Carnegie Mellon University

Self-Improving Vision-Language-Action Models with Data Generation via Residual RL

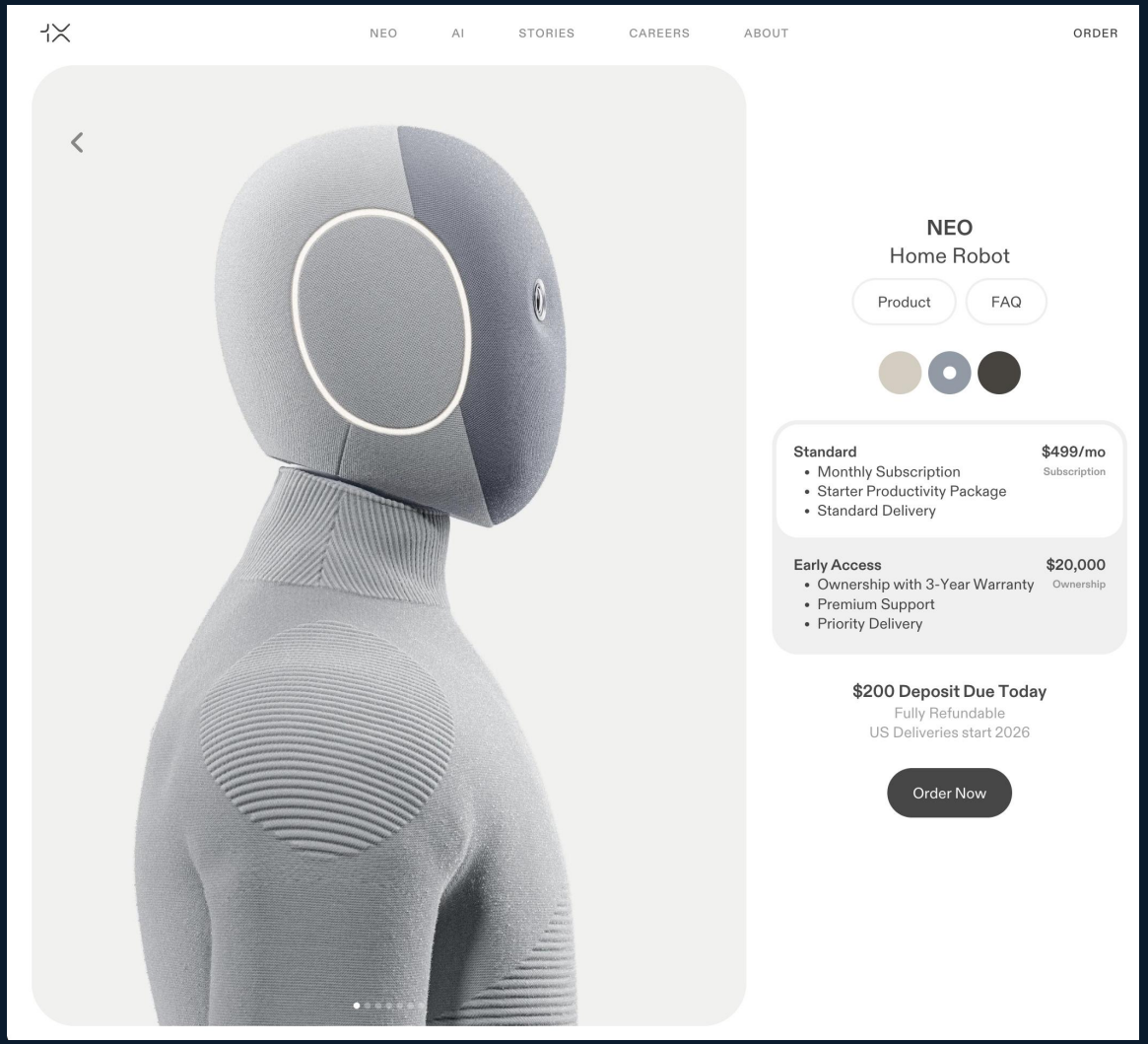
NOV 2025

Wenli Xiao **Haotian Lin**

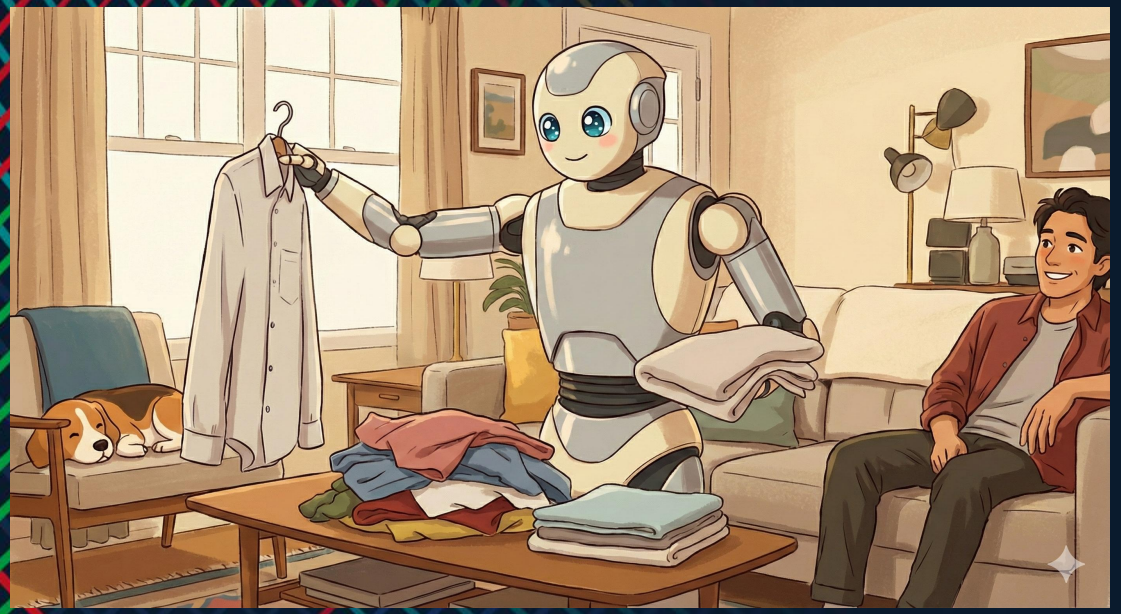
LeCAR@CMU, NVIDIA GEAR

Agenda

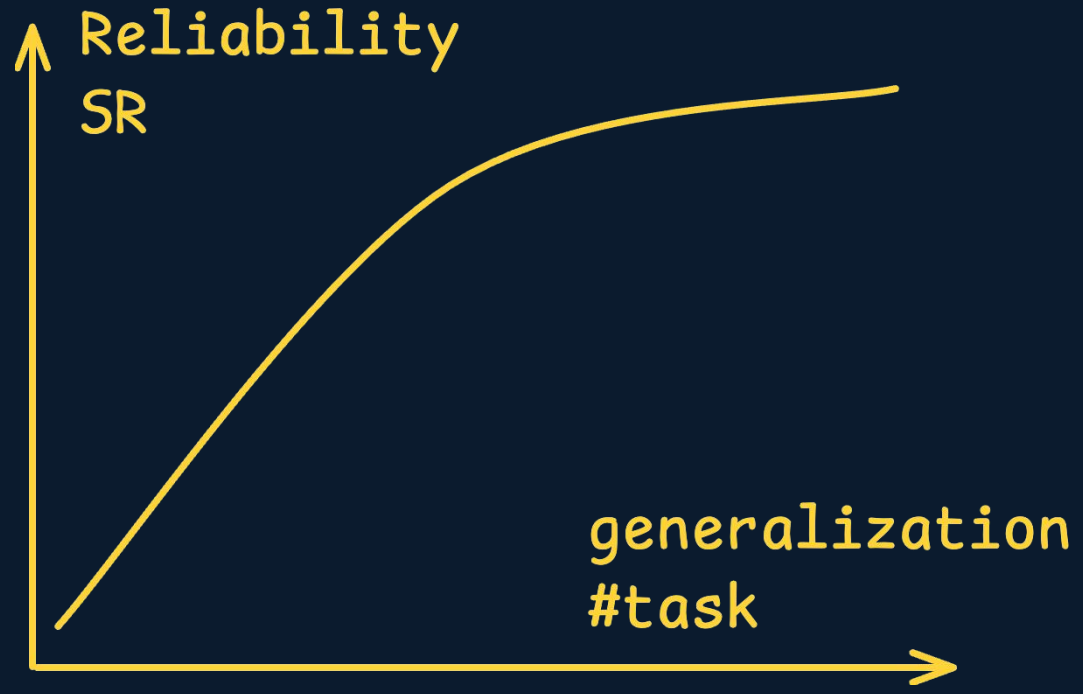
1. Background
2. Motivation
3. Method
4. Experiments
5. Conclusion



