World Action Models are Zero-shot Policies

DreamZero: a 14B video-diffusion model that predicts future video and robot actions together

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, et al. · NVIDIA
Presenter: Haoyu Kang

VLAs know what to do, but not how to do it

Vision-Language-Action models (VLAs) start from a vision-language model. They get semantic knowledge from static image-text data, but no knowledge of physical dynamics.

WORKS

Semantic generalization

“Move coke can to Taylor Swift”: web knowledge finds the target, and the *move* skill is in the robot data.

FAILS

New motions and skills

“Untie the shoelace”: if the skill is not in the robot data, the VLA cannot do it.

CONSEQUENCE

Repetitive data collection

VLAs need many repeated demonstrations for each task and each environment.

The idea: a World Action Model (WAM)



Start from a video diffusion model pretrained on web-scale video. Train it to predict **future video and actions together**.

1. Diverse data. The model learns from non-repetitive robot data.

2. Open-world generalization. It does unseen tasks in unseen environments.

3. Cross-embodiment. It learns from video-only data of humans or other robots.

4. Few-shot adaptation. It moves to a new robot with 30 minutes of data.

Figure 1 of the paper.

Main results in four numbers

DreamZero is a 14B autoregressive video diffusion transformer, trained on about 500 hours of diverse teleoperation data.

>2×

Higher task progress than the best pretrained VLA on new tasks and new environments

38×

Inference speedup: 5.7 s to 150 ms for each action chunk, real-time control at 7 Hz

+42%

Relative gain on unseen tasks from 10–20 minutes of video-only data (human or other robot)

30 min

Play data necessary to adapt to a new robot (YAM), with zero-shot generalization kept

The model weights, inference code, and benchmark code are open-source.

Formulation: video prediction plus inverse dynamics

DREAMZERO (ONE MODEL)

p(future video, actions | past video, language, robot state)

= VIDEO PREDICTION

p(future video | past video, language, robot state)

× INVERSE DYNAMICS (IDM)

p(actions | past and future video, robot state)

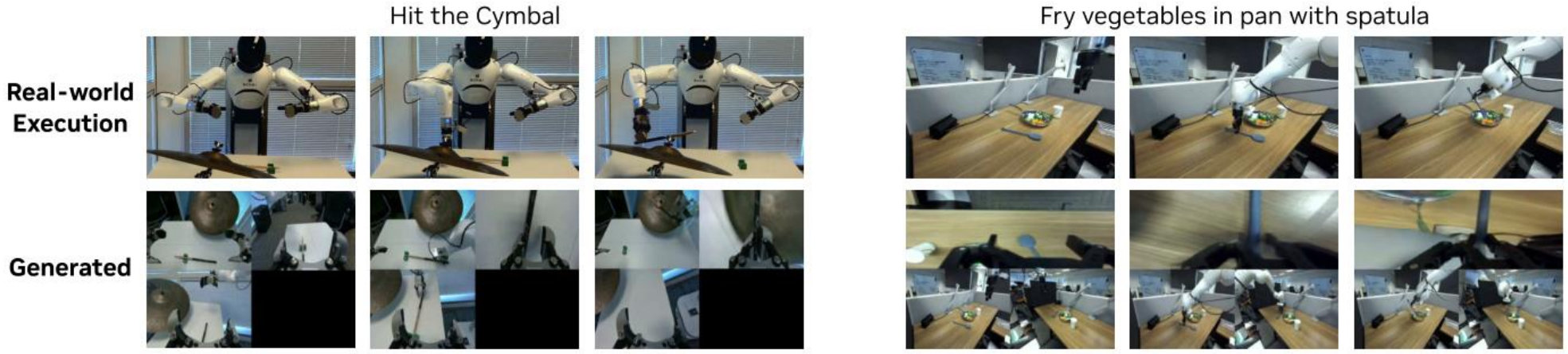
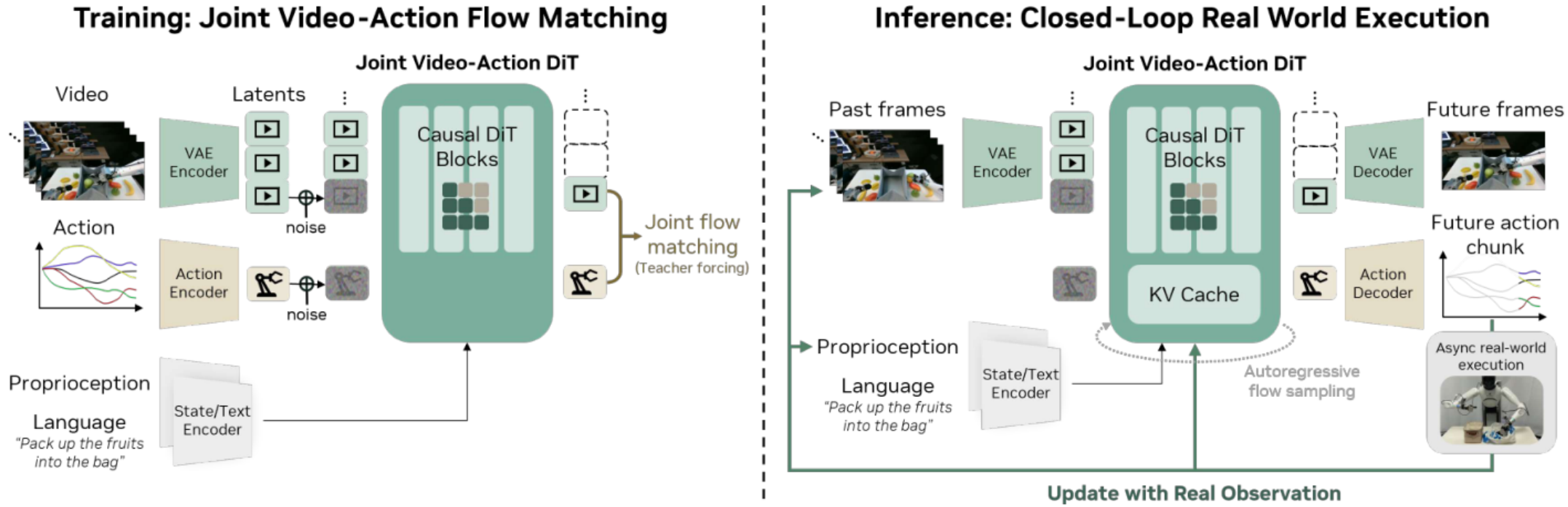


