THE SURPRISING EFFECTIVENESS OF FEW-STEP ADAPTATION FOR ROBOTIC MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language-action (VLA) models are pretrained on large robot datasets and fine-tuned on hundreds of demonstrations per task, yet the natural way to teach a robot something new is to show it once. The prevailing answer for using that one demonstration is in-context learning, which retrains the policy to read demonstrations at inference. We ask whether something far simpler will do: keep the pretrained VLA as it is and update it with a few gradient steps of its own training loss on the single demonstration. We call this Few-step One-Shot Imitation Learning (FOSIL). It works, and surprisingly well. One gradient step suffices when a familiar skill meets a new object or scene, ten for skills composed in a new order, and thirty for skills the policy could not perform before. We show this on a two-arm robot, with every result backed by 20 physical trials per task over 250 hours of evaluation, and on LIBERO-PRO, where the same demonstration given in context to the same backbone does not improve on the unadapted model. We then ask why so few steps are enough: robot pretraining decides the speed of adaptation, the useful update lives in a single direction per weight matrix, and a demonstration that moves differently from the policy can be refined with the policy's own motion. A pretrained VLA, it turns out, is already a few gradient steps away from its next task. Physical robot rollouts and videos: https://fosilvla.github.io.

1 INTRODUCTION

Vision-language-action (VLA) models are pretrained on large robot datasets and then fine-tuned on hundreds of demonstrations of each new task. Yet the most natural way to teach a robot something new is to show it once: throw this heavier object, place the cup where the basket now is, hand the object across and close the drawer. What can a pretrained VLA learn from a single demonstration?

In this one-demonstration setting, the prevailing answer is in-context learning: retrain the policy so that it reads the demonstration at inference, as ICRT, RiCL, and Instant Policy do (Fu et al., 2025; Sridhar et al., 2025; Vosylius & Johns, 2025). Robot data makes this harder than it sounds: in most datasets the observation alone predicts the action, so a policy trained to read context can reach a low loss while ignoring it (Wen et al., 2020). Making context matter therefore requires a new architecture, a retraining run, and a training set designed so that the demonstration carries information the observation does not, and the resulting policy is costlier to train and to run, since every context frame is encoded at each call (App. D). We ask whether something simpler will do. **Can a pretrained VLA learn from one demonstration with nothing more than a few gradient steps?**

It can, and surprisingly well. We keep the VLA exactly as it is, architecture and objective, and update it with a few steps of its own training loss on the single demonstration; we call this Few-step One-Shot Imitation Learning (FOSIL). There is no meta-training, no context pathway, and no environment interaction, and the adapted policy runs at the cost of the base policy (App. D). The only preparation is to train the initialization on *pretext behaviors*, demonstrations of unrelated behaviors collected on the target robot, so that the one demonstration teaches a new behavior rather than a new embodiment. **With one to thirty gradient steps, pretrained VLAs learn unseen manipulation tasks from one demonstration, and do so more reliably than the same demonstration given in context to the same backbone.** Recent work on one-demonstration adaptation meta-trains for it or builds machinery for context (Bianchi et al., 2026; Fu et al., 2025; Sridhar et al., 2025); the plain

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

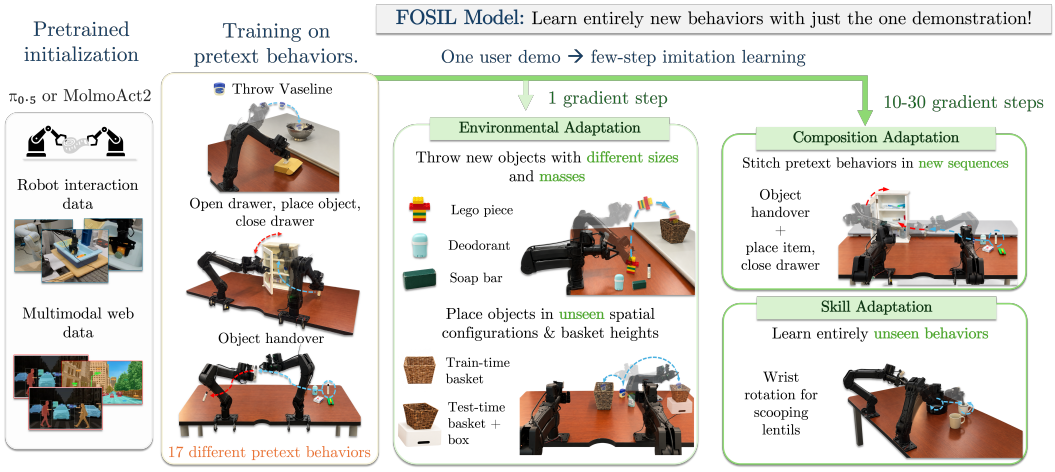


Figure 1: **FOSIL: one demonstration, a few gradient steps.** A pretrained VLA is trained on pretext behaviors, then adapted to a held-out task from one demonstration with its own training loss. What changes across tasks is their distance from the pretext behaviors, not the method: a familiar skill in a new setting takes one gradient step; a new sequence of skills takes ten; a skill the pretext behaviors do not contain, thirty.

route has not been examined on its own, and we find that it is not only competitive but also open to a principled analysis of why it works.

How far from the pretext behaviors can a task be? We organize tasks by that distance, following Gao et al. (2026), and Fig. 1 shows one task of each kind: *environmental adaptation* changes the object or layout but not the skill, such as throwing an object of a new weight; *compositional adaptation* sequences familiar skills in a new way; and *skill adaptation* asks for a skill the pretext behaviors do not contain, such as scooping or sweeping. One gradient step suffices for the first, ten for the second, and thirty for the third, skills the policy could not perform before the demonstration.

We test these findings on physical robots with extensive rigor. On a two-arm platform, every result rests on 20 trials per task and method, over 250 hours of evaluation in all. The LIBERO-PRO benchmark then lets us repeat the study densely, with the backbone held fixed: there, the same demonstration given in context to the same $\pi_{0.5}$ architecture leaves success where the unadapted model had it. Rollouts of every task are on the project webpage.

That so little suffices runs against the common wisdom that one demonstration is too little for gradient descent, so the second half of the paper asks what makes it possible. Our analysis yields three findings. First, the initialization decides the speed of adaptation: across four VLA initializations trained on the same pretext behaviors, those with robot pretraining gain the most from the first gradient step, and a controlled variant of $\pi_{0.5}$ with its action expert re-initialized barely moves. Robot pretraining supplies, for free, the initialization that meta-learning optimizes for. Second, we explain why: for pretrained VLAs a single direction per weight matrix carries almost the entire adaptation gain, and the update concentrates in a few layers of the backbone, whereas the same architecture without robot pretraining needs an order of magnitude more directions. A demonstration adjusts a direction the model already has rather than constructing one. Third, the demonstration itself bounds the gain: it can specify the right task while executing it as the policy never would, so a few steps fit the wrist or the pace instead of the task. Refining the demonstration with the policy's own motion, keeping only where the end effector goes and when the gripper acts, restores the gain without additional data. The same lens gives insight into the in-context results: in our experiments, those methods use the demonstration only when their training data contains tasks that make it necessary, and adding ten times more ordinary tasks does not help, whereas a handful of tasks designed to make context necessary does. Gradient steps need no such design, because the demonstration changes the parameters directly.

In summary, in the one-demonstration setting where in-context learning is the status quo, we (1) show the surprising finding that pretrained VLAs, with architecture and objective unchanged, learn environmental, compositional, and skill changes from one to thirty gradient steps, on hardware and

in simulation, more reliably than the same demonstration given in context; (2) explain it through the concentration of the adaptation update under robot pretraining, established across four initializations and by rank and layer analysis, and through the compatibility of the demonstration with the policy, which a refinement step improves without additional data; and (3) offer insight into when in-context imitation can use a demonstration at all, and why more data alone does not get it there.

2 Related Work

Pretrained VLA policies and their generalization. VLAs pair a pretrained vision-language backbone with an action decoder and train on large robot datasets, which gives them broad zero-shot competence and a sharp drop once the instruction, object, or scene leaves the training distribution (Intelligence et al., 2025; Fang et al., 2026; StarVLA Community, 2026). The standard remedy is fine-tuning on a new dataset, which transfers well but presumes data the deployment usually lacks (Kim et al., 2024). Benchmarks have moved from suites that current VLAs have largely solved to perturbed variants that isolate how a task changes, and taxonomies separate changes in observation, instruction, and required behavior (Liu et al., 2023; Zhou et al., 2025; Gao et al., 2026). These efforts measure what a frozen policy generalizes to. We build directly on them: FOSIL takes these pretrained VLAs as its initializations, adopts their perturbation axes and taxonomy to organize its tasks, and asks how much one demonstration recovers.

In-context imitation learning. The route that promises adaptation without parameter updates conditions the policy on demonstrations at inference, an idea that dates to one-shot imitation with context-conditioned policies (Duan et al., 2017). ICRT prompts a causal transformer with a demonstrated trajectory, RiCL retrieves the nearest demonstration frames and interpolates toward their actions, Instant Policy diffuses actions over a graph of demonstration and current observations, and Behavior Prompting encodes a whole trajectory as context for a diffusion policy (Fu et al., 2025; Sridhar et al., 2025; Vosylius & Johns, 2025; Patel et al., 2026). Each retrains the policy with a pathway for context, and each depends on training data in which the demonstration carries information the observation does not; otherwise the policy learns to ignore its context or copy it, a shortcut that behavioral cloning does not penalize (Wen et al., 2020). In contrast, FOSIL never trains the policy to read a demonstration; the demonstration enters through a few gradient steps on the policy's own loss, and we compare the two routes on one backbone by porting ICRT and RiCL to $\pi_{0.5}$. Whether a policy can adapt from so little supervision is the question meta-learning was built to answer.

Post-training VLAs from few demonstrations. Updating a model on the test input itself has a long history as test-time training, most recently by fine-tuning on the in-context examples of a few-shot prompt (Sun et al., 2020; 2024; Akyürek et al., 2025). For VLAs, the dominant few-demonstration route is reinforcement learning after a short supervised warm start, from sparse success rewards, human intervention, or online progress rewards (Tan et al., 2025; Chen et al., 2025a; Li et al., 2025; Lu et al., 2025; Chen et al., 2025b; Liu et al., 2026). Each of these buys its gains with rewarded environment interaction. Complementary to them, FOSIL is the supervised warm start they begin from, studied on its own: one demonstration, no interaction, and a fixed budget of gradient steps.

3 FOSIL: Few-Step One-Shot Imitation Learning

This section lays out what we run and why. We first describe the FOSIL protocol, which is deliberately minimal so that the only variables are the initialization, the pretext behaviors, and the number of gradient steps. We then describe the tasks, which we arrange by their distance from the pretext behaviors so that the results read as a ladder rather than a list. Finally, we describe the in-context baselines, which receive the same demonstration through a different route and let us ask, on one backbone, whether the demonstration is better used through the context or through the parameters.

3.1 Protocol

Initializations. FOSIL starts from a pretrained VLA and changes nothing about it. For our main results, we initialize from $\pi_{0.5}$ or MolmoAct2 (Intelligence et al., 2025; Fang et al., 2026), both of which pair a VLM backbone with a dedicated action expert trained with a continuous flow-matching objective. Because our analysis asks what robot pretraining contributes, we also use two

initializations without it: StarVLA-OFT and StarVLA-π (StarVLA Community, 2026), which start from pretrained VLMs and learn action prediction entirely from the pretext behaviors.

Training on pretext behaviors. A single demonstration cannot be asked to overcome a new embodiment, camera, and scene at the same time as a new behavior. We therefore train each initialization on pretext behaviors collected with the same robot, camera configuration, and environment as the tasks, with every task held out, so that the demonstration has only the behavior left to teach. The result is the FOSIL model. Before any task-specific update we call it the *unadapted model*: it is where every adaptation starts and the zero-shot reference against which every result is read.

Adaptation. For each task, we start from the unadapted model and update all parameters by minimizing the training objective on frames from the demonstration; with a $\pi_{0.5}$ initialization this is the flow-matching loss. We use AdamW with freshly initialized moments, discarding the optimizer state from training on pretext behaviors. Each gradient step uses a large batch of frames sampled uniformly in time over the demonstration, so the number of steps is the only quantity that changes with the adaptation budget. We call the success gained per gradient step the *speed of adaptation*; it is a property of the initialization and the demonstration, not of wall-clock time or compute. We select the learning rate per architecture from three candidates around the learning rate used to train the model on the pretext behaviors, reduce it when adaptation takes more than one gradient step (App. B), and apply the same recipe to every task and suite with no per-task tuning. We evaluate each reported cell over 20 physical trials per task on the robot, or 60 on throwing with 20 per object, and 50 episodes per task in simulation; App. B gives the remaining details.

3.2 Tasks

Real world. We collect 254 demonstrations across 17 pretext behaviors, including throwing an object into a basket, bimanual object handover, and opening a drawer, placing an object inside, and closing it (Fig. 7). None of these is a task we evaluate (videos of all 17 are on the project webpage). The tasks are built from them, by changing how the FOSIL model must execute a skill, by combining skills, or by requiring a skill that no pretext behavior contains. App. A lists all pretext behaviors and the dataset composition.

Adaptation categories. We group tasks by their relationship to the pretext behaviors, following the adaptation-gap terminology of Gao et al. (2026), and Fig. 2 shows a real-world example of each category from left to right. *Environmental adaptation* changes the instruction, goal, object, or spatial configuration while largely preserving the skill structure of a pretext behavior; this includes semantic changes, such as a different object or goal, and changes in object properties, such as mass or shape, that require different execution. In the left column, the FOSIL model must throw a new object into the basket, or place an object in a basket under a new spatial configuration, adjusting the motion while keeping the skill. *Compositional adaptation* combines skills that appear individually in the pretext behaviors but never together: in the middle column, the right arm hands an object to the left arm, which places it in the shelf and closes the shelf; the pretext behaviors contain the handover and the shelf placement, but never in sequence. *Skill adaptation* requires a skill absent from the pretext behaviors: in the right column, the FOSIL model must scoop and sweep, and none of the 17 pretext behaviors involve a spoon or a sweeping motion. We define skill novelty relative to the pretext behaviors, not to the initialization's pretraining data.

Simulation. Simulation lets us evaluate more densely than hardware allows and hold the backbone fixed across methods. We use LIBERO (Liu et al., 2023), a benchmark that current VLAs have largely solved, through LIBERO-PRO (Zhou et al., 2025), which perturbs its instructions, objects, and object positions so that success requires more than reproducing the training demonstrations. It was designed for out-of-distribution evaluation; here its perturbations define the tasks for one-shot adaptation. For environmental adaptation, we train on the pretext behaviors of each of the Object, Spatial, and Goal suites, collect a single demonstration in each perturbed setting, and focus on task perturbation and object swap, the two most demanding perturbations. For compositional adaptation, following previous work (Nguyen et al., 2026), we split LIBERO-90 into 75 pretext behaviors and 15 held-out tasks, six of which are short-horizon compositions of skills in the pretext behaviors, and report results on this subset; from the same unadapted model we also evaluate LIBERO-10, whose tasks are long-horizon compositions of the same skills. App. A describes the distribution shifts in these tasks.

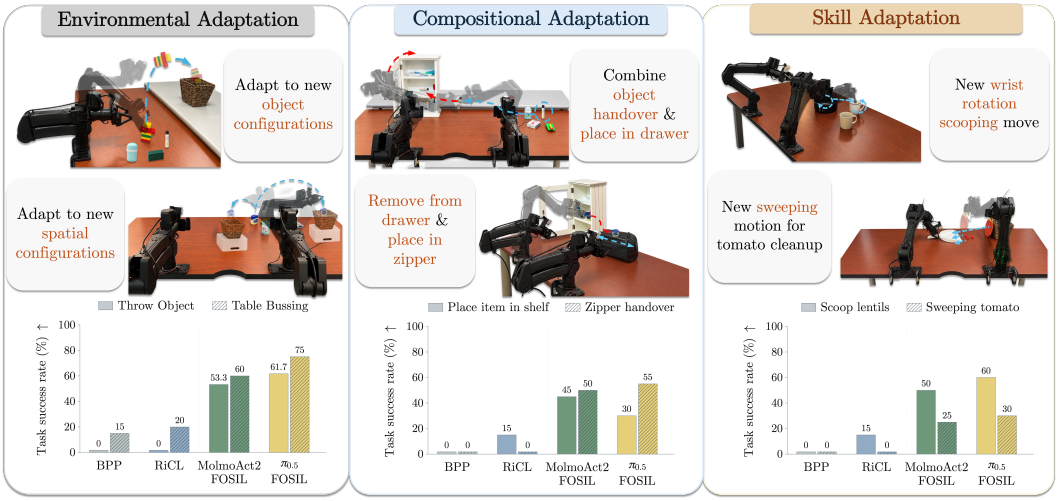


Figure 2: **Main real-world results.** Success over 20 rollouts per task (60 on throwing, 20 per object) with one demonstration per task, for BPP, RiCL ($\pi_{0.5}$), and FOSIL (green) from the MolmoAct2 and $\pi_{0.5}$ initializations, on the three adaptation categories of Fig. 7: environmental (left, one gradient step), compositional (middle, ten), and skill (right, thirty). Rollouts for every task and method are on the project webpage.

3.3 Baselines

Every baseline receives the same demonstration as FOSIL and is trained on the same pretext behaviors; the only difference is that the demonstration enters through the policy's context at inference rather than through its parameters (App. D compares the compute of the two routes). Alongside the unadapted model as the zero-shot reference, we compare against three in-context methods.

- **ICRT** (Fu et al., 2025) (Trajectory prompting): a causal transformer over interleaved observation and action tokens that predicts discrete action tokens by next-token prediction, conditioned on the demonstration and the current observation.
- **Retraining for In-Context Learning (RiCL)** (Sridhar et al., 2025) (Retrieval and interpolation): retrieves the demonstration frames nearest to the query as context and interpolates the predicted action toward the retrieved action with a weight that decays with retrieval distance.
- **Behavior Prompting Policy (BPP)** (Patel et al., 2026) (Trajectory encoding): encodes the complete demonstration into a context that conditions a diffusion model over actions.

ICRT and RiCL were built on earlier architectures, so we port both to $\pi_{0.5}$ and compare the two routes on one backbone; we call these ports ICL ($\pi_{0.5}$), which conditions on the single nearest demonstration frame, and RiCL ($\pi_{0.5}$), and App. B describes them. We also adapt rank-16 LoRA adapters (Hu et al., 2022) on $\pi_{0.5}$ for comparison against Parameter Efficient FineTuning (PEFT).

4 Main Results

We evaluate FOSIL on the three adaptation categories of Sec. 3 and ask whether a single demonstration is used better through the parameters or through the context. Tab. 1 reports the simulation results and Fig. 2 the real-world tasks.

One gradient step recovers environmental adaptation (Tab. 1, rows 1–2 vs. 9–10; Fig. 2). In the real world, throwing a new object requires a different wrist velocity, and table bussing changes the target object and the basket's placement; both build on pretext behaviors (Fig. 7). On throwing, one gradient step raises the success of FOSIL ($\pi_{0.5}$) from 23% to 62% across three objects, by adjusting the planned wrist speed to the new object, and on table bussing it reaches 75%; FOSIL (MolmoAct2) reaches 53% and 60%. The in-context methods stay at or below 20% on both tasks: BPP succeeds on no throw and on 15% of the bussing trials, RiCL on no throw and on 20%. In simulation, one gradient step roughly doubles the average success of $\pi_{0.5}$ on LIBERO-PRO task perturbation (25% $\rightarrow$ 49%) and raises it under object swap (20% $\rightarrow$ 34%), with gains on every suite; FOSIL (MolmoAct2) likewise improves on every reported suite.

Table 1: **Main simulation results.** Success rate (%) with one demonstration per task. Rows: unadapted models (gray), in-context methods (blue), rank-16 LoRA adaptation (uncolored), and FOSIL (green , ours), the last two with one gradient step, or ten on the compositional tasks (*). Bold and italic: best and second-best one-shot method per column.

		Task perturbation				Object swap				Compositional*	
#	Method	Object	Spatial	Goal	Avg.	Object	Spatial	Goal	Avg.	LIBERO-10	LIBERO-90
	Zero-shot (unadapted model, no demonstration)										
1	$\pi_{0.5}$ (Intelligence et al., 2025)	10	51	14	25	10	31	18	20	5	18
2	MolmoAct2 (Fang et al., 2026)	22	37	16	25	15	10	14	13	25	24
	One-shot (one demonstration per task)										
3	ICRT (Fu et al., 2025)	0	12	2	5	0	0	0	0	1	0
4	ICL [$\pi_{0.5}$ init]	18	30	14	21	16	24	15	18	26	53
5	RiCL [π_0-FAST init] (Sridhar et al., 2025)	0	16	24	13	*49*	4	22	25	12	33
6	RiCL [$\pi_{0.5}$ init]	17	42	16	25	22	18	18	19	19	41
7	BPP (Patel et al., 2026)	0	15	11	8	3	4	8	5	2	12
8	LoRA r16 [$\pi_{0.5}$ init]	10	*55*	15	27	10	**40**	14	21	37	46
9	FOSIL [$\pi_{0.5}$ init] *(ours)*	**52**	**58**	*36*	**49**	36	36	*30*	*34*	*52*	**66**
10	FOSIL [MolmoAct2 init] *(ours)*	*47*	47	**48**	*48*	**52**	*38*	**34**	**41**	**53**	*58*

The same demonstration is worth less as context than as a gradient step (Tab. 1, rows 4 and 6 vs. 9). ICL ($\pi_{0.5}$) and RiCL ($\pi_{0.5}$) share the initialization and pretext behaviors of FOSIL ($\pi_{0.5}$), so they isolate whether the demonstration enters through the context or through a gradient step. Under task perturbation, neither improves over the unadapted model on average (21% and 25%, against 25%), whereas FOSIL reaches 49%. ICRT and BPP, designed for in-context imitation, stay below 15% on every environmental column. We examine this behavior more deeply in Sec. 5.3.

The same gradient steps through LoRA adapters leave the model where it was (Tab. 1, row 8 vs. 9). Rank-16 LoRA gains little over the unadapted model (27% and 21% on average, versus 25% and 20%), even with its first step matched in norm to the full update. The bottleneck is initialization: with $B = 0$, the gradient with respect to A vanishes, so the first step updates B alone and the resulting update is constrained to the row space of the random A. A random rank-r subspace in d_{in} dimensions retains only an r/d_{in} fraction of the gradient's squared norm in expectation, i.e., a cosine of about $\sqrt{r/d_{\text{in}}}$; empirically, the cosine with the full update is just 0.03. Subsequent steps can move A, but only through the now-nonzero B, leaving the adapter poorly aligned with the useful update directions. Sec. 5.1 shows that FOSIL's update is concentrated in the leading singular direction.

Compositions need ten steps and keep improving (Tab. 1, last two columns; Fig. 3). In simulation we evaluate the compositional LIBERO-90 held-out tasks and the long-horizon LIBERO-10 tasks, both from a model trained on the 75 LIBERO-90 pretext behaviors (App. A). With ten gradient steps, FOSIL ($\pi_{0.5}$) reaches 66% on LIBERO-90 and 52% on LIBERO-10, against 53% and 26% for ICL ($\pi_{0.5}$), the strongest in-context baseline; FOSIL (MolmoAct2) reaches 58% and 53%. Unlike environmental adaptation, compositions need a few more steps, partly to locate the skills among the pretext behaviors and partly to combine them with robust transitions between skills. In the real world (Fig. 2, middle) we test two compositions. They reach 30% and 55% for FOSIL ($\pi_{0.5}$) and 45% and 50% for FOSIL (MolmoAct2) with ten gradient steps; RiCL reaches 15% on the shelf task and does not succeed on the zipper task, and BPP succeeds on neither.

Thirty steps teach a skill absent from the pretext behaviors (Fig. 2). Scooping and sweeping require motions that none of the real-world pretext behaviors contain, and need thirty gradient steps at a learning rate of 10^{-5} (App. B). With them, FOSIL ($\pi_{0.5}$) reaches 60% on scooping and 30% on sweeping, and FOSIL (MolmoAct2) 50% and 25%. BPP succeeds on neither skill, and RiCL reaches 15% on scooping and does not succeed on sweeping.

Speed of adaptation depends on robot pretraining (Fig. 3). Fig. 3 repeats these evaluations across initializations. FOSIL models from pretrained VLAs ($\pi_{0.5}$, MolmoAct2) gain more from the first gradient step than those from VLMs without VLA pretraining (StarVLA-π, StarVLA-OFT), and stay ahead even after ten. These initializations also differ in backbone, action representation, and data, so we isolate robot pretraining with $\pi_{0.5}$-VL: $\pi_{0.5}$ with its action expert re-initialized and trained on the same pretext behaviors. On the environmental tasks, one gradient step barely changes its success, whereas $\pi_{0.5}$ gains 27–42 points. We analyze this behavior further in our Sec. 5, as it provides us clean test-bed of comparing different initialization under same network architecture.

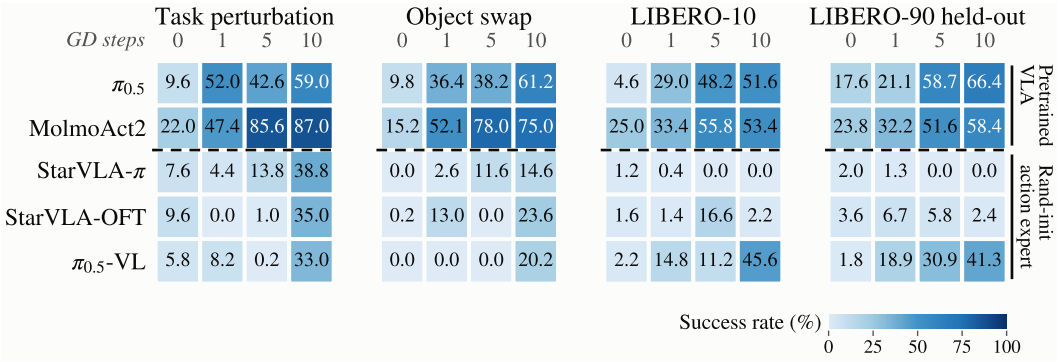


Figure 3: **Speed of adaptation depends on the initialization.** Success rate (%) as a function of the number of gradient steps on the single demonstration, for FOSIL models from the $\pi_{0.5}$, MolmoAct2, StarVLA-π, and StarVLA-OFT initializations and from $\pi_{0.5}$-VL, each trained on the same pretext behaviors.

> FINDING 1 Across four initializations trained on the same pretext behaviors, those with robot pretraining gain the most from the first gradient step, and re-initializing the action expert of $\pi_{0.5}$ removes most of that gain. Robot pretraining supplies the initialization that meta-learning optimizes for explicitly.

5 WHY DO FEW STEPS SUFFICE? FINDINGS AND INSIGHTS

Why does FOSIL work at all? One demonstration produces a large change in a pretrained VLA after a single update, yet the same update does far less for a policy whose action expert is trained from scratch.

5.1 WHY DOES PRETRAINING MAKE ONE GRADIENT STEP ENOUGH?

One possibility is that the pretrained model already contains most of what is needed, accessible through small changes in parameters. To test this, following Yu et al. (2026), we keep only the leading k singular directions of each weight matrix's update at matched norm and measure the retained fraction of the success gain,

$$R_{\text{gain}}(k) = \frac{A_k - A_{\text{unadapted}}}{A_{\text{adapted}} - A_{\text{unadapted}}}, \qquad (1)$$

where A is the success rate of each model (App. E).

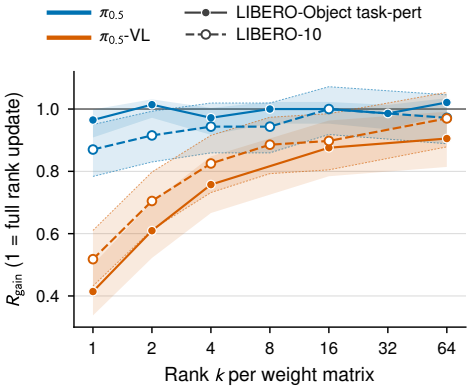


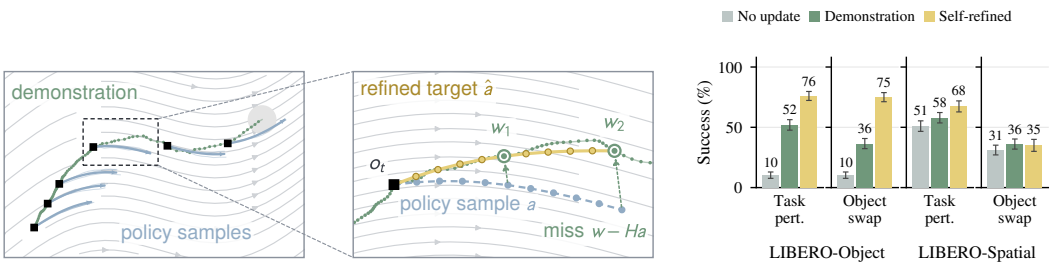
Figure 4: **VLA pretraining concentrates the update in one direction.** Retained gain $R_{\text{gain}}(k)$ at update rank k (App. E); bands: 95% Wilson intervals.

With VLA pretraining, we find a single direction per matrix carries almost the whole gain (Fig. 4): at $k = 1$, $\pi_{0.5}$ retains 0.96 of the gain on LIBERO-Object and 0.87 on LIBERO-10, against 0.41 and 0.52 for $\pi_{0.5}$-VL, which needs $k = 64$ and $k = 16$ to reach 0.9. We find that these updates are also highly local with most of the energy of gradients concentrated in specific layers of VLM (see App. E.1), which also correspond to same layers that recent analysis by Swann et al. (2026) find salient and steerable features in $\pi_{0.5}$.

> INSIGHT 1 VLA pretraining places task-relevant directions close to the initialization, so that a demonstration adjusts an existing direction rather than constructing one over many gradient steps.

5.2 WHICH DEMONSTRATIONS ADAPT WELL, AND CAN WE REPAIR THE ONES THAT DO NOT?

Average success after adaptation stays near 50% on most suites, and part of the cause is the demonstration rather than the optimization. A demonstration can specify the correct task while differing



(a) Refinement procedure.

(b) Effect on adaptation.

Figure 5: **Refining the demonstration toward the unadapted model's action distribution.** *(a)* The demonstration supplies sparse task constraints (end-effector waypoints and gripper commands); the refined chunk $\hat{a}$ is a policy sample minimally corrected to satisfy them. *(b)* Success on the perturbed task after adapting on the demonstration and on its refinement, with one gradient step for both, or eight on LIBERO-Spatial task perturbation (App. C).

from the unadapted model in how it executes it, for example in speed, wrist orientation, or trajectory shape. A gradient step sees all of these differences as targets. Fitting such actions can move the policy away from its pretext behaviors, while also encouraging it memorize the behavior leaving behind generalization.

Refinement keeps what the demonstration specifies and takes the rest from the policy. For environmental adaptation, we therefore refine the demonstration before adaptation, using the unadapted model as a local motion prior, akin to constrained inverse problems in the diffusion literature. From the demonstration we keep only sparse task constraints: the end-effector position at a few waypoints and the gripper commands. At each such state we sample an action chunk a from the unadapted model and correct it minimally to satisfy them,

$$\hat{a} = a + H^{+}(w - Ha), \tag{2}$$

where $w = Ha^{\star}$ collects the constraints read from the demonstrated chunk $a^{\star}$ and H^{+} is a smoothed pseudoinverse of H. The refined chunk $\hat{a}$ matches the demonstration exactly on the constraints and follows the policy sample everywhere else, with the correction spread smoothly over time (Fig. 5a). App. C gives the construction of H and H^{+}.

Refinement improves environmental adaptation. Fig. 5b compares adaptation on the demonstration as recorded with adaptation on its refinement. On LIBERO-Object, refinement makes a single gradient step markedly more effective, roughly doubling success under object swap. On LIBERO-Spatial task perturbation, one step does not separate the two, so we compare both at the same budget of eight steps, within the ten-step limit, where the gap opens (App. C); on the other axes both arms use a single step. In rollouts, the former inherits details such as the wrist orientation and the latter keeps the policy's own motion (App. C.1).

> INSIGHT 2 Few-step adaptation is most reliable when the demonstrated motion lies close to the unadapted model's own action distribution; keeping only the demonstration's task constraints and sampling the rest from the policy recovers the gain with no additional data.

5.3 WHY DO IN-CONTEXT LEARNING METHODS FAIL WITH ROBOT DATA?

Robot data lets an in-context policy ignore its context. An in-context policy must read *what* behavior the demonstration specifies and use the current observation to decide *how* to execute it. Robot data admits a shortcut (Wen et al., 2020): in most pretext behaviors, the observation and instruction alone already predict the action, so a policy can reach a low training loss while ignoring the demonstration, $\pi(x, D) \approx \pi(x)$, where x is the query (observation, instruction, and proprioception) and D the demonstration in context. The behavioral-cloning loss does not distinguish this shortcut from the intended use of the context. The shortcut fails exactly under environmental adaptation, where the demonstration asks for a behavior that the query alone does not predict.

Relation support measures whether training rules out the shortcut. Training rules out the shortcut only when some pretext behaviors pair a query with a demonstrated behavior that differs

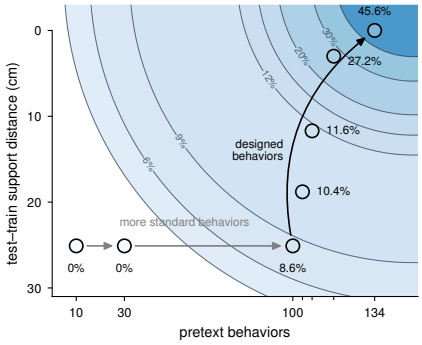


Figure 6: **ICL needs the right relations, not more pretext behaviors.** BPP success on LIBERO-Object task perturbation. Adding standard behaviors (grey) leaves support distance unchanged and success low; a few designed behaviors (black) close the distance and raise success to 45.6% (App. F).

from what the query alone predicts. We call this difference the *relation* of an episode, $r = \psi(\tau) - \phi(x)$, where $\psi(\tau)$ is a feature of the demonstrated behavior τ and $\phi(x)$ the behavior that the training data associates with the query x on its own. The relation is zero when the demonstration only repeats what the query predicts. We summarize a training set by its *support distance* d_{supp}: the average distance from each test pair $(\phi(x), \psi(\tau))$ to its nearest training pair (App. F). A small d_{supp} means that training contains relations like those the tasks require.

Adding pretext behaviors helps only when it adds relations. When we add ten times more standard LIBERO pretext behaviors while leaving relation support unchanged, and BPP succcess rates stays below 9% (Fig. 6, grey). However, when we add a few designed behaviors that lowers d_{supp}, success rises to 45.6% (black). What matters is data in which the context is necessary, not the diversity of the tasks. ICL ($\pi_{0.5}$) behaves the same way: at a fixed number of demonstrations, replacing standard pretext behaviors with designed ones takes it from near 8% to a around 65.6%.

> INSIGHT 3 In-context learning benefits from pretext behaviors only when they supply the *relations* the demonstration must convey, and scaling robot data does not supply them by default. Adaptation does not need them: the demonstration changes the parameters directly rather than through a learned use of context.

6 LIMITATIONS AND FUTURE DIRECTIONS

To the best of our knowledge, this is the first study to examine plain gradient steps on a single demonstration as a route to adapting pretrained VLAs, to compare that route with in-context learning on one backbone, and to ask why so few steps suffice. Several extensions follow from the structure we have found.

The clearest is the demonstration itself. Adaptation inherits whatever the demonstration carries, and our refinement, which keeps the demonstrated intent and lets the policy supply the execution, is a first instance of a broader idea: treating the demonstration as a noisy label and denoising it with the model's own prior, as self-distillation and constraint-guided sampling do in other settings (Shenfeld et al., 2026; Black et al., 2025; Wang et al., 2023). Extending it to longer horizons, and characterizing how much the outcome varies with the choice of demonstration, are the obvious next steps. The second concerns the rest of the policy: one of our main finding is related to low-rank nature of the update and usually how it is confined to a few layers (App. E.1), which suggests composing several one-demonstration updates into one policy and confining the backward pass to those layers, might be a useful area of study for the future. This might also address catastrophic forgetting under explicit gradient step, as localization of update allow for a model to adapt to new situation without moving too far from its pretraining base. Current popular methods in robotics like LoRA although might fail to capture the true structure of this update as shown in our experiments, borrowing ideas from other works (Zhang et al., 2025) and improving them further will also be an interesting future direction.

AI USE STATEMENT

We used large language models to assist with checking the manuscript, reviewing code generation and experimental work, and polishing the writing and presentation. This assistance included feedback on the experimental workflow, code review, and editorial and formatting revisions. The authors retain responsibility for verifying the AI-assisted work and for the final manuscript, code, experiments, and reported claims.

REPRODUCIBILITY STATEMENT

App. A describes the tasks and pretext behaviors, App. B the adaptation protocol and in-context baselines, App. C the demonstration refinement and rollouts, App. D the compute comparison of in-context conditioning and few-step adaptation, App. E the rank truncation and layer locality of the update, and App. F the relation support for in-context learning. Code and rollout videos are at https://fosilvla.github.io.

REFERENCES

Ekin Akyürek, Mehul Damani, Adam Zweiger, Linlu Qiu, Han Guo, Jyothish Pari, Yoon Kim, and Jacob Andreas. The surprising effectiveness of test-time training for few-shot learning. In *International Conference on Machine Learning (ICML)*, 2025.

Christian Bianchi, Siamak Yousefi, Alessio Sampieri, Andrea Roberti, Luca Rigazio, Fabio Galasso, and Luca Franco. Robotic policy adaptation via weight-space meta-learning. *arXiv preprint arXiv:2606.07217*, 2026.

Kevin Black, Manuel Y. Galliker, and Sergey Levine. Real-time execution of action chunking flow policies. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, pp. 37596–37620, 2025.

Kang Chen, Zhihao Liu, Tonghe Zhang, et al. π_{RL}: Online RL fine-tuning for flow-based vision-language-action models. *arXiv preprint*, 2025a.

Yuhui Chen, Shuai Tian, Shugao Liu, Yingting Zhou, Haoran Li, and Dongbin Zhao. ConRFT: A reinforced fine-tuning method for VLA models via consistency policy. In *Robotics: Science and Systems (RSS)*, 2025b.

Yan Duan, Marcin Andrychowicz, Bradly C. Stadie, Jonathan Ho, Jonas Schneider, Ilya Sutskever, Pieter Abbeel, and Wojciech Zaremba. One-shot imitation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Haoquan Fang, Jiafei Duan, Donovan Clay, Sam Wang, Shuo Liu, Weikai Huang, Xiang Fan, Wei-Chuan Tsai, Shirui Chen, Yi Ru Wang, Shanli Xing, Jaemin Cho, Jae Sung Park, Ainaz Eftekhar, Peter Sushko, Karen Farley, Angad Wadhwa, Cole Harrison, Winson Han, Ying-Chun Lee, Eli VanderBilt, Rose Hendrix, Suveen Ellawela, Lucas Ngoo, Joyce Chai, Zhongzheng Ren, Ali Farhadi, Dieter Fox, and Ranjay Krishna. Molmoact2: Action reasoning models for real-world deployment. *ArXiv*, abs/2605.02881, 2026. URL https://api.semanticscholar.org/CorpusID:287950425.

Letian Fu, Huang Huang, Gaurav Datta, Lawrence Yunliang Chen, Will Panitch, Fangchen Liu, Hui Li, and Kenneth Y. Goldberg. ICRT: In-context imitation learning via next-token prediction. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5937–5944, 2025.

Jensen Gao, Suneel Belkhale, Sudeep Dasari, Ashwin Balakrishna, Dhruv Shah, and Dorsa Sadigh. A taxonomy for evaluating generalist robot manipulation policies. *IEEE Robotics and Automation Letters*, 11:3182–3189, 2026. URL https://api.semanticscholar.org/CorpusID:284737614.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Rich Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. OpenVLA: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://arxiv.org/abs/1904.06991.

Haozhan Li, Yuxin Zuo, Jiale Yu, et al. SimpleVLA-RL: Scaling VLA training via reinforcement learning. *arXiv preprint*, 2025.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

Changyu Liu et al. On-the-fly VLA adaptation via test-time reinforcement learning. *arXiv preprint arXiv:2601.06748*, 2026.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *ArXiv*, abs/2205.05638, 2022. URL https://api.semanticscholar.org/CorpusID:248693283.

Guanxing Lu, Wenkai Guo, Chubin Zhang, Yuheng Zhou, Haonan Jiang, Zifeng Gao, Yansong Tang, and Ziwei Wang. VLA-RL: Towards masterful and general robotic manipulation with scalable reinforcement learning. *arXiv preprint*, 2025.

Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable fidelity and diversity metrics for generative models. In *International Conference on Machine Learning*, 2020. URL https://arxiv.org/abs/2002.09797.

Toan Nguyen, Weiduo Yuan, Songlin Wei, Hui Li, Daniel Seita, and Yue Wang. ICLR: In-context imitation learning with visual reasoning, 2026. URL https://api.semanticscholar.org/CorpusID:286377375.

Austin Patel, Ben Pekarek, Joel Enrique Castro Hernandez, and Shuran Song. Behavior prompting policy: Demonstrations as prompts for manipulation. *ArXiv*, abs/2606.30457, 2026. URL https://api.semanticscholar.org/CorpusID:289685244.

Johannes Schwab, Stephan Antholzer, and Markus Haltmeier. Deep null space learning for inverse problems: convergence analysis and rates. *Inverse Problems*, 35, 2018. URL https://api.semanticscholar.org/CorpusID:119152310.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. *ArXiv*, abs/2601.19897, 2026. URL https://api.semanticscholar.org/CorpusID:285071839.

Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *International Conference on Learning Representations*, 2023.

Kaustubh Sridhar, Souradeep Dutta, Dinesh Jayaraman, and Insup Lee. RICL: Adding in-context adaptability to pre-trained vision-language-action models. In *Conference on Robot Learning (CoRL)*, 2025.

StarVLA Community. StarVLA: A lego-like codebase for vision-language-action model developing. *ArXiv*, abs/2604.05014, 2026. URL https://api.semanticscholar.org/CorpusID:287208646.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A. Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning (ICML)*, 2020.

Yu Sun, Xinhao Li, Karan Dalal, Jiarui Xu, Arjun Vikram, Genghan Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, Sanmi Koyejo, Tatsunori Hashimoto, and Carlos Guestrin. Learning to (learn at test time): RNNs with expressive hidden states. *arXiv preprint arXiv:2407.04620*, 2024.

Aiden Swann, Lachlain McGranahan, Hugo Buurmeijer, Monroe Kennedy, and Mac Schwager. Sparse autoencoders reveal interpretable and steerable features in vla models. *CoRL*, abs/2603.19183, 2026. URL https://api.semanticscholar.org/CorpusID:286673529.

Shuhan Tan, Kairan Dou, Yue Zhao, and Philipp Krähenbühl. Interactive post-training for vision-language-action models. *arXiv preprint*, 2025.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bash-lykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wen-Yin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, A. S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng-Xu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melissa Hall, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnić, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288, 2023. URL https://api.semanticscholar.org/CorpusID:259950998.

Vitalis Vosylius and Edward Johns. Instant policy: In-context imitation learning via graph diffusion. In *ICLR*, 2025.

Yinhuai Wang, Jiwen Yu, and Jian Zhang. Zero-shot image restoration using denoising diffusion null-space model. In *International Conference on Learning Representations*, 2023.

Chuan Wen, Jierui Lin, Trevor Darrell, Dinesh Jayaraman, and Yang Gao. Fighting copycat agents in behavioral cloning from observation histories. *NeurIPS*, abs/2010.14876, 2020. URL https://api.semanticscholar.org/CorpusID:225094546.

Guo Yu, Wenlin Liu, Yulan Hu, Hao-Xuan Ma, Jun-Peng Jiang, and Han-Jia Ye. Dense supervision, sparse updates: On the sparsity and geometry of on-policy distillation. *ArXiv*, abs/2606.13657, 2026. URL https://api.semanticscholar.org/CorpusID:289187037.

Yuanhe Zhang, Fanghui Liu, and Yudong Chen. Lora-one: One-step full gradient could suffice for fine-tuning large language models, provably and efficiently. In *International Conference on Machine Learning (ICML)*, 2025. arXiv:2502.01235.

Xueyang Zhou, Yangming Xu, Guiyao Tie, Yongchao Chen, Guowen Zhang, Duanfeng Chu, Pan Zhou, and Lichao Sun. LIBERO-PRO: Towards robust and fair evaluation of vision-language-action models beyond memorization. *ArXiv*, abs/2510.03827, 2025. URL https://api.semanticscholar.org/CorpusID:281843691.