组会分享
Pi0.6star & Gen-0 & RLinf

李佩泽 2025-11-22

$\pi^*_{0.6}$: a VLA That Learns From Experience

Physical Intelligence

Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Kevin Black, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Jared DiCarlo, Danny Driess, Michael Equi, Adnan Esmail, Yunhao Fang, Chelsea Finn, Catherine Glossop, Thomas Godden, Ivan Goryachev, Lachy Groom, Hunter Hancock, Karol Hausman, Gashon Hussein, Brian Ichter, Szymon Jakubczak, Rowan Jen, Tim Jones, Ben Katz, Liyiming Ke, Chandra Kuchi, Marinda Lamb, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Yao Lu, Vishnu Mano, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Charvi Sharma, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, Will Stoeckle, Alex Swerdlow, James Tanner, Marcel Torne, Quan Vuong, Anna Walling, Haohuan Wang, Blake Williams, Sukwon Yoo, Lili Yu, Ury Zhilinsky, Zhiyuan Zhou

https://pi.website/blog/pistar06

***Abstract*—We study how vision-language-action (VLA) models can improve through real-world deployments via reinforcement learning (RL). We present a general-purpose method, RL with Experience and Corrections via Advantage-conditioned Policies (RECAP), that provides for RL training of VLAs via advantage conditioning. Our method incorporates heterogeneous data into the self-improvement process, including demonstrations, data from on-policy collection, and expert teleoperated interventions provided during autonomous execution. RECAP starts by pre-training a generalist VLA with offline RL, which we call $\pi^*_{0.6}$, that can then be specialized to attain high performance on downstream tasks through on-robot data collection. We show that the $\pi^*_{0.6}$ model trained with the full RECAP method can fold laundry in real homes, reliably assemble boxes, and make espresso drinks using a professional espresso machine. On some of the hardest tasks, RECAP more than doubles task throughput and roughly halves the task failure rate.**

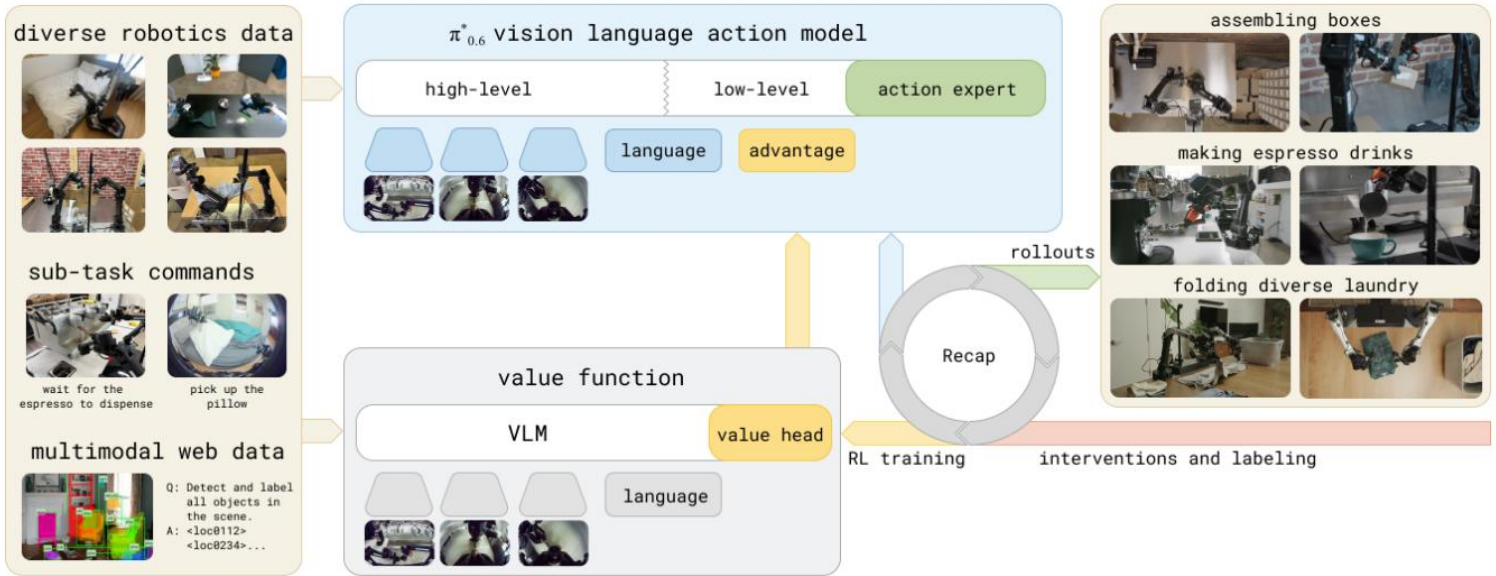


Fig. 1: **RECAP enables training VLAs with reward feedback and interventions.** Our system starts with a pre-trained VLA that incorporates *advantage conditioning*, allowing the model to learn effectively from real-world experience. For each task, we deploy the model and collect both autonomous rollouts and online human corrections. We then fine-tune the value function on this online data, improving its estimates of how actions influence performance. Fine-tuning and conditioning the VLA on these updated advantage estimates in turn improves policy behavior.

模型结构与伪代码

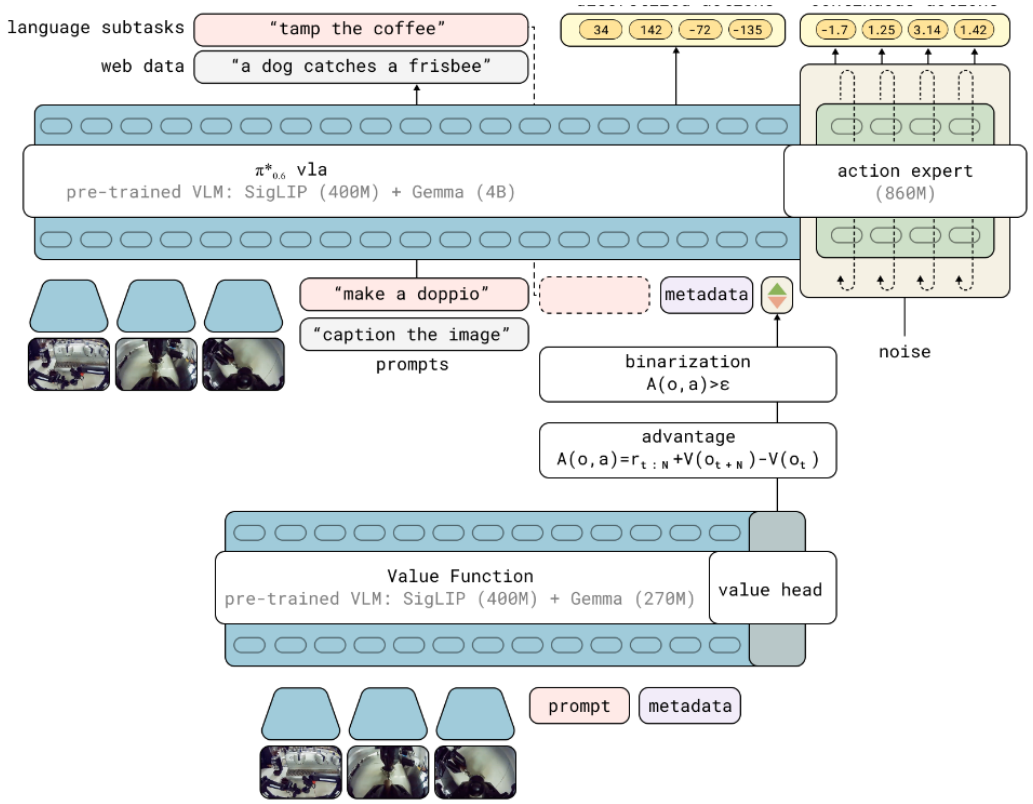


Fig. 3: **Interaction between the $\pi^*_{0.6}$ VLA and value function during RECAP training.** The $\pi^*_{0.6}$ VLA uses a pre-trained VLM backbone. Training follows the KI recipe [73], with next-token prediction on many data sources in pre-training, and an flow-matching action-expert with stop gradient. The VLA is conditioned on a binarized advantage indicator, obtained from a separate value function initialized from a pre-trained but smaller VLM model.

