket	Put sauce on tray	Water Flowers	Lift Basket	Swirl Around Spoon	Use Whisk	Close soup container	Uncover Pot	Cover Pot	Average
GR00T N1	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0
w/ DREAMGEN	**30**	**10**	**20**	**45**	**35**	**45**	**15**	**55**	**15**	**25**	**55**	**30**	**35**	**28.5**
Examples														

in generating videos of totally unseen behaviors (examples shown in Figure 12). We recommend referring to the website [8] for better visualizations. Leveraging this capability, we generate 50 neural trajectories for each of the 14 novel behavior tasks and train our downstream visuomotor robot policy only on the neural trajectories. As shown in Table 1, we first show the result of GR00T N1 fine-tuned on the 2,885 pick-and-place trajectories, which also gets a somewhat non-trivial performance (11.8%), due to some of the tasks giving partial points for picking up the object (e.g. for example, we give 0.5 success for picking up the bottle for the "Pour Water" task). Nonetheless, we see a non-trivial performance gain when trained with neural trajectories (11.2% $\rightarrow$ 43.2%), showing that our pipeline enables learning totally new verbs.

Environment Generalization To our surprise, when prompted with initial frames of totally new environments, we observe that video world models can still generalize and generate very realistic robot videos, following the kinematics it learned during fine-tuning, while retaining the internet-video knowledge learned during pretraining. We follow the same proposed pipeline and train visuomotor robot policies *solely* on neural trajectories, and observe that we can get non-trivial success rates on both seen behaviors (variants of pick-and-place) and unseen behaviors (e.g., watering flowers, closing containers, stirring whisk, etc.) as shown in Table 1. Importantly, unlike previous work that showed environment generalization by scaling the total number of environments in the training data [6], our approach did not require any physical data collection beyond a single environment (i.e., lab setup)—we only capture initial frames, effectively implementing a *zero-shot* transfer methodology. Lastly, the baseline model trained only on pick-and-place in a single environment shows 0% Success Rate, since it does not have the ability to generalize beyond the environment it was trained in.

4 Conclusion

We introduce a novel pipeline for robot learning that taps into the power of SOTA video generative models. By generating synthetic videos and extracting pseudo-actions, we enable training visuomotor policies without relying solely on manual demonstrations. This approach not only augments existing tasks but also unlocks the ability to learn entirely new behaviors in unseen environments. DREAMGEN serves as a solid stepping stone towards unleashing the full potential of world models in robotics.

5 Limitation

Our approach is complementary to existing methods that learn from videos, although we do not directly benchmark against them. Many of these works focus on learning from human demonstration

[8] https://research.nvidia.com/labs/gear/dreamgen

videos. Since DREAMGEN helps bridge the human-robot domain gap, we believe it can serve as a useful foundation for improving such methods and enabling broader generalization. Our tasks are relatively simple and cover a limited portion of the robot's full kinematic capabilities. Supporting more complex, dexterous behaviors that require richer control remains an important direction for future work. Increasing the diversity of training behaviors, along with broader video-language pairings, may allow the video world model to take on more of the representational burden and improve generalization to challenging tasks.

DREAMGEN currently requires significant compute. For instance, generating the 240k-sample RoboCasa dataset took 54 hours on 1500 NVIDIA L40 GPUs. While feasible in a large-scale research setting, reducing computational cost without sacrificing the strength of video priors remains an important challenge. The method also relies on manually providing initial frames, which introduces operational overhead. Developing automated ways to generate or select initial frames is a promising future direction.

Finally, the automatic evaluator used in DREAMGENBENCH is based on lightweight open-source models to keep the benchmark accessible. These models can occasionally hallucinate, especially when evaluating physical realism in videos, which remains a difficult and evolving problem. We acknowledge this limitation and leave improvements in evaluation to future work.

References

[1] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, P. Florence, C. Fu, M. G. Arenas, K. Gopalakrishnan, K. Han, K. Hausman, A. Herzog, J. Hsu, B. Ichter, A. Irpan, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, L. Lee, T.-W. E. Lee, S. Levine, Y. Lu, H. Michalewski, I. Mordatch, K. Pertsch, K. Rao, K. Reymann, M. Ryoo, G. Salazar, P. Sanketi, P. Sermanet, J. Singh, A. Singh, R. Soricut, H. Tran, V. Vanhoucke, Q. Vuong, A. Wahid, S. Welker, P. Wohlhart, J. Wu, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *arXiv preprint arXiv:2307.15818*, 2023.

[2] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, et al. $\pi 0$: A vision-language-action flow model for general robot control, 2024. *URL https://arxiv. org/abs/2410.24164*.

[3] G. R. Team, S. Abeyruwan, J. Ainslie, J.-B. Alayrac, M. G. Arenas, T. Armstrong, A. Balakrishna, R. Baruch, M. Bauza, M. Blokzijl, et al. Gemini robotics: Bringing ai into the physical world. *arXiv preprint arXiv:2503.20020*, 2025.

[4] Q. Bu, J. Cai, L. Chen, X. Cui, Y. Ding, S. Feng, S. Gao, X. He, X. Huang, S. Jiang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025.

[5] J. Bjorck, F. Castañeda, N. Cherniadev, X. Da, R. Ding, L. Fan, Y. Fang, D. Fox, F. Hu, S. Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

[6] P. Intelligence, K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

[7] N. Agarwal, A. Ali, M. Bala, Y. Balaji, E. Barker, T. Cai, P. Chattopadhyay, Y. Chen, Y. Cui, Y. Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

[8] Z. Yang, J. Teng, W. Zheng, M. Ding, S. Huang, J. Xu, Y. Yang, W. Hong, X. Zhang, G. Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.

[9] A. Wang, B. Ai, B. Wen, C. Mao, C.-W. Xie, D. Chen, F. Yu, H. Zhao, J. Yang, J. Zeng, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

[10] W. Kong, Q. Tian, Z. Zhang, R. Min, Z. Dai, J. Zhou, J. Xiong, X. Li, B. Wu, J. Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

[11] Z. Lin, W. Liu, C. Chen, J. Lu, W. Hu, T.-J. Fu, J. Allardice, Z. Lai, L. Song, B. Zhang, et al. Stiv: Scalable text and image conditioned video generation. *arXiv preprint arXiv:2412.07730*, 2024.

[12] J. Xiang, G. Liu, Y. Gu, Q. Gao, Y. Ning, Y. Zha, Z. Feng, T. Tao, S. Hao, Y. Shi, et al. Pandora: Towards general world model with natural language actions and video states. *arXiv preprint arXiv:2406.09455*, 2024.

[13] S. Ye, J. Jang, B. Jeon, S. J. Joo, J. Yang, B. Peng, A. Mandlekar, R. Tan, Y.-W. Chao, B. Y. Lin, L. Liden, K. Lee, J. Gao, L. Zettlemoyer, D. Fox, and M. Seo. Latent action pretraining from videos. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=VYOe2eBQeh.

[14] B. Baker, I. Akkaya, P. Zhokov, J. Huizinga, J. Tang, A. Ecoffet, B. Houghton, R. Sampedro, and J. Clune. Video pretraining (vpt): Learning to act by watching unlabeled online videos. *Advances in Neural Information Processing Systems*, 35:24639–24654, 2022.

[15] Y. Du, S. Yang, B. Dai, H. Dai, O. Nachum, J. Tenenbaum, D. Schuurmans, and P. Abbeel. Learning universal policies via text-guided video generation. *Advances in neural information processing systems*, 36:9156–9172, 2023.