Figure 2: the executed motion (top) agrees with the generated video (bottom), on unseen tasks. One end-to-end model does the two steps, not two separate models.

Model architecture



Inputs: video (VAE), language (text encoder), proprioception (state encoder).

Backbone: Wan2.1 I2V 14B DiT. Only small state and action encoders and decoders are new.

Outputs: future video frames and an action chunk, denoised together.

Three design choices

ARCHITECTURE

Autoregressive, chunk by chunk

Each chunk has a fixed number of latent frames that agrees with the action horizon.

The native frame rate is kept, thus video and actions stay aligned.

A KV cache makes inference 3–4× faster than a bidirectional model.

TRAINING

Joint flow matching

One velocity target for the video latents and the actions.

Video and action share the denoising timestep.

Teacher forcing: denoise the current chunk with clean earlier chunks as context.

INFERENCE

Closed loop with real frames

After the robot executes an action chunk, real observations replace the generated frames in the KV cache.

This removes the compounding error of autoregressive video generation.

Real-time execution: a 38× speedup

5.7 s → 150 ms

Latency for one action chunk. The naive model is too slow for closed-loop control.

Asynchronous execution. The robot executes the latest chunk while the next inference runs. A chunk is 48 steps at 30 Hz (1.6 s), thus the target latency is about 200 ms.

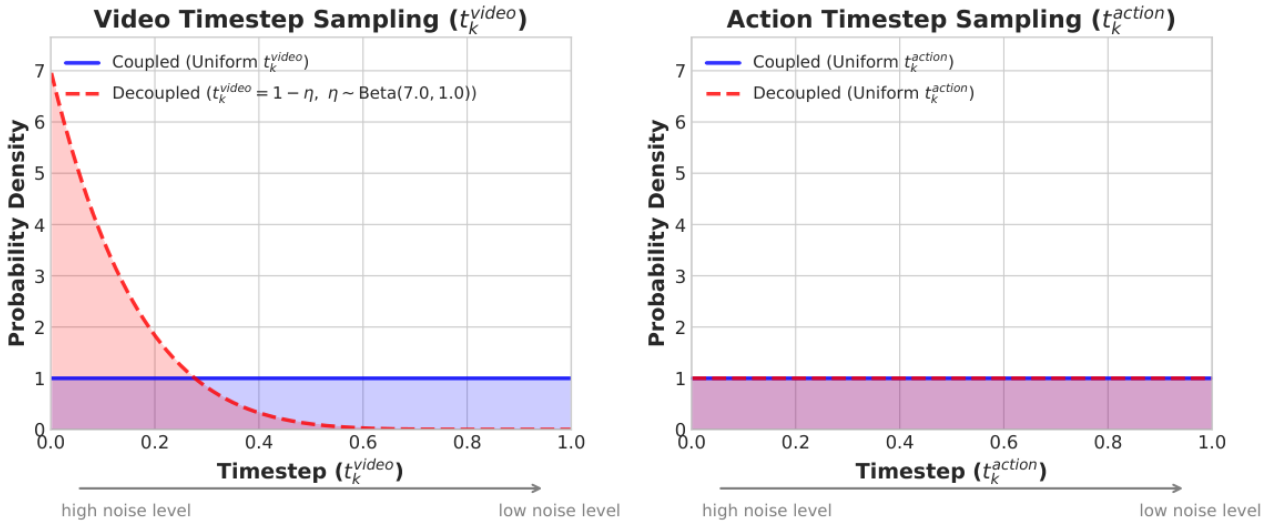
DiT caching. If successive velocities are almost parallel, the model uses the cached velocity again: 16 steps become 4.