Carnegie Mellon University

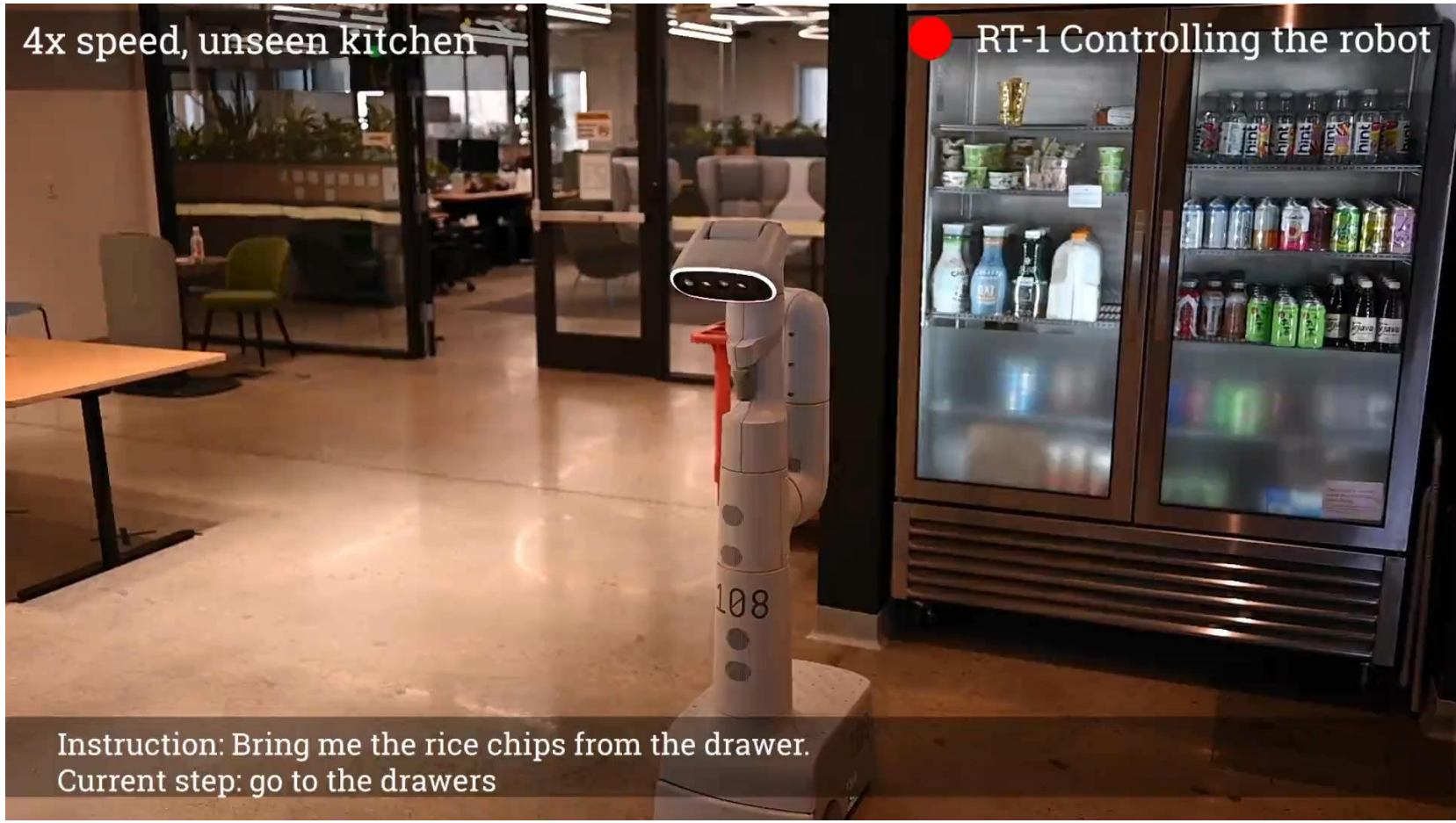


How far are we from general robots in our home?



How far are we from general robots in our home?

Robotics in 2022



RT–1, Google

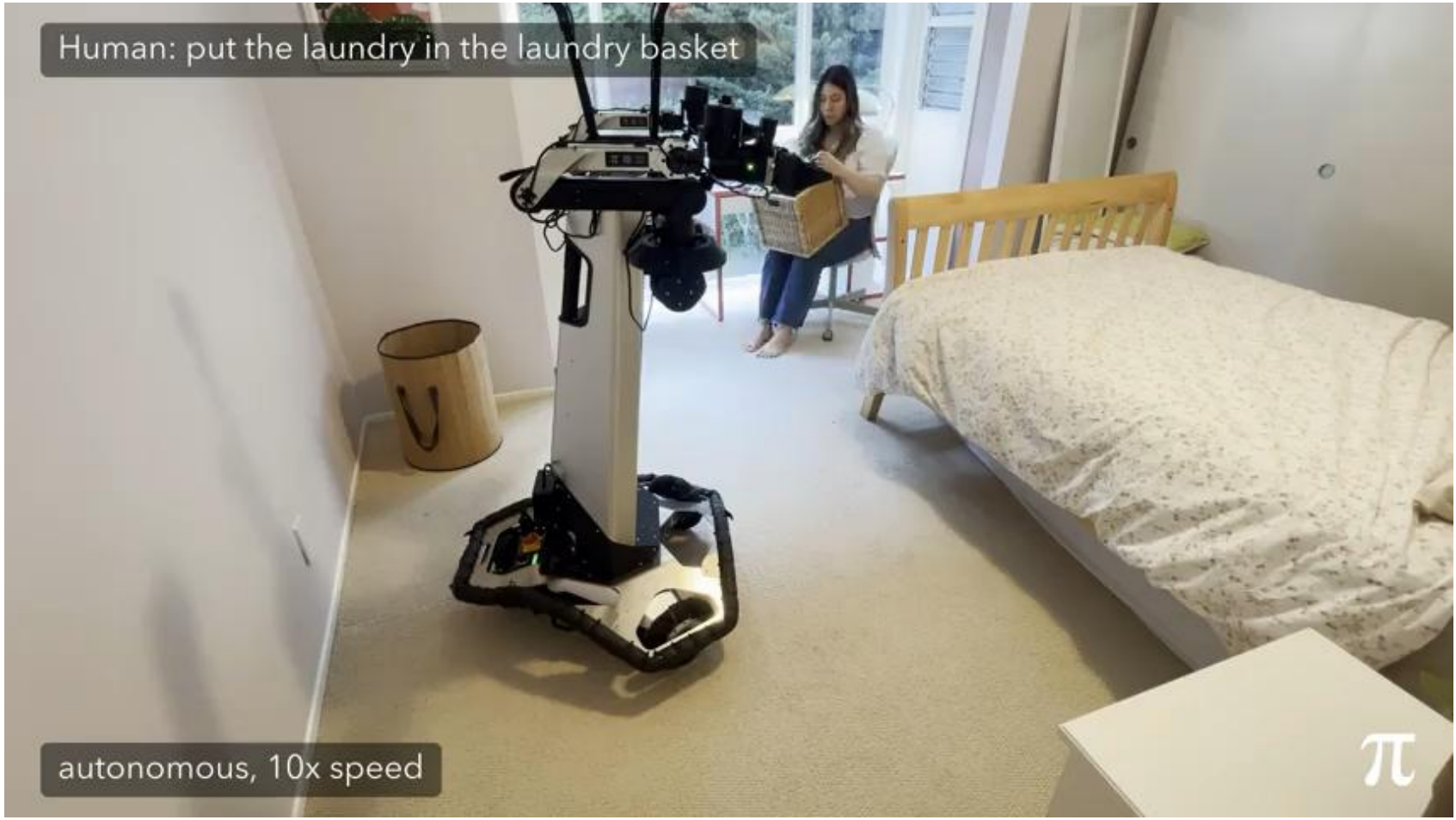
Pick and place in structured env

Robotics in 2024

Pi–0, Physical Intelligence

Laundry

Robotics in 2025



Pi–0.5, Physical Intelligence

Open–world Mobile Manipulation

Robotics this week

ACT-1,
Open-world Mobile Manipulation
Sunday

More data, better generalization

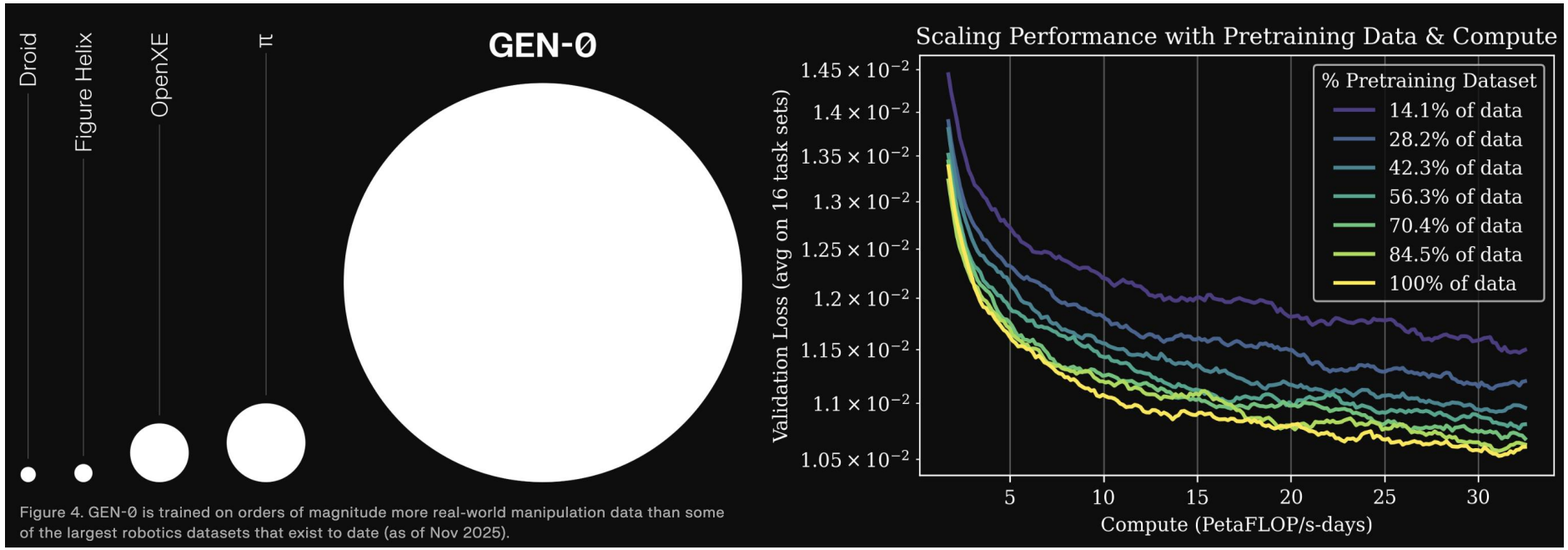


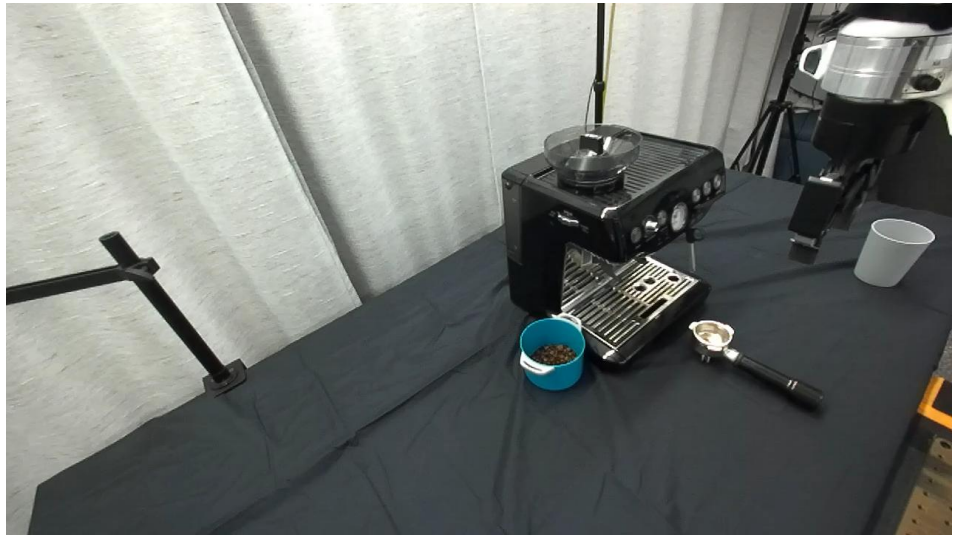
Figure 4. GEN-0 is trained on orders of magnitude more real-world manipulation data than some of the largest robotics datasets that exist to date (as of Nov 2025).

Data Scaling Law shown by Generalist AI[1]

[1]https://generalistai.com/blog/nov-04-2025-GEN-0

More data, better generalization

But not reliable in new Environment!

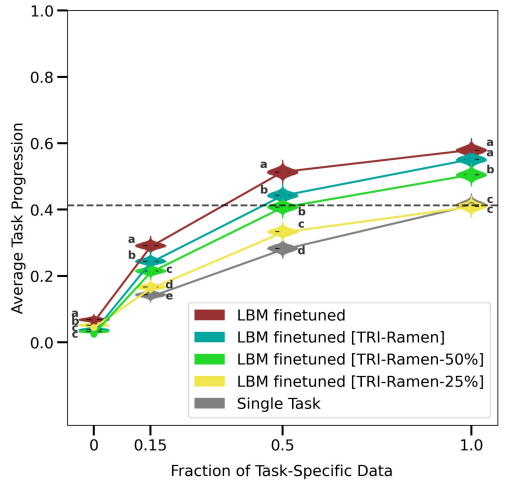


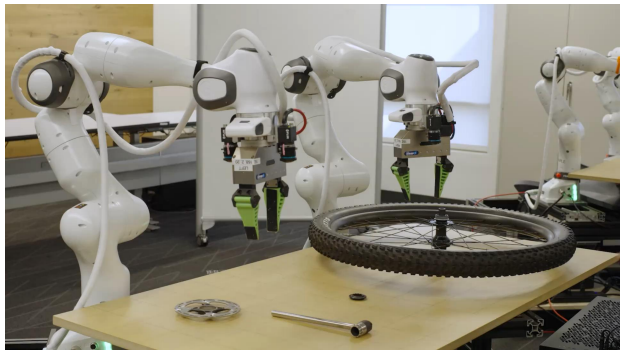
Pi0 Zero-shot deployment in GRASP Lab[1]

Success Rate doesn't scale with more data

	ShirtEasy	ShirtMessy	Number of Demonstrations
Shirt-100%	75%	70%	8,658
Shirt-75%	75%	70%	6,493
Shirt-50%	85%	20%	4,329
Shirt-25%	30%	0%	2,164
Shirt-25%-LongFilter	30%	-	1,623
Shirt-25%-MediumFilter	55%	-	1,082
Shirt-25%-ShortFilter	40%	-	541

ALOHA Unleashed[2]

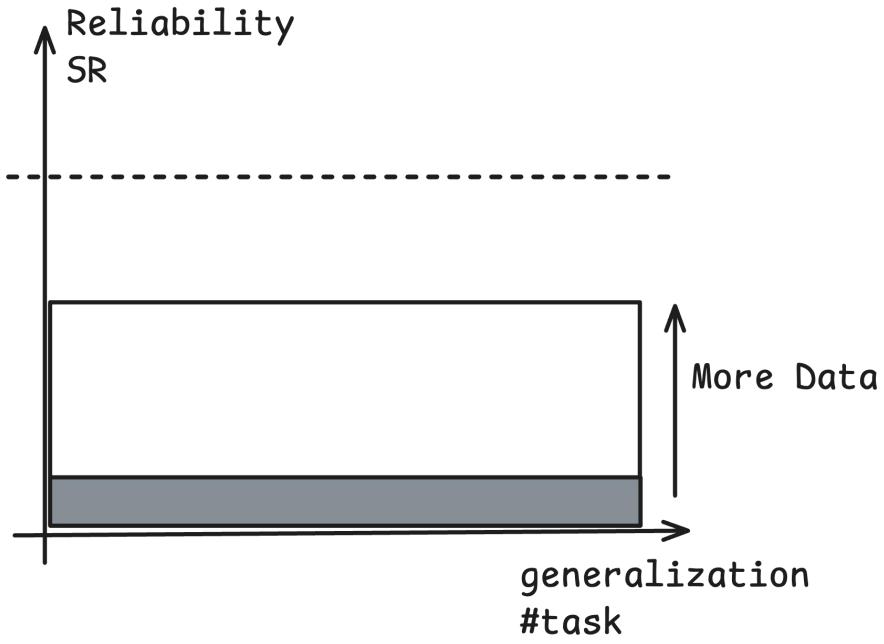




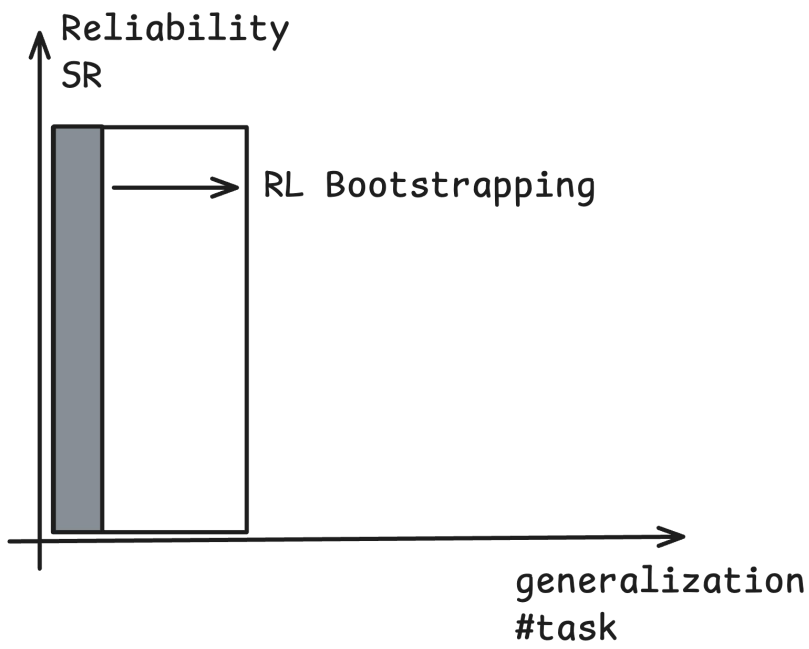
Large Behavior Model[3]

[1]https://penn-pal-lab.github.io/Pi0-Experiment-in-the-Wild/
[2]https://aloha-unleashed.github.io/
[3]https://toyotaresearchinstitute.github.io/lbm1/

Master skills via RL Bootstrapping

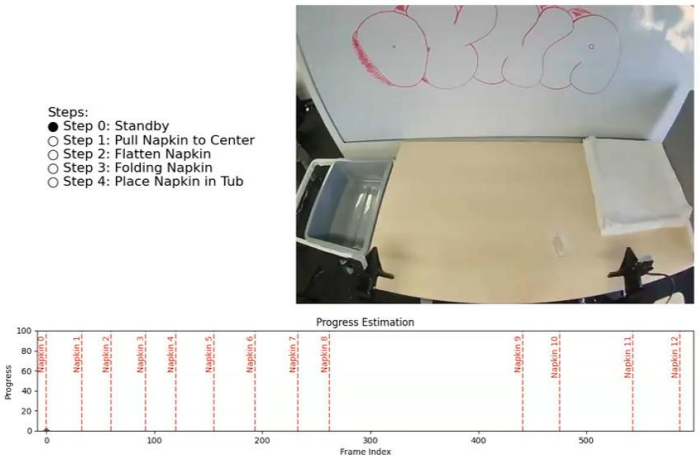


“ChatGPT”

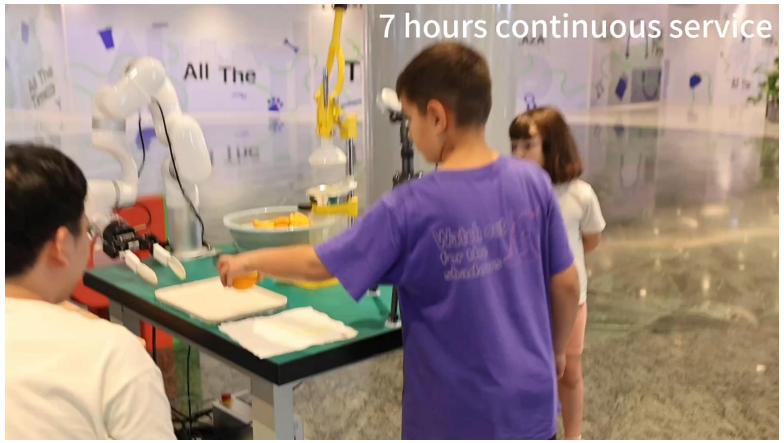


“Deepseek-R1”

Master skills via RL Bootstrapping



Dyna-1[1]

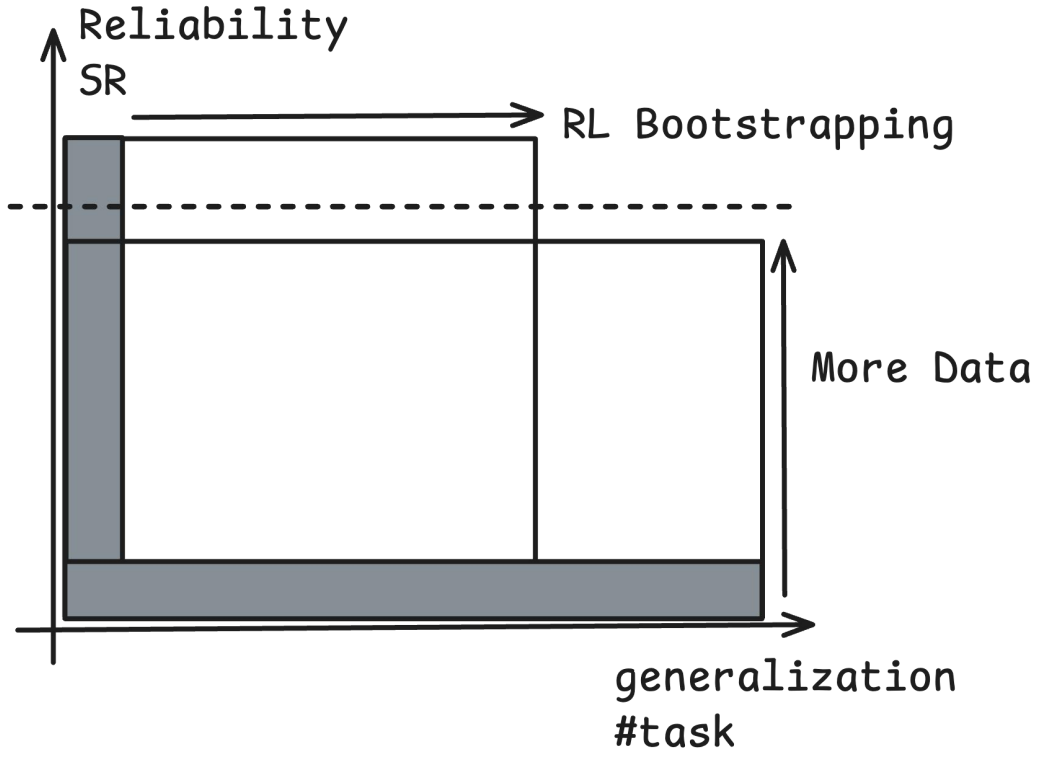


RL-100[2]

HIL-SERL[3]

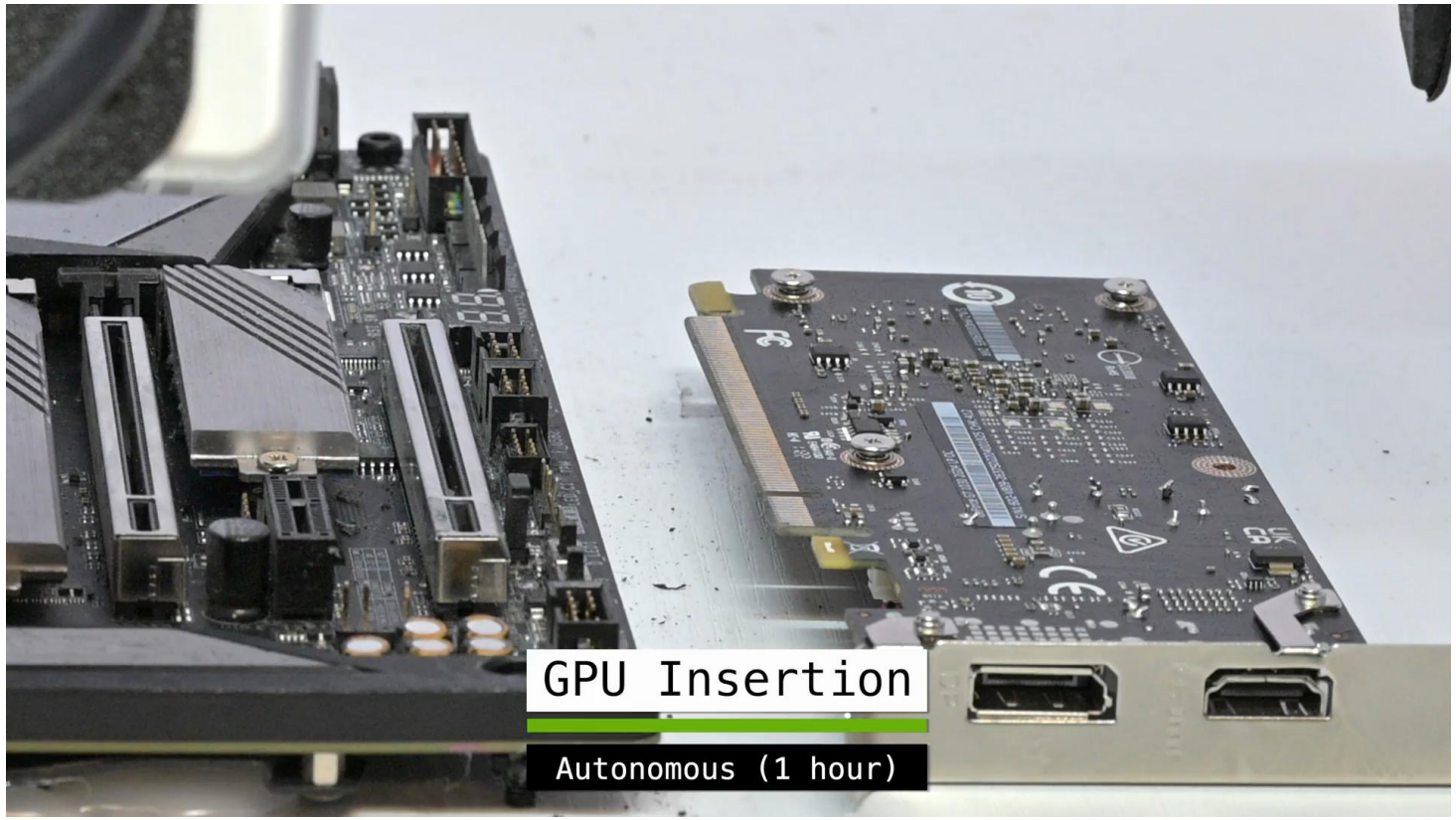
[1]https://www.dyna.co/dyna-1/research
[2]https://lei-kun.github.io/RL-100/
[3]https://hil-serl.github.io/

Can we achieve both?




```
>>> PLD.VLA = ... #OpenVLA, Pi0, Octo, ...
>>> PLD.robot = YAM_REAL_01()
>>> PLD.task = GPU_Insertion()
>>> PLD.Reward = binary()
>>> PLD.train()
VLA_star
```

Can we achieve both?



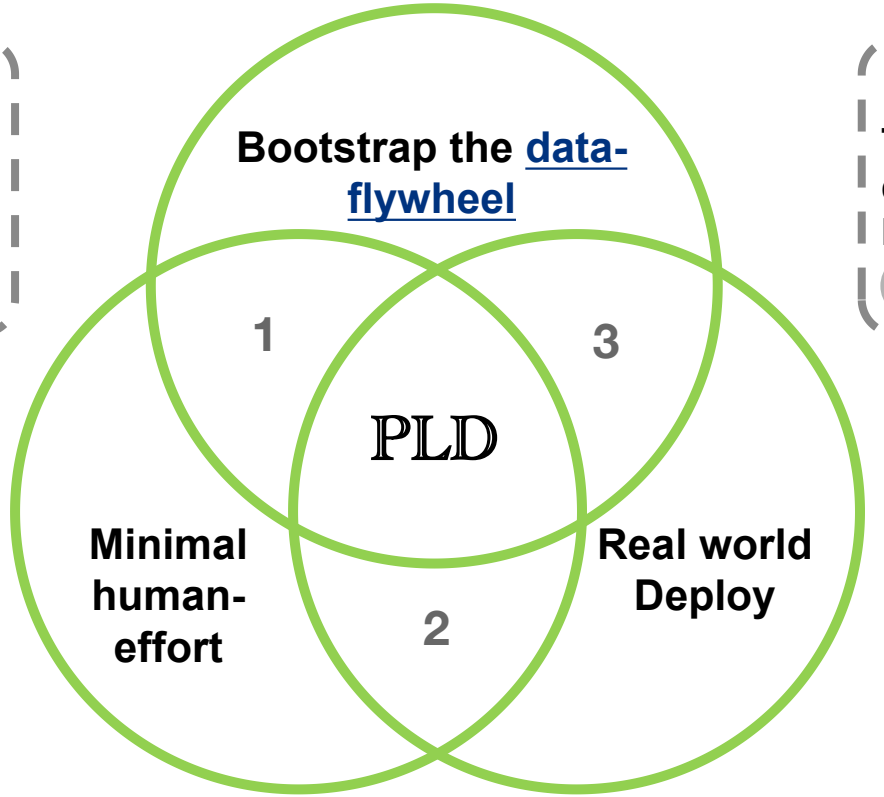

```
>>> PLD.VLA = ... #OpenVLA, Pi0, Octo, ...
>>> PLD.robot = YAM_REAL_01()
>>> PLD.task = GPU_Insertion()
>>> PLD.Reward = binary()
>>> PLD.train()
VLA_star
```

Method

Design Philosophy

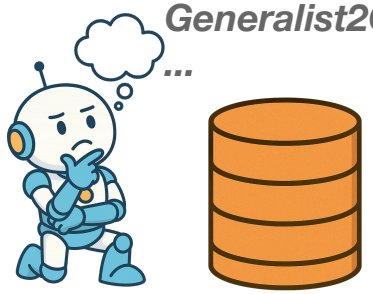
On–policy RL post training + Sim2Real:
VLA–RL, SimpleVLA–RL, π–RL
1

Real-World RL / Off2On:
SERL, Warm-start RL, RL-100
2



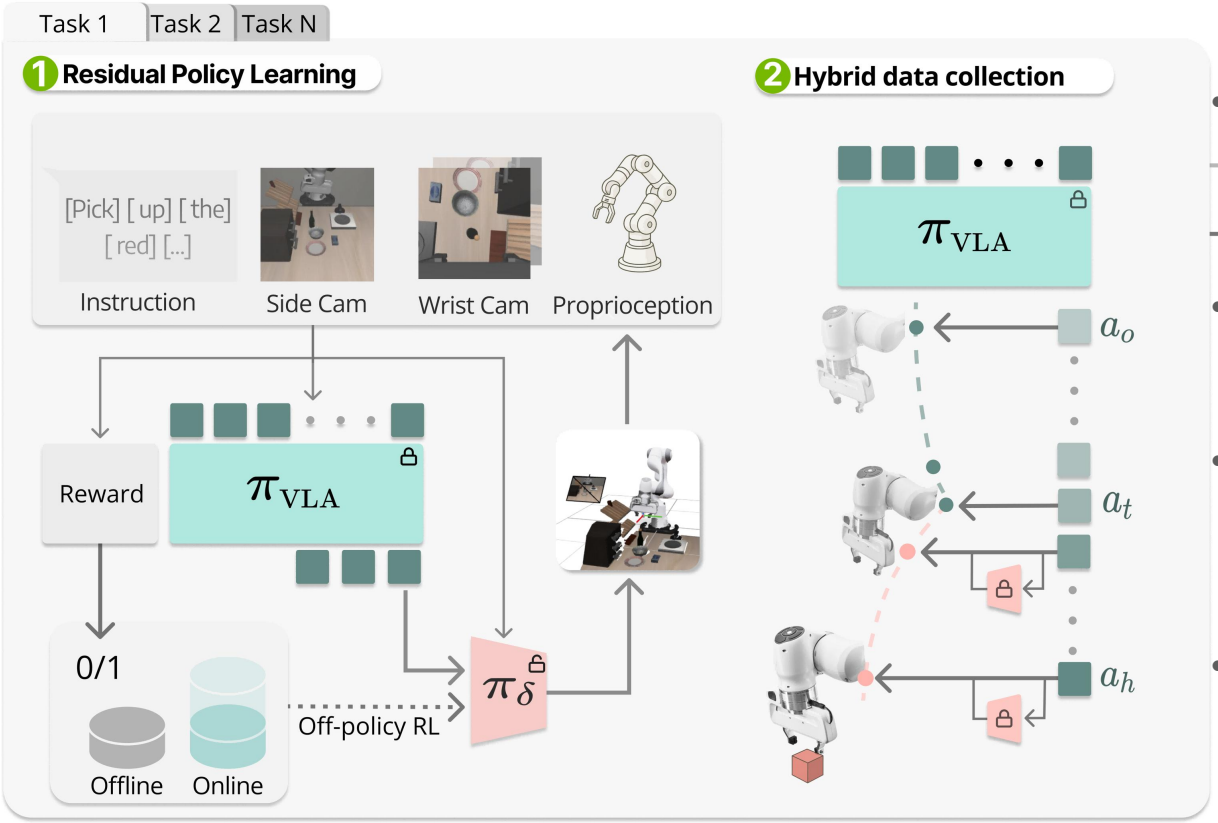
Two-way Communication in data collection:
RaC, HG-Dagger, DexFlyWheel
3

Policy Agnostic, Resource efficiency, Generalist2Generalist, ...

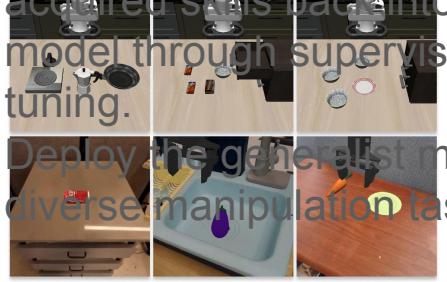


Method

Pipeline in a Nutshell



- Stage 1 (**Learn**): Train a specialist residual policy for each task through online real-world RL.
- Stage 2 (**Probe**): Collect recovery data by probing the failure modes of the base policy.
- Stage 3 (**Distill**): Distill the acquired skills back into VLA model through supervised fine-tuning.
- Deploy the generalist model in diverse manipulation tasks



Diverse Manipulation Tasks

Method

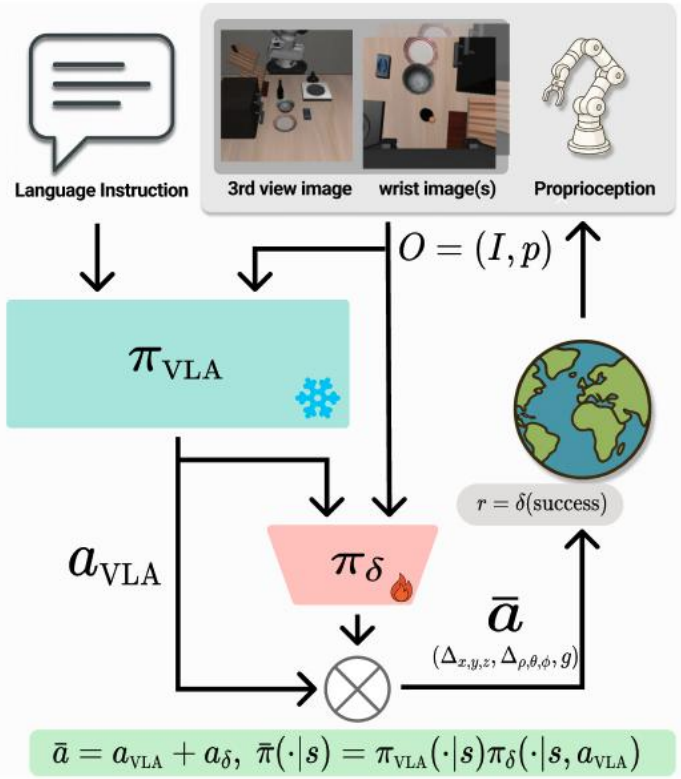
Sample–Efficient Real–World RL

We have seen on–policy RL tuning VLA with hundreds or parallel simulation (GRPO, PPO), what about just one env that deployable on single physical hardware?

Key Components

Offline Warm–start: Collect success rollouts of the base–model to create a small offline dataset. Leverage Calibrated–QL[3] for conservative **critic initialization** (approximately 80k steps) while preventing underestimation.

Oversampling: Symmetric sampling from offline & online replay buffer as in Hybrid–RL[1] to increase high value state–action visitation.



Method

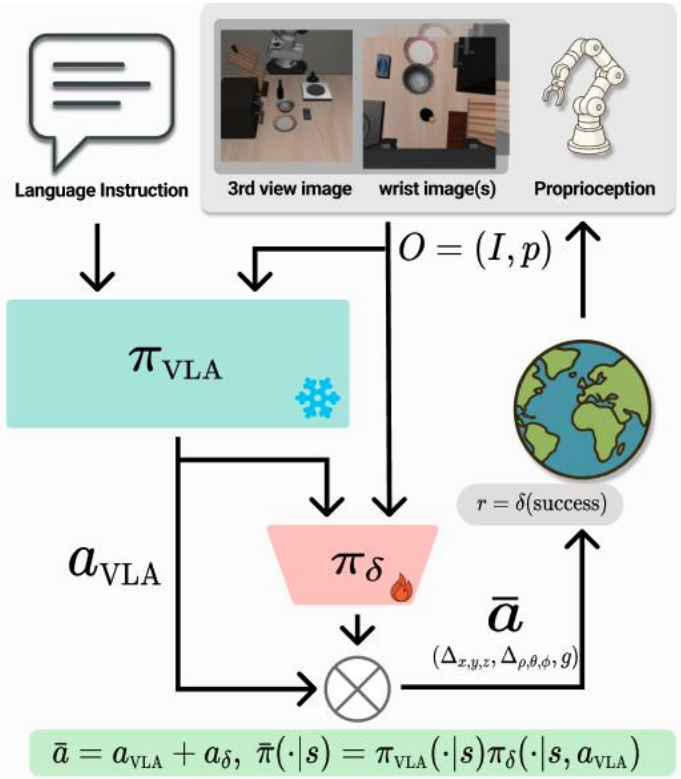
Sample–Efficient Real–World RL

Offline warm–started, Off–policy, Residual RL, w/ Sparse Binary Reward

Key Components

Online Exploration :

Offline performance is bottlenecked when *dataset lack sufficient corrective behavior or task diversity*; Online RL with controlled exploration and on–the–fly refinement solves of better policy with improved **reactivity** and **dexterity** that is *absent* from the chunking policy.



Method

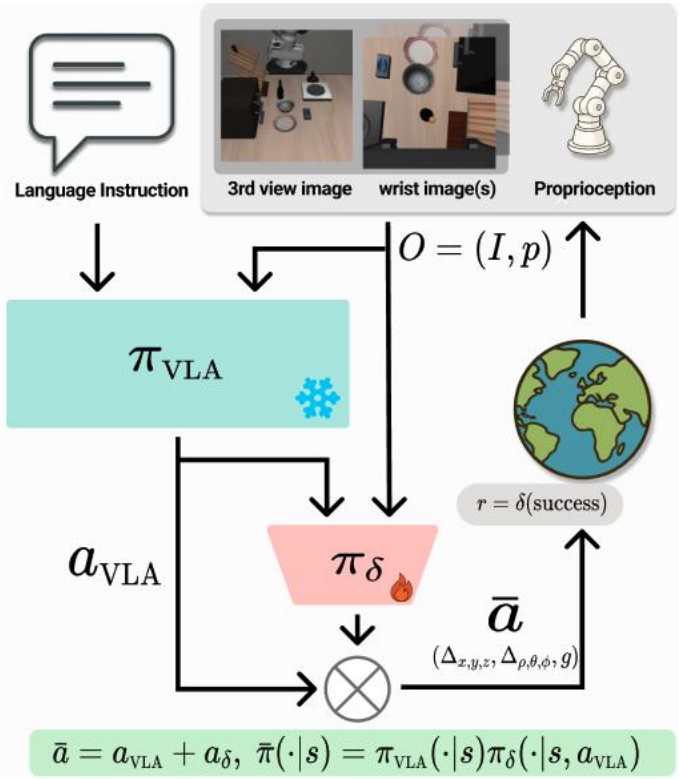
Sample–Efficient Real–World RL

Offline warm–started, Off–policy, Residual RL, w/ Sparse Binary Reward

Key Components

Online Exploration :

Accelerate exploration leveraging the prior knowledge from the base policy: Sampling **50%** from offline buffer(self–bootstrapped data); **Zero initialization** and controll delta actor scale; **State Distribution Shaping**: Using base policy rollouts to initialize exploration (“Jump–start”[4])



Method

Sample–Efficient Real–World RL

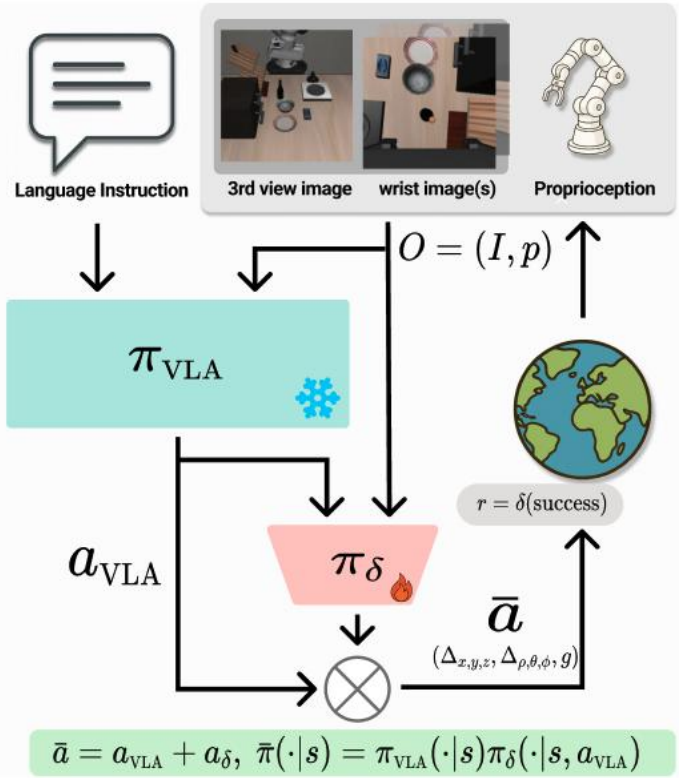
Offline warm–started, Off–policy, Residual RL, w/ Sparse Binary Reward

Architecture:

Policy: VLA base policy (OpenVLA, OCTO, π_0) + light weight residual (ResNet + MLP)

Critic: Shared visual encoder (ResNet + MLP), Evaluates combined action

Reward: Learned success classifier (Could be replaced by foundation reward models)



Method

Sample–Efficient Real–World RL

Offline warm–started, Off–policy, Residual RL, w/ Sparse Binary Reward

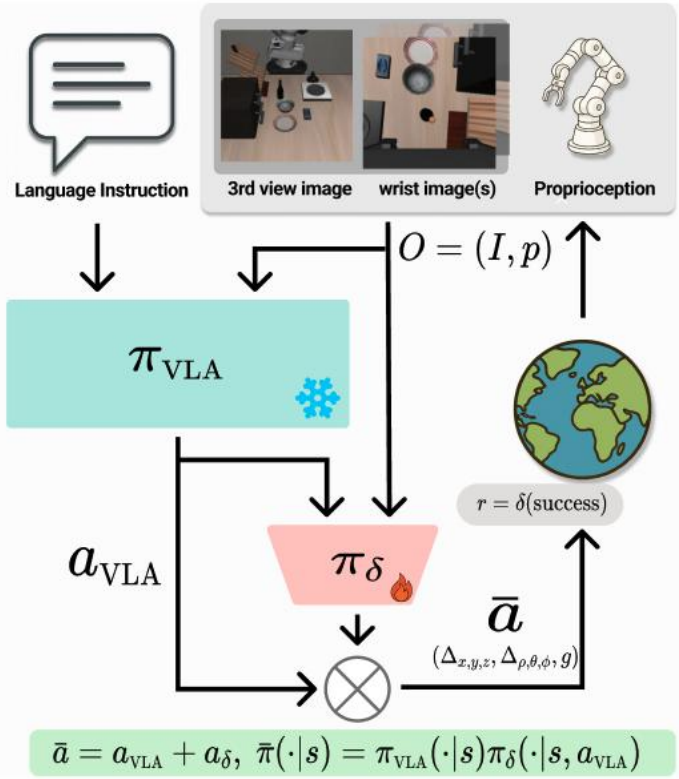
Objectives

Policy: MaxEnt RL (SAC)

$$\pi_\delta = \arg\max_\pi \mathbb{E}_{a_\delta \sim \pi, a_b \sim \pi_b} Q^{\bar{\pi}}(a_b + a_\delta) - \alpha \log \pi(a_\delta | s)$$

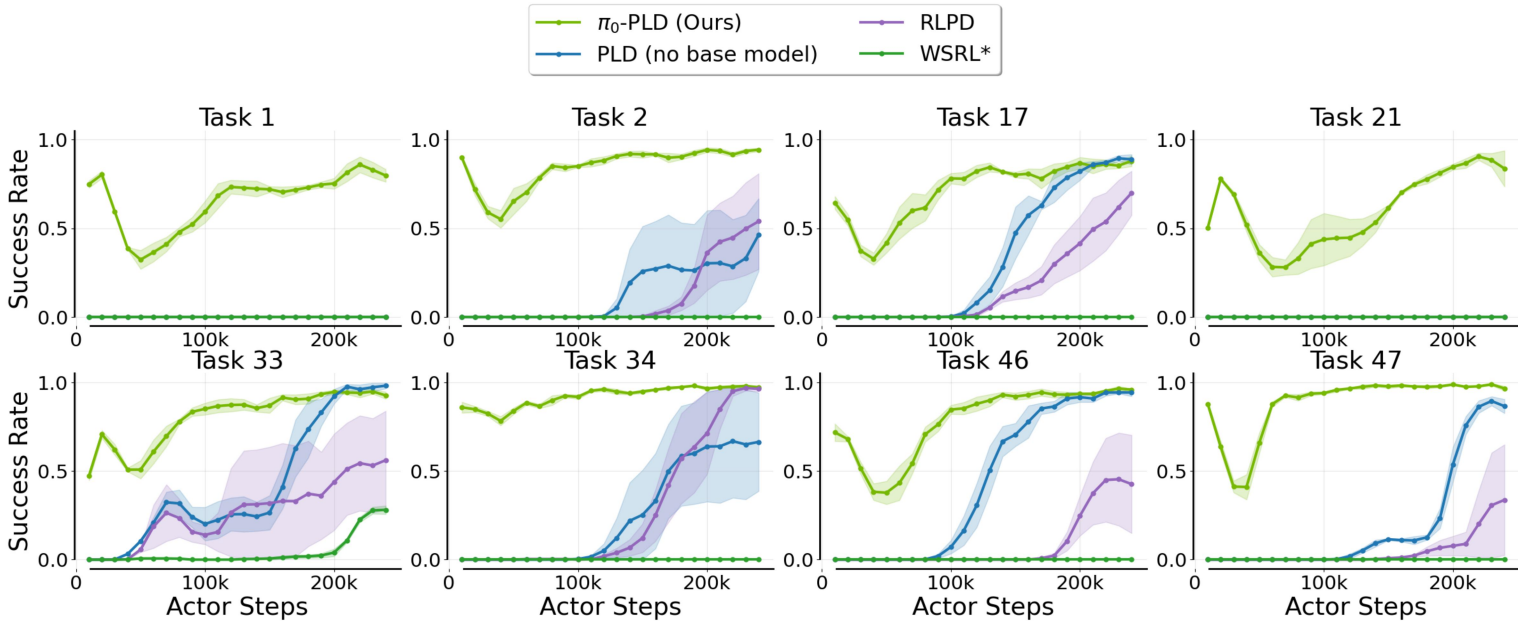
Critic: Standard TD–loss with ensembled Q function

$$Q^{\bar{\pi}}(s_t, \bar{a}_t) \leftarrow r(s,a) + \gamma \mathbb{E}_{s_{t+1} \sim p(\cdot|s_t, \bar{a}_t)}[Q^{\bar{\pi}}_{target}(s_{t+1}, \bar{a}_{t+1})]$$



Method

Sample–Efficient Real–World RL



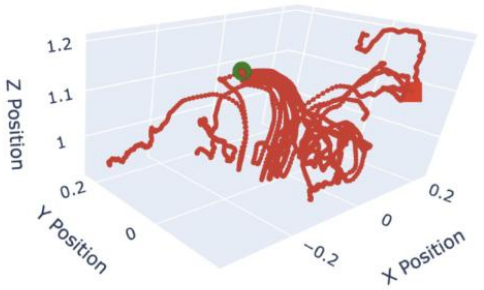
We compare PLD–RL with baseline algorithms that either leverage policy prior or data prior. We report mean rollout performance and 95% CIs for 3 seeds across 8 manipulation tasks selected from LIBERO–90. Performan on all tasks can surpass 95% SR when converge.

Method

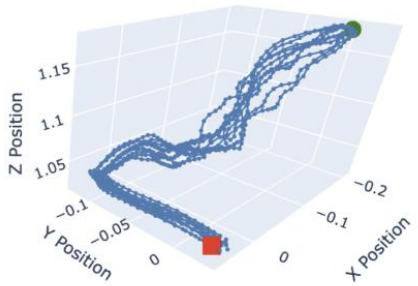
Scaling "Self-Curated" Data

Data flywheel with residual RL expert: RL data is highly optimal, with consistent and smooth behavior, no hesitation, shorter horizon.

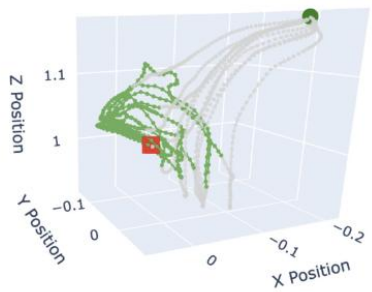
A straightforward way is to learn from this high-quality data, will surely result in improved task-specific performance. It's good but...



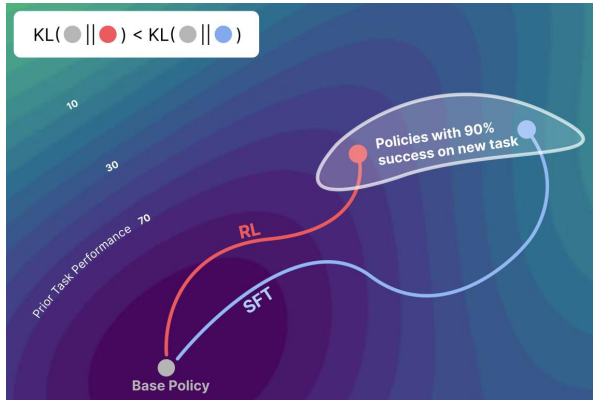
Base Policy Rollout Data



RL Rollout Data



PLD Rollout Data (Ours)

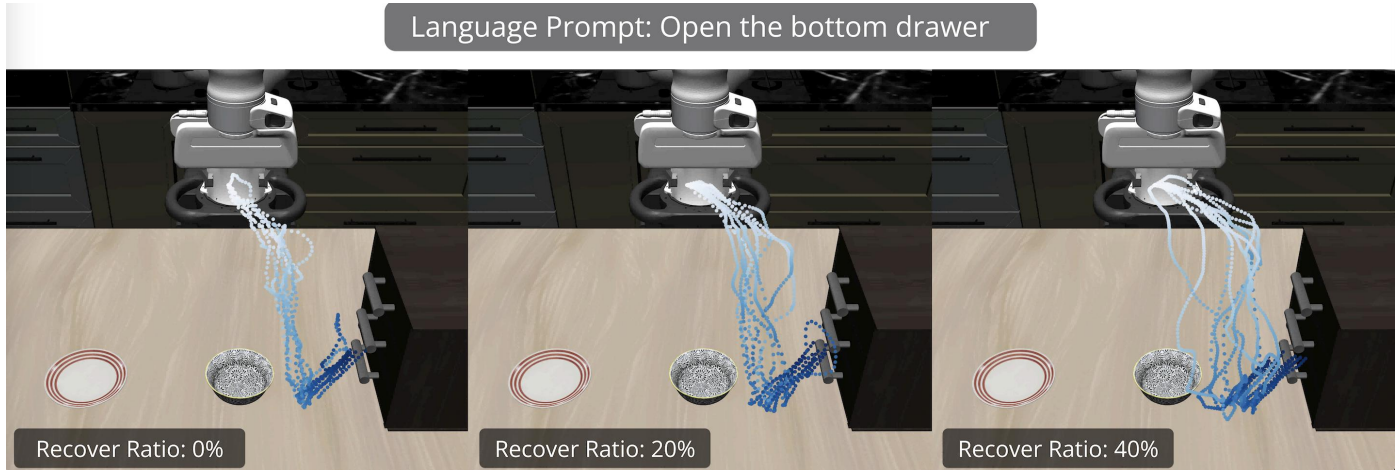


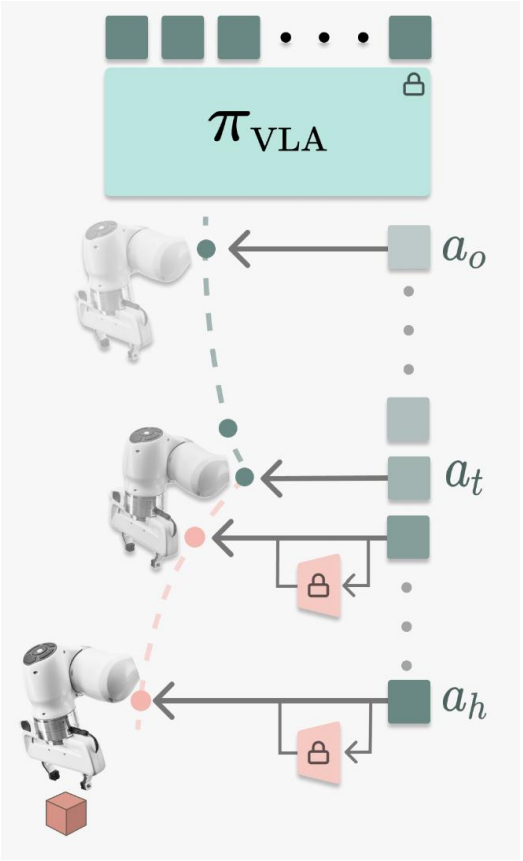
Method

Scaling "Self-Curated" Data

Hybrid data collection scheme: Incorporates base-policy initialization: We first rollout the base policy for random steps, then let the learned residual RL policy to take over.

$$\tau_{demo} = \{(s_1, a_{b,1}), \ldots, (s_{t-1}, a_{b,t-1})\} \cup \{(s_t, a_{b,t} + \bar{a}_t), \ldots\}$$





Method

Distillation via Supervised Fine-tuning (offline)

The entire pipeline is policy agnostic, trivally adopts to different VLA architecture: Flow-based policy, autoregressive...

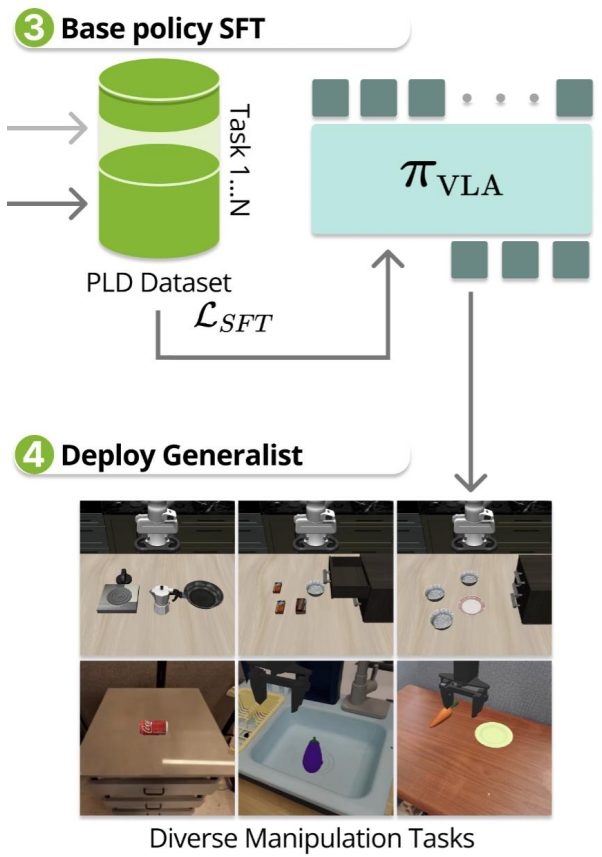
Auto-regressive: sequential NLL

$$\mathcal{L}_{\mathrm{AR}}(\theta) = -\,\mathbb{E}_{k\sim[K]}\left[\log p_\theta\big(u_k \mid u_{<k}, x\big)\right]$$

Diffusion/Flow: Score matching/Velocity matching

$$\mathcal{L}_{\mathrm{diff}}(\theta) = \mathbb{E}_{t,\epsilon,(x,a)}\left[\left\|\epsilon - \epsilon_\theta(a_t^{(\mathrm{noisy})}, x, t)\right\|_2^2\right]$$

Fine-tune modes: Head-only, full parameter, **LoRA**

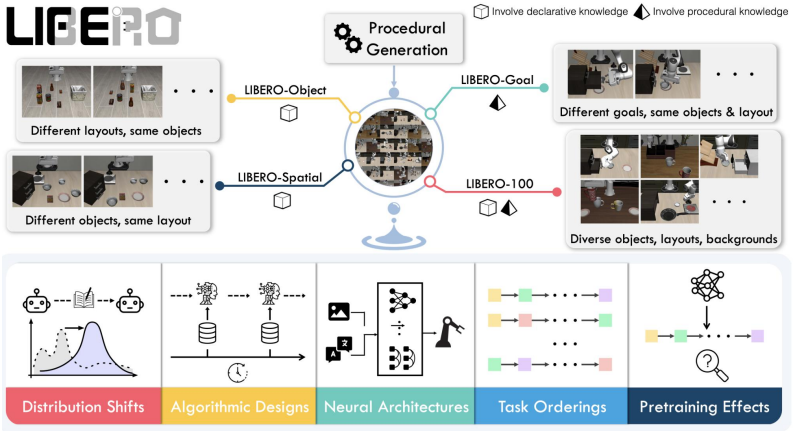


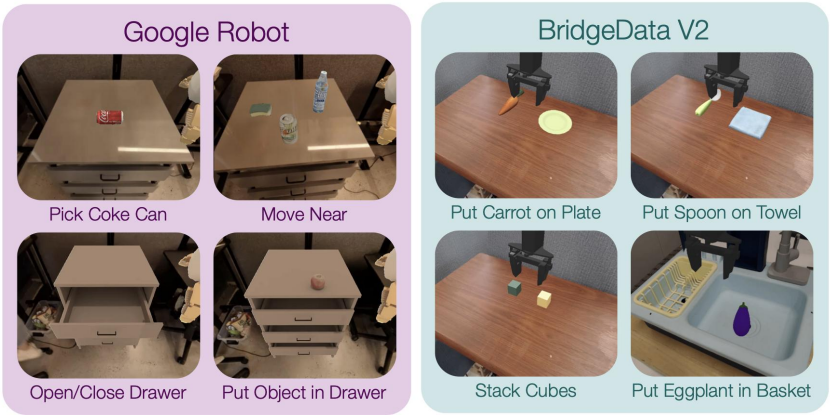
Experiments

Performance on Simulation Benchamarks

LIBERO (https://libero-project.github.io/intro.html): Lifelong learning benchmark focused on language-guided manipulation tasks. It comprises 130 tasks grouped into four suites that stress object distribution, spatial arrangement, task goals

SimplerEnv (https://simpler-env.github.io): A suite of open-source simulated evaluation environments for common real robot manipulation setups (google robot in RT-series and WidowX in Bridge Dataset), aims for high sim-to-real correlation.





Experiments

Performance on Simulation Benchamarks

Can PLD improve state-of-the-art VLA on in-domain tasks?

Table 1: Performance on LIBERO benchmark of VLA models fine-tuned on **PLD** data.

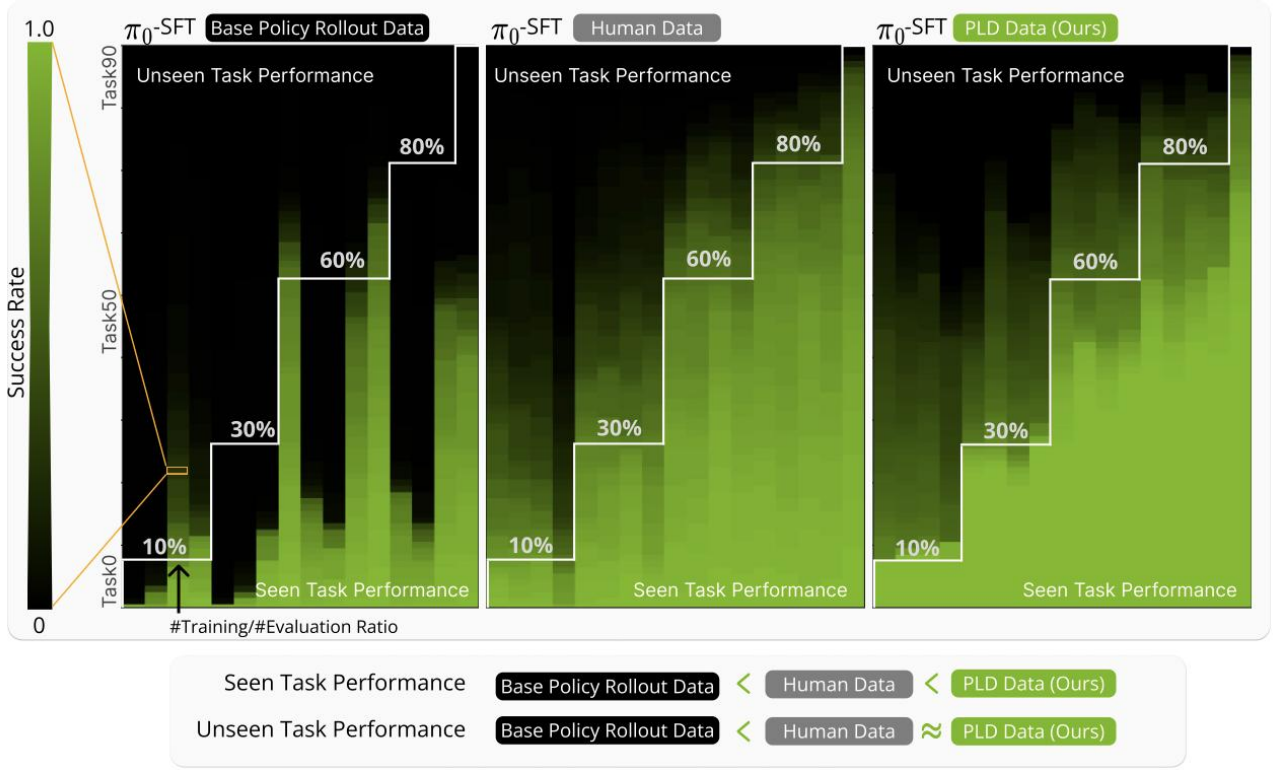
	π_0				OpenVLA			
Model	**Spatial**	**Object**	**Goal**	**Avg**	**Spatial**	**Object**	**Goal**	**Avg**
Baseline (SFT/OFT)	95.2	97.6	87.4	93.4	92.9	99.1	83.25	91.8
w/ PLD	**97.7**	**98.5**	**95.3**	**97.2**	**99.5**	**99.1**	**98.9**	**99.2**
Δ	+2.5	+0.9	+7.9	+3.8	+6.6	+0.0	+15.7	+7.4

Table 3: Evaluate **PLD** on SimplerEnv

Model	**WidowX Pick Eggplant**	**WidowX Pick Carrot**	**Google Open Drawer**	**Google Coke Can**	**Avg**
Octo-SFT	65.5	43.3	92.5	85.7	71.8
w/ ours	97.8	93.9	99.3	95.5	96.6
Δ	+32.3	+50.6	+6.8	+9.8	+24.9

Experiments

Deep Dive: What does PLD brings to generalization



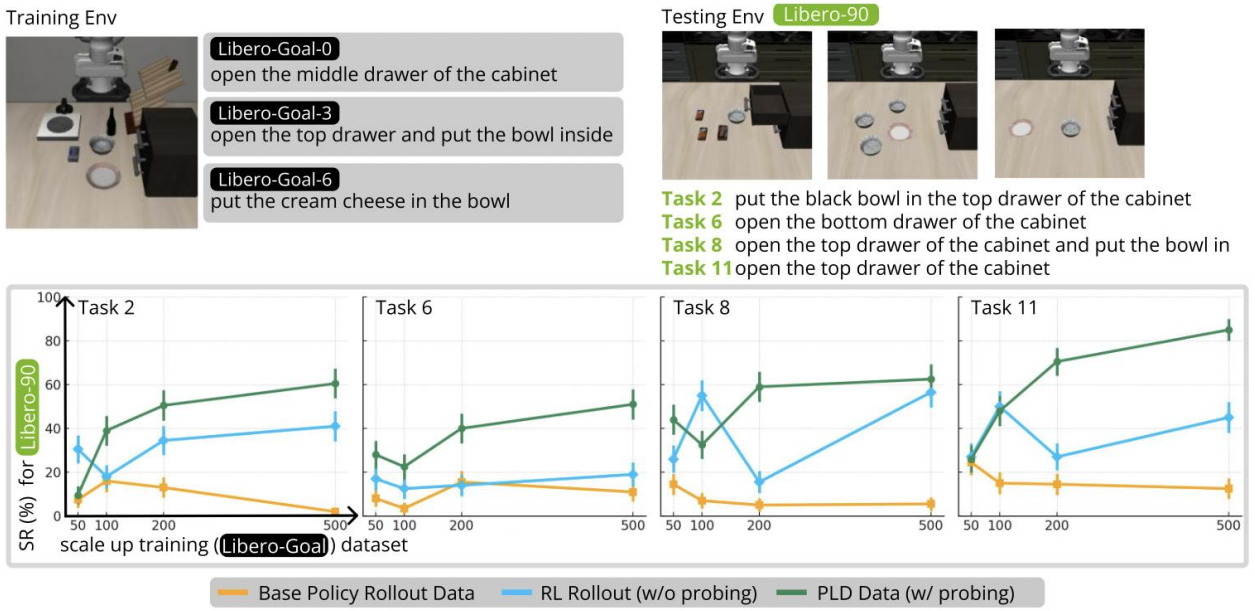
We have been talking about generalist2generalist... But what is wrong about self-bootstrapping successful behaviors / 0-1 REINFORCE ?

Generalization to unseen task: For each ration, random select tasks to form source domain; **SFT** on different data source; **Zero-shot evaluation** on all tasks (LIBERO-90)

PLD is as good as human oracle, if not better

Experiments

Scaling "Self-Curated" Data



Scaling in-domain PLD data yields better few-shot performance:

π_0 SFT: Different scales of data from source tasks (plus 10 oracle demos of unseen tasks).

Monotonic improvements in SFT performance as PLD data scales from 50 to 500 trajectories.

Experiments

Resource Efficiency

VLA base policy remains frozen and we only optimize a lightweight residual MLP, the GPU memory footprint is significantly reduced compared to direct RL fine-tune.

Peak VRAM ~5BG per task during **online RL training** (π_0 inference).

CPU System RAM for replay buffer up to 100GB per task

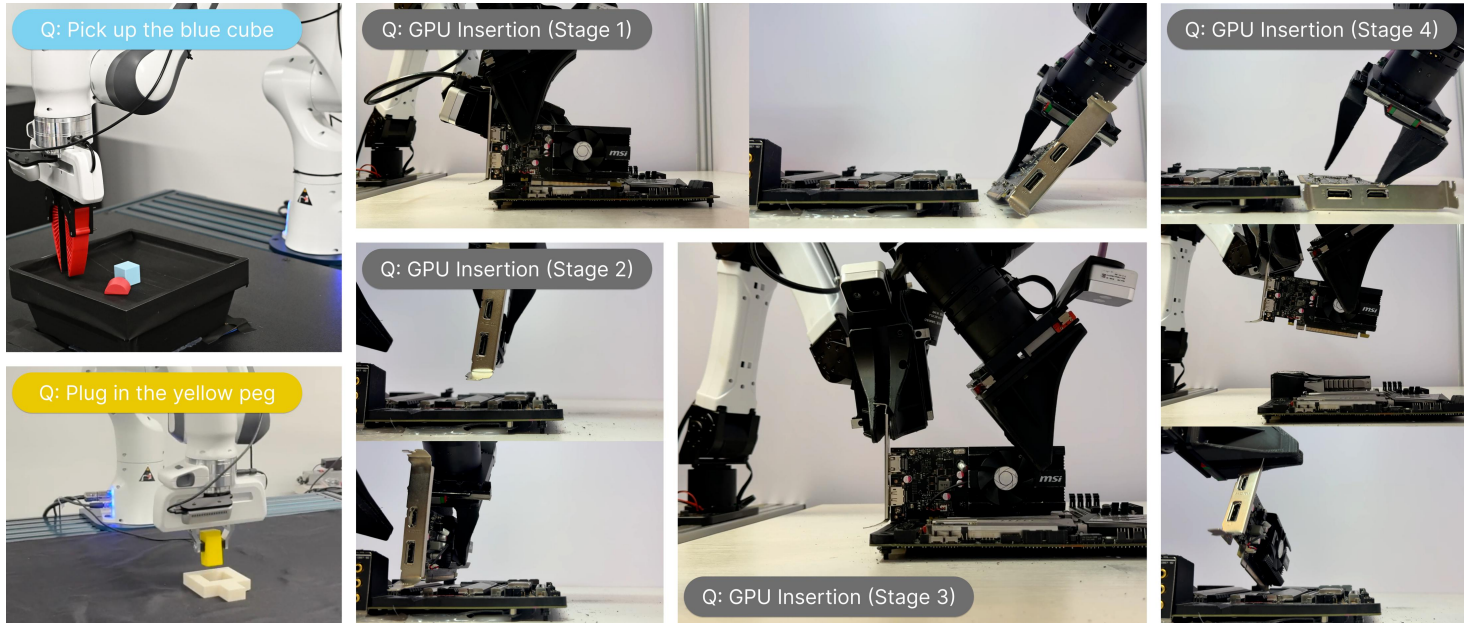
Linear scalability for multi-task learning: We successfully parallelized the LIBERO-90 experiment by distributing 90 tasks across a cluster node with **90 L40 GPUs** and **10TB CPU memory**.

Pipeline implemented with **JAX**, accelerating using Just-in-time. We can easily deploy it on consumer-level compute (single 4090).

Experiments

Real-World Experiments

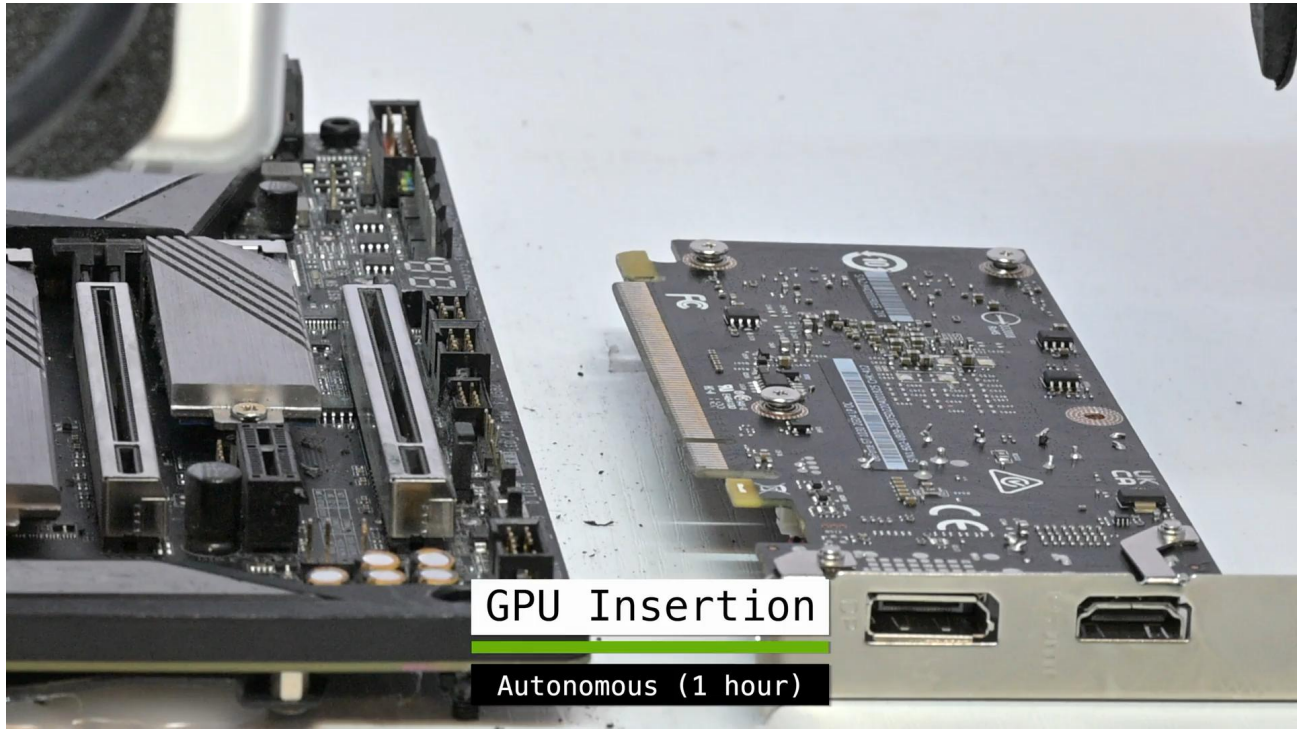
We deploy PLD on a 7-DoF Franka Emika Panda and 6-dom YAM ARM with end-effector delta pose control at 20~Hz.



*Pease check out our website: https://www.wenlixiao.com/self-improve-VLA-PLD

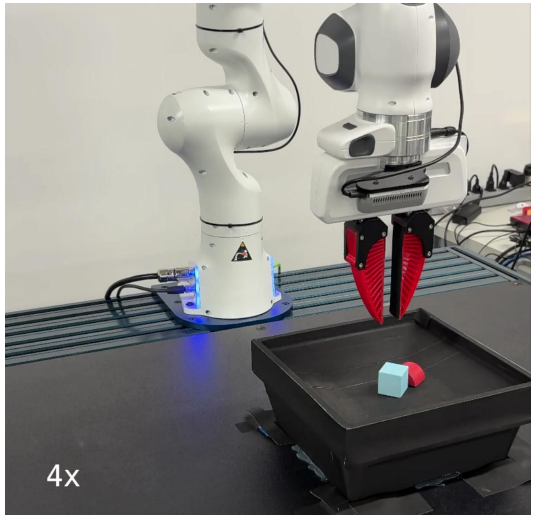
Experiments

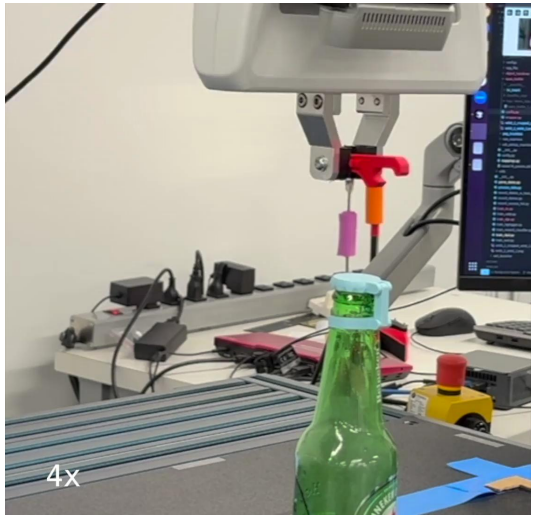
Real-World Experiments



Experiments

Real-World Experiments





PLD improves data diversity by capuring **recovery behavior** that are neighter available in Human teleop data nor base-policy rollout data.

PLD can scale to long-horizon multi-stage tasks, the distilled action-chunking policy preserves reactivity and dexterity of the closed loop RL expert.

*Pease check out our website: https://www.wenlixiao.com/self-improve-VLA-PLD

Conclusion

- We propose **PLD**, a three-stage post-training pipeline that enables VLA models to improve autonomously without relying on additional oracle human demonstrations. PLD has the potential of a self-improving data flywheel.
- Across large-scale simulation experiments and real-world deployment, PLD improves without additional human demonstration, achieving near-saturated ~99% success on LIBERO.
- Ablations identify residual policy probing and distribution-aware replay as key to stability, sample efficiency, and generalization.


Q&A