[16] S. Zhou, Y. Du, J. Chen, Y. Li, D.-Y. Yeung, and C. Gan. Robodreamer: Learning compositional world models for robot imagination. *arXiv preprint arXiv:2404.12377*, 2024.

[17] P.-C. Ko, J. Mao, Y. Du, S.-H. Sun, and J. B. Tenenbaum. Learning to act from actionless videos through dense correspondences. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Mhb5fpA1T0.

[18] S. Yang, Y. Du, S. K. S. Ghasemipour, J. Tompson, L. P. Kaelbling, D. Schuurmans, and P. Abbeel. Learning interactive real-world simulators. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=sFyTZEqmUY.

[19] Y. Du, S. Yang, P. Florence, F. Xia, A. Wahid, brian ichter, P. Sermanet, T. Yu, P. Abbeel, J. B. Tenenbaum, L. P. Kaelbling, A. Zeng, and J. Tompson. Video language planning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=9pKtcJcMP3.

[20] S. Nasiriany, A. Maddukuri, L. Zhang, A. Parikh, A. Lo, A. Joshi, A. Mandlekar, and Y. Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. In *Robotics: Science and Systems (RSS)*, 2024.

[21] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

[22] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

[23] B. Baker, I. Akkaya, P. Zhokhov, J. Huizinga, J. Tang, A. Ecoffet, B. Houghton, R. Sampedro, and J. Clune. Video pretraining (vpt): Learning to act by watching unlabeled online videos, 2022. URL https://arxiv.org/abs/2206.11795.

[24] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023.

[25] A. Mandlekar, S. Nasiriany, B. Wen, I. Akinola, Y. Narang, L. Fan, Y. Zhu, and D. Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. In *Conference on Robot Learning*, 2023.

[26] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.

[27] M. Dalal, A. Mandlekar, C. R. Garrett, A. Handa, R. Salakhutdinov, and D. Fox. Imitating task and motion planning with visuomotor transformers. In *Conference on Robot Learning*, pages 2565–2593. PMLR, 2023.

[28] J. Gu, F. Xiang, X. Li, Z. Ling, X. Liu, T. Mu, Y. Tang, S. Tao, X. Wei, Y. Yao, et al. Maniskill2: A unified benchmark for generalizable manipulation skills. In *The Eleventh International Conference on Learning Representations*, 2023.

[29] H. Ha, P. Florence, and S. Song. Scaling up and distilling down: Language-guided robot skill acquisition. In *Conference on Robot Learning*, pages 3766–3777. PMLR, 2023.

[30] Z. Jiang, Y. Xie, K. Lin, Z. Xu, W. Wan, A. Mandlekar, L. Fan, and Y. Zhu. Dexmimicgen: Automated data generation for bimanual dexterous manipulation via imitation learning. 2024.

[31] Y. Wang, Z. Xian, F. Chen, T.-H. Wang, Y. Wang, K. Fragkiadaki, Z. Erickson, D. Held, and C. Gan. Robogen: Towards unleashing infinite data for automated robot learning via generative simulation. In *International Conference on Machine Learning*, 2024.

[32] Y. Su, S. Zhou, Y. Wu, T. Su, D. Liang, J. Liu, D. Zheng, Y. Wang, J. Yan, and X. Hu. Dynamic multi-path neural network. *arXiv preprint arXiv:1902.10949*, 2019.

[33] C. Garrett, A. Mandlekar, B. Wen, and D. Fox. Skillmimicgen: Automated demonstration generation for efficient skill learning and deployment. *arXiv preprint arXiv:2410.18907*, 2024.

[34] L. Yang, H. Suh, T. Zhao, B. P. Graesdal, T. Kelestemur, J. Wang, T. Pang, and R. Tedrake. Physics-driven data generation for contact-rich manipulation via trajectory optimization. *arXiv preprint arXiv:2502.20382*, 2025.

[35] Z. Mandi, H. Bharadhwaj, V. Moens, S. Song, A. Rajeswaran, and V. Kumar. Cacti: A framework for scalable multi-task multi-scene visual imitation learning. *arXiv preprint arXiv:2212.05711*, 2022.

[36] T. Yu, T. Xiao, A. Stone, J. Tompson, A. Brohan, S. Wang, J. Singh, C. Tan, J. Peralta, B. Ichter, et al. Scaling robot learning with semantically imagined experience. *arXiv preprint arXiv:2302.11550*, 2023.

[37] Z. Chen, S. Kiami, A. Gupta, and V. Kumar. Genaug: Retargeting behaviors to unseen situations via generative augmentation. *arXiv preprint arXiv:2302.06671*, 2023.

[38] L. Y. Chen, C. Xu, K. Dharmarajan, M. Z. Irshad, R. Cheng, K. Keutzer, M. Tomizuka, Q. Vuong, and K. Goldberg. Rovi-aug: Robot and viewpoint augmentation for cross-embodiment robot learning. *arXiv preprint arXiv:2409.03403*, 2024.

[39] H. A. Alhaija, J. Alvarez, M. Bala, T. Cai, T. Cao, L. Cha, J. Chen, M. Chen, F. Ferroni, S. Fidler, et al. Cosmos-transfer1: Conditional world generation with adaptive multimodal control. *arXiv preprint arXiv:2503.14492*, 2025.

[40] J. Liang, R. Liu, E. Ozguroglu, S. Sudhakar, A. Dave, P. Tokmakov, S. Song, and C. Vondrick. Dreamitate: Real-world visuomotor policy learning via video generation. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=InT87E5sr4.

[41] H. Bharadhwaj, D. Dwibedi, A. Gupta, S. Tulsiani, C. Doersch, T. Xiao, D. Shah, F. Xia, D. Sadigh, and S. Kirmani. Gen2act: Human video generation in novel scenarios enables generalizable robot manipulation. *arXiv preprint arXiv:2409.16283*, 2024.

[42] C. Luo, Z. Zeng, Y. Du, and C. Sun. Solving new tasks by adapting internet video knowledge. In *The Thirteenth International Conference on Learning Representations*, 2025.

[43] H. Wu, Y. Jing, C. Cheang, G. Chen, J. Xu, X. Li, M. Liu, H. Li, and T. Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. *arXiv preprint arXiv:2312.13139*, 2023.

[44] C.-L. Cheang, G. Chen, Y. Jing, T. Kong, H. Li, Y. Li, Y. Liu, H. Wu, J. Xu, Y. Yang, et al. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024.

[45] Y. Guo, Y. Hu, J. Zhang, Y.-J. Wang, X. Chen, C. Lu, and J. Chen. Prediction with action: Visual policy learning via joint denoising process. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[46] S. Li, Y. Gao, D. Sadigh, and S. Song. Unified video action model. *arXiv preprint arXiv:2503.00200*, 2025.

[47] C. Zhu, R. Yu, S. Feng, B. Burchfiel, P. Shah, and A. Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. *arXiv preprint arXiv:2504.02792*, 2025.

[48] Q. Zhao, Y. Lu, M. J. Kim, Z. Fu, Z. Zhang, Y. Wu, Z. Li, Q. Ma, S. Han, C. Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. *arXiv preprint arXiv:2503.22020*, 2025.

[49] R. McCarthy, D. C. Tan, D. Schmidt, F. Acero, N. Herr, Y. Du, T. G. Thuruthel, and Z. Li. Towards generalist robot learning from internet video: A survey. *arXiv preprint arXiv:2404.19664*, 2024.

[50] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[51] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022.

[52] S. Dasari, M. K. Srirama, U. Jain, and A. Gupta. An unbiased look at datasets for visuo-motor pre-training. In *Conference on Robot Learning*, 2023.

[53] J. Zeng, Q. Bu, B. Wang, W. Xia, L. Chen, H. Dong, H. Song, D. Wang, D. Hu, P. Luo, et al. Learning manipulation by predicting interaction. *arXiv preprint arXiv:2406.00439*, 2024.

[54] S. Bahl, R. Mendonca, L. Chen, U. Jain, and D. Pathak. Affordances from human videos as a versatile representation for robotics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[55] A. Kannan, K. Shaw, S. Bahl, P. Mannam, and D. Pathak. Deft: Dexterous fine-tuning for real-world hand policies. *arXiv preprint arXiv:2310.19797*, 2023.

[56] M. K. Srirama, S. Dasari, S. Bahl, and A. Gupta. Hrp: Human affordances for robotic pre-training. *arXiv preprint arXiv:2407.18911*, 2024.

[57] K. Shaw, S. Bahl, and D. Pathak. Videodex: Learning dexterity from internet videos. In *Conference on Robot Learning*, 2023.

[58] C. Wen, X. Lin, J. So, K. Chen, Q. Dou, Y. Gao, and P. Abbeel. Any-point trajectory modeling for policy learning. *arXiv preprint arXiv:2401.00025*, 2023.

[59] H. Bharadhwaj, R. Mottaghi, A. Gupta, and S. Tulsiani. Track2act: Predicting point tracks from internet videos enables diverse zero-shot robot manipulation. *arXiv preprint arXiv:2405.01527*, 2024.

[60] C. Wang, L. Fan, J. Sun, R. Zhang, L. Fei-Fei, D. Xu, Y. Zhu, and A. Anandkumar. Mimicplay: Long-horizon imitation learning by watching human play. *arXiv preprint arXiv:2302.12422*, 2023.

[61] Y. Zhu, A. Lim, P. Stone, and Y. Zhu. Vision-based manipulation from single human video with open-world object graphs. *arXiv preprint arXiv:2405.20321*, 2024.

[62] H. Bharadhwaj, A. Gupta, S. Tulsiani, and V. Kumar. Zero-shot robot manipulation from passive human videos. *arXiv preprint arXiv:2302.02011*, 2023.

[63] J. Ye, J. Wang, B. Huang, Y. Qin, and X. Wang. Learning continuous grasping function with a dexterous hand from human demonstrations. *IEEE Robotics and Automation Letters*, 8(5): 2882–2889, 2023.

[64] Y. Qin, Y.-H. Wu, S. Liu, H. Jiang, R. Yang, Y. Fu, and X. Wang. Dexmv: Imitation learning for dexterous manipulation from human videos. In *European Conference on Computer Vision*, 2022.

[65] J. Yang, Z.-a. Cao, C. Deng, R. Antonova, S. Song, and J. Bohg. Equibot: Sim (3)-equivariant diffusion policy for generalizable and data efficient learning. *arXiv preprint arXiv:2407.01479*, 2024.

[66] J. Bruce, M. Dennis, A. Edwards, J. Parker-Holder, Y. Shi, E. Hughes, M. Lai, A. Mavalankar, R. Steigerwald, C. Apps, Y. Aytar, S. Bechtle, F. Behbahani, S. Chan, N. Heess, L. Gonzalez, S. Osindero, S. Ozair, S. Reed, J. Zhang, K. Zolna, J. Clune, N. de Freitas, S. Singh, and T. Rocktäschel. Genie: Generative interactive environments, 2024. URL https://arxiv.org/abs/2402.15391.

[67] Y. Chen, Y. Ge, W. Tang, Y. Li, Y. Ge, M. Ding, Y. Shan, and X. Liu. Moto: Latent motion token as the bridging language for learning robot manipulation from videos, 2025. URL https://arxiv.org/abs/2412.04445.

[68] D. Schmidt and M. Jiang. Learning to act without actions. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=rvUq3cxpDF.

[69] Z. Ren, Y. Wei, X. Guo, Y. Zhao, B. Kang, J. Feng, and X. Jin. Videoworld: Exploring knowledge learning from unlabeled videos, 2025. URL https://arxiv.org/abs/2501.09781.

[70] Q. Bu, Y. Yang, J. Cai, S. Gao, G. Ren, M. Yao, P. Luo, and H. Li. Univla: Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2505.06111*, 2025.

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

[72] B. Kang, Y. Yue, R. Lu, Z. Lin, Y. Zhao, K. Wang, G. Huang, and J. Feng. How far is video generation from world model: A physical law perspective. *arXiv preprint arXiv:2411.02385*, 2024.

[73] H. Bansal, Z. Lin, T. Xie, Z. Zong, M. Yarom, Y. Bitton, C. Jiang, Y. Sun, K.-W. Chang, and A. Grover. Videophy: Evaluating physical commonsense for video generation. *arXiv preprint arXiv:2406.03520*, 2024.

[74] S. Motamed, L. Culp, K. Swersky, P. Jaini, and R. Geirhos. Do generative video models learn physical principles from watching videos? *arXiv preprint arXiv:2501.09038*, 2025.

[75] H. Duan, H.-X. Yu, S. Chen, L. Fei-Fei, and J. Wu. Worldscore: A unified evaluation benchmark for world generation. *arXiv preprint arXiv:2504.00983*, 2025.

[76] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang, H. Zhong, Y. Zhu, M. Yang, Z. Li, J. Wan, P. Wang, W. Ding, Z. Fu, Y. Xu, J. Ye, X. Zhang, T. Xie, Z. Cheng, H. Zhang, Z. Yang, H. Xu, and J. Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[77] H. Bansal, Y. Bitton, I. Szpektor, K.-W. Chang, and A. Grover. Videocon: Robust video-language alignment via contrast captions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13927–13937, 2024.

[78] R. Cadene, S. Alibert, A. Soare, Q. Gallouedec, and T. Wolf. Lerobot: Making ai for robotics more accessible with end-to-end learning, 2024. URL https://github.com/huggingface/lerobot. Accessed: 2025-04-30.

A Related Work

Synthetic Data Generation in Robotics. Real-world robot data collection through human teleoperation requires large amounts of time and considerable human cost. As an alternative, collecting synthetic data in simulation can be more efficient and automated with minimal human effort [25, 26, 27, 28, 29, 20, 30, 31, 32, 33, 34]. However, using these trajectories can be challenging due to the following factors: (1) the sim-to-real gap, (2) difficulty in simulating objects such as liquid and articulated objects, and (3) being bounded by either Task and Motion Planning (TAMP) based systems or the interpolation of human teleoperation data. Another direction is to use neural generative models to augment existing sets of robot demonstrations [35, 36, 37, 38], using in-painting, image diffusion models, or even video2video models [39]. However, the diversity of the generated data is limited, especially in terms of robot motions, and the augmented data is only used to increase visual robustness to distribution shifts.

Video World Modeling for Robotics. Video generative models can be used to generate synthetic robot trajectories and extract executable actions during test-time via inverse-dynamics models (IDM) [15, 16], optical flow as dense correspondence [17], or trajectories as high-level plans [18, 19]. Another work generates human videos along with 3D tracking during test-time [40], or human videos for novel scenes and motions [41], and trains a policy with a point tracking objective. A concurrent work explores adapting text-to-video models for task generalization [42] by generating synthetic trajectories and extracting executable actions via an IDM or using it to extract rewards to guide a reinforcement learning policy. However, the scope of the work is bounded by simulation tasks. Some recent work aims to either train a robot policy initialized from a video generative model [43, 44] or perform policy training, inverse dynamics, and forward dynamics together, enabling co-training with both robot and video data [45, 46, 47, 48]. Our approach deliberately separates these components to fully make use of the state-of-the-art video generative models, which is currently not feasible to run in adjacent with a robot policy real time to ensure the strongest generalization capabilites.

Learning Robot Policies from Videos Videos provide abundant information for training robots, yet most do not come with labeled actions [49]. To enhance visual representations, prior work has used pretraining of vision encoders on egocentric videos of human activity [50], which has proven beneficial in downstream tasks [51, 52]. Several approaches extract various forms of information from human-centric videos, including human-object interactions [53], object affordances [54, 55, 56, 57], and visual trajectories [58, 59]. Other lines of research focus on translating human motions into robotic behaviors, employing hand pose estimators [60, 61, 57, 62, 63, 64] or motion capture systems [65]. Another line of work extracts *latent* actions to train downstream robot policies from visual deltas between the current and future frames [13, 66, 67, 68, 69, 4, 70, 71]. In this work, we use *synthetic* videos generated by a world model as the source instead of human videos, and explore using latent actions by co-training latent actions with real-world actions.

B DreamGenBench: A Video Generation Benchmark for Robotics

Motivated by recent work benchmarking the capabilities of video generative models as world models [72, 73, 74, 75], we introduce DreamGenBench, a systematic world modeling benchmark that aims to quantify the capacity of existing video generative models to adapt to a specific robot embodiment, internalizing the rigid body physics of the given robot, while generalizing to new objects, behaviors, and environments. We measure two key metrics: *instruction following* and *physics following*.

First, the *instruction following* metric is used to assess whether the generated video strictly adheres to given instructions to generate a video of the robot *completing* the specific task. The generated videos are fed into Qwen-VL-2.5 [76] with specific prompts to give a binary score (0 or 1) for quantifying the consistency between the video content and the task instructions, thereby ensuring

Table 2: `DreamGenBench` **Statistics and Results**. IF represents Instruction Following, and PA represents Physics Alignment. GPT represents the evaluation from GPT4o, Qwen represents the evaluation from Qwen2.5VL, and Hu represents the human evaluation. -zero represents zero-shot inference and -sft represents fine-tuned variants. Best is **bolded** and second best is underlined.

Dataset Statistics																
Dataset	RoboCasa				GR1											
Train (# trajs)	1200				100											
Eval (# frames)	48				Object: 50				Behavior: 47				Env: 30			
Results																
	IF			**PA**	**IF**			**PA**	**IF**			**PA**	**IF**			**PA**
	GPT	Qwen	Hu		GPT	Qwen	Hu		GPT	Qwen	Hu		GPT	Qwen	Hu	
Hunyuan-zero	1.0	0.0	-	0.0	0.0	0.0	-	0.0	0.0	2.1	-	2.1	0.0	0.0	-	0.0
CogVideoX-zero	0.0	0.0	-	0.0	0.0	0.0	-	0.0	0.0	0.0	-	0.0	0.0	0.0	-	0.0
WAN2.1-zero	0.0	0.0	-	0.0	0.0	2.0	-	2.0	0.0	2.1	-	2.1	0.0	6.7	-	6.7
Cosmos-zero	**4.2**	**22.9**	-	**22.9**	0.0	**32.0**	-	**32.0**	**6.4**	**31.9**	-	**31.9**	**3.5**	**24.1**	-	**24.1**
Hunyuan-sft	68.8	8.3	81.3	44.8	38.0	26.0	52.0	39.0	38.3	10.6	14.9	12.8	27.6	27.6	43.2	35.4
CogVideoX-sft	72.9	10.4	79.2	44.8	72.0	38.0	72.0	55.0	44.0	28.0	21.3	24.7	55.2	41.4	61.1	51.3
WAN2.1-sft	77.1	18.8	91.7	55.3	72.0	58.0	80.0	69.0	**72.3**	55.3	**74.5**	**64.9**	48.3	**65.5**	**67.4**	**66.5**
Cosmos-sft	**79.2**	**29.2**	**93.8**	**61.5**	**90.0**	**62.0**	**84.0**	**73.0**	59.6	**61.7**	68.1	**64.9**	**69.0**	**65.5**	53.3	59.4

that the actions and scenes in the video match the intended objectives. We provide the exact prompt we use for the evaluation in Appendix J.1. We also provide human evaluations in addition to the model-based evaluation, showing an average Pearson correlation of $> 90\%$, ensuring that the model-based evaluation metric is aligned to human judgment in Appendix J.3.

Next, we quantify the *physics alignment* to evaluate the physical plausibility of the generated videos, so that the videos are actually useful for downstream robot learning. For this purpose, we first employ the VideoCon-Physics [73], a VLM specifically trained to give scores for physics adherence of generated videos. Specifically, we get a 0 to 1 score from VideoCon-Physics. In practice, we find the model has not been trained on multiview videos (RoboCasa) and diverse robot environments, so we use a general VLM: Qwen-VL-2.5 to also score each video based on our instruction and then calculate the average score of these two scores for each video generation model on each dataset. We provide more details of VideoCon-Physics in Appendix J.2.

Using these two metrics, we benchmark 4 different video world models, Hunyuan [10], CogVideoX [8], WAN 2.1 [9], and Cosmos [7], on 2 different training and evaluation setups, one in simulation on the Franka Emika robot and one in real on the Fourier GR1 Humanoid. We also quantify the *zero-shot* capability of the models, evaluated without adapting to the specific embodiment. Results and dataset statistics are shown in Table 2. In addition to these two metrics, we also replay the IDM actions in simulation to empirically see the quality of the IDM actions, where we have access to the digital twin of the Fourier GR1. See Section J.4 for more details.

`DreamGenBench` **shows positive correlation to downstream robot policy performance.** To measure whether DREAMGENBENCH could be a proxy evaluation for the performance of the downstream robot policy, we measure the performance of the RoboCasa benchmark by *only* training on neural trajectories generated from the different video world models. A positive correlation between DREAMGENBENCH and RoboCasa would indicate that building a better world model that can follow language instruction and model world physics leads to better performance on the downstream robot manipulation tasks. We compare all the models in Table 2 with 7K neural trajectories per model. For DREAMGENBENCH score, we use the average of IF (GPT) and PA from Table 2. The results are illustrated in Figure 6. As shown, the correlation between DREAMGENBENCH and RoboCasa shows a positive correlation, indicating that building a stronger video world model could lead to larger performance enhancement.

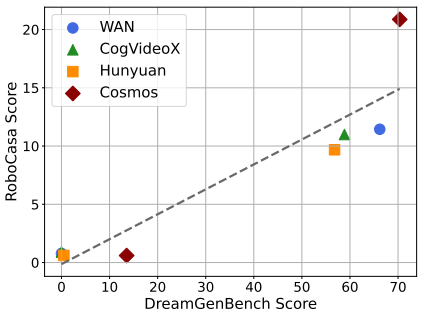


Figure 6: Performance correlation between DREAMGENBENCH and RoboCasa.