DreamZero-Flash. 4 steps become 1.

Level	Optimization (cumulative)	H100	GB200
	Baseline	1×	1.1×
System	+ CFG parallelism (2 GPUs)	1.9×	1.8×
System	+ DiT caching	5.5×	5.4×
Implementation	+ Torch compile, CUDA graphs	8.9×	10.9×
Implementation	+ Kernel and scheduler	9.6×	14.8×
Implementation	+ Quantization (NVFP4)	n/a	16.6×
Model	+ DreamZero-Flash	n/a	38×

Table 1 of the paper. Each row includes all rows above it. “n/a”: not applicable to that hardware.

DreamZero-Flash: decoupled noise schedules



Method	Steps	Task progress	Latency
DreamZero	4	83% ± 6.1%	350 ms
DreamZero	1	52% ± 10.2%	150 ms
DreamZero-Flash	1	74% ± 10.1%	150 ms

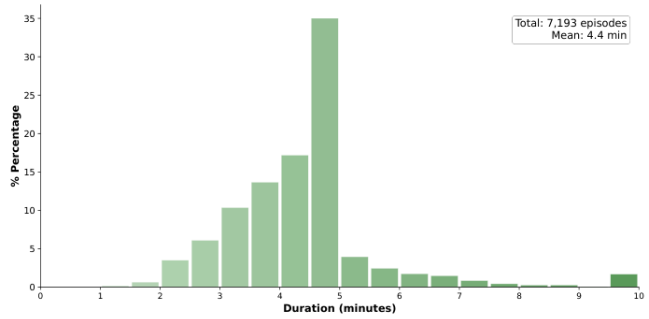
Problem. With one denoising step, the video is still noisy when the action must be clean. Training with a shared timestep does not show this case to the model.

Fix. During training, sample the video timestep from a Beta(7, 1) distribution (mostly high noise). Keep the action timestep uniform.

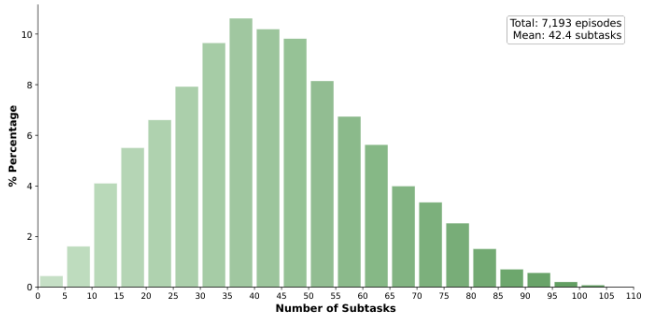
Result. One step gives 74% task progress on table bussing, 9 points below the 4-step model, at 2.33× the speed.

Figure 5 and Table 3 of the paper.

