A PROBE CRITIC MATCHES A POLICY-SIZED CRITIC IN AGENTIC PPO

Anonymous authors
Paper under double-blind review

ABSTRACT

Proximal policy optimization (PPO) for language models typically uses a separate value network comparable in size to the policy, and agentic reinforcement learning has kept that design for trajectories that run to thousands of tokens across many turns. These policy-sized critics add considerable memory and compute and are pretrained before use. We show that a small probe trained on a language model agent's own activations can serve as its critic, at four orders of magnitude fewer parameters. We demonstrate that training the policy with such a probe nearly matches the performance of a critic the size of the policy, and holds that performance when the trajectories it learns from are several updates off-policy. We further find probe critics robust to how they are fit. A cold-start probe learns value alongside the policy, and a warm-start probe that never co-learns can still train the policy.

1 INTRODUCTION

Large language models (LLMs) increasingly act as agents that take many actions to complete a task, in domains such as web navigation, software engineering, and multi-step tool use (Sun et al., 2026; Jimenez et al., 2024; Yao et al., 2023). Post-training methods such as reinforcement learning drive much of the recent progress (Ouyang et al., 2022; Shao et al., 2024; Z.ai, 2026).

Proximal policy optimization (PPO) (Schulman et al., 2017) optimizes a clipped surrogate of the policy gradient objective and estimates the advantage against a learned value function. Classical actor-critic computes that value function with a small head on a backbone shared with the policy (Mnih et al., 2016). With transformer language models the critic became a policy-sized value network (Ouyang et al., 2022; Stiennon et al., 2022; Bai et al., 2022), and agentic post-training has kept that design (Hou et al., 2026). Agentic tasks are long-horizon and multi-turn. PPO computes a separate advantage for each token of such a trajectory. Critic-free group methods such as Group Relative Policy Optimization (GRPO) (Shao et al., 2024) give every token of a trajectory the same advantage. GLM-5.3 was post-trained with PPO (Hou et al., 2026; Z.ai, 2026). A policy-sized critic roughly doubles the model memory and requires additional compute for each update, and it is pretrained on the policy's own rollouts before reinforcement learning begins (Hou et al., 2026).

Recent agentic recipes set the GAE λ close to one over a whole trajectory (Yue et al., 2025; Hou et al., 2026). The return then reaches every token, and the advantage of a token is mostly the return minus the critic's value. The critic acts mainly as a baseline. A baseline does not change the expected policy gradient. It only lowers the gradient's variance. That is a smaller job than its policy-sized network suggests.

Linear probes read properties off intermediate layers that the network was never trained to expose (Alain & Bengio, 2018). Hidden states carry the board state of a game the model saw only as move sequences (Li et al., 2024), and carry directions linear enough to steer generation (Nanda et al., 2023; Park et al., 2024; Zou et al., 2023). Sparse probes localize such features to a handful of neurons (Gurnee et al., 2023). A linear probe on hidden states predicts a model's own uncertainty well enough to reduce hallucination (Ji et al., 2025). Despite the advent of more complex methods such as sparse autoencoders (Huben et al., 2024), supervised probes have been shown to match or beat them on concept detection (Kantamneni et al., 2025).

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.

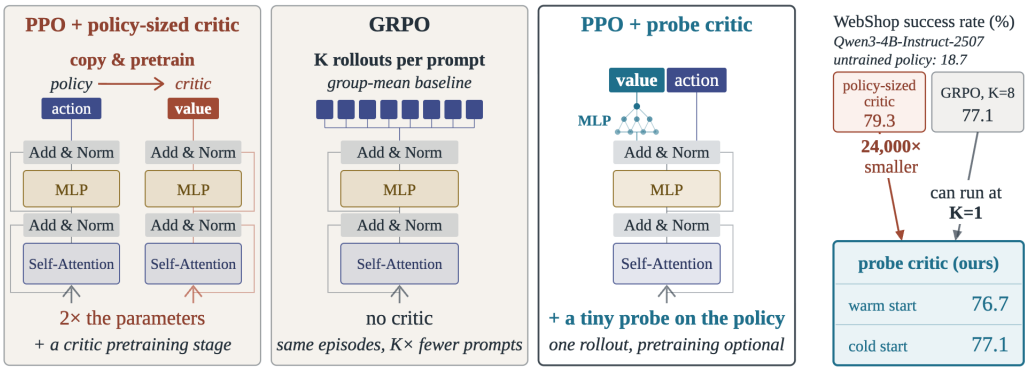

Figure 1: Three ways to estimate the advantage in agentic reinforcement learning. PPO with a policy-sized critic copies the policy and pretrains the copy before reinforcement learning. GRPO drops the critic and draws K rollouts per prompt. The probe critic is a small MLP on the policy's final hidden state, trained from one rollout per prompt, with a cold or warm start. Right: WebShop test success, mean of three seeds (Table 2).

We investigate whether a probe critic can read value estimates from the last layer of the policy model. Following the pretraining of policy-sized critics (Hou et al., 2026) and the random initialization of classical actor-critic (Mnih et al., 2016), we train language agents with warm-start and cold-start probe critics. Warm-start probes are pretrained on the training distribution before training the policy while cold-start probes train the policy from a Kaiming initialization and a zeroed head. The probe critic trains policies as good as those of a policy-sized critic on both models and both environments, at four orders of magnitude fewer parameters, and from a single rollout per prompt where GRPO uses eight.

Contributions. Our contributions are as follows.

1. We replace the policy-sized PPO critic with a probe critic on the policy's own activations, four orders of magnitude smaller (Figure 1, Table 2).
2. We show a cold-start probe critic, trained only alongside the policy, reaches parity with a policy-sized critic.
3. We show the probe critic trains on WebShop rollouts up to sixteen updates behind the learner without degrading task success, as does the policy-sized critic.
4. A probe fit once on the untrained policy and never updated still trains the policy in every setting, matching a co-learned probe on Qwen3-4B and trailing it on granite-4.1-3b (Section 5, Figure 3).

2 Method

2.1 Preliminaries

A policy π_θ acting over many steps receives one scalar from a verifier at the end of the episode, and every token of the trajectory is credited from that single number. A value function estimates the expected return from each state, and each token's advantage is measured against that estimate. We train language agents under this outcome supervision. We optimize the policy with the clipped surrogate objective of PPO (Schulman et al., 2017),

$$\mathcal{J}(\theta) = \mathbb{E}\Big[\tfrac{1}{T}\textstyle\sum_t \min\big(\rho_t(\theta)\,\hat{A}_t,\ \mathrm{clip}(\rho_t(\theta), 1-\epsilon, 1+\epsilon)\,\hat{A}_t\big)\Big], \tag{1}$$

with $\rho_t(\theta)$ the probability ratio between the current and sampling policies. The methods we compare differ only in $\hat{A}_t$, and specifically in whether estimating $\hat{A}_t$ needs a learned value function. Group relative methods replace the value function with the mean reward of several rollouts drawn from

the same state (Shao et al., 2024; Ahmadian et al., 2024). A group baseline requires visiting each state more than once. PPO learns a critic V_ϕ instead. A state-dependent baseline leaves the policy gradient unbiased (Williams, 1992; Greensmith et al., 2004). The critic reduces the variance of the gradient estimate without changing its expectation. Reinforcement learning from human feedback (RLHF) and agentic PPO make that critic a second network the size of the policy (Ouyang et al., 2022; Stiennon et al., 2022; Hou et al., 2026).

The critic we use departs from that one in two ways. The probe critic reads frozen features the policy has already computed and can start from a random initialization with no fit in advance. The probe's architecture is fixed by a decodability measurement made before any policy training begins (Section 2.4).

2.2 Advantage estimation

Writing s_t for the prefix at step t and a_t for the emitted token, a trajectory $\tau = (s_0, a_0, \ldots, s_{T-1}, a_{T-1})$ is scored by a verifier at termination, $r_{T-1} = R(\tau)$. The agent loop returns a single scalar per episode, placed at the final token. Every earlier reward is zero, $r_t = 0$ for $t < T-1$. We set $\gamma = 1$ throughout. We form $\hat{A}_t$ by generalized advantage estimation (Schulman et al., 2018), with $\delta_t = r_t + V_\phi(s_{t+1}) - V_\phi(s_t)$ and

$$\hat{A}_t^{(\lambda)} = \sum_{l=0}^{T-1-t} \lambda^l \, \delta_{t+l}. \tag{2}$$

We follow Hou et al. (2026) in the two choices that fix λ: the policy's λ is set per trajectory to the length-adaptive form of VAPO, Yue et al. (2025), $1 - 1/(1.5\,\ell)$ for a trajectory of ℓ generated tokens, and the critic's λ is held at 1.0. Yue et al. (2025) define ℓ for a single response, and we apply it to the whole multi-turn trajectory. The critic is regressed on the Monte Carlo return, the outcome the verifier returned for that trajectory. Unlike a bootstrapped target, it does not depend on the critic's own predictions. Following the same work, advantages are computed across action boundaries and environment observation tokens are skipped, since they are not the policy's actions.

The critic acts mainly as a baseline. Expanding Equation 2 with $r_t = 0$ before the final token and $V(s_T) = 0$ gives

$$\hat{A}_t = \lambda^{T-1-t}\big(R(\tau) - V(s_t)\big) + (1-\lambda) \sum_{j=1}^{T-1-t} \lambda^{j-1}\big(V(s_{t+j}) - V(s_t)\big), \tag{3}$$

where t and T index generated tokens, since the recursion skips environment observations. The first term is the return minus the critic's value at s_t. Its weight λ^{T-1-t} is about 0.51 at the first token of a trajectory and reaches one at the last. Averaged over a trajectory it is about 0.73. In this term the critic is the state-dependent baseline introduced above. It leaves the expected gradient unchanged and lowers its variance, which mainly shows up as faster learning. The second term carries the rest of the weight. It depends on how the critic's value changes over the rest of the episode.

With λ this close to one, most of what the critic contributes is a baseline. Serving as a baseline is a smaller task than bootstrapping accurate values over a long horizon as in traditional PPO. This suggests that recent PPO implementations demand less of the critic, which raises the question of whether it can be compressed.

2.3 A critic read from the policy

A probe on the policy's own features. Classical actor-critic reads the value function off representations the policy already computes (Mnih et al., 2016). We return to that arrangement. Let $h_t \in \mathbb{R}^d$ be the policy's final hidden state at s_t: the residual stream after the last transformer block and the final normalization. The final hidden state is the representation the language model head reads to choose a_t. The critic is a small multilayer perceptron probe on that vector,

$$V_\phi(s_t) = f_\phi\big(\mathrm{sg}[h_t]\big), \qquad f_\phi : \mathbb{R}^d \to \mathbb{R}, \tag{4}$$

where $\mathrm{sg}[\cdot]$ is the stop-gradient. The probe critic reads a hidden state the policy has already computed and needs no second copy of the model. The probe critic has $166{,}017$ parameters against the policy's

three to four billion. Probing studies find different information at different depths, with the later layers carrying the more abstract structure (Alain & Bengio, 2018; Tenney et al., 2019). We read at the last layer. The method assumes the return is recoverable from the hidden state by a probe. Section 2.4 measures how much of the return is.

The policy receives no critic gradient. In classical actor-critic the value head shares a backbone with the policy, and the value loss updates the representation the policy acts on. The stop-gradient in Equation 4 removes that path, and the policy's features are shaped only by the policy loss. This keeps value-regression gradients out of the policy. Hou et al. (2026) report that these gradients destabilize the attention layers, and freeze those layers during value training.

The critic is refit as the policy changes. The probe is fit to the returns of π_θ, which changes at every update. Both the return distribution and the features h_t change with it. After each policy step the probe takes M gradient steps on

$$\mathcal{L}(\phi) = \frac{1}{|\mathcal{B}_i|} \sum_{(s_t,\tau)\in\mathcal{B}_i} \big(f_\phi(h_t^{\theta_i}) - R(\tau)\big)^2 \tag{5}$$

over the rollouts of that step. Following Hou et al. (2026), who find one critic update per policy update too few when the policy changes quickly, we set $M = 2$. The mean and scale of $R(\tau)$ change as the policy improves. We regress a standardized target and denormalize at use, $V_\phi(s_t) = \sigma f_\phi(\mathrm{sg}[h_t]) + \mu$, with (μ, σ) running estimates of the return mean and scale (Hessel et al., 2018). The standardization reparameterizes V_ϕ and leaves the estimator unchanged.

Trajectories the learner did not generate. Asynchronous training separates the policy that produced a trajectory from the one being updated. A staleness bound η limits how far apart the two policies may be. A sample consumed at update i was generated under θ_j with $i - \eta \le j \le i$, and $\eta = 0$ recovers synchronous on-policy training. Whatever the bound, the learner recomputes h_t under θ_i rather than caching it from generation under θ_j. The features the probe reads are always the current policy's.

Initialization. Agentic value models are pretrained by regression before policy training begins (Hou et al., 2026). Pretraining requires a corpus of the policy's own rollouts collected in advance. Classical actor-critic instead initializes its value head at random and learns it alongside the policy (Mnih et al., 2016). We run the probe under both. A cold-start probe begins from a random initialization and learns value only during policy training. A warm-start probe is first fit on that corpus.

The cold-start probe starts with Kaiming hidden layers and a zero output layer and biases, so the untrained critic adds no random value bias. It learns value only from rollouts collected during training. A zero output layer passes no gradient to the hidden layers until the output weights become nonzero. At a critic learning rate of 5×10^{-6} the output weights stay near zero for the whole run. We train both initializations at a critic learning rate of 3×10^{-4}, which moves the output weights off zero within the first few updates (Appendix E). The warm-start probe is fit by the procedure of Section 2.4, on the same corpus, against the same λ=1 target, with the same $[64, 32]$ architecture. The two runs are otherwise configured identically, and both reach comparable task success (Section 4).

2.4 Choosing the probe

We fix the architecture before training any policy. We follow the usual actor-critic arrangement in which the critic is fit on the same task distribution the policy will train on. We run the frozen initial policy on every training task, at least two rollouts per task on WebShop's 3,072 tasks, 6,371 in total, and four per task on the 240 training tasks of the Berkeley Function Calling Leaderboard (BFCL). Each task's rollouts are split between fitting and scoring. One of the two on WebShop is held out, and 30% on BFCL. Validation and test tasks are never rolled out for this corpus. Features are the final-layer hidden states of generated tokens, paired with that rollout's λ=1 return, and each architecture is fit under three seeds (Appendix B).

Selection. We score each candidate on eight evenly spaced tokens per held-out rollout, so that every rollout counts equally, by its explained variance,

$$\mathrm{EV} = 1 - \frac{\min_c \mathbb{E}\big[(y - \hat{V} - c)^2\big]}{\mathrm{Var}(y)}, \tag{6}$$

Critic	Params	WebShop		BFCL	
		Qwen3-4B	granite-4.1-3b	Qwen3-4B	granite-4.1-3b
Linear readout	2.6K	0.091 ± 0.002	0.183 ± 0.001	0.268 ± 0.009	0.332 ± 0.003
[32]	82K	0.201 ± 0.009	0.281 ± 0.004	0.546 ± 0.010	0.521 ± 0.011
[64,32] *(deployed)*	166K	0.239 ± 0.015	0.288 ± 0.010	0.617 ± 0.014	0.540 ± 0.004
[128,64]	336K	0.217 ± 0.008	0.291 ± 0.002	0.629 ± 0.011	0.535 ± 0.013
[256,128]	689K	0.222 ± 0.024	0.290 ± 0.014	0.639 ± 0.014	0.563 ± 0.016
Causal attention	214K	0.196 ± 0.040	0.334 ± 0.005	0.644 ± 0.031	0.483 ± 0.022
no sequence mixing	197K	0.127 ± 0.007	0.263 ± 0.007	0.393 ± 0.047	0.385 ± 0.033
Policy-sized critic	3–4B	0.445	0.270	0.446	0.158

Table 1: Selection scores of candidate probe critics and the pretrained policy-sized critic, fit on the frozen policy's rollouts. Entries are explained variance, the fraction of return variance the critic explains. The deployed probe explains more return variance than the policy-sized critic in three of the four settings. Table 3 adds Spearman correlations. Every score is taken on eight evenly spaced tokens per held-out rollout, so every rollout counts equally. Probes are scored at the best held-out epoch, with one standard error over 3 seeds. The policy-sized critic is one pretrained build, scored on its own held-out half of the same corpus.

and by its Spearman correlation with the return (Appendix B, Table 3). Subtracting the minimizing constant makes EV invariant to a shift in the critic's output, and $\mathrm{EV} = 0$ is the score of predicting the mean. This is also how the policy-sized critic is scored. Pooling every token instead weights a rollout by its length. On Qwen3-4B WebShop, failed episodes are about four times longer than successful ones ($1{,}276$ against 325 generated tokens), so each failed rollout is almost four times as influential as a successful one. Explained variance levels off at $[64, 32]$ in all four settings, and we deploy $[64, 32]$, the smallest architecture on that plateau (Table 1). Causal attention over the sequence does not consistently improve the probe.

Both critics score comparably. We score the pretrained policy-sized critic of Appendix C on its own held-out rollouts of the same corpora under the same protocol (Table 1, last row). The probe matches or exceeds it in three of four settings, and the policy-sized critic scores higher on Qwen3-4B WebShop. Task success does not separate the two critics in any setting (Section 4).

3 Experimental Setup

Environments. We train on two agentic environments that differ in horizon, action space, and how a task is scored. WebShop (Yao et al., 2023) places an agent in a product catalogue and judges whether it buys the item an instruction specifies, within a 15-turn budget. BFCL v4 multi-turn (Patil et al., 2025) runs an agent against a stateful set of APIs over several user turns and checks the resulting API state against a reference execution. In both environments the verifier returns a single score at the end of the episode, and the critic predicts that score. We measure transfer on τ^2-bench retail (Barres et al., 2025), which adds a simulated user, played by `gpt-oss-120b` (OpenAI et al., 2025), that the agent must interact with and scores the database the episode leaves behind. No policy is trained on τ^2-bench. Each environment is reported on the metric its reward defines, and the metrics are not comparable to each other.

WebShop is split disjointly into $3072/128/500$ train, validation and test tasks. For BFCL we take the 600 entries of the three multi-turn variants that share an initial configuration space, and split them $240/120/240$ so that no initial configuration seen in training reappears at test. τ^2-bench uses its published retail split of 114 tasks, reported as avg@4, the mean per-task success rate over four trials, and scored on the database state alone. Split construction, sampling temperatures and validation cadence are in Appendix A. The policies are Qwen3-4B-Instruct-2507 (Yang et al., 2025a) and granite-4.1-3b (IBM Granite Team, 2026), each trained on both environments. Comparisons are within each model.

Training protocol. Every method except GRPO trains from a single rollout per prompt (Hou et al., 2026) on the same tasks, shares all other hyperparameters, and differs only in how it estimates per-

Table 2: Synchronous (η=0) test-split performance, in percent, with the gain over the untrained policy beneath each row. The τ^2 row is a transfer evaluation of the BFCL-trained policies. BFCL uses our own configuration-disjoint split and is not comparable to the public leaderboard. Every trained cell is the mean of three independent seeds, the untrained column is three sampled evaluations of the same policy, and errors are the standard error across them. No comparison between the probe critic, GRPO and the policy-sized critic separates at three seeds.

	Untrained	GRPO (K=8)	Probe critic (ours)		Policy-sized critic
			cold start	warm start	
Model: *Qwen3-4B-Instruct-2507*					
WebShop *(success, avg@4)*	18.70 ± 0.05	77.08 ± 2.18	77.08 ± 1.78	76.72 ± 1.28	79.28 ± 0.29
Δ *untrained*		↑58.4	↑58.4	↑58.0	↑60.6
BFCL *(pass rate, avg@16)*	12.97 ± 1.99	22.42 ± 0.72	24.70 ± 1.51	21.90 ± 0.92	23.39 ± 1.52
Δ *untrained*		↑9.5	↑11.7	↑8.9	↑10.4
τ^2-bench retail *(avg@4, transfer)*	35.53 ± 3.28	37.06 ± 1.22	37.86 ± 1.69	36.40 ± 1.52	36.99 ± 1.02
Δ *untrained*		↑1.5	↑2.3	↑0.9	↑1.5
Model: *granite-4.1-3b*					
WebShop *(success, avg@4)*	12.30 ± 1.27	80.10 ± 0.98	80.80 ± 2.21	79.40 ± 2.15	82.15 ± 1.27
Δ *untrained*		↑67.8	↑68.5	↑67.1	↑69.9
BFCL *(pass rate, avg@16)*	8.44 ± 1.47	29.61 ± 2.18	27.89 ± 3.24	25.19 ± 0.60	23.66 ± 1.01
Δ *untrained*		↑21.2	↑19.5	↑16.8	↑15.2
τ^2-bench retail *(avg@4, transfer)*	36.84 ± 3.61	39.84 ± 0.32	41.89 ± 2.13	39.40 ± 0.19	37.72 ± 0.91
Δ *untrained*		↑3.0	↑5.1	↑2.6	↑0.9

state value. The budget is 64 episodes per update over 200 updates on WebShop and 80 over 100 on BFCL, at staleness η=0 except where Section 4 sweeps it. Every task-success result we report is the mean of three seeds. Appendix I states the scope of these comparisons.

Critics. The probe critic is the $[64, 32]$ probe of Section 2, run under both initializations. The policy-sized critic is the value network of RLHF and agentic PPO (Ouyang et al., 2022; Stiennon et al., 2022; Hou et al., 2026). The policy-sized critic is initialized from the policy, pretrained by regression on returns over the same frozen-policy corpus the probe sweep uses with its attention layers frozen as in Hou et al. (2026), and then co-learned with the policy with no layers frozen. The probe critic trains at 3×10^{-4} (Appendix E) and the policy-sized critic at 10^{-5}. Appendix C describes it in full.

We run the policy-sized critic only with pretraining, since Hou et al. (2026) find initializing a value network this size without pretraining a major bottleneck. GRPO (Shao et al., 2024) replaces the critic with the mean return of a group of rollouts drawn from the same prompt. We compare against GRPO at the matched episode budget of Hou et al. (2026), with K=8 rollouts from each of one eighth as many prompts.

4 Results

4.1 Synchronous training

A probe critic on the policy's final hidden state trains policies as good as those of a critic the size of the policy. Table 2 reports test success for every method. The cold-start probe critic and the policy-sized critic are not separable on either environment or model. The ordering differs by environment, with the policy-sized critic ahead on WebShop and the cold-start probe ahead on BFCL. Pooling each environment's runs, normalized by the gain over the untrained policy, the cold-start probe attains 97% of the policy-sized critic's gain on WebShop, with a 95% interval of 94% to 101%, and 120% on BFCL, with an interval of 97% to 148%. Both exclude a shortfall larger than about 6% of the critic's gain (Appendix G).

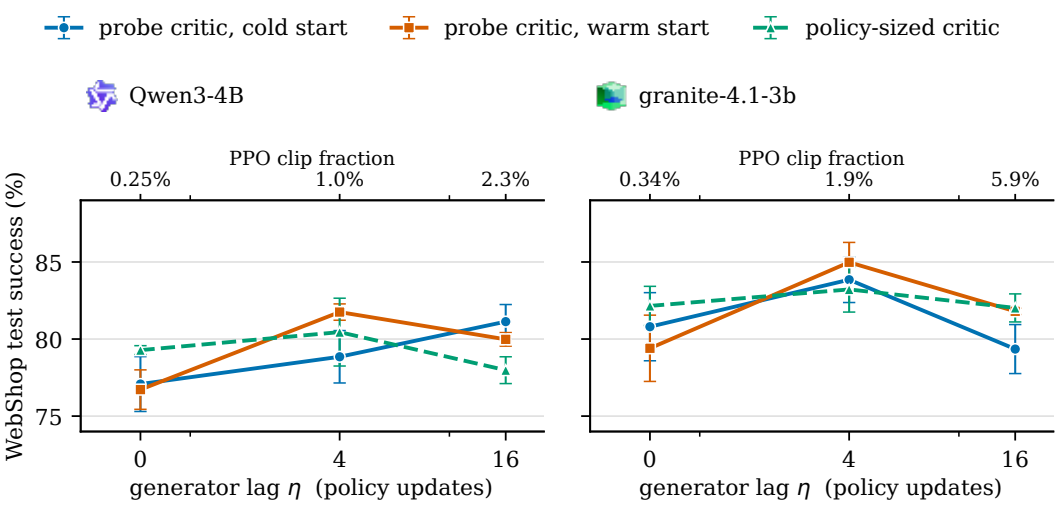

Figure 2: WebShop test success against generator lag η, the number of policy updates by which the rollout generator trails the trainer, on both models at three seeds. The cold-start and warm-start probe critic and the policy-sized critic are not separable at three seeds. η=0 is synchronous. The rise at η=4 on granite-4.1-3b is within seed noise. The top axis gives the mean PPO clip fraction over training at each lag. Points are the mean of three seeds and bars are the standard error across them.

The probe critic also matches the strongest critic-free baseline. GRPO reaches the same task success within the seed spread on both environments and both models, drawing eight rollouts from each of one eighth as many prompts. A per-state value estimate at one rollout per prompt does as well as a within-prompt group baseline at eight. We also evaluate a critic-free single-rollout method and find it unstable. Appendix D reports REINFORCE++, which replaces the critic with a scalar baseline. REINFORCE++ scores below every run in Table 2 on BFCL with its policy entropy rising throughout training, stays at the untrained policy's success on granite-4.1-3b, and collapses on WebShop with or without the entropy bonus.

The parity holds on both environments and both model families, over episodes that average 9 turns on WebShop and 14 on BFCL. The three critics also learn at about the same rate (Figure 5, Appendix H). A critic that acts mainly as a baseline mostly changes how fast the policy learns. Neither the probe nor the policy-sized critic learns clearly faster. On τ^2-bench retail, which none of the policies trained on, every BFCL-trained policy gains little over the untrained policy and no method separates from another.

The probe critic can be trained from a cold start. The cold-start probe does as well as the warm-start one, and scores above it on all four training settings of Table 2 and on both transfer evaluations, though no single comparison separates. The cold-start probe reaches parity with the policy-sized critic without a corpus collected from the frozen policy or a fit before reinforcement learning. The probe critic also removes the policy-sized critic's 64 GB of VRAM and about 160 GPU-minutes of pretraining (Appendix F).

4.2 Asynchronous training

In asynchronous settings, the probