Extended Abstract

Motivation Vision-language-action (VLA) models for humanoid robots are trained almost exclusively via supervised imitation learning, bounding peak performance by demonstration quality and giving no mechanism to differentiate better trajectories from worse ones at deployment Wei et al. (2026); Physical Intelligence et al. (2025b); NVIDIA et al. (2025). Physical Intelligence's RECAP Physical Intelligence et al. (2025a) proposes advantage-conditioned fine-tuning to remedy this, doubling task throughput on long-horizon manipulation without requiring tractable policy log-likelihoods. But RECAP has only been shown on a closed-source proprietary VLA, leaving open whether it transfers to other humanoid architectures and survives the order-of-magnitude compute and data reductions typical of academic research.

Method We apply RECAP's three-component recipe to Ψ_0 Wei et al. (2026), a recently open-sourced humanoid VLA, providing the first reproduction of RECAP outside Physical Intelligence's stack. Our pipeline is: (1) collect 500 outcome-labeled on-policy trajectories in MuJoCo simulation on the G1WholebodyBendPickMP-v0 task from the Ψ_0 SIMPLE benchmark; (2) train a small distributional value function Physical Intelligence et al. (2025a); Bellemare et al. (2017) predicting time-to-task-completion; and (3) compute advantages, threshold to a binary `advantage: high/low` indicator, inject the indicator as a text prefix token into Ψ_0's prompt, and fine-tune with the standard flow-matching loss Lipman et al. (2023). This sidesteps the intractable log-likelihood problems that hinder online RL methods such as ConRFT Chen et al. (2025) and VLA-RL Lu et al. (2025) on flow-matching policies. We target BendPickMP—a hard whole-body reach–bend–pick task—after the SFT policy saturated at 98.2% on TabletopGraspMP and left no headroom there.

Implementation The base model is Ψ_0's released `postpre.1by1.pad36` checkpoint. SFT ran on a single H100, rollout collection on an RTX 5000 Ada (MuJoCo), and RECAP fine-tuning on 4×A100s—all via Modal. The core change is ≈200 lines: the distributional value-function head and the advantage-token prompt-injection pipeline.

Results We isolate the effect of advantage conditioning with two controlled comparisons that hold the training data fixed. First, a *within-model* contrast: the single RECAP policy, evaluated with only the inference-time prefix token changed, succeeds on **30.8%** of BendPickMP episodes when conditioned `advantage: high` versus **21.5%** when conditioned `advantage: low` (+9.3 points, two-proportion $z = 3.35$, $p < 0.001$; $N = 500$ each). Because nothing but the token differs, this is direct causal evidence that Ψ_0 learns to condition on the injected advantage signal. Second, a *matched-data* comparison against behavior cloning on the identical 500 rollouts but with no advantage token (BC-all): RECAP-high reaches **30.8%** versus **18.2%** for BC-all (+12.6 points, $z = 4.63$, $p < 0.001$), showing advantage conditioning extracts more task success from the same mixed-quality data than plain imitation. The supervised baseline (SFT@40k, **48.6%**) is reported for context, not as a head-to-head comparison: it trains on 100 clean expert demonstrations rather than the 500 mixed-quality rollouts (64% failures) that RECAP and BC-all share, so SFT-vs-RECAP conflates data source with method. The value function that produces the advantage labels reaches an episode-level AUC of **0.703** and a held-out return MSE of **0.062** (vs. 0.13 naive).

Discussion The two controlled results establish the central claim: advantage conditioning transfers to an open-source flow-matching VLA, yielding a statistically significant gain over both its low-advantage counterpart and a matched-data behavior-cloning baseline. The remaining gap to SFT@40k is a data-regime gap, not a method failure—SFT consumes clean expert demonstrations while RECAP learns from a predominantly-negative on-policy distribution. The value function is a coarse but useful classifier (AUC 0.703), accurate enough that its labels drive a real behavioral gain; scaling rollouts and iterating the value function Peng et al. (2019); Kostrikov et al. (2022) are the clearest levers for closing the remaining gap.

Conclusion We present the first reproduction of RECAP on an open-source humanoid VLA, showing that advantage conditioning works on Ψ_0—it significantly outperforms both its `advantage: low` counterpart and a matched-data behavior-cloning baseline under an academic budget (500 rollouts, 4×A100). This provides the community with a reproducible RL post-training baseline for open-source humanoid foundation models and a roadmap for scaling it.

RECAP-Ψ: Advantage-Conditioned Fine-Tuning for Open-Source Humanoid VLAs

Karthik Pythireddi
Department of Electrical Engineering
Stanford University
karthik9@stanford.edu

Aaditya Shah
Department of Computer Science
Stanford University
aadi2@stanford.edu

Jonathan Lu
Department of Physics
Stanford University
jlu2@stanford.edu

Abstract

Vision-language-action (VLA) models for humanoid robots are trained almost exclusively via supervised imitation learning, which bounds peak performance by demonstration quality and provides no mechanism to learn from successes versus failures. Physical Intelligence's RECAP introduces advantage-conditioned fine-tuning to address this limitation, doubling task throughput on long-horizon manipulation, but has only been validated on a closed-source proprietary VLA. We present RECAP-Ψ, the first reproduction of RECAP on an open-source humanoid VLA, applying the method to Ψ_0 Wei et al. (2026) under academic compute and data budgets. Our pipeline collects 500 on-policy rollouts in MuJoCo simulation on the G1WholebodyBendPickMP-v0 task, trains a distributional value function Physical Intelligence et al. (2025a); Bellemare et al. (2017) for credit assignment, and injects binary advantage tokens into Ψ_0's language prompt prefix before fine-tuning with the standard flow-matching loss Lipman et al. (2023). Our value function achieves an episode-level AUC of 0.703 and an MSE of 0.062 on held-out trajectories (versus a naive baseline of 0.13), confirming that meaningful credit assignment is learned. Two controlled comparisons that hold the training data fixed establish that advantage conditioning works: conditioning the policy `advantage: high` yields 30.8% task success versus 21.5% for `advantage: low` (+9.3 points, $p < 0.001$), and RECAP outperforms a behavior-cloning baseline trained on the identical 500 rollouts without advantage tokens (30.8% vs. 18.2%, $p < 0.001$). We deliberately do not treat the expert-demonstration SFT baseline (48.6%) as a head-to-head comparison, since it is trained on clean expert data rather than the mixed-quality on-policy rollouts RECAP consumes. This work provides the open-source robotics community with a reproducible RL post-training baseline for humanoid foundation models.

1 Introduction

Recent advances in large-scale learning have produced vision-language-action (VLA) models that leverage vision-language pre-training to generate robot actions directly from image observations and natural language instructions NVIDIA et al. (2025); Physical Intelligence et al. (2025b); Wei et al. (2026). Trained on large human-demonstration datasets via supervised imitation learning (behavior cloning), they acquire diverse, generalizable manipulation behaviors. However, the imitation

paradigm has a fundamental ceiling: the policy is bounded by its demonstrations, with no training-time mechanism to distinguish successful trajectories from failures or to improve through interaction with the environment.

Reinforcement learning (RL) offers a principled framework for learning from outcomes rather than demonstrations, and several recent works have explored applying RL to fine-tune pretrained VLA models Chen et al. (2025); Lu et al. (2025). However, a core technical challenge stands in the way: modern high-performing VLAs, including Ψ_0 Wei et al. (2026) and $\pi_{0.5}$ Physical Intelligence et al. (2025b), use flow-matching action experts Lipman et al. (2023) rather than autoregressive discrete-token policies, and standard RL objectives such as policy gradient or PPO require tractable per-action log-likelihoods that flow-matching policies do not provide. Approaches like ConRFT Chen et al. (2025) and VLA-RL Lu et al. (2025) must therefore work around this issue with approximations or by restricting themselves to autoregressive architectures.

Physical Intelligence's $\pi^*_{0.6}$ work introduces RECAP Physical Intelligence et al. (2025a), an elegant solution that sidesteps the log-likelihood problem entirely by using *advantage conditioning*: rather than modifying the policy gradient, RECAP trains a value function for credit assignment, thresholds the resulting advantages into a binary indicator, and injects this indicator as a text token into the policy's prompt prefix. The policy is then fine-tuned with the standard supervised flow-matching loss, conditioned on the advantage token. RECAP demonstrates strong results on $\pi^*_{0.6}$, more than doubling task throughput and roughly halving failure rates on difficult long-horizon tasks. However, RECAP has only been demonstrated on Physical Intelligence's closed-source proprietary VLA stack, leaving open two critical questions: (1) whether the method generalizes to other VLA architectures, and (2) whether its benefits hold under the order-of-magnitude reductions in compute and data typical of academic research.

We apply RECAP to Ψ_0 Wei et al. (2026), a recently open-sourced humanoid VLA trained from 800 hours of egocentric video and only 30 hours of robot data. We target G1WholebodyBendPickMP-v0 from its SIMPLE benchmark—a whole-body reach–bend–grasp task with a clean binary success signal and headroom over the SFT baseline—and follow RECAP's three-component recipe: on-policy rollout collection in MuJoCo, distributional value-function training Physical Intelligence et al. (2025a); Bellemare et al. (2017), and advantage-conditioned flow-matching fine-tuning with binary `advantage: high/low` prefix tokens.

We make three contributions. First, the first reproduction of RECAP on an open-source humanoid VLA, establishing that advantage conditioning is architecturally compatible with Ψ_0's flow-matching policy. Second, two confound-controlled comparisons that separate the effect of advantage conditioning from that of the training data—a within-model contrast varying only the inference-time token, and a matched-data comparison against behavior cloning on identical rollouts—both statistically significant (`high` 30.8% vs. `low` 21.5%; RECAP 30.8% vs. matched-data BC 18.2%; both $p < 0.001$). Third, an evaluation in the small-data regime (500 rollouts), far below the original RECAP scale, with a characterization of the value function quality (AUC 0.703, MSE 0.062) behind these gains and a path to closing the remaining gap to expert-data SFT via scaled rollouts and value-function iteration.

2 Related Work

2.1 Humanoid Vision-Language-Action Models

Recent work has produced a new generation of humanoid VLA models trained via large-scale imitation learning. GR00T N1 NVIDIA et al. (2025) pairs a vision-language module with a diffusion transformer action expert, trained on a mixture of real robot trajectories, human videos, and synthetic data, and generalizes across multiple embodiments. $\pi_{0.5}$ Physical Intelligence et al. (2025b) extends π_0 with cross-task co-training and high-level semantic subtask prediction for broad household generalization. Ψ_0 Wei et al. (2026) argues that data quality matters more than scale, pre-training on human egocentric video before fine-tuning on a small high-quality robot dataset for whole-body loco-manipulation. All are trained purely with imitation objectives, so they have no mechanism to learn from outcomes and are bounded by demonstration quality.

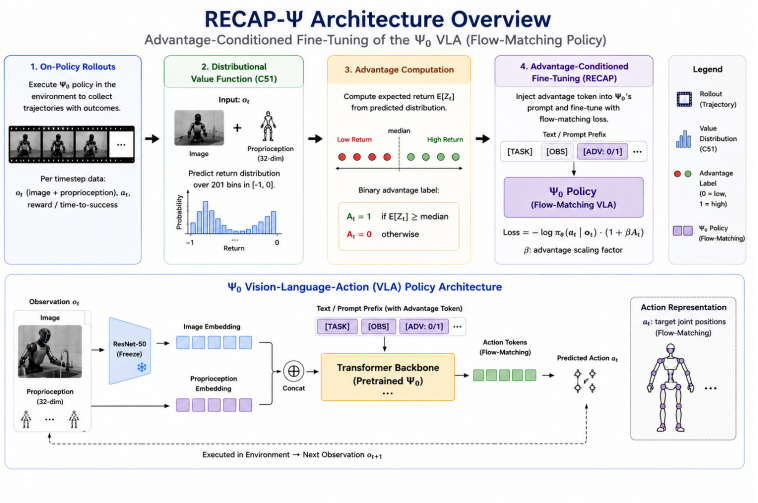

Figure 1: RECAP-Ψ pipeline: on-policy rollouts from the SFT policy feed a distributional value function, whose top-30% advantages are injected as `advantage: high/low` prefix tokens into Ψ_0's prompt before flow-matching fine-tuning Lipman et al. (2023).