Algorithm 1 RL with Experience and Corrections via Advantage-conditioned Policies (RECAP)

Require: multi-task demonstration dataset $\mathcal{D}_{\text{demo}}$

Train V_{pre} on $\mathcal{D}_{\text{demo}}$ using Eq. 1
Train π_{pre} on $\mathcal{D}_{\text{demo}}$ using Eq. 3 and V_{pre}
Initialize $\mathcal{D}_\ell$ with demonstrations for ℓ
Train V_ℓ^0 from V_{pre} on $\mathcal{D}_\ell$ using Eq. 1
Train π_ℓ^0 from π_{pre} on $\mathcal{D}_\ell$ using Eq. 3 and V_ℓ^0
for $k = 1$ to K **do**
    Collect data with π_ℓ^{k-1}, add it to $\mathcal{D}_\ell$
    Train V_ℓ^k from V_{pre} on $\mathcal{D}_\ell$ using Eq. 1
    Train π_ℓ^k from π_{pre} on $\mathcal{D}_\ell$ using Eq. 3 and V_ℓ^k
end for

模型结构与伪代码

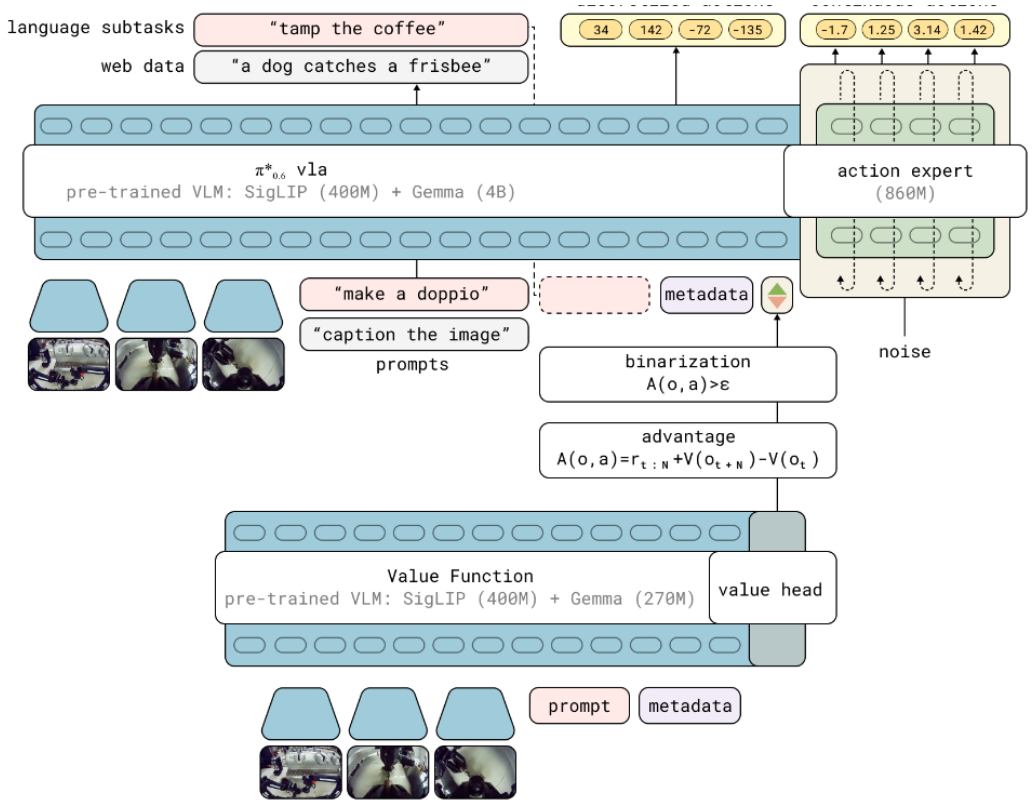
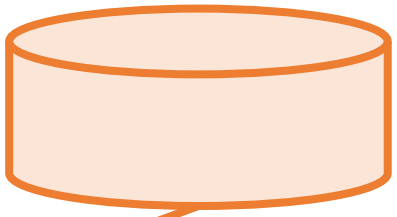


Fig. 3: **Interaction between the $\pi^*_{0.6}$ VLA and value function during RECAP training.** The $\pi^*_{0.6}$ VLA uses a pre-trained VLM backbone. Training follows the KI recipe [73], with next-token prediction on many data sources in pre-training, and an flow-matching action-expert with stop gradient. The VLA is conditioned on a binarized advantage indicator, obtained from a separate value function initialized from a pre-trained but smaller VLM model.



Algorithm 1 RL with Experience and Corrections via Advantage-conditioned Policies (RECAP)

Require: multi-task demonstration dataset $\mathcal{D}_{\text{demo}}$

Train V_{pre} on $\mathcal{D}_{\text{demo}}$ using Eq. 1
Train π_{pre} on $\mathcal{D}_{\text{demo}}$ using Eq. 3 and V_{pre}
Initialize $\mathcal{D}_\ell$ with demonstrations for ℓ
Train V_ℓ^0 from V_{pre} on $\mathcal{D}_\ell$ using Eq. 1
Train π_ℓ^0 from π_{pre} on $\mathcal{D}_\ell$ using Eq. 3 and V_ℓ^0
for $k = 1$ to K **do**
    Collect data with π_ℓ^{k-1}, add it to $\mathcal{D}_\ell$
    Train V_ℓ^k from V_{pre} on $\mathcal{D}_\ell$ using Eq. 1
    Train π_ℓ^k from π_{pre} on $\mathcal{D}_\ell$ using Eq. 3 and V_ℓ^k
end for

模型结构与伪代码

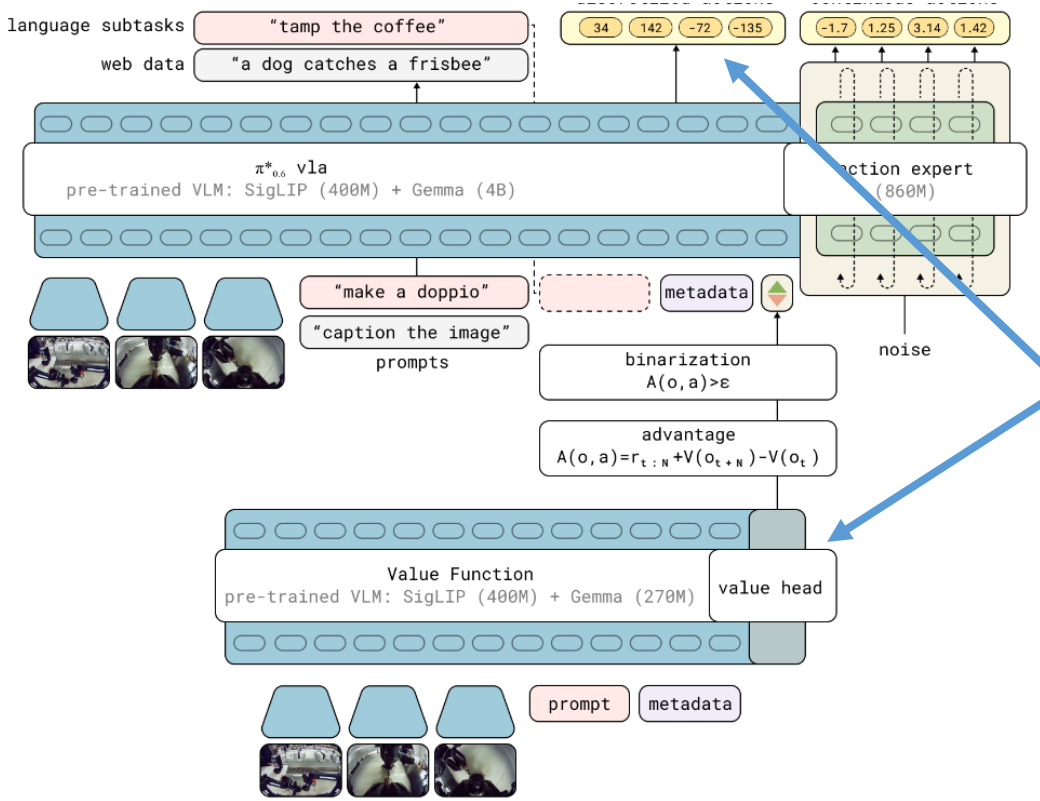


Fig. 3: **Interaction between the $\pi^*_{0.6}$ VLA and value function during RECAP training.** The $\pi^*_{0.6}$ VLA uses a pre-trained VLM backbone. Training follows the KI recipe [73], with next-token prediction on many data sources in pre-training, and an flow-matching action-expert with stop gradient. The VLA is conditioned on a binarized advantage indicator, obtained from a separate value function initialized from a pre-trained but smaller VLM model.