Table 3: LAPA Training Dataset Statistics

Dataset	Length (Frames)	Duration (hr)	FPS	Category
GR-1 Teleop Pre-Training	6.4M	88.4	20	Real robot
DexMG	4.4M	61.64	20	Simulation
DROID (OXE)	23.1M	428.3	15	Real robot
RT-1 (OXE)	3.7M	338.4	3	Real robot
Language Table (OXE)	7.0M	195.7	10	Real robot
Bridge-v2 (OXE)	2.0M	111.1	5	Real robot
RoboCasa	19.3M	268.0	20	Simulation
Agibot-Alpha	213.8M	1,979.4	30	Real robot
Sth-v2	4.0M	105.7	30	Human
Ego4D	154.4M	2,144.7	20	Human
Total	438.1M	5,721.3	–	–

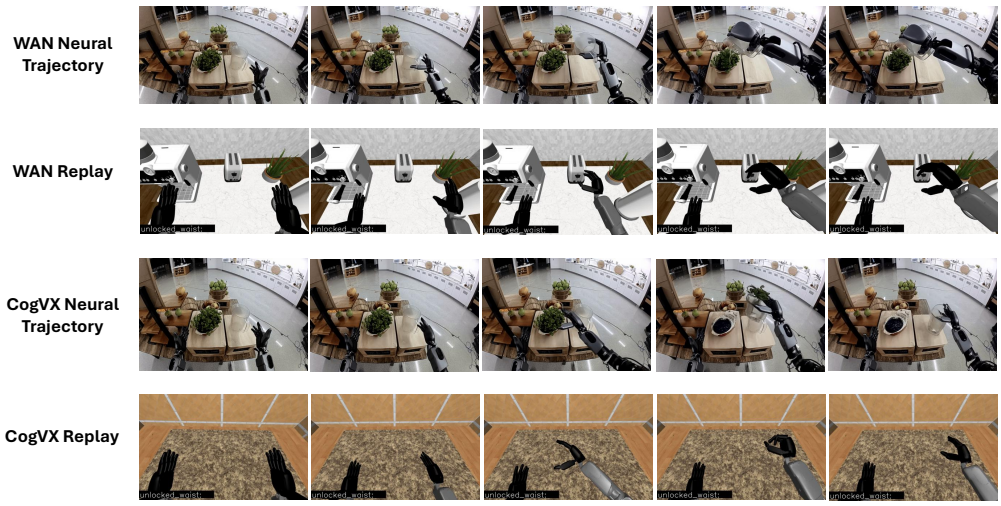


Figure 7: **Neural Trajectories and Replay Videos for WAN and CogVideoX model**. The language instruction is to "Use the right hand to pick up the plastic pitcher and pour water onto the green plant."

C Extracting Pseudo Actions from Synthetic Videos

For IDM, if we have a digital cousin of the real robot embodiment in simulation, we can also replay the pseudo actions in simulation and do intermediate checking whether the neural trajectory quality is not good enough or the bottleneck is on the IDM model (as shown in Figure 7). Empirically, we observe that most of the bottleneck is from the quality of the neural trajectories, which indicates that future video models that can generate videos with better language following and physics alignment could lead to a significant boost on the downstream task. For LAPA training, we trained a collection of datasets that include real robots, simulation, and human videos. The detailed statistics are shown in Table 3. We use a codebook size of 8 and a sequence length of 16 for vector quantization. We train 100K steps with a batch size of 1024.

D Environment for Teleoperation and Evaluation

We provide some sample images of the environment where we collected all of our GR1 humanoid teleoperation data in Figure 8 and all of the 10 environments where we conducted environment generalization results in Figure 9, respectively.

Figure 8: **Seen Environment.** Sample images for the environment where we collected the pick-and-place GR1 data.

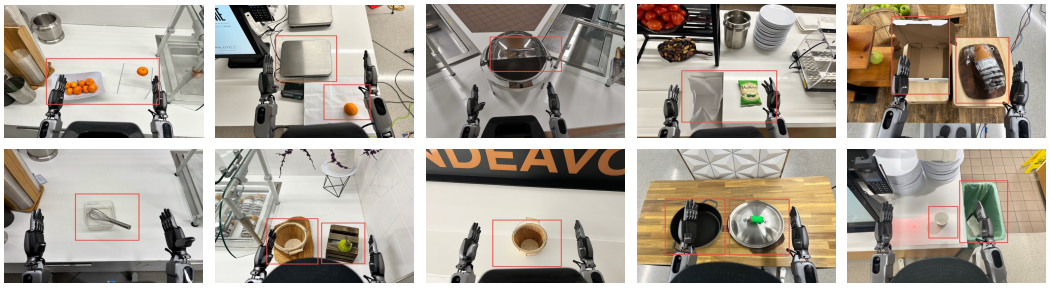


Figure 9: **Unseen Environment.** All of the 10 environments for our environment generalization experiments.

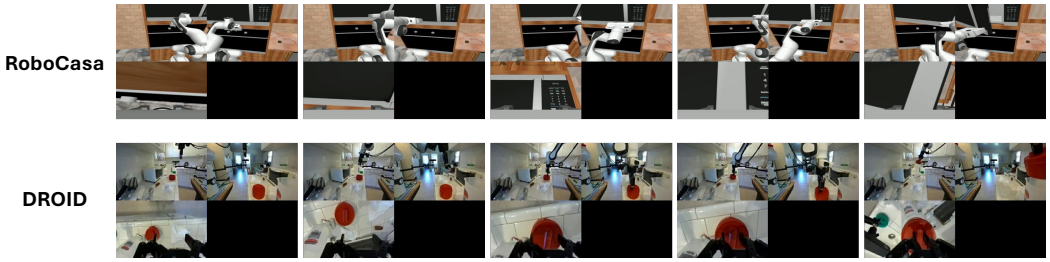


Figure 10: **Multiview Examples.** The top row shows a trajectory from RoboCasa and the bottom shows a trajectory from the DRIOD dataset.

E Examples of Multiview Robot Data Processing

We provide examples of how we process multiview training data, RoboCasa, and DROID, for video world model fine-tuning in Figure 10. Specifically, we arrange the viewpoints into a 2×2 grid: the left camera view is placed at the top-left, the right camera view at the top-right, and the wrist camera view at the bottom-left. A black image is inserted in the bottom-right to complete the grid.

F Video World Model Training Hyperparameters

For all of the WAN 2.1 fine-tuning experiments, we used a learning rate of 1e-4, LoRA rank 4, and LoRA alpha 4. For RoboCasa finetuning, we trained the model for 100 epochs with a batch size of 32. For GR1 finetuning, we trained the model for 75 epochs with a batch size of 64. For DROID fine-tuning, we trained the model for 5 epochs with a batch size of 64. For both of the two tasks in SO-100 finetuning, we trained the model for 200 epochs with batch size 8.

G Detailed Experimental Results on RoboCasa

Table 4 shows all of the experimental results on RoboCasa. As seen in the chart, ONLY neural trajectories also achieves 20.55% average success rate across the 24 tasks, showcasing how close neural trajectories are to ground truth trajectories.

Table 4: **Experimental Results on RoboCasa**. *NT* stands for 240k neural trajectories.

Task		GR00T N1						
		30 traj.	100 traj.	300 traj.	30 traj. + NT	100 traj. + NT	300 traj. + NT	ONLY NT
Pick and Place	PnPCabToCounter	0.93	3.92	19.61	5.77	13.46	25.00	1.96
	PnPCounterToCab	1.85	6.86	36.27	3.85	19.23	50.96	16.67
	PnPCounterToMicrowave	0.00	0.00	12.75	0.00	9.62	19.23	0.00
	PnPCounterToSink	0.00	0.98	9.80	0.00	12.50	33.65	1.96
	PnPCounterToStove	0.00	0.00	23.53	0.00	12.50	42.31	8.82
	PnPMicrowaveToCounter	0.00	0.00	15.69	0.00	14.42	28.85	0.00
	PnPSinkToCounter	0.00	5.88	33.33	3.85	28.85	60.58	0.98
	PnPStoveToCounter	0.00	0.00	29.41	0.96	9.62	58.65	5.88
Open/Close Doors	CloseDoubleDoor	0.00	43.14	74.51	9.62	52.88	82.69	2.94
	OpenDoubleDoor	0.00	12.75	14.71	0.00	8.65	28.85	0.00
	CloseSingleDoor	49.07	67.65	83.33	51.92	80.77	94.23	52.94
	OpenSingleDoor	20.37	54.90	58.82	44.23	55.77	47.12	15.69
Open/Close Drawers	CloseDrawer	76.85	96.08	99.02	88.46	98.08	98.08	82.35
	OpenDrawer	9.26	42.16	79.41	33.65	68.27	74.04	33.33
Twisting Knobs	TurnOnStove	14.81	25.49	55.88	21.15	27.88	51.92	17.65
	TurnOffStove	4.63	15.69	26.47	7.69	13.46	25.96	6.86
Turning Levers	TurnOffSinkFaucet	49.07	67.65	72.55	51.92	69.23	95.19	59.80
	TurnSinkSpout	24.07	42.16	52.94	37.50	45.19	59.62	28.43
	TurnOnSinkFaucet	33.33	59.80	62.75	48.08	67.31	72.12	25.49
Pressing Buttons	TurnOffMicrowave	47.22	57.84	70.59	55.77	75.96	76.92	29.41
	TurnOnMicrowave	55.56	73.53	78.43	49.04	52.88	72.12	48.04
	CoffeePressButton	27.78	56.86	85.29	34.62	63.46	83.65	48.04
Insertion	CoffeeServeMug	3.70	34.31	72.55	11.54	48.08	74.04	2.94
	CoffeeSetupMug	0.00	1.96	22.55	0.00	10.58	26.92	2.94
Average		17.44	32.07	49.59	23.32	39.94	57.61	20.55

H Fine-tuning Data for Video World Models and IDMs

In this section, we provide some detailed information about the protocol we followed to train the video world models and the IDM for each experimental setup.

Four dexterous tasks on Real-world GR1. To train our video world model, we follow the same protocol outlined in Section 2, and train on 2,884 GR1 trajectories of pick-and-place collected in a single lab environment. Since these four tasks differ significantly from the target task, we further fine-tune the model on the *low data* trajectories for each task. For each task, we collect 100 trajectories, but only utilize 10 trajectories for Hammering, Wiping, Stacking, and 25 trajectories for Folding to test data efficiency. We utilize the IDM trained only on the 2,884 GR1 pick-and-place data for all experimentsl.

3 tasks on Franka. Following protocol in Section 2, we train our video world model on 49,895 DROID data examples, and further fine-tune the model on the *low data* trajectories for each task. We found that utilizing the model trained only from the DROID dataset results in dreams that show generalization to the new environment, but produced trajectories that made mistakes on fine-graed details (e.g. grasping). We use 11, 10, and 8 trajectories for putting milk in bowl, cube stacking, and scooping M&Ms, respectively. Similarly to GR1, we use the IDM trained on 49,895 trajectories and do not do any specific post-training.

2 tasks on SO-100. The original SO-100 videos concatenate multiple trajectories with identical actions into a single video. For fine-tuning, we manually trim and split these into separate videos, each corresponding to an individual trajectory. Specifically, we sample 10 and 13 videos for the two tasks, which yield 68 and 44 trajectories, respectively, after trimming.

I Full Real-world Experimental Results

Table 5 shows the entire experimental results, including the model performance when trained on the "High Data" variant of each experimental setup.

Table 5: **Success Rate (%) of Real-world Data Augmentation Experiments.**.

Model	GR1					Franka			SO-100	
	Hammering	Wiping	Folding	Stacking	Average	Pick&Place	Cube Stacking	Tool Usage	Pick&Place	Tic-Tac-Toe
DP	35.0	23.3	6.6	25.0	22.0	20.0	0.0	10.0	-	-
π_0	-	-	-	-	-	30.0	10.0	20.0	-	-
GR00T N1	60.0	36.6	27.0	25.0	37.0	40.0	10.0	20.0	17.0	25.0
DP + Neural Traj.	15.0	33.3	26.4	35.0	27.0	30.0	20.0	10.0	-	-
π_0 + Neural Traj.	-	-	-	-	-	40.0	20.0	20.0	-	-
GR00T N1 + Neural Traj.	65.0	49.0	37.0	35.0	46.0	60.0	20.0	30.0	26.0	65.0
DP (High Data)	60.0	36.0	43.3	75.0	54.0	30.0	20.0	20.0	-	-
π_0 (High Data)	-	-	-	-	-	50.0	40.0	40.0	-	-
GR00T N1 (High Data)	75.0	50.0	66.6	85.0	69.0	80.0	50.0	40.0	36.0	40.0

J Video World Model Evaluation

J.1 Success Rate

Specifically, we use the following prompts to Qwen2.5-VL-7B-Instruct [76] to judge whether a video follows the instruction to complete a specific task or not.

Prompt Template for Success Rate

User: {Video: `<vid_path>`}{Text: "The video shows a robot arm completing a specific task. Please evaluate: if the video follows the instruction to finish the task '{prompt}', give a positive score. Reply only '0' for No or '1' for Yes."}
Assistant: 0 or 1

Prompt Template for Success Rate (Zeroshot)

User: {Video: `<vid_path>`}{Text: "You are evaluating if a robot arm correctly follows this instruction: '{prompt}'
CRITICAL EVALUATION PROCESS: 1. FIRST CHECK: If you see HUMAN HANDS instead of robot arms, IMMEDIATELY ANSWER 0. 2. SECOND CHECK: Only if robot arms confirmed, verify if the instruction is followed exactly. 3. For videos with multiview clip (4 grids), verify if the instruction is followed exactly in each view. Only if all the view is following instruction, answer 1, otherwise, answer 0.
Remember: human hands = automatic failure (0). Be extremely strict in your judgment. For videos with multiview clip (4 grids), check if the human arm is present in any view, if so, make sure to answer 0.
Reply ONLY with a single digit: 0 for failure or 1 for success."}
Assistant: 0 or 1

J.2 Physics Alignment

While human evaluation provides accurate benchmarking, it is time-consuming and costly at scale. To enable model developers with limited resources to use our benchmark, we use **VideoCon-Physics**, an open video-text language model with 7B parameters trained on real videos for physics alignment evaluation [73]. Specifically, they finetune VideoCon [77] using human annotations collected for physics alignment on generated videos. We prompt it to generate binary responses conditioned on multimodal templates. They evaluate this auto-rater by computing ROC-AUC between human judgments and model predictions on videos generated with testing prompts, and show that they have a strong correlation with human evaluation results. In addition to it, we use Qwen2.5-VL-7B-Instruct [76] to judge whether a video follow physics or not with the following prompt:

Prompt Template for Physics Alignment

User: {Video: <vid_path>}{"The video shows a robot arm completing a specific task. Does the video show good physics dynamics that is aligned with the physical world? Answer 0 for No or 1 for Yes. Reply only 0 or 1."}
Assistant: 0 or 1

We finally compute the average of two scores together for each video.

J.3 Human Evaluation

To verify the reliability of our automatic benchmark on success rate, we compare it with human evaluation results and calculate the AUC-ROC between them. In detail, we perform human evaluations of all of the instances from the 3 fine-tuned video world models from Table 2, to show that the model-based metrics indeed do correlate with human-based judgement of success rate (SR) and physics alignment (PA). For SR, similar to the model-based metric, humans give a binary signal, 0 or 1, whether the trajectory has successfully completed the task specified by the language. For PA, instead of giving a fine-grained score, humans rank the model's output, given the same initial frame, and see the ranking corresponds to the ranking by the scores of the model.

Dataset	Metric	Hunyuan-sft	CogVideoX-sft	WAN2.1-sft	Cosmos-sft	Pearson r
RoboCasa	IF	68.8	72.9	77.1	79.2	0.94
	IF-human	81.3	79.2	91.7	93.8	
GR1-Object	IF	38.0	72.0	72.0	90.0	0.93
	IF-human	52.0	72.0	80.0	84.0	
GR1-Behavior	IF	38.3	44.0	72.3	59.6	0.96
	IF-human	14.9	21.3	74.5	68.1	
GR1-Env	IF	27.6	55.2	48.3	69.0	1.00
	IF-human	20.0	30.0	43.3	53.3	

Table 6: Pearson correlation coefficients between automatic IF (GPT-4o) and human IF-human scores across different datasets and model variants.

Dataset	Metric	Hunyuan-sft	CogVideoX-sft	WAN2.1-sft	Cosmos-sft	Pearson r
RoboCasa	IF	8.3	10.4	18.8	29.2	0.92
	IF-human	81.3	79.2	91.7	93.8	
GR1-Object	IF	26.0	38.0	58.0	62.0	0.95
	IF-human	52.0	72.0	80.0	84.0	
GR1-Behavior	IF	10.6	28.0	55.3	61.7	0.97
	IF-human	14.9	21.3	70.2	68.1	
GR1-Env	IF	27.6	41.4	65.5	65.5	0.96
	IF-human	20.0	30.0	43.3	53.3	

Table 7: Pearson correlation coefficients between automatic IF (Qwen2.5-VL) and human IF-human scores for each dataset.

Table 6 and Table 7 present the Pearson correlation coefficients between our automatic evaluation metric (IF) and the corresponding human-annotated scores (IF-human) for three model variants on each dataset. The correlations of IF evaled by GPT-4o are uniformly high—0.94 for RoboCasa, 0.93 for GR1-Object, 0.96 for GR1-Behavior, and essentially 1.00 for GR1-Env—indicating a near-perfect linear relationship across all cases. These results confirm that the IF metric faithfully captures human judgments and can serve as a reliable proxy for resource-intensive manual evaluation.

J.4 Intermediary Step for Checking Downstream Performance

The most straightforward way to truly quantify the capabilities of the video world models is to use them to generate neural trajectories and use the generated trajectories for downstream visuomotor policy training. In fact, we generate 7k neural trajectories for each of the video world models (zero-shot and fine-tuned) from Table 2 and show that benchmark numbers directly correlate to downstream robot policy performances. However, this is very resource-intensive, since verifying a new video world model beyond benchmark numbers requires generating 7k new videos. As an intermediary step, we utilize a *cheaper* way of quantifying the quality of the dreams. After extracting the IDM actions from the generated videos (see Section 2.3), we replay the IDM actions in simulation, where we have access to the digital twin of the Fourier GR1. Some examples of replayed IDM actions in simulation are shown in Appendix C.

K Robot Experiment Evaluation

K.1 GR1 Humanoid Experiments

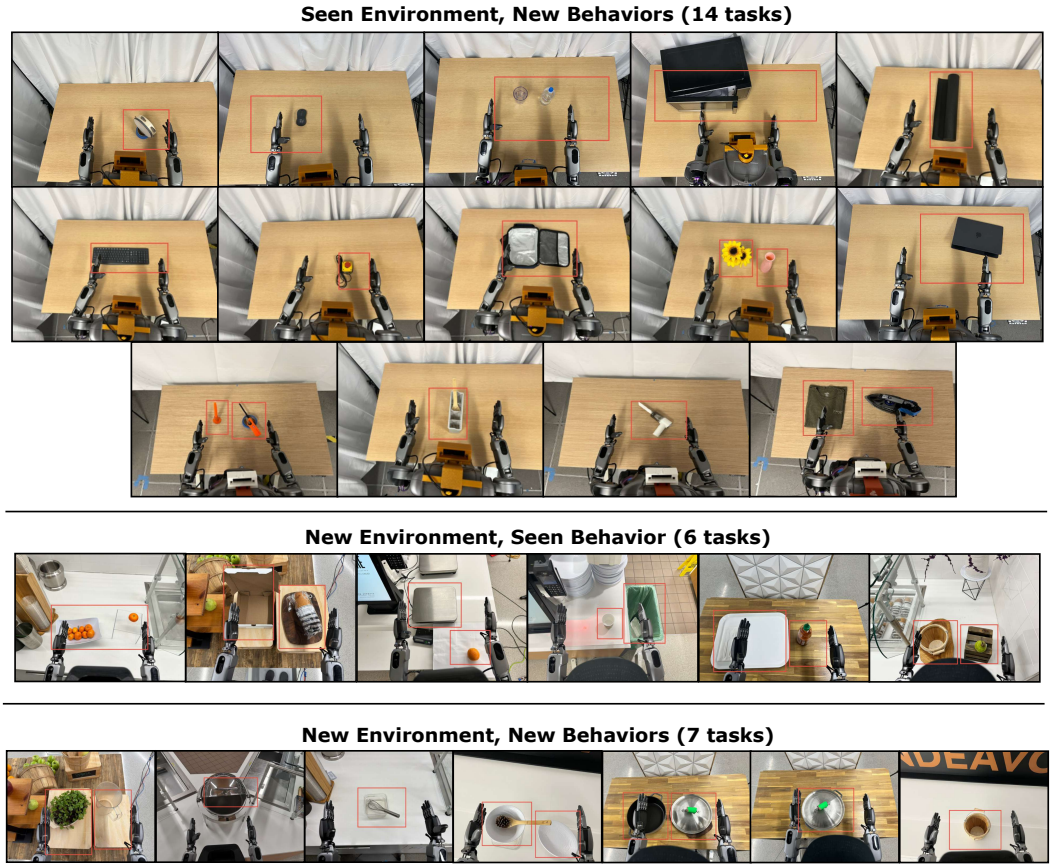


Figure 11: **Evaluations for all Real-world GR1 Experiments.** The rectangular box represents the region where we randomize the target object during evaluation.

Data Augmentation We have 4 tasks for the data augmentation experiments using the GR1 Humanoid: Hammering, Wiping, Folding, and Stacking. For each task, we collect 100 trajectories, while randomizing the target object locations in the rectangular box as shown in Figure 5.

For Hammering, we give 0.5 for picking up the hammer, and 1.0 for actually hitting the nail. For Wiping, 0.33 for grabbing the rag, 0.66 for taking the rag to the stain, and 1.0 for actually wiping

the stain. For Folding, we give 0.33 for folding the first fold, but imperfectly, 0.66 for completing the first fold, and 1.0 for completing the second fold. Lastly, for Stacking, we give 0.5 for stacking the left bowl, and 1.0 for stacking the right bowl. We perform 10 eval rollouts per checkpoint.

Behavior and Environment Generalization Table 8 shows the criterion we use to measure the performance on behavior and environment generalization. We performed 10 rollouts per checkpoint while randomizing the initial location of the target object across all trials to ensure fair, direct comparisons between models. The region of target object randomization is shown in Figure 11.

Table 8: **Task Evaluation Criteria for GR1 Generalization Experiments**

Seen Environments, Novel Behaviors

Task	Criteria
Open Microwave	0.33 grasp handle 0.66 do closing motion 1.0 close microwave
Open Macbook	0.5 opening motion 1.0 open laptop
Close Lunchbox	0.5 contact lid 1.0 close lunchbox
Hit Tambourine	0.5 grab tambourine 1.0 hit with left hand
Hit Keyboard	0.5 going to keyboard 1.0 pressing
Grab Button	0.5 go to button 1.0 grab button
Pour Water	0.5 picking up 1.0 pouring
Water Flowers	0.5 grasp pink bottle 1.0 pour
Light Candle	0.5 grasp lighter 1.0 approach candle
Use Vacuum	0.5 pick up vacuum 1.0 do sweeping motion
Iron Shirt	0.5 grasp iron 1.0 press shirt
Take Spoon Out	0.33 grasp spoon 0.66 pick up spoon 1.0 place spoon
Unroll Mat	0.5 go to mat 1.0 unroll
Move Mouse	0.5 grab the mouse 1.0 move it around

Novel Environments

Seen Behaviors	
Task	**Criteria**
Pick up Tangerine	0.5 pick up 1.0 place in bowl
Box Sandwich	0.5 grab the sandwich 1.0 place in box
Weigh the Orange	0.5 pick up 1.0 place on scale
Put Cup in Trash	0.5 grab cup 1.0 throw it away
Put Pear in Basket	0.5 grab pear 1.0 put in bucket
Put Sauce on Tray	0.5 grab bottle 1.0 place bottle on tray
Novel Behaviors	
Task	**Criteria**
Water Flowers	0.5 pick up pitcher 1.0 water the plants
Lift Basket	0.5 grab handle 1.0 lift bucket
Swirl Around Spoon	0.5 grab spoon 1.0 scoop to plate
Use Whisk	0.5 grab whisk 1.0 mix
Close Soup Container	0.5 use handle 1.0 close
Uncover Pot	0.5 grab cover 1.0 uncover pot
Cover Pot	0.5 grab cover 1.0 cover pot

K.2 DROID (Franka) Experiments

We carry out our second real-world study on the Franka Emika Panda arm, collecting 100 teleoperation data for three manipulation tasks, pick-and-place, cube stacking, and tool use (Figure 5.). We also have a *low*-data regime, where we only train on 10 trajectories, except for the folding task,

where we train on 25 trajectories. Following our proposed pipeline, we train our video world model and the IDM model on the DROID dataset [22],

To ensure rigorous evaluation, we executed 10 rollouts per checkpoint for each model and enforced identical initial state configurations across models, enabling fair, head-to-head comparisons. Within each batch of rollouts, we further randomized object poses to probe policy robustness. Results show that conditioning on neural trajectories consistently boosts the performance of Diffusion Policy, π_0, and GR00T N1 across all tasks.

K.3 SO-100 Experiments

We also present fine-tuning experiments with real and neural trajectories on a LeRobot SO-100 [78], serving as a new embodiment with a foundation robot policy (GR00T N1 VLA). The first task, "Picking 3 Strawberries," consists of 10 real-world trajectories and 30 neural trajectories. The second task is "Tic-Tac-Toe", which requires the correct language prompt to execute the task, and includes 13 real-world trajectories and 40 neural trajectories.

For the "Picking 3 Strawberries" task, the evaluation criteria involve 10 trials. The goal of each trial is to pick up all three strawberries from various locations on the table and place them on the plate. Each trial lasts 1 minute, with each successful pick and place contributing 33% to the score for that trial. To ensure randomness, strawberries are placed on the left, center, and right sides of the table. In the "Tic-Tac-Toe" task, we evaluated the policy by prompting it with 5 tasks, each corresponding to placing an "X" in different boxes on the grid. With a total of 10 trials, the grid is randomized with varying "X" and "O" placements across the trials, each lasting 1 minute. Each successful pick and place corresponds to 0.5 points.

We observed that with co-training using neural trajectories, the policy overfits less to the proprioceptive states and conditions more effectively to the current visual state of the environment. Additionally, we noticed that the policy augmented with neural trajectories is less likely to get stuck at the initial home position, which is a common failure case of our baseline policy. Detailed results are shown in Figure 5.

L Examples of Generated Neural Trajectories

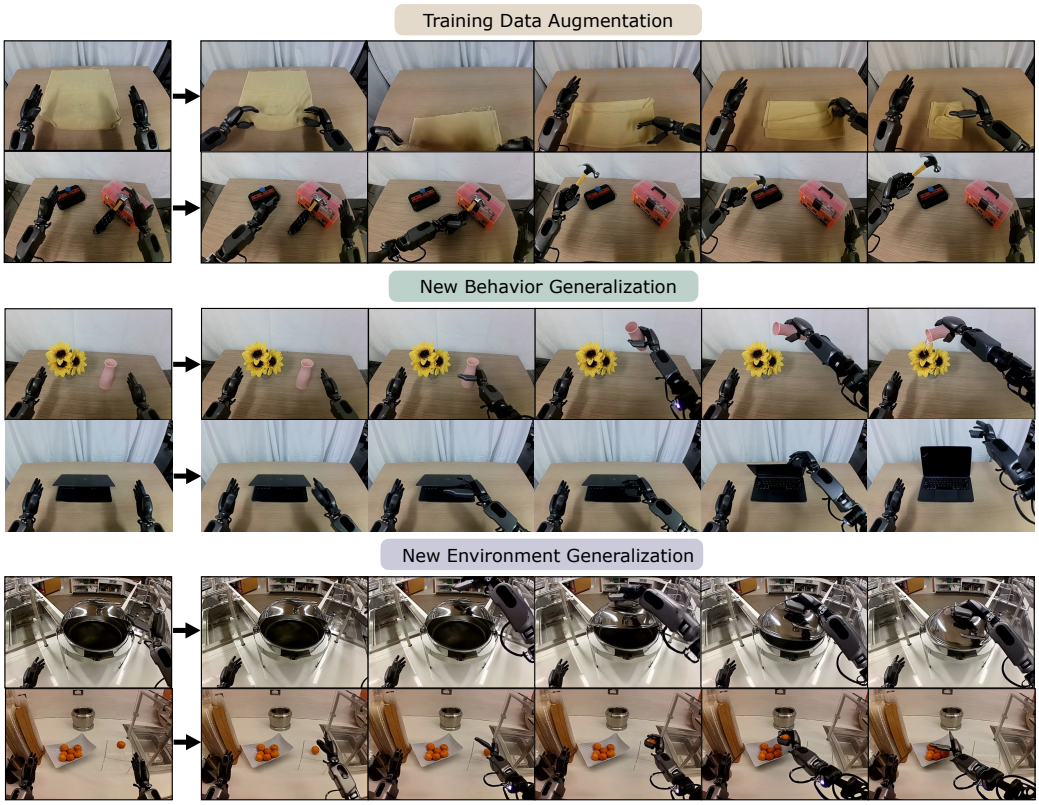


Figure 12: **Examples of Neural Trajectories.**