2.2 Reinforcement Learning for VLA Fine-Tuning

Several works apply RL to push VLAs past the imitation ceiling. ConRFT Chen et al. (2025) combines offline Q-learning with an online consistency-policy objective for contact-rich manipulation, but constrains actions in-distribution and targets autoregressive policies. VLA-RL Lu et al. (2025) uses trajectory-level online RL with a pretrained process reward model, again for autoregressive VLAs. Both struggle with flow-matching action experts, which lack the tractable per-action log-likelihoods these objectives require. RECAP Physical Intelligence et al. (2025a) sidesteps this with advantage conditioning—an indirect policy-improvement operator Frans et al. (2025) that trains a value function, thresholds advantages into a binary token, and fine-tunes with the standard supervised loss conditioned on that token, making it naturally compatible with flow matching. Our work is the first to reproduce RECAP outside Physical Intelligence's stack, and the first on Ψ_0.

2.3 Offline RL and Advantage-Weighted Methods

The offline RL literature provides the foundations for RECAP's advantage conditioning. Advantage-Weighted Regression (AWR) Peng et al. (2019) casts policy improvement from a fixed dataset as advantage-weighted supervised learning. Implicit Q-Learning (IQL) Kostrikov et al. (2022) advances this by avoiding out-of-distribution action evaluation through expectile regression on the value function. RECAP can be viewed as a flow-matching-compatible instantiation of this advantage-weighted recipe for continuous actions and large pre-trained policies. Both AWR and IQL are sensitive to data coverage and value function quality—directly relevant to our small-data regime (500 rollouts). The distributional value function we use Bellemare et al. (2017) gives richer uncertainty estimates than a point estimate and matches the original RECAP formulation.

3 Method

3.1 Problem Formulation

We consider single-task manipulation where a Ψ_0 policy π_θ maps an observation $o_t = (\mathbf{I}_t, \mathbf{s}_t)$—an egocentric RGB image and a 32-d proprioceptive state—to a whole-body action chunk $\mathbf{a}_t \in \mathbb{R}^{H\times d}$ (H=30, d=36). The policy also receives a fixed natural language task description and, during RECAP fine-tuning and inference, a binary advantage prefix token (`advantage: high` or `advantage: low`). Ψ_0 is three-tier: a Qwen3-VL-2B-Instruct vision-language backbone (System-2); a flow-matching diffusion-transformer action expert (System-1, ≈500M parameters) Lipman et al. (2023); and an RL tracking controller (System-0) for lower-body stability Wei et al. (2026). Our goal is to

improve an expert-fine-tuned Ψ_0 from on-policy outcome feedback without modifying System-1's flow-matching objective.

3.2 Supervised Fine-Tuning Baseline

We fine-tune the publicly released Ψ_0 `postpre.1by1.pad36` checkpoint on 100 expert demonstrations of the BendPickMP task. We evaluate checkpoints at 20k and 40k gradient steps, obtaining success rates of 36.0% and 48.6% respectively on BendPickMP over 500 closed-loop MuJoCo evaluation episodes. The 20k checkpoint initializes on-policy rollout collection, as its 36.0% success rate produces a diverse mix of successes and failures well-suited for value function training. The SFT baseline is *not* a controlled comparison for RECAP—it trains on clean expert data, while RECAP trains on mixed-quality rollouts—so we instead use the matched-data baseline described next (Section 3.3).

3.3 Matched-Data Behavior Cloning Baseline

To isolate the contribution of advantage conditioning from the contribution of the rollout data itself, we train a behavior-cloning baseline (**BC-all**) on exactly the same 500 on-policy rollouts used for RECAP, with the standard flow-matching loss and *no* advantage token in the prompt. BC-all and RECAP thus differ in exactly one factor—the presence of the advantage-conditioning token—making BC-all the correct head-to-head control for measuring what advantage conditioning adds on top of plain imitation of the same data. The expert-demonstration SFT baseline cannot play this role: it is fit to a fundamentally different data distribution (100 all-successful demos vs. 500 rollouts that are 64% failures), so a SFT-vs-RECAP gap would conflate data quality with the learning method (Appendix C, Fig. 5).

3.4 Task Selection

We evaluated multiple tasks from the Ψ_0 SIMPLE benchmark and selected G1WholebodyBendPickMP-v0 as our primary evaluation environment. The task provides a balanced mixture of successful and unsuccessful rollouts, enabling meaningful credit assignment while leaving substantial room for policy improvement. In contrast, some benchmark tasks were either nearly solved by the supervised baseline or too difficult to generate sufficient successful trajectories for learning. Detailed task-screening results and selection rationale are provided in Appendix A.

3.5 On-Policy Rollout Collection

We deploy the 20k SFT checkpoint in closed-loop MuJoCo using the Ψ_0 SIMPLE benchmark harness. Each episode runs up to $T = 600$ simulation steps, terminating early on success, and stores the camera images and 32-d proprioceptive states at each VLA query point along with the binary success label and step count n_{steps}. Of 500 collected episodes, 180 (36.0%) succeed; the 320 failures all time out at the 600-step limit.

3.6 Distributional Value Function

We train a distributional value function Z_ϕ following the RECAP Physical Intelligence et al. (2025a) framework, which builds on categorical distributional RL Bellemare et al. (2017) to predict the distribution over normalized time-to-task-completion from each observation. For a successful episode, the return at VLA decision point $i \in \{0, \ldots, T_{\text{obs}} - 1\}$ is:

$$r_i = -\frac{(n_{\text{steps}} - 1)}{599} \cdot \left(1 - \frac{i}{T_{\text{obs}} - 1}\right), \tag{1}$$

which maps all returns to $[-1, 0]$, where 0 indicates the decision point immediately preceding task success and -1 indicates maximum distance from completion. Failed episodes receive $r_i = -1.0$ at all decision points. This formulation rewards efficiency: earlier decision points in faster episodes receive higher returns than the same points in slower ones. Because exact VLA query timestamps are not recorded, we assume uniform spacing within each episode; in practice queries occur approximately every 24 simulation steps.

The backbone is a frozen ImageNet-pretrained ResNet-50 He et al. (2016) that encodes each camera image into a 2048-d feature; concatenated with the 32-d proprioceptive state, this feeds a 3-layer MLP producing logits over 201 uniformly-spaced return bins spanning $[-1, 0]$. We use a frozen ResNet-50 rather than Ψ_0's own Qwen3-VL-2B-Instruct backbone because the full encoder is prohibitively expensive for iterative value-function development under our budget, while ResNet-50 features suffice for return prediction. The full MLP architecture and training setup (cross-entropy loss, Adam, 50 epochs, stratified 90/10 split, seed 42) are detailed in Appendix B.

3.7 Advantage Labeling

Given the trained value function, we form the scalar baseline as the mean of the predicted return distribution and define the per-step advantage as the realized return minus that baseline:

$$\hat{V}(o_i) = \mathbb{E}\big[Z_\phi(o_i)\big] = \sum_k p_k \, b_k, \qquad A_i = r_i - \hat{V}(o_i), \tag{2}$$

where $p_k = \mathrm{softmax}(Z_\phi(o_i))_k$ are the predicted bin probabilities and b_k are the fixed bin centers. The advantage A_i measures whether the observed return at decision point i exceeded the model's expectation. We convert it into the binary conditioning label that RECAP injects into the policy by thresholding at the global 70th percentile $\eta = Q_{0.7}\big(\{A_j\}_{j \in \mathcal{D}}\big)$ over all decision points in the rollout dataset $\mathcal{D}$:

$$\ell_i \;=\; \begin{cases} \texttt{high} & \text{if } A_i \geq \eta \quad (\text{top } 30\%), \\ \texttt{low} & \text{otherwise.} \end{cases} \tag{3}$$

The percentile rule fixes a $30/70$ split independent of the raw advantage scale, making the labeling robust to miscalibration of $\hat{V}$ and avoiding a hand-tuned threshold at zero. The threshold is computed globally rather than per-episode.

3.8 RECAP Fine-Tuning

Advantage-conditioned flow-matching objective. The label ℓ_i from Eq. (3) is rendered as a plain-text prefix concatenated with the task instruction to form the conditioning context $c(o_i, \ell_i)$—e.g. `advantage: high. Pick up the object on the floor`—processed by the System-2 backbone alongside the image. The System-1 action expert is a conditional flow-matching model Lipman et al. (2023) over action chunks $a \in \mathbb{R}^{H \times d}$: for a flow time $\tau \sim \mathcal{U}(0, 1)$ and noise $\varepsilon \sim \mathcal{N}(0, I)$, the straight-line interpolant $a^\tau = (1 - \tau)\, a + \tau\, \varepsilon$ has constant target velocity $\varepsilon - a$. RECAP fine-tunes the velocity field v_θ with the standard flow-matching regression, conditioned on the advantage label:

$$\mathcal{L}_{\text{RECAP}}(\theta) \;=\; \mathbb{E}_{(o,a,\ell)\sim\mathcal{D}}\, \mathbb{E}_{\tau,\,\varepsilon} \Big\| v_\theta\big(a^\tau, \tau,\, o,\, c(o, \ell)\big) - \big(\varepsilon - a\big) \Big\|_2^2. \tag{4}$$

The loss is identical to supervised behavior cloning; the *only* change is the advantage token inside $c(o, \ell)$. The matched-data BC-all baseline (Section 3.3) strips the conditioning to $c(o)$, so the two differ in exactly one input.

Inference as indirect policy improvement. At deployment we fix $\ell = \texttt{high}$ and sample an action chunk by integrating the learned ODE from noise $\varepsilon \sim \mathcal{N}(0, I)$ to data. Because the policy was trained to reproduce the action distribution associated with each label, conditioning on `high` targets the top-advantage slice of the data,

$$\pi_\theta(a \mid o, \texttt{high}) \;\approx\; p_{\mathcal{D}}\big(a \mid o,\; A(o, a) \geq \eta\big), \tag{5}$$

i.e. behavior cloning restricted to the highest-advantage actions—an *indirect* advantage-weighted policy-improvement step Peng et al. (2019); Frans et al. (2025) that never requires a tractable per-action log-likelihood, which is precisely what makes it compatible with the flow-matching action expert.

Training configuration. We fine-tune on the advantage-labeled 500-rollout dataset alone—without mixing in the expert demonstrations—so that RECAP and BC-all share identical data. Both start from the same base checkpoint and train for 40k gradient steps with the AdamW optimizer, learning rate 1×10^{-4}, cosine schedule with 1k warmup steps, and a global batch size of 128. RECAP fine-tuning runs on $4\times$A100 GPUs via Modal.

4 Experimental Setup

4.1 Simulation Environment

All experiments are conducted in MuJoCo simulation using the Ψ_0 SIMPLE benchmark harness Wei et al. (2026). The target task is G1WholebodyBendPickMP-v0, described in Section 3.4. Episodes terminate upon task success or after a maximum of $T = 600$ simulation steps. All policies are evaluated in closed-loop: the policy receives observations at each VLA query point (approximately every 24 simulation steps) and executes 24-step action chunks until termination.

4.2 Compared Conditions

All four conditions start from the same Ψ_0 `postpre.1by1.pad36` checkpoint Wei et al. (2026). **SFT@20k** and **SFT@40k** are fine-tuned on 100 expert demonstrations of BendPickMP (20k/40k steps); SFT@20k initializes rollout collection, and SFT@40k is reported for context only (different data regime, see Section 3.3). **BC-all** and **RECAP** are trained for 40k steps on the identical 500 on-policy rollouts and differ only in the advantage token (BC-all has none). At inference RECAP is run under both prefixes, **RECAP-high** and **RECAP-low**, to measure the within-model token effect. All training uses AdamW, learning rate 1×10^{-4}, and batch size 128.

4.3 Evaluation Protocol

We measure performance using task **success rate**—the fraction of evaluation episodes in which the robot successfully picks up the object within $T = 600$ steps. Every condition is evaluated over 500 closed-loop MuJoCo episodes. We report two-proportion z-tests for the controlled comparisons and Wilson 95% confidence intervals for all success rates. The within-model contrast (RECAP-high vs. RECAP-low) evaluates the same trained policy with only the prefix token changed, isolating the causal effect of the advantage signal.

4.4 Compute Resources

SFT fine-tuning runs on a single H100 GPU provisioned via Modal cloud compute. On-policy rollout collection runs on an RTX 5000 Ada GPU. RECAP fine-tuning runs on 4×A100 GPUs via Modal. Value function training runs on a Modal cloud GPU instance as described in Section 3.6.

5 Results

We evaluate RECAP-Ψ on G1WholebodyBendPickMP-v0 across two axes: (1) closed-loop task success rate in MuJoCo simulation, and (2) value function calibration quality on the held-out validation split. Our analysis centers on two confound-controlled comparisons—the within-model advantage contrast (RECAP-high vs. RECAP-low) and the matched-data comparison (RECAP vs. BC-all)—with the expert-data SFT baseline reported for context only.

5.1 Quantitative Evaluation

Advantage conditioning works (within-model contrast). The cleanest test of the method holds everything fixed except the inference-time advantage token. Evaluating the single trained RECAP policy under each prefix, success rises from **21.5%** (`advantage: low`) to **30.8%** (`advantage: high`)—a 9.3-point gain that is statistically significant (two-proportion $z = 3.35$, $p < 0.001$; $N = 500$ each). Since the weights, observations, and evaluation episodes are identical and only the prompt token differs, this is direct causal evidence that Ψ_0 learned to condition its behavior on the injected advantage signal.

Advantage conditioning beats matched-data BC. Against BC-all—trained on the identical 500 rollouts but with no advantage token—RECAP-high reaches **30.8%** versus **18.2%** ($z = 4.63$, $p < 0.001$). With data held fixed, adding the advantage-conditioning token improves task success by 12.6 points, showing the mechanism extracts more from the same mixed-quality on-policy data than plain imitation. Notably, even RECAP-low (21.5%) edges out BC-all, consistent with the advantage token giving the policy a usable axis of behavioral variation that undifferentiated cloning lacks.

Table 1: Task success rate on BendPickMP with Wilson 95% confidence intervals. The two confound-controlled comparisons are bracketed: RECAP-high vs. RECAP-low varies only the inference token ($p < 0.001$); RECAP vs. BC-all holds the 500-rollout training data fixed and varies only the advantage token ($p < 0.001$). SFT (expert demos) is reported for context, not as a controlled comparison.

Method	Training data	N	Success Rate	95% CI
SFT@20k (rollout init)	100 expert demos	500	36.0%	[31.9, 40.3]
SFT@40k (context)	100 expert demos	500	48.6%	[44.2, 53.0]
BC-all	500 rollouts, no token	500	18.2%	[15.1, 21.8]
RECAP-low	500 rollouts + token	500	21.5%	[18.1, 25.3]
RECAP-high	500 rollouts + token	500	**30.8%**	[26.9, 35.0]

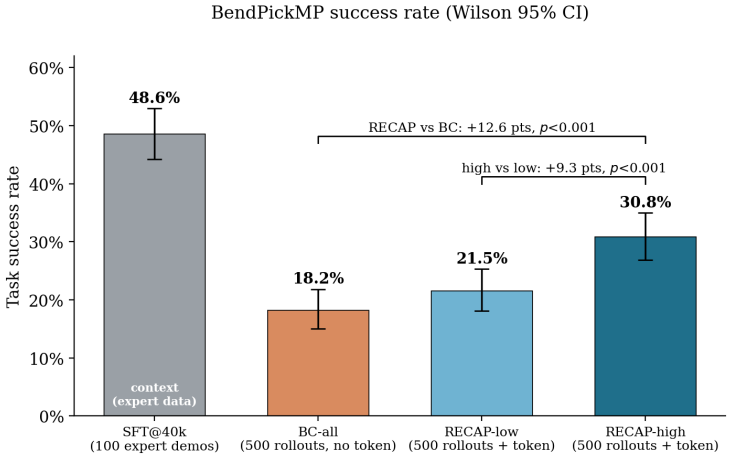

Figure 2: BendPickMP success rate (Wilson 95% CI). The matched-data trio (BC-all, RECAP-low, RECAP-high; identical 500 rollouts) isolates the advantage token: RECAP-high beats RECAP-low and the no-token BC-all baseline (both $p < 0.001$). SFT@40k (grey) uses clean expert demonstrations and is shown for context only.

Relation to the expert-data SFT baseline. SFT@40k attains 48.6%, above RECAP-high, but this is not a head-to-head result: SFT trains on 100 clean expert demonstrations while RECAP and BC-all train on 500 rollouts that are 64% failures. The gap therefore measures the value of clean expert data under our budget, not a failure of advantage conditioning (see Discussion).

Value Function Calibration. We evaluate the value function on four metrics over the held-out 10% validation split (50 episodes, stratified by outcome). Step-level MSE between predicted $\hat{V}(o_t)$ and true return r_t is **0.062** versus a 0.13 naive baseline (predicting the dataset mean return at every step)—a 52% error reduction. Episode-level AUC, ranking successful episodes above failed ones from $\hat{V}(o_0)$, is **0.703**; step-level AUC is **0.837**, showing the value signal is consistent throughout the trajectory rather than only at initialization. Kendall's τ between $\hat{V}(o_0)$ and the true continuous return is only **0.294** ($p = 0.007$), however: the VF reliably discriminates success from failure but ranks episodes by continuous return weakly, behaving more as a coarse classifier than a fine-grained return predictor (calibration plot in Appendix C, Fig. 4).

5.2 Qualitative Analysis

Success rate alone does not reveal *how* policies fail or succeed on BendPickMP. The task requires the Unitree G1 to coordinate a deep whole-body bend, a forward reach toward an object below its nominal standing envelope, and a stable two-hand grasp while maintaining balance—a sequence where small timing errors compound into mis-reaches, partial grasps, or 600-step timeouts. We therefore inspected closed-loop rollouts across all compared conditions. Among the 500 on-policy training episodes, 64% end in timeout without a successful pick; the remaining failures we observed by hand include repeated

Figure 3: Qualitative comparison of SFT and RECAP grasping behavior on BendPickMP. Top: SFT@20k successful grasp (episode 383, 176 steps). Bottom: RECAP advantage-conditioned successful grasp (episode 013, 142 steps). Frames sampled from the grasp phase (30–98% of episode).

reach corrections near the object and grasps that never fully close. These modes are informative for credit assignment—they occupy long stretches of negative return before termination—but they also explain why undifferentiated behavior cloning on the same rollouts (BC-all) struggles to extract a reliable success signal.

The within-model RECAP contrast makes the advantage token's behavioral effect visible even when an episode ultimately fails. Evaluating the *same* fine-tuned weights with only the prefix changed, `advantage: high` rollouts tend to commit earlier to the bend-and-reach phase and spend less time hovering above the target before attempting closure, whereas `advantage: low` rollouts more often resemble the hesitant, correction-heavy trajectories common in the raw rollout buffer. This pattern aligns with the training objective: high-advantage labels preferentially attach to decision points from faster-completing episodes (Eqs. (1)–(3)), so conditioning on `high` at inference asks the policy to imitate that higher-return slice of the data rather than the dataset average.

Figure 3 illustrates the contrast on two representative *successful* episodes, with frames taken from the grasp phase (30–98% of episode length) after the robot has already bent and reached the object. The SFT@20k policy (episode 383, 176 steps) executes a smooth but comparatively slow sequence: it approaches the box cautiously, adjusts hand pose over multiple query steps, and only then lifts. The RECAP-high policy (episode 013, 142 steps) follows the same high-level structure—reach, contact, lift—but closes the grasp in fewer decision cycles and transitions to lift with less intermediate repositioning. Both episodes succeed against table clutter (distractor objects vary by initialization), so the difference is not explained by an easier scene; rather, RECAP appears to have internalized a more decisive closing behavior consistent with upweighting trajectories that complete the task sooner. The 19% reduction in steps on this paired example (176 $\rightarrow$ 142) is directionally consistent with the value function's efficiency-sensitive return definition and with the quantitative RECAP-high advantage over both RECAP-low and BC-all.

6 Discussion

The advantage-conditioning mechanism transfers to Ψ_0. The two controlled comparisons converge on one conclusion: advantage conditioning yields a real, statistically significant gain on an open-source flow-matching humanoid VLA. The within-model contrast ($+9.3$ pts) rules out any explanation but the token itself, since weights and evaluation episodes are identical; the matched-data contrast ($+12.6$ pts over BC-all) rules out the rollout data alone driving the gain. Together they show Ψ_0 reads the injected token and shifts toward higher-return actions—the property RECAP claims, shown for the first time outside Physical Intelligence's stack.

Why we do not compare directly to SFT. SFT@40k (48.6%) exceeds RECAP-high, but the two train on different data: 100 curated, all-successful demonstrations versus 500 predominantly-failed on-policy rollouts. A direct comparison would confound data quality with learning method, so we control instead against baselines that share RECAP's data (BC-all) or weights (RECAP-low). The remaining gap reads as headroom a larger or cleaner rollout set could recover, not as evidence against advantage conditioning.

The value function is a coarse but sufficient classifier. The value function separates successes from failures well (episode AUC 0.703, step AUC 0.837) but ranks continuous return only weakly (Kendall's $\tau = 0.294$), behaving more as a binary success classifier than a fine-grained return estimator. Crucially, this is already enough to produce advantage labels that drive a significant behavioral gain: at this data budget the binary advantage signal is the operative ingredient, and sharpening the value function's continuous ranking is a lever for further gains rather than a prerequisite for any gain at all.

Limitations. Our controlled comparisons rest on $N = 500$ evaluation episodes per condition and a single training seed on a single task in simulation; while the difference confidence intervals clearly exclude zero (high$-$low: $+9.3$ pts, 95% CI $[3.9, 14.7]$; RECAP$-$BC: $+12.6$ pts, 95% CI $[7.3, 17.9]$), we report no cross-seed variance. The value function uses a frozen ResNet-50 rather than Ψ_0's own vision backbone for compute reasons, the advantage threshold is a fixed global 30th percentile, and we evaluate a single RECAP iteration rather than the iterative collect-relabel-finetune loop of the original method. These choices trade fidelity for tractability under an academic budget and bound the strength of our claims to "the mechanism works and helps" rather than "the method is fully reproduced at scale."

7 Conclusion

We presented RECAP-Ψ, the first reproduction of advantage-conditioned fine-tuning on an open-source humanoid VLA. Through two confound-controlled comparisons we showed that advantage conditioning works on Ψ_0: conditioning `advantage: high` significantly outperforms `advantage: low` on the same policy (30.8% vs. 21.5%, $p < 0.001$), and RECAP outperforms a behavior-cloning baseline trained on identical data without the advantage token (30.8% vs. 18.2%, $p < 0.001$). The expert-data SFT baseline remains higher (48.6%), but is not a controlled comparison because it consumes clean expert demonstrations rather than mixed-quality on-policy rollouts. Our value function, while only a coarse return ranker, is accurate enough to produce labels that drive these gains. The clearest path to closing the remaining gap to expert-data SFT is to scale on-policy rollout collection and iterate the value function—now feasible with the GPU-accelerated rollout pipeline we built. We release our pipeline as a reproducible RL post-training baseline for open-source humanoid foundation models.

8 Team Contributions

- **Karthik Pythireddi:** Worked on the project proposal and created and contributed to the poster write-up. Evaluated the G1WholebodyTabletopGraspMP-v0, FaucetMP, and BendPickMP tasks. Performed SFT rollouts and calculated the success rates for all three tasks. Did the rollouts collection on the advantage high and the advantage low for the BendPickMP task. Coordinated the communication with the Project Mentor and the teammates.

  Repository: `https://github.com/karthikpythireddi/cs224_final_project`
- **Jonathan Lu**: Worked on the project proposal and contributed to the poster write-up. Designed and implemented the distributional value function pipeline, including the return normalization scheme, model architecture, training script, and four-metric calibration evaluation suite (step-level MSE, episode-level AUC, step-level AUC, and Kendall's τ). Generated the value function calibration figure. Created and contributed to the final report.

  Repository: `https://github.com/fozziejonathan/recap-dvf`
- **Aaditya Shah:** Wrote the project proposal and contributed to the poster. Implemented the RECAP advantage-conditioning pipeline—the prompt-injection of advantage tokens into Ψ_0's prefix and the advantage-conditioned flow-matching fine-tuning—and ran the 40k-step

RECAP training on Modal. Collected the `advantage: low` on-policy rollouts, trained the matched-data BC-all baseline, and collected its rollouts. Wrote the Results and Discussion sections of the report and prepared the LaTeX manuscript.

Repository: `https://github.com/aadityashah/Psi0-Recap`

Changes from Proposal Our experimental plan changed in three ways as the project progressed. First, we pivoted the target task: the proposal targeted G1WholebodyTabletopGraspMP-v0, but the SFT policy saturated at 98.2% success on that task, leaving no headroom for RECAP to demonstrate improvement, so we moved to the harder BendPickMP task (and screened out OpenFaucetTeleop, which the SFT policy could not solve at all). Second, we redesigned the evaluation around confound-controlled comparisons: an initial SFT-vs-RECAP comparison conflated data source (clean expert demonstrations vs. mixed-quality on-policy rollouts) with the learning method, so we added a matched-data behavior-cloning baseline (BC-all) and a within-model high-vs-low advantage contrast as the rigorous tests of the mechanism. Third, to keep the matched-data comparison clean, we fine-tuned RECAP on the on-policy rollouts alone rather than mixing in the expert demonstrations, ensuring RECAP and BC-all differ only in the presence of the advantage token.

AI Tools Disclosure

We used AI assistants in supporting roles only; all core experiments, implementations, and scientific claims were designed and executed by the team. During the literature review, AI tools helped summarize background papers for faster orientation. Claude was used to debug CUDA and local Ψ_0 environment setup, including package and dependency issues; it was not used for heavy implementation of the RECAP pipeline, value function, or training code. When developing the rollout collection script, we referenced public GitHub examples for structure and used ChatGPT for debugging assistance. For writing, ChatGPT and Writefull were used occasionally to fix LaTeX typos and grammatical errors in Overleaf. None of these tools generated the experimental results or substantive technical content of this report.

References

Marc G. Bellemare, Will Dabney, and Rémi Munos. 2017. A Distributional Perspective on Reinforcement Learning. In *International Conference on Machine Learning (ICML)*. PMLR, 449–458. `https://arxiv.org/abs/1707.06887` arXiv:1707.06887.

Yuhui Chen, Shuai Tian, Shugao Liu, Yingting Zhou, Haoran Li, and Dongbin Zhao. 2025. ConRFT: A Reinforced Fine-tuning Method for VLA Models via Consistency Policy. In *Proceedings of Robotics: Science and Systems (RSS)*. doi:`10.15607/RSS.2025.XXI.019` arXiv:2502.05450.

Kevin Frans, Seohong Park, Pieter Abbeel, and Sergey Levine. 2025. Diffusion Guidance Is a Controllable Policy Improvement Operator. *arXiv preprint arXiv:2505.23458* (2025). doi:`10.48550/arXiv.2505.23458`

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 770–778. doi:`10.1109/CVPR.2016.90` arXiv:1512.03385.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. 2022. Offline Reinforcement Learning with Implicit Q-Learning. In *International Conference on Learning Representations (ICLR)*. `https://arxiv.org/abs/2110.06169` arXiv:2110.06169.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. 2023. Flow Matching for Generative Modeling. In *International Conference on Learning Representations (ICLR)*. `https://arxiv.org/abs/2210.02747` arXiv:2210.02747.

Guanxing Lu, Wenkai Guo, Chubin Zhang, Yuheng Zhou, Haonan Jiang, Zifeng Gao, Yansong Tang, and Ziwei Wang. 2025. VLA-RL: Towards Masterful and General Robotic Manipulation with Scalable Reinforcement Learning. *arXiv preprint arXiv:2505.18719* (2025). doi:`10.48550/arXiv.2505.18719`

NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. 2025. GR00T N1: An Open Foundation Model for Generalist Humanoid Robots. *arXiv preprint arXiv:2503.14734* (2025). doi:10.48550/arXiv.2503.14734

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. 2019. Advantage-Weighted Regression: Simple and Scalable Off-Policy Reinforcement Learning. *arXiv preprint arXiv:1910.00177* (2019). doi:10.48550/arXiv.1910.00177

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, and Zhiyuan Zhou. 2025a. $\pi^*_{0.6}$: a VLA That Learns From Experience. *arXiv preprint arXiv:2511.14759* (2025). doi:10.48550/arXiv.2511.14759

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 Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. 2025b. $\pi_{0.5}$: a Vision-Language-Action Model with Open-World Generalization. *arXiv preprint arXiv:2504.16054* (2025). doi:10.48550/arXiv.2504.16054

Songlin Wei, Hongyi Jing, Boqian Li, Zhenyu Zhao, Jiageng Mao, Zhenhao Ni, Sicheng He, Jie Liu, Xiawei Liu, Kaidi Kang, Sheng Zang, Weiduo Yuan, Marco Pavone, Di Huang, and Yue Wang. 2026. Ψ_0: An Open Foundation Model Towards Universal Humanoid Loco-Manipulation. arXiv:2603.12263 [cs.RO] https://arxiv.org/abs/2603.12263

A Task Screening and Selection

An advantage-conditioning study requires a task that is neither saturated nor unsolvable under our data budget: a saturated task leaves no headroom for the method to demonstrate improvement, while an unsolvable one yields no successful rollouts and thus no learnable advantage signal. We screened two endpoints of the Ψ_0 SIMPLE benchmark and ruled both out—TabletopGraspMP, where the SFT policy saturated at 98.2% success, and OpenFaucetTeleop, where it achieved 0.0%.

We conduct all reported experiments on G1WholebodyBendPickMP-v0, which sits in the productive middle of this range and satisfies every requirement of the study.

1. **Headroom with signal:** the SFT policy reaches an intermediate ceiling, and the rollout-collection checkpoint succeeds on 36.0% of episodes, so on-policy data contains both successes and failures.
2. **Task difficulty:** the Unitree G1 humanoid must coordinate bending, reaching, and grasping motions to retrieve an object below its nominal reach envelope.
3. **Clean evaluation:** success is defined by a clear binary outcome, providing an unambiguous reward signal.
4. **Informative failures:** failures include mis-reaches, failed grasps, and timeouts, yielding useful credit-assignment signals.

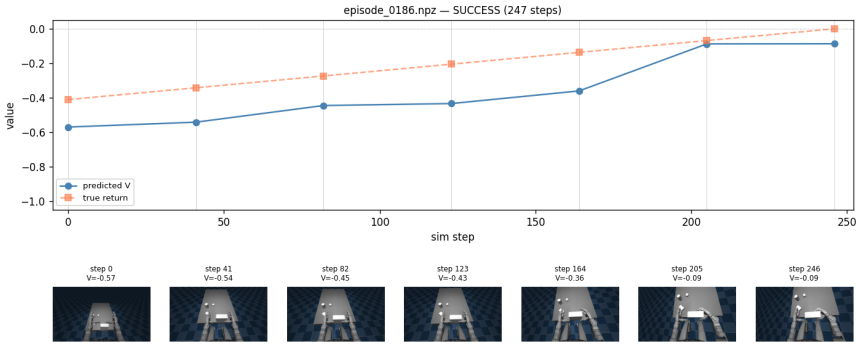

Figure 4: Predicted value $\hat{V}(o_t)$ vs. true return r_t for a held-out successful episode. The value function correctly assigns increasing values as the episode progresses toward task completion, consistent with the step-level MSE of 0.062 and step-level AUC of 0.837.

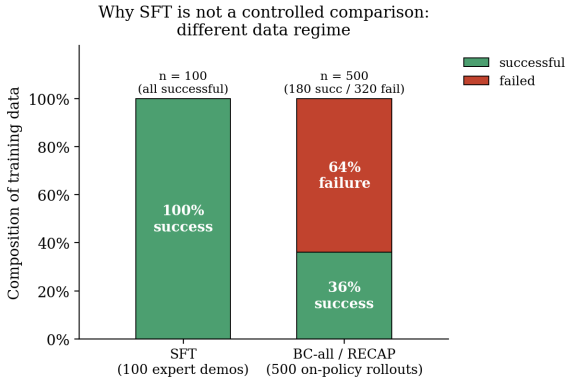

Figure 5: Training-data composition. SFT uses 100 all-successful expert demos; BC-all and RECAP use the same 500 on-policy rollouts (64% failures). Because the distributions differ, only BC-all (matched data) and RECAP-low (matched weights) are valid controls for the advantage-conditioning effect.

B Implementation Details

Distributional value-function architecture. The frozen ResNet-50 He et al. (2016) encodes each camera image (resized from 360×640 to 224×224 with ImageNet normalization) into a 2048-dimensional feature, which is concatenated with the 32-dimensional proprioceptive state and passed through a 3-layer MLP: $\texttt{Linear}(2080 \to 256) + \texttt{LayerNorm} + \texttt{ReLU} \to \texttt{Linear}(256 \to 256) + \texttt{ReLU} \to \texttt{Linear}(256 \to 201)$, producing logits over 201 uniformly-spaced return bins spanning $[-1, 0]$.

Value-function training. We minimize cross-entropy against the discretized true-return bin using Adam (learning rate 3×10^{-4}, batch size 256) for 50 epochs on a Modal cloud GPU. The dataset is split at the episode level 90/10, stratified by success/failure outcome with random seed 42 for reproducibility; the best checkpoint by validation cross-entropy is retained.

C Additional Experiments

Figures 5 and 6 supplement the confound-controlled comparisons in Section 5. Figure 5 visualizes why the expert-demonstration SFT baseline is not a valid control for RECAP; Figure 6 summarizes the two primary effect sizes with difference confidence intervals.

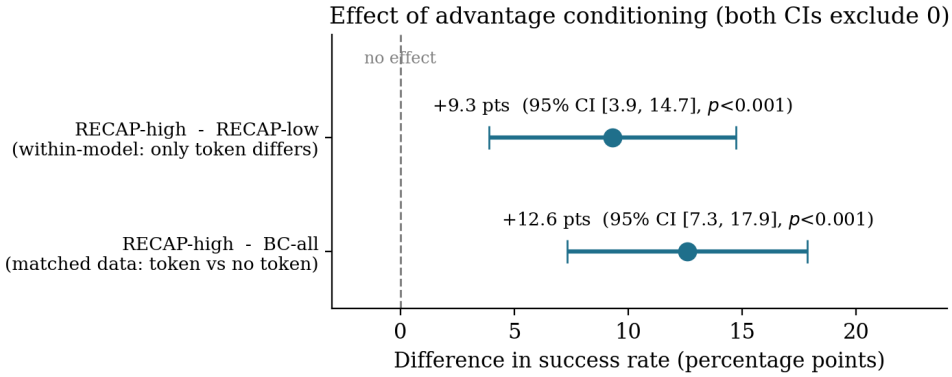

Figure 6: The two confound-controlled effects of advantage conditioning, as differences in success rate (percentage points) with 95% confidence intervals. Both intervals exclude zero. The within-model contrast (top) varies only the inference-time prefix token on a single policy; the matched-data contrast (bottom) holds the 500-rollout training set fixed and varies only the presence of the advantage token.