Algorithm 1 RL with Experience and Corrections via Advantage-conditioned Policies (RECAP)

Require: multi-task demonstration dataset $\mathcal{D}_{\text{demo}}$

Train V_{pre} on $\mathcal{D}_{\text{demo}}$ using Eq. 1
Train π_{pre} on $\mathcal{D}_{\text{demo}}$ using Eq. 3 and V_{pre}
Initialize $\mathcal{D}_\ell$ with demonstrations for ℓ
Train V_ℓ^0 from V_{pre} on $\mathcal{D}_\ell$ using Eq. 1
Train π_ℓ^0 from π_{pre} on $\mathcal{D}_\ell$ using Eq. 3 and V_ℓ^0
for $k = 1$ to K **do**
    Collect data with π_ℓ^{k-1}, add it to $\mathcal{D}_\ell$
    Train V_ℓ^k from V_{pre} on $\mathcal{D}_\ell$ using Eq. 1
    Train π_ℓ^k from π_{pre} on $\mathcal{D}_\ell$ using Eq. 3 and V_ℓ^k
end for

模型结构与伪代码

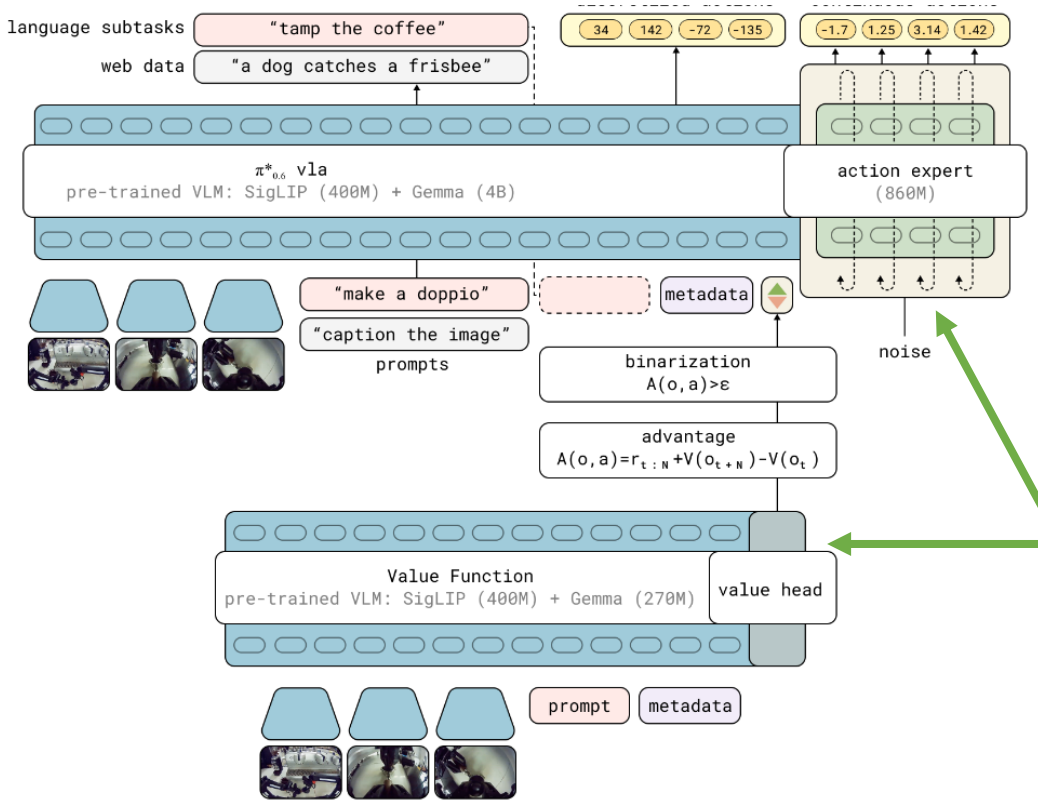


Fig. 3: **Interaction between the $\pi^*_{0.6}$ VLA and value function during RECAP training.** The $\pi^*_{0.6}$ VLA uses a pre-trained VLM backbone. Training follows the KI recipe [73], with next-token prediction on many data sources in pre-training, and an flow-matching action-expert with stop gradient. The VLA is conditioned on a binarized advantage indicator, obtained from a separate value function initialized from a pre-trained but smaller VLM model.

Algorithm 1 RL with Experience and Corrections via Advantage-conditioned Policies (RECAP)

Require: multi-task demonstration dataset $\mathcal{D}_{\text{demo}}$

Train V_{pre} on $\mathcal{D}_{\text{demo}}$ using Eq. 1
Train π_{pre} on $\mathcal{D}_{\text{demo}}$ using Eq. 3 and V_{pre}
Initialize $\mathcal{D}_\ell$ with demonstrations for ℓ
Train V_ℓ^0 from V_{pre} on $\mathcal{D}_\ell$ using Eq. 1
Train π_ℓ^0 from π_{pre} on $\mathcal{D}_\ell$ using Eq. 3 and V_ℓ^0
for $k = 1$ to K **do**
    Collect data with π_ℓ^{k-1}, add it to $\mathcal{D}_\ell$
    Train V_ℓ^k from V_{pre} on $\mathcal{D}_\ell$ using Eq. 1
    Train π_ℓ^k from π_{pre} on $\mathcal{D}_\ell$ using Eq. 3 and V_ℓ^k
end for

模型结构与伪代码

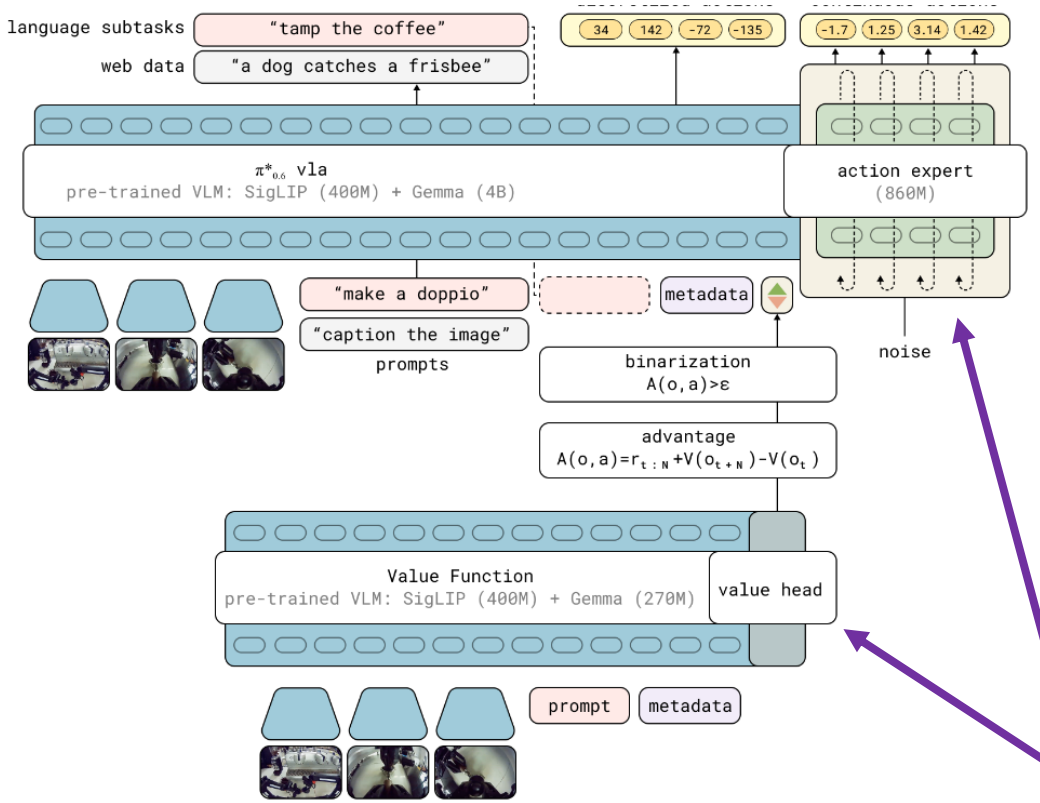
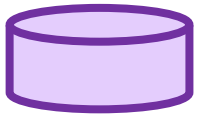


Fig. 3: **Interaction between the $\pi^*_{0.6}$ VLA and value function during RECAP training.** The $\pi^*_{0.6}$ VLA uses a pre-trained VLM backbone. Training follows the KI recipe [73], with next-token prediction on many data sources in pre-training, and an flow-matching action-expert with stop gradient. The VLA is conditioned on a binarized advantage indicator, obtained from a separate value function initialized from a pre-trained but smaller VLM model.