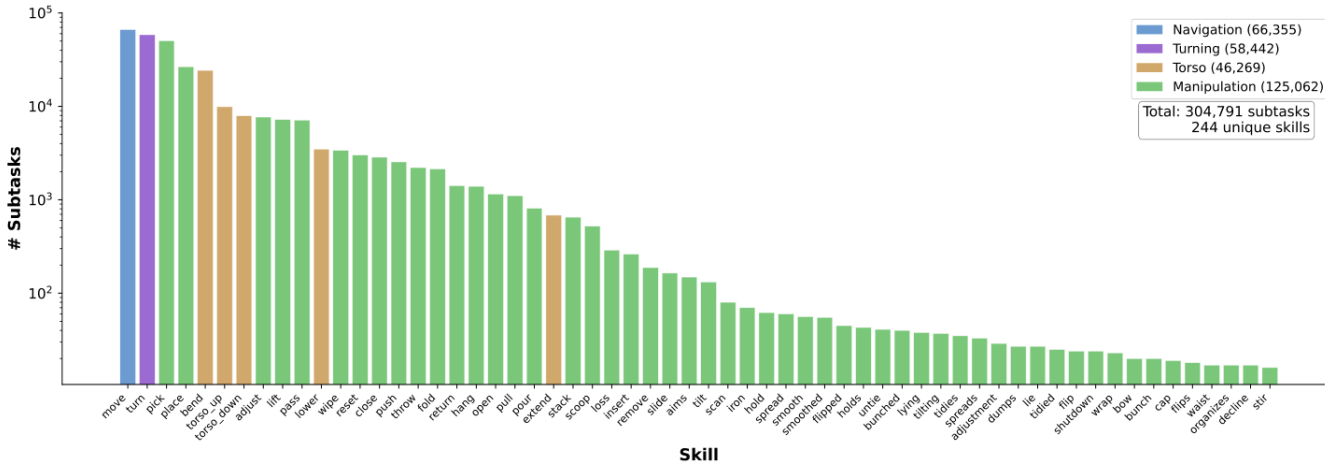
Pretraining data: diversity before repetition



(a) Episode Duration Distribution



(b) Subtask Count Distribution per Episode



(c) Skill Distribution

~500 h

AgiBot G1 teleoperation

22

real environments

4.4 min

mean episode duration

42

subtasks in one episode (mean)

7.2K long episodes from homes, restaurants, supermarkets, coffee shops and offices. The episodes are not repeated demonstrations of one task.

Second embodiment: Franka, with the public DROID dataset.

Figure 6 of the paper.

Evaluation setup

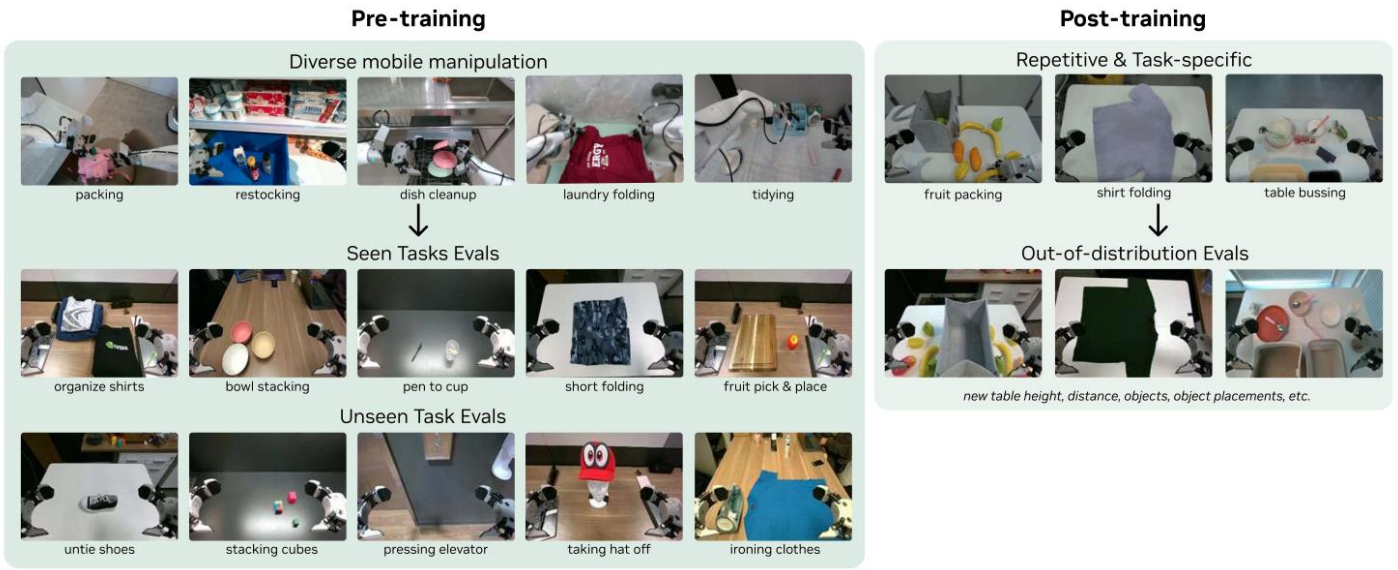


Figure 7 of the paper.

Default condition: unseen environment and unseen objects. The evaluation sites are in a different geographic location from the data collection sites.

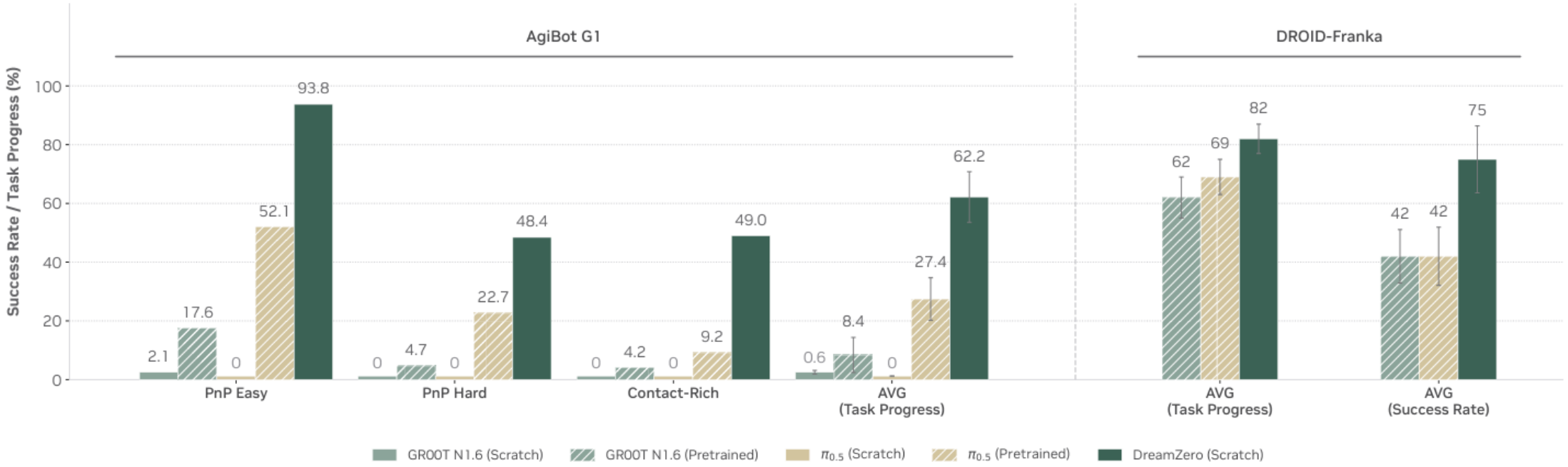
Baselines. GR00T N1.6 and π0.5, each from scratch (VLM weights only) and from the official pretrained checkpoint.

Same data. All models train on the same data, with the same batch size and gradient steps.

Seen and unseen tasks. A task is a motion plus an object type. Folding socks is unseen if the data has only shirts.

Rollouts. AgiBot: 10 tasks × 8 rollouts for each group. DROID: 20 + 20 tasks × 2 rollouts.

Q1. Do WAMs learn better from diverse data?

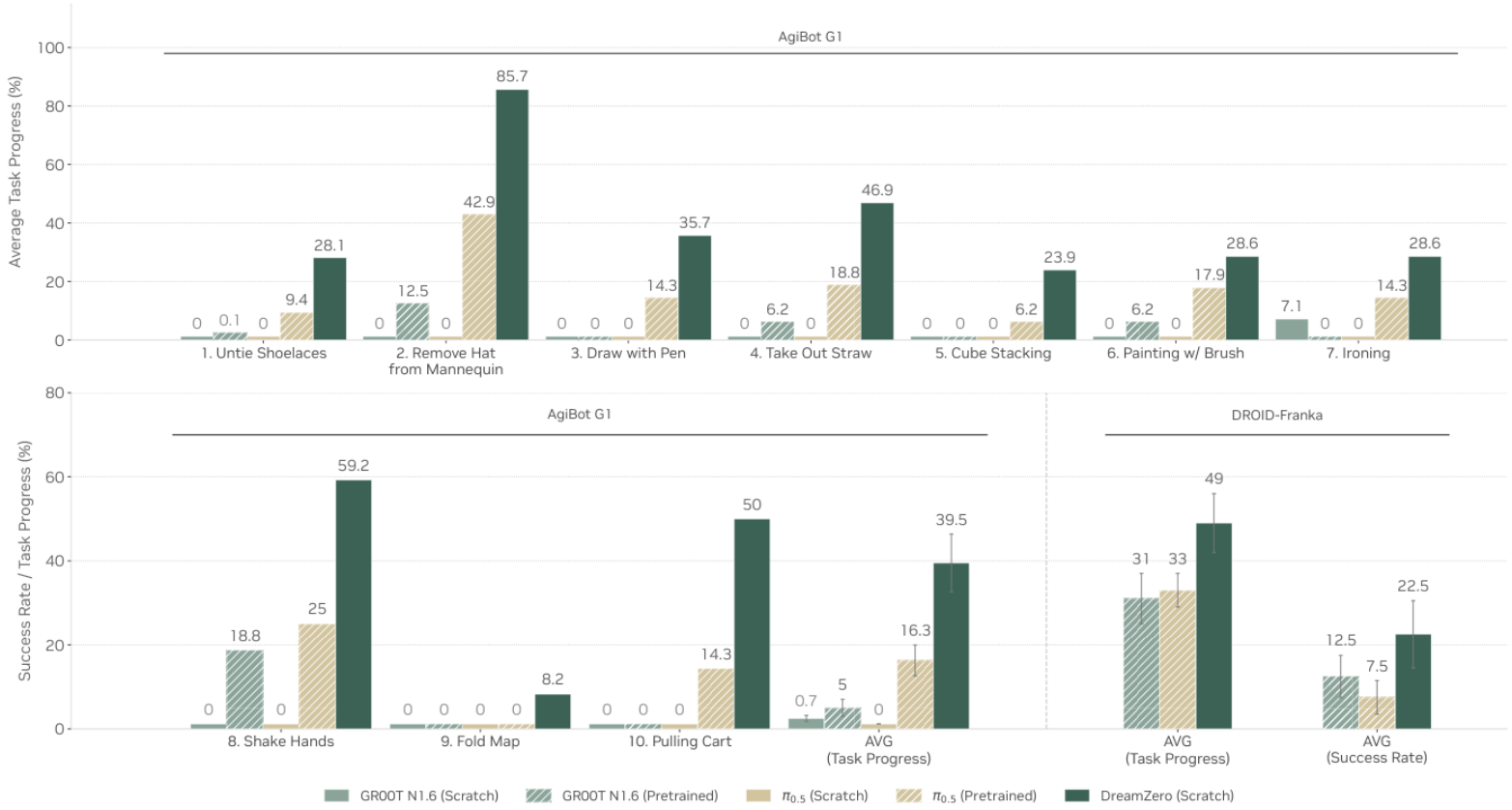


AgiBot G1, seen tasks: DreamZero 62.2% mean task progress. Best pretrained VLA: 27.4%.

From-scratch VLAs: near 0%. They move to the object but do not interact correctly.

DROID-Franka: DreamZero 82% task progress, 75% success. Best VLA: 69% and 42%.

Q2. Do WAMs generalize to unseen tasks?



39.5%

DreamZero on 10 unseen AgiBot tasks. Best pretrained VLA: 16.3%. From-scratch VLAs: below 1%.

85.7%

“Remove hat from mannequin”. “Shake hands”: 59.2%.

49%

DROID-Franka task progress. GR00T N1.6: 31%. π0.5: 33%.

Free-form evaluation on more than 100 tasks



The operators give verbal instructions with no fixed task list.

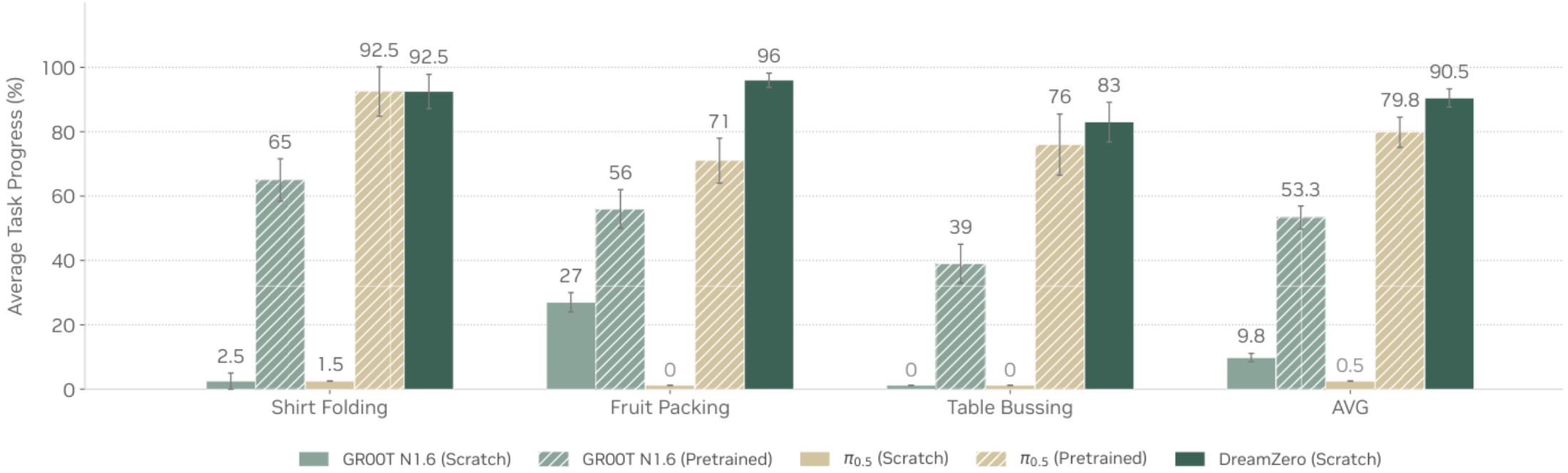
Object manipulation: stack bowls, put mango on tray.

Tool use: wipe a whiteboard, pour sauce.

Human-robot interaction: fist bump, hand a tool to a person.

Figure 3 of the paper.

Q3. Do WAMs improve post-training performance?

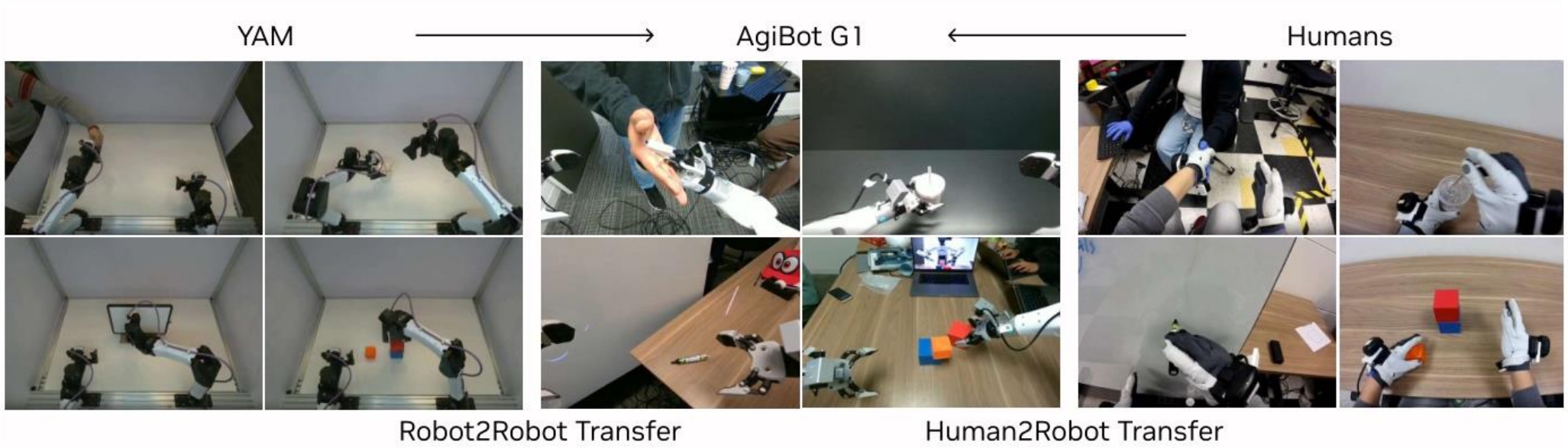


Mean task progress: DreamZero 90.5%, π0.5 (pretrained) 79.8%, GR00T N1.6 (pretrained) 53.3%.

Fruit packing: the largest gain, 96% compared with 71%. Shirt folding is equal at 92.5%.

Meaning: environment generalization stays after task-specific fine-tuning.

Q4. Cross-embodiment transfer from video only



38.3%

DreamZero baseline, 9 unseen tasks (± 7.6%)

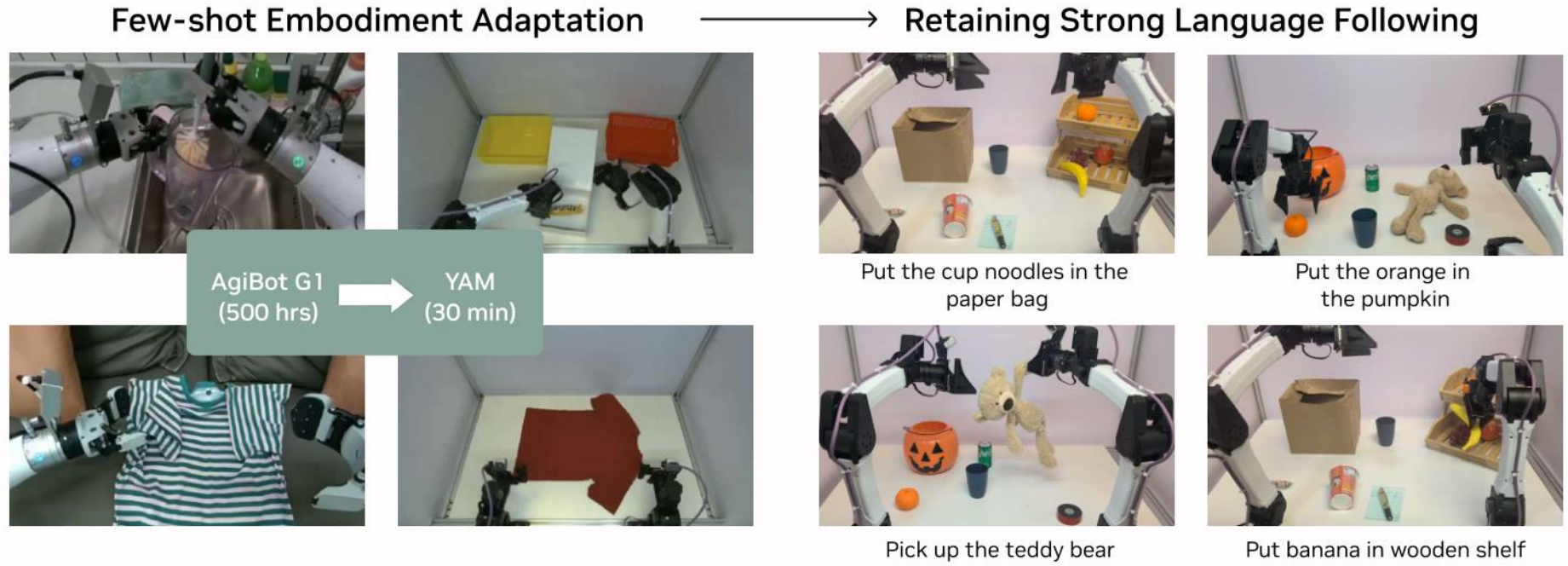
54.3%

+ 12 minutes of human egocentric video (± 10.4%)

55.4%

+ 20 minutes of YAM robot video (± 9.5%)

Q5. Few-shot adaptation to a new robot



Data: 55 trajectories, 11 tasks, about 30 minutes of play data on the YAM robot.

Result: the policy follows language and handles objects that are not in the YAM data.

Why: the model only has to learn a new mapping from predicted video to actions.

Ablations: data, model size, architecture

Question	Model	Size	Data	Task progress
Data diversity	DreamZero (AR)	14B	Repetitive	33% ± 4.2%
	DreamZero (AR)	14B	Diverse	50% ± 6.3%
Model scale	DreamZero (AR)	5B	Diverse	21% ± 4.2%
	VLA	5B	Diverse	0% ± 0.0%
	VLA	14B	Diverse	0% ± 0.0%
Architecture	DreamZero (BD)	14B	Diverse	50% ± 14.4%

Table 4 of the paper. PnP Easy tasks. 50K steps, batch size 32. AR = autoregressive, BD = bidirectional. The highlighted row is the default model.

Diverse data helps. 500 hours of diverse data are better than 500 hours of repetitive data (70 tasks).

A larger video model helps. The 5B model hallucinates more, and wrong video gives wrong actions.

A larger VLA does not help. It stays at 0% on diverse data.

AR equals BD in task progress. AR gives smoother motion and 3–4× faster inference.

Limitations and future work

Inference cost	7 Hz needs two GB200 GPUs. Current VLAs run at more than 20 Hz on consumer GPUs.
Short horizon	The visual memory is about 6 seconds. Long tasks need a System 2 planner or a longer context.
High precision	Tasks with sub-centimetre tolerance (key insertion, fine assembly) stay difficult.
Scaling laws	The relation between model size, data size, compute and performance is not measured yet.
Human video	The human data is only 12 minutes from a lab. Large in-the-wild human video is the next step.
Embodiment design	More degrees of freedom need more play data. Human-like robots can use human video better.