

Figure 5: **Extracting Pseudo Actions.** (a) shows the architecture of our IDM model and (b) shows the architecture of our latent action model.

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	—	—

A Extracting Pseudo Actions from Synthetic Videos

Figure 5 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). 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 6). 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.

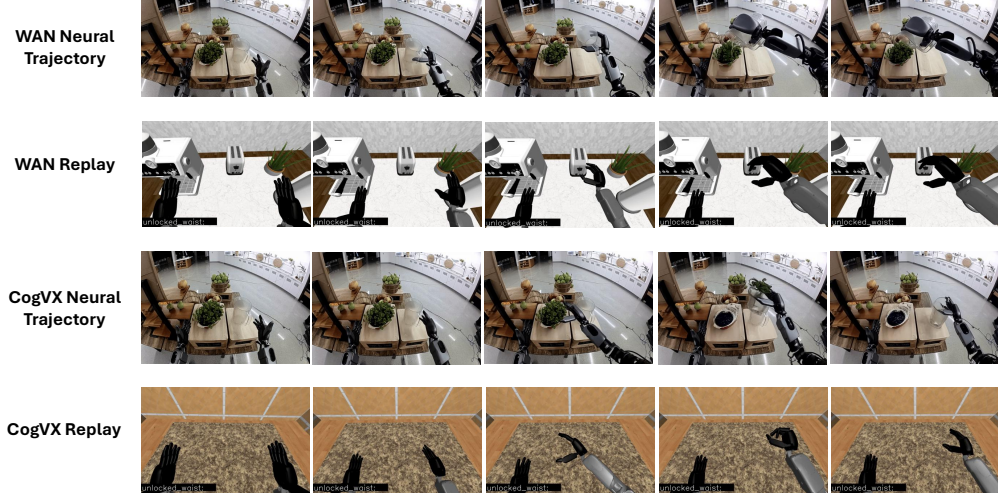


Figure 6: **Neural Trajectory videos and replay videos for WAN and CogVideoX model.** The language instruction is to “Use the right hand to pick up the plastic pitcher and pour water onto the green plant.”



Figure 7: Sample images for the environment where we collected GR1 data.

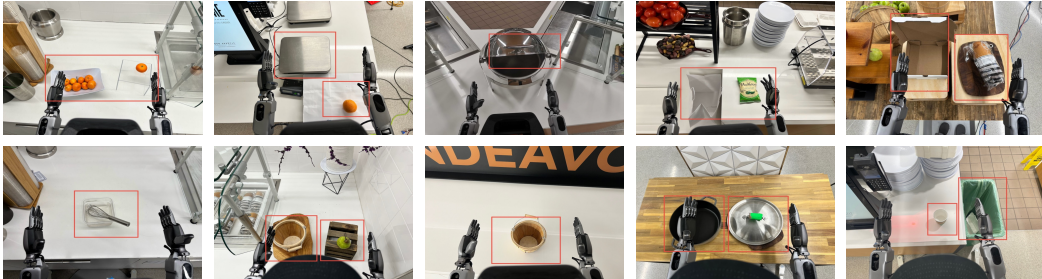


Figure 8: All of the 10 environments for our environment generalization experiments.

B 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 7 and all of the 10 environments where we conducted environment generalization results in Figure 8, respectively.

C 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 9. 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.

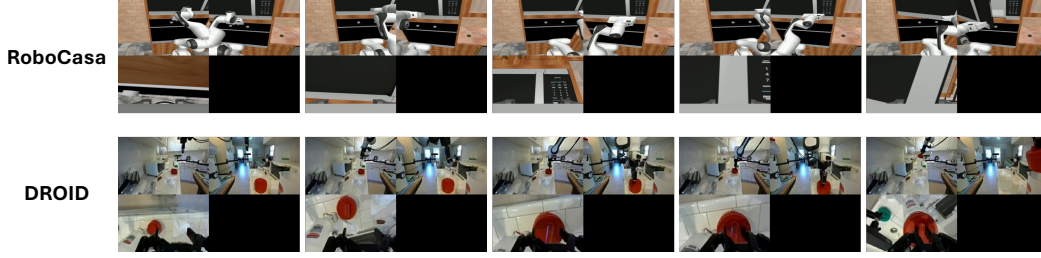


Figure 9: Examples of multiview robot videos for video world model finetuning.

D 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 finetuning, 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.

E 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 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	PnP CabToCounter	0.93	3.92	19.61	5.77	13.46	25.00	1.96
	PnP CounterToCab	1.85	6.86	36.27	3.85	19.23	50.96	16.67
	PnP CounterToMicrowave	0.00	0.00	12.75	0.00	9.62	19.23	0.00
	PnP CounterToSink	0.00	0.98	9.80	0.00	12.50	33.65	1.96
	PnP CounterToStove	0.00	0.00	23.53	0.00	12.50	42.31	8.82
	PnP MicrowaveToCounter	0.00	0.00	15.69	0.00	14.42	28.85	0.00
	PnP SinkToCounter	0.00	5.88	33.33	3.85	28.85	60.58	0.98
	PnP StoveToCounter	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

F 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 experiments.

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-grained 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.

G Full Real-world Experimental Results

Table 5: Full real-world experimental results including “High Data” setting.

Model	GR1					Franka			SO-100	
	Hammering	Wiping	Folding	Stacking	Average	Pick&Place	Cube Stacking	Tool Usage	Pick&Place	Tic-Tac-Toe
DP	35.00	23.30	6.60	25.00	22.00	20.00	0.00	10.00	-	-
Pi-zero	-	-	-	-	-	30.00	10.00	20.00	-	-
GR00T N1	60.00	36.60	27.00	25.00	37.00	40.00	10.00	20.00	17.00	25.00
DP + neural	15.00	33.30	26.40	35.00	27.00	30.00	20.00	10.00	-	-
Pi-zero + neural	-	-	-	-	-	40.00	20.00	20.00	-	-
GR00T N1 + neural	65.00	49.00	37.00	35.00	46.00	60.00	20.00	30.00	26.00	65.00
DP (High Data)	60.00	36.00	43.30	75.00	54.00	30.00	20.00	20.00	-	-
Pi-zero (High Data)	-	-	-	-	-	50.00	40.00	40.00	-	-
GR00T N1 (High Data)	75.00	50.00	66.60	85.00	69.00	80.00	50.00	40.00	36.00	40.00

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

H Video World Model Evaluation

H.1 Success Rate

Specifically, we use the following prompts to Qwen2.5-VL-7B-Instruct [28] 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. Does the video follow the instruction: '{prompt}'? Answer 0 for No or 1 for Yes. Reply only 0 or 1."}
Assistant: 0

H.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 [25]. Specifically, they finetune VideoCon [73] 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 [28] 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

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

H.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	Pearson r
RoboCasa	IF	8.3	10.4	18.8	0.94
	IF-human	81.3	79.2	91.7	
GR1-Object	IF	26	38	58	0.93
	IF-human	52	72	80	
GR1-Behavior	IF	10.6	28	55.3	0.96
	IF-human	14.9	21.3	70.2	
GR1-Env	IF	27.6	41.4	65.5	1.00
	IF-human	20	30	43.3	

Table 6: Pearson correlation coefficients between automatic IF and human IF-human scores across different datasets and model variants.

Table 6 presents 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 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.

H.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 24k 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 24k 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 A.

I Robot Experiment Evaluation

I.1 GR1 Humanoid Experiments

Training Data Augmentation We performed 10 rollouts per checkpoint while maintaining identical initial state configurations across all trials to ensure fair, direct comparisons between models. Neural trajectories are generated for four dexterous tasks on the GR1 Humanoid robot. The red rectangular box shows the range of object randomization during training and evaluation. *Low* denotes training with 10 real-world trajectories for Hammering, Wiping, and Stacking, and 25 real-world trajectories for Folding. and *High* denotes training with 100 real-world trajectories. *Low + Neural Traj.* denotes co-training with 10/25 real-world trajectories and 300 neural trajectories. As shown in Figure 3, neural trajectories consistently improve the success rate for both Diffusion Policy and GR00T N1.

Behavior and Environment Generalization Table 7 shows the criterion we use to measure the performance on behavior and environment generalization.

Table 7: Task evaluation criteria for various behaviors across different environments

Seen Environments, Novel Behaviors		Novel Environments, Seen Behaviors	
Task	Criteria	Task	Criteria
Open Microwave	0.33 grasp handle 0.33 do closing motion 1 close microwave	Pick up Tangerine	0.5 pick up 0.5 place in bowl
Open Macbook	0.5 opening motion 1 open laptop	Box Sandwich	0.5 grab the sandwich 0.5 place in box
Close Lunchbox	0.5 contact lid 0.5 close lunchbox	Weigh the Orange	0.5 pick up 0.5 place on scale
Hit Tambourine	0.5 grab tambourine 0.5 hit with left hand	Put Cup in Trash	0.5 grab cup 0.5 throw it away
Hit Keyboard	0.5 going to keyboard 0.5 pressing	Put Pear in Basket	0.5 grab pear 0.5 put in bucket
Grab Button	0.5 go to button 0.5 grab button	Put Sauce on Tray	0.5 grab bottle 0.5 place bottle on tray
Pour Water	0.5 picking up 0.5 pouring	Novel Environments, Novel Behaviors	
Water Flowers	0.5 grasp pink bottle 0.5 pour	Task	Criteria
Light Candle	0.5 grasp lighter 0.5 approach candle	Water Flowers	0.5 pick up pitcher 0.5 water the plants
Use Vacuum	0.5 pick up vacuum 0.5 do sweeping motion	Lift Basket	0.5 grab handle 0.5 lift bucket
Iron Shirt	0.5 grasp iron 0.5 press shirt	Swirl Around Spoon	0.5 grab spoon 0.5 scoop to plate
Take Spoon Out	0.33 grasp spoon 0.66 pick up spoon 1.0 place spoon	Use Whisk	0.5 grab whisk 0.5 mix
Unroll Mat	0.5 go to mat 0.5 unroll	Close Soup Container	0.5 use handle 0.5 close
		Uncover Pot	0.5 grab cover 0.5 uncover pot
		Cover Pot	0.5 grab cover 0.5 cover pot

I.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 3.). 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, Pi-Zero, and GR00T N1 across all tasks.

I.3 SO-100 Experiments

We also present fine-tuning experiments with real and neural trajectories on a LeRobot SO-100 [74], 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 3.

J Examples of Generated Neural Trajectories

Figure 10 shows some examples of generated neural trajectories.



Figure 10: Examples of Neural Trajectories.