Algorithm 1 RL with Experience and Corrections via Advantage-conditioned Policies (RECAP)

Require: multi-task demonstration dataset $\mathcal{D}_{\text{demo}}$

Train V_{pre} on $\mathcal{D}_{\text{demo}}$ using Eq. 1
Train π_{pre} on $\mathcal{D}_{\text{demo}}$ using Eq. 3 and V_{pre}
Initialize $\mathcal{D}_\ell$ with demonstrations for ℓ
Train V_ℓ^0 from V_{pre} on $\mathcal{D}_\ell$ using Eq. 1
Train π_ℓ^0 from π_{pre} on $\mathcal{D}_\ell$ using Eq. 3 and V_ℓ^0
for $k = 1$ to K **do**
  Collect data with π_ℓ^{k-1}, add it to $\mathcal{D}_\ell$
  Train V_ℓ^k from V_{pre} on $\mathcal{D}_\ell$ using Eq. 1
  Train π_ℓ^k from π_{pre} on $\mathcal{D}_\ell$ using Eq. 3 and V_ℓ^k
end for

模仿学习训练

Algorithm 1 RL with Experience and Corrections via Advantage-conditioned Policies (RECAP)

Require: multi-task demonstration dataset $\mathcal{D}_{\text{demo}}$

Train V_{pre} on $\mathcal{D}_{\text{demo}}$ using Eq. 1
Train π_{pre} on $\mathcal{D}_{\text{demo}}$ using Eq. 3 and V_{pre}
Initialize $\mathcal{D}_\ell$ with demonstrations for ℓ
Train V_ℓ^0 from V_{pre} on $\mathcal{D}_\ell$ using Eq. 1
Train π_ℓ^0 from π_{pre} on $\mathcal{D}_\ell$ using Eq. 3 and V_ℓ^0
for $k = 1$ to K **do**
    Collect data with π_ℓ^{k-1}, add it to $\mathcal{D}_\ell$
    Train V_ℓ^k from V_{pre} on $\mathcal{D}_\ell$ using Eq. 1
    Train π_ℓ^k from π_{pre} on $\mathcal{D}_\ell$ using Eq. 3 and V_ℓ^k
end for

- Loss

在进行人为干预时 It置true

$$\min_\theta \mathbb{E}_{\mathcal{D}_{\pi_{\text{ref}}}}\Big[-\log \pi_\theta(\mathbf{a}_t|\mathbf{o}_t,\ell) - \alpha \log \pi_\theta(\mathbf{a}_t|I_t,\mathbf{o}_t,\ell)\Big], \quad (3)$$
$$\text{where } I_t = \mathbb{1}\big(A^{\pi_{\text{ref}}}(\mathbf{o}_t,\mathbf{a}_t,\ell) > \epsilon_\ell\big).$$

1. 对(s,a)对的学习；
2. 针对Advantage标注学习。

$$\log \pi_\theta\big(\mathbf{a}_{t:t+H}, a^\ell_{t:t+H}, \hat{\ell}|\mathbf{o}_t,\ell\big) = \log \pi_\theta\big(\hat{\ell}|\mathbf{o}_t,\ell\big) + \log \pi_\theta\big(a^\ell_{t:t+H}|\mathbf{o}_t,\ell,\hat{\ell}\big) + \log \pi_\theta\big(\mathbf{a}_{t:t+H}|\mathbf{o}_t,\ell,\hat{\ell}\big).$$

1. 子任务预测
2. FAST tokenizer输出的动作预测
3. 动作专家实际输出的动作预测

强化学习训练

Algorithm 1 RL with Experience and Corrections via Advantage-conditioned Policies (RECAP)

Require: multi-task demonstration dataset $\mathcal{D}_{\text{demo}}$

Train V_{pre} on $\mathcal{D}_{\text{demo}}$ using Eq. 1
Train π_{pre} on $\mathcal{D}_{\text{demo}}$ using Eq. 3 and V_{pre}
Initialize $\mathcal{D}_\ell$ with demonstrations for ℓ
Train V_ℓ^0 from V_{pre} on $\mathcal{D}_\ell$ using Eq. 1
Train π_ℓ^0 from π_{pre} on $\mathcal{D}_\ell$ using Eq. 3 and V_ℓ^0
for $k = 1$ to K **do**
    Collect data with π_ℓ^{k-1}, add it to $\mathcal{D}_\ell$
    Train V_ℓ^k from V_{pre} on $\mathcal{D}_\ell$ using Eq. 1
    Train π_ℓ^k from π_{pre} on $\mathcal{D}_\ell$ using Eq. 3 and V_ℓ^k
end for

- Reward、Return与Value

$$R(\tau) = \sum_{t=0}^{T} r_t$$

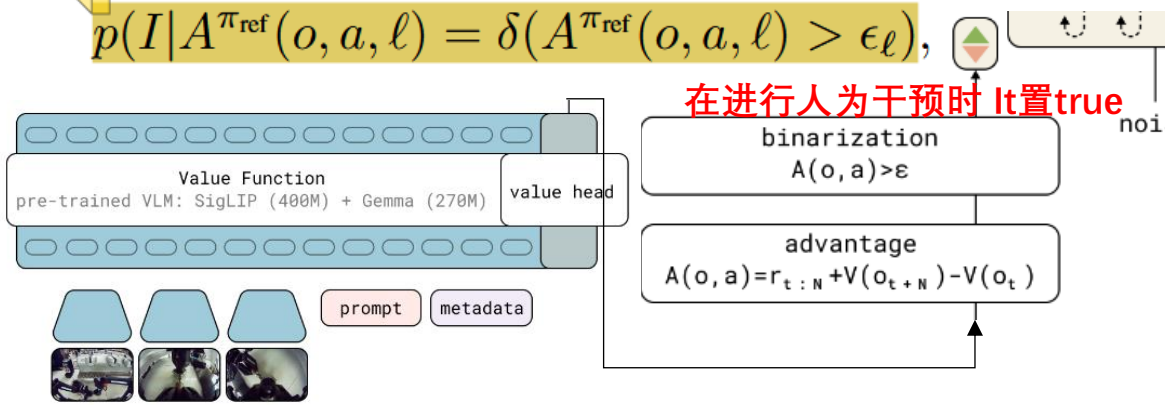
$$r_t = \begin{cases} 0 & \text{if t = T and success} \\ -C_{\text{fail}} & \text{if t = T and failure} \\ -1 & \text{otherwise.} \end{cases} \quad (5)$$

$$\min_\phi \mathbb{E}_{\tau \in \mathcal{D}} \left[\sum_{\mathbf{o}_t \in \tau} H(R_t^B(\tau), p_\phi(V|\mathbf{o}_t, \ell)) \right]. \quad (1)$$

- 训练方法

$$\hat{\pi}(\mathbf{a}, |\mathbf{o}, \ell) \propto \pi_{\text{ref}}(\mathbf{a}|\mathbf{o}, \ell) \left(\frac{\pi_{\text{ref}}(\mathbf{a}|I, \mathbf{o}, \ell)}{\pi_{\text{ref}}(\mathbf{a}|\mathbf{o}, \ell)} \right)^\beta. \quad (2)$$

$$p(I|A^{\pi_{\text{ref}}}(o, a, \ell) = \delta(A^{\pi_{\text{ref}}}(o, a, \ell) > \epsilon_\ell),$$



实验评估

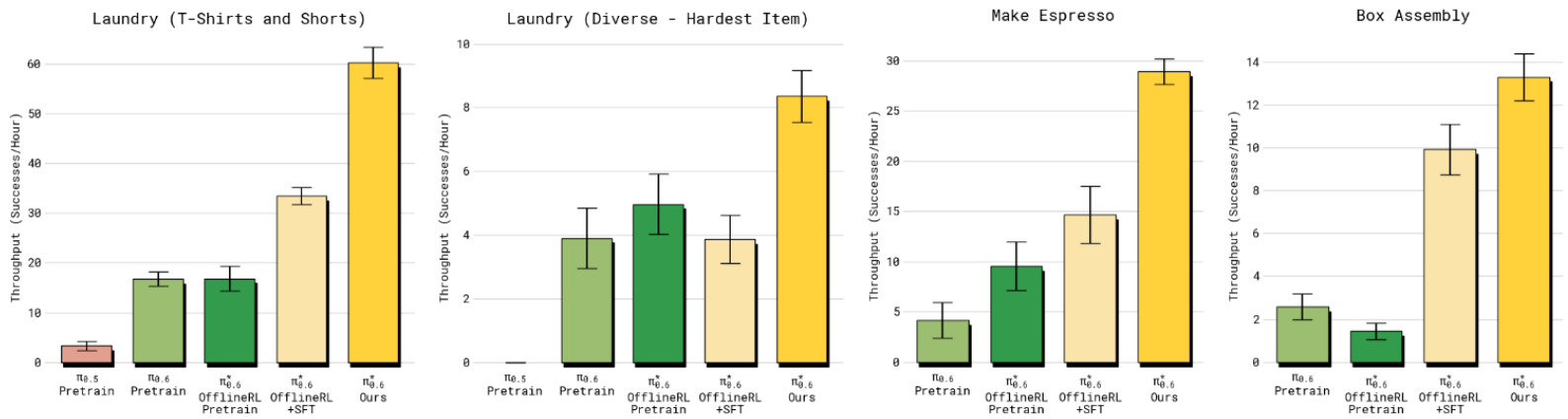


Fig. 7: **Throughput.** We show the number of successfully completed tasks *per hour* for laundry (simple and diverse), espresso making, and box assembly. Error bars show standard error. This metric measures both success and speed. In all cases, RECAP applied to $\pi^*_{0.6}$ (Ours) leads to substantial improvements in throughput. RECAP has the highest impact on throughput for diverse laundry and espresso tasks, more than doubling successful completions per hour.

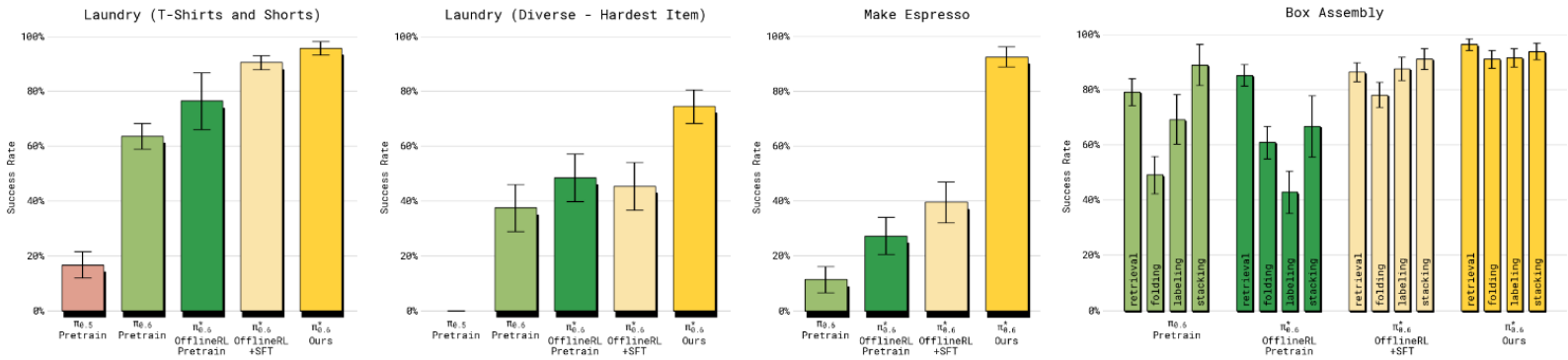


Fig. 8: **Success rates.** We show the absolute success rates with standard error. Each stage of RECAP improves performance across the tasks, with the challenging diverse laundry and espresso tasks seeing the largest gains success rate, corresponding to more than 2× reduction in failure rates. For the box assembly task we show the success rate for the different subtasks. RECAP leads to the most consistent (and highest) success across all subtasks.

实验评估

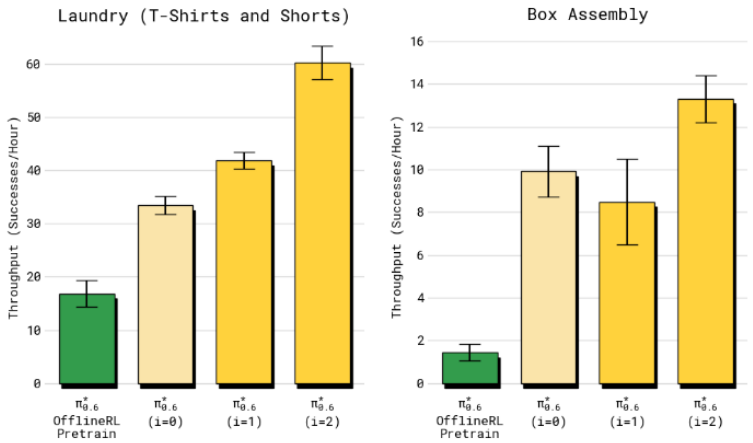


Fig. 9: **Improvement in throughput over multiple iterations.** Both tasks improve significantly in throughput as we take more iterations of RECAP, with box assembling first dropping and then improving significantly.

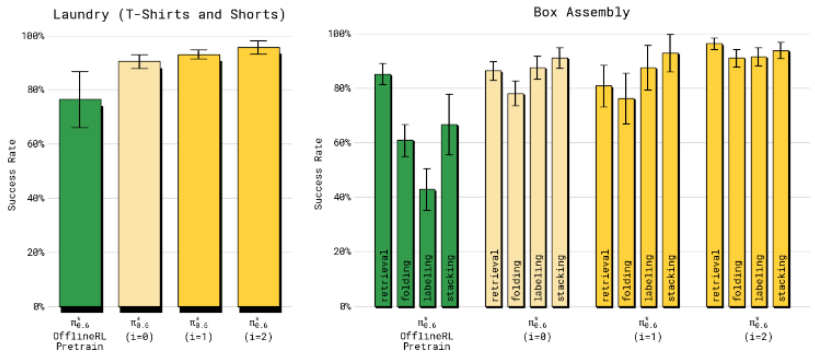


Fig. 10: **Improvement in success rate over multiple iterations.** The laundry task quickly reaches the maximum success rate (but continues to improve in throughput as shown in Figure 9, while box assembly continues to improve.

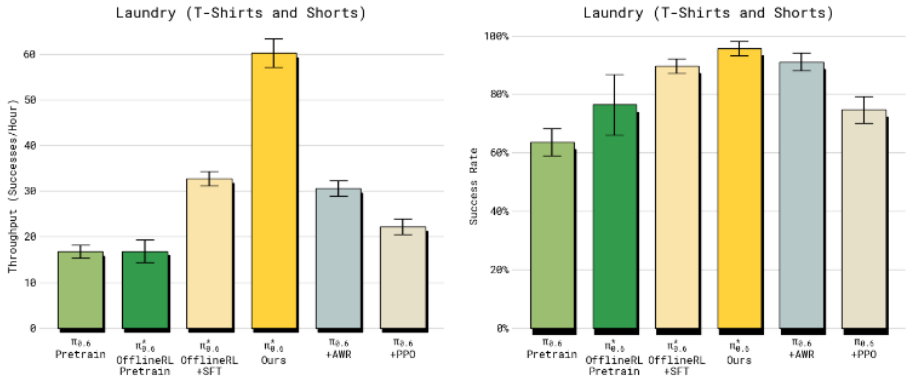


Fig. 11: **Comparison of different policy extraction methods.** RECAP applied to $\pi^*_{0.6}$ achieves by far the highest throughput for the laundry task compared to AWR and PPO.

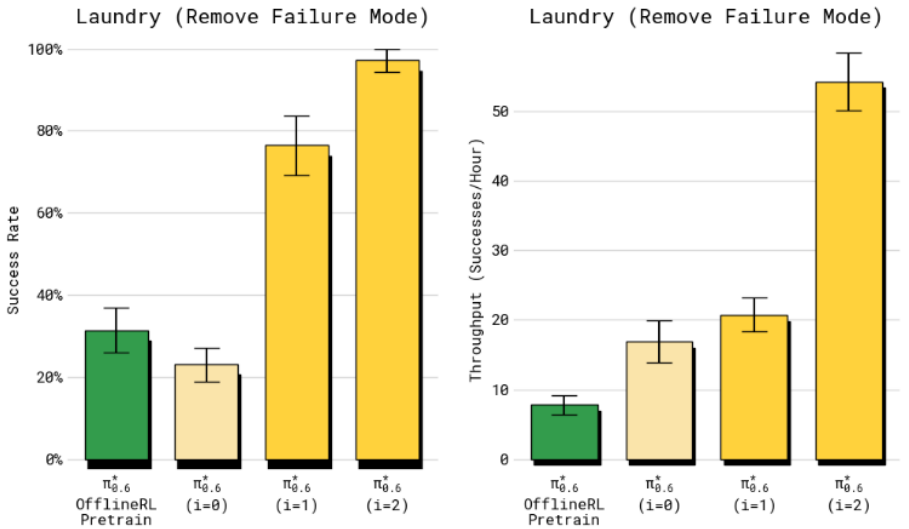


Fig. 12: **Failure mode removal.** Here we apply RECAP on a variant of the laundry task with one item but a very strict success criteria. RECAP is particularly effective at removing failure modes that would be considered non successful under the strict criteria. Therefore, our method can also be used to alter a policy's behavior with relatively little data effectively.

PI系列的发展轨迹

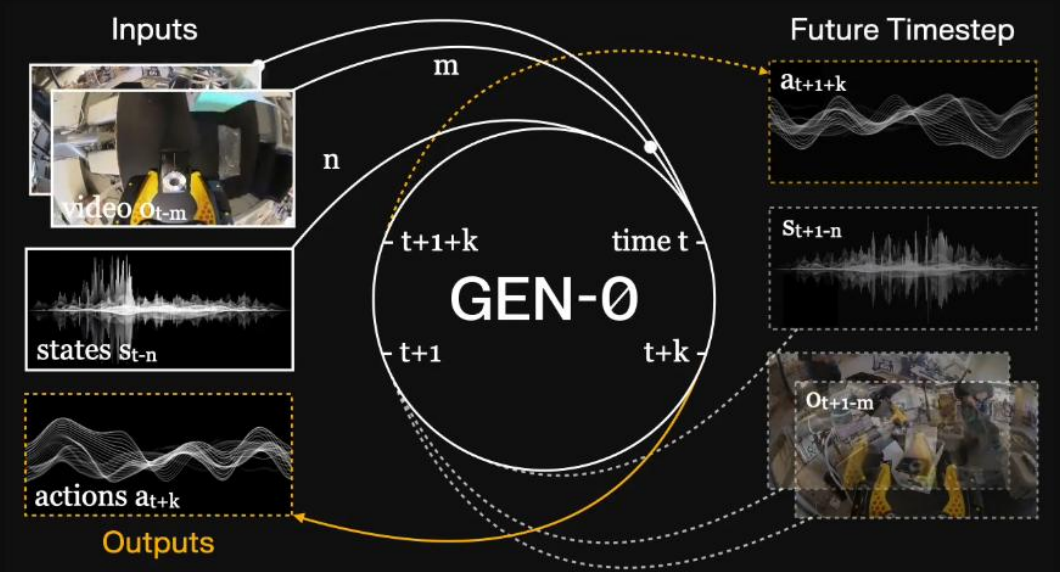
工作	时间	主要创新
PI-0	2024-11	基于Block-wise的VLM-AE连接 基于流匹配的AE动作生成
PI-FAST	2025-01	频域离散的动作编码加速监督学习的序列生成
PI-0.5	2025-04	异质数据监督学习预训练+微调
PI-0.6*	2025-11	人在回路的强化学习 扩大的模型参数

Blog / Research

GEN-0 / Embodied Foundation Models That Scale with Physical Interaction

Generalist AI Team

Nov 4, 2025



For years, foundation models in robotics have primarily used vision-language pretraining as the stepping stone towards scaling robotics, allowing us to transfer[1] the benefits of semantic generalization from existing large multimodal models. But what's been missing is how to effectively scale large multimodal model training in the domain of robotics itself—to establish scaling laws that

[1] PaLM-E: An Embodied Multimodal Language Model (Driess et al., 2023)

Gen0

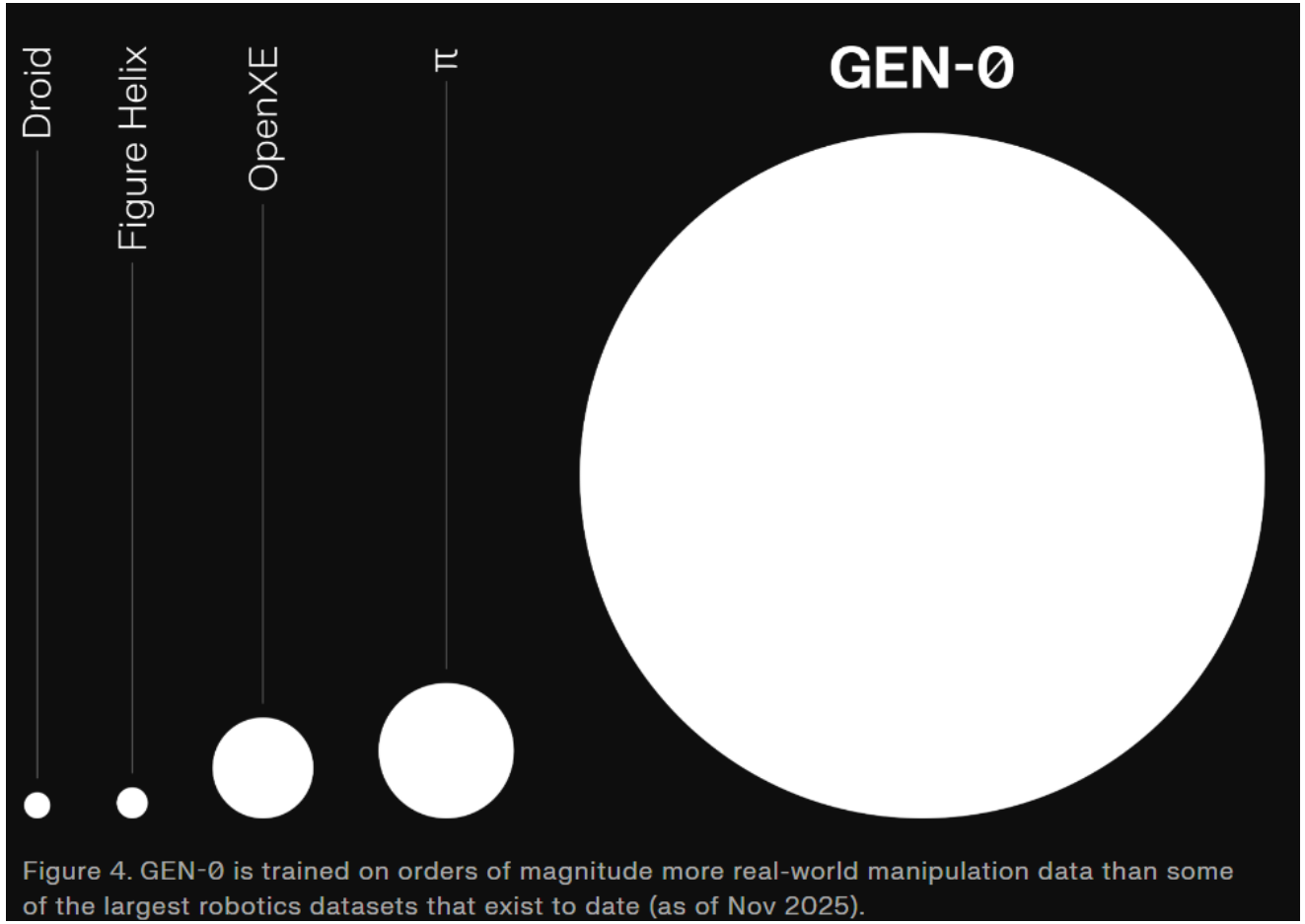
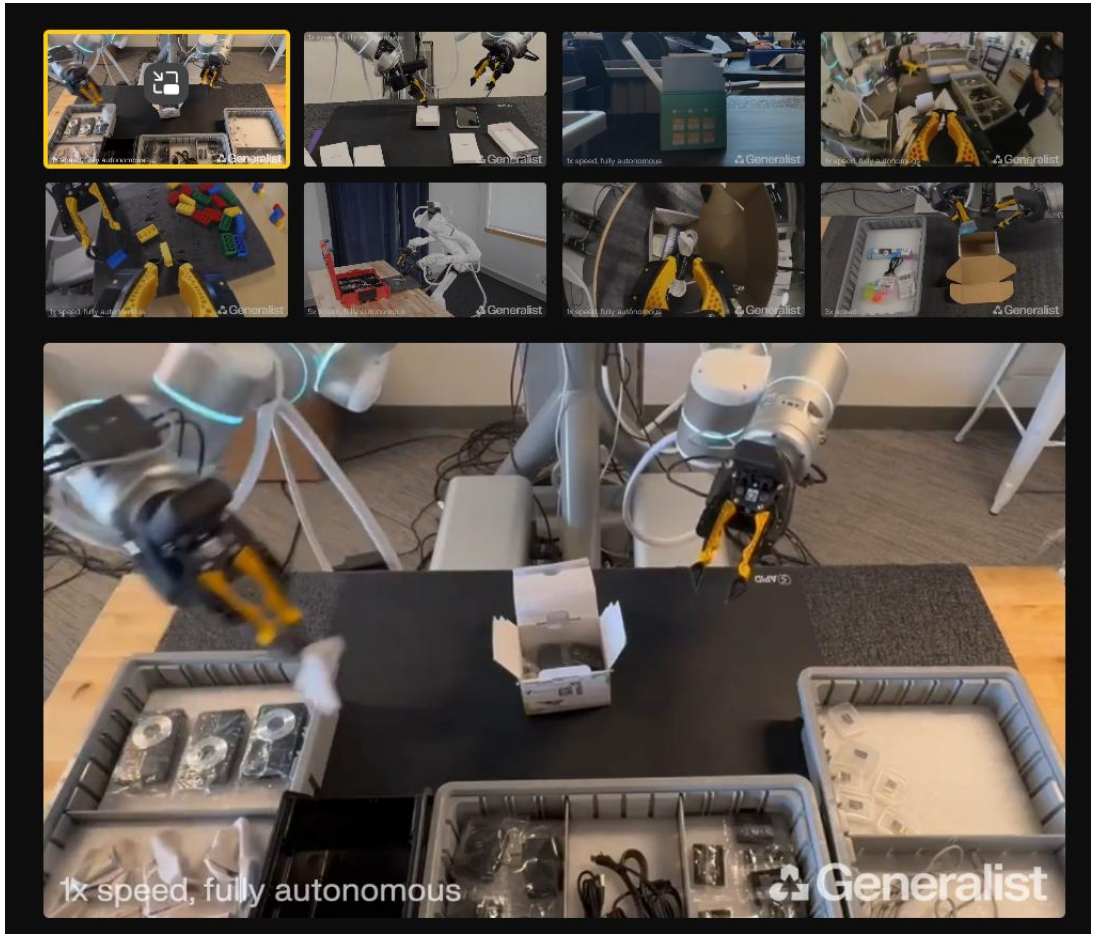


Figure 4. GEN-0 is trained on orders of magnitude more real-world manipulation data than some of the largest robotics datasets that exist to date (as of Nov 2025).



Gen0

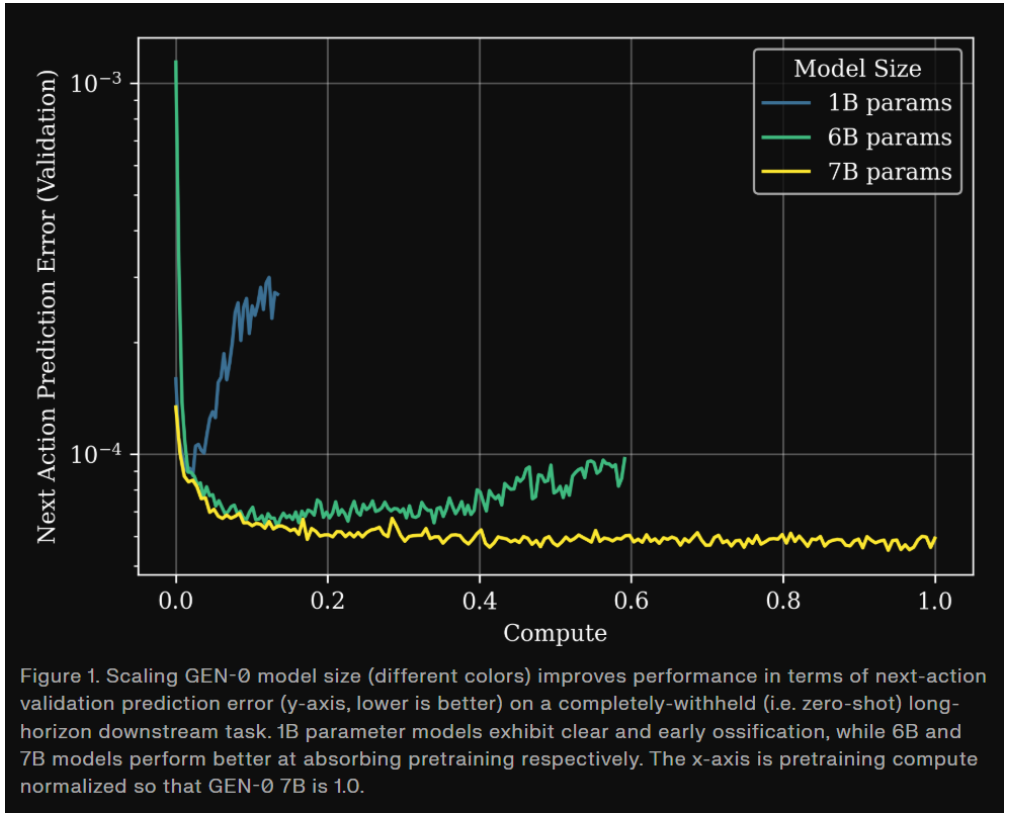
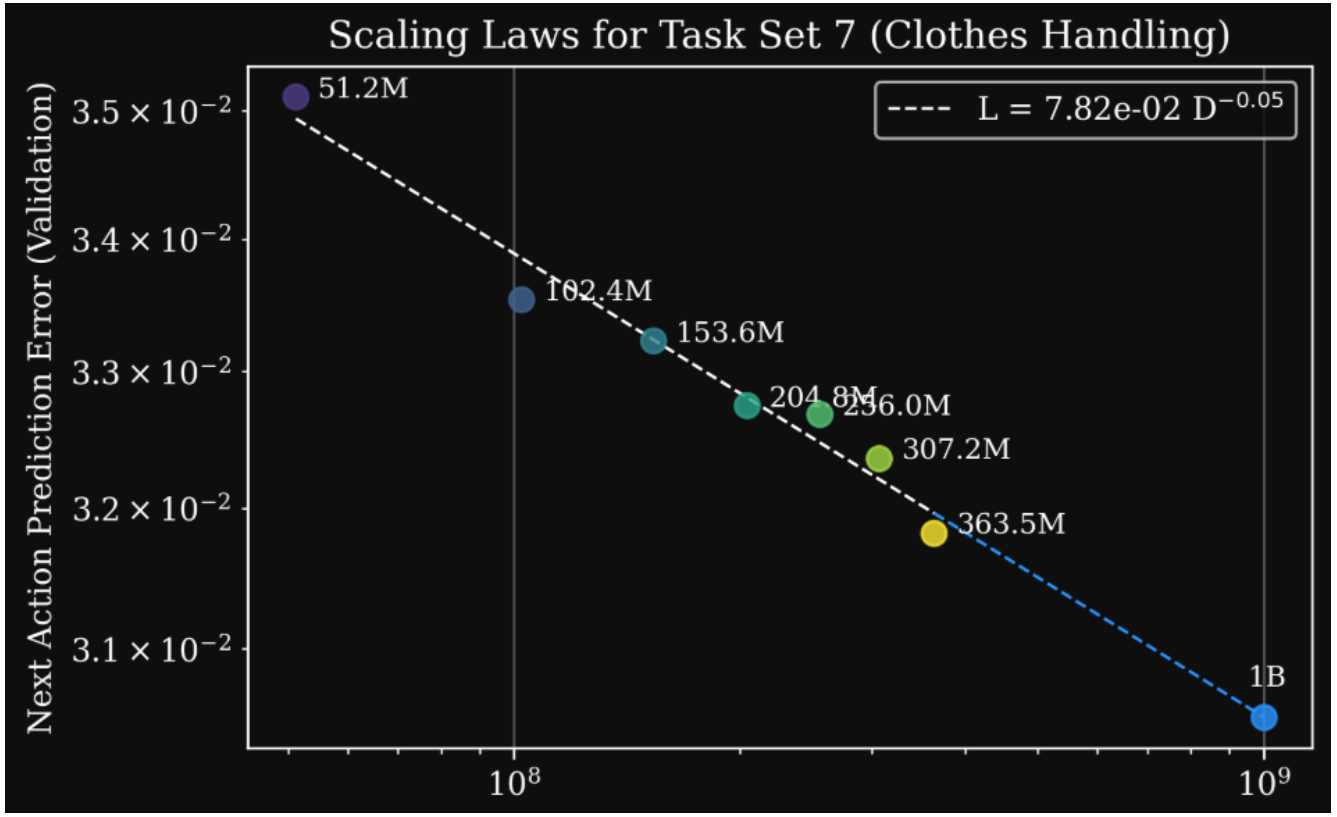
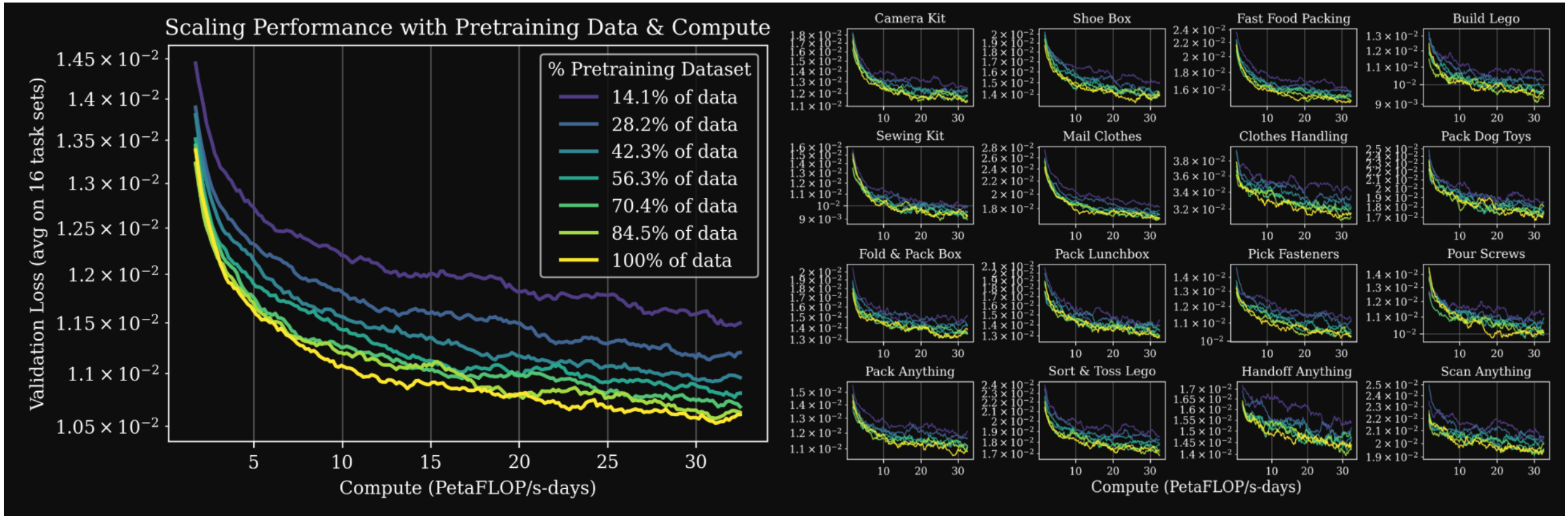


Figure 1. Scaling GEN-0 model size (different colors) improves performance in terms of next-action validation prediction error (y-axis, lower is better) on a completely-withheld (i.e. zero-shot) long-horizon downstream task. 1B parameter models exhibit clear and early ossification, while 6B and 7B models perform better at absorbing pretraining respectively. The x-axis is pretraining compute normalized so that GEN-0 7B is 1.0.



Gen0


Scaling Performance with Pretraining Data & Compute
Validation Loss (avg on 16 task sets)
Compute (PetaFLOP/s-days)
% Pretraining Dataset
14.1% of data
28.2% of data
42.3% of data
56.3% of data
70.4% of data
84.5% of data
100% of data
Camera Kit
Shoe Box
Fast Food Packing
Build Lego
Sewing Kit
Mail Clothes
Clothes Handling
Pack Dog Toys
Fold & Pack Box
Pack Lunchbox
Pick Fasteners
Pour Screws
Pack Anything
Sort & Toss Lego
Handoff Anything
Scan Anything
Compute (PetaFLOP/s-days)

RLinf: Flexible and Efficient Large-scale Reinforcement Learning via Macro-to-Micro Flow Transformation

Chao Yu[12], Yuanqing Wang[34], Zhen Guo[3], Hao Lin[3], Si Xu[3], Hongzhi Zang[1], Quanlu Zhang[3], Yongji Wu[5], Chunyang Zhu[3], Junhao Hu[3], Zixiao Huang[1], Mingjie Wei[2], Yuqing Xie[1], Ke Yang[2], Bo Dai[6], Zhexuan Xu[1], Xiangyuan Wang[4], Xu Fu[3], Zhihao Liu[2], Kang Chen[42], Weilin Liu[3], Gang Liu[1], Boxun Li[3], Jianlei Yang[6], Zhi Yang[4], Guohao Dai[73], Yu Wang[1*]

[1]Tsinghua University [2]Zhongguancun Academy [3]Infinigence AI
[4]Peking University [5]UC Berkeley [6]Beihang University [7]Shanghai Jiaotong University
[*]Corresponding Author: yu-wang@tsinghua.edu.cn
GitHub Repo: https://github.com/RLinf/RLinf

Abstract

Reinforcement learning (RL) has demonstrated immense potential in advancing artificial general intelligence, agentic intelligence, and embodied intelligence. However, the inherent heterogeneity and dynamicity of RL workflows often lead to low hardware utilization and slow training on existing systems. In this paper, we present RLinf, a high-performance RL training system based on our key observation that the major roadblock to efficient RL training lies in *system flexibility*. To maximize flexibility and efficiency, RLinf is built atop a novel RL system design paradigm called *macro-to-micro flow transformation* (M2Flow), which automatically breaks down high-level, easy-to-compose RL workflows at both the temporal and spatial dimensions, and recomposes them into optimized execution flows. Supported by RLinf worker's adaptive communication capability, we devise *context switching* and *elastic pipelining* to realize M2Flow transformation, and a profiling-guided scheduling policy to generate optimal execution plans. Extensive evaluations on both reasoning RL and embodied RL tasks demonstrate that RLinf consistently outperforms state-of-the-art systems, achieving 1.1×∼2.13× speedup in end-to-end training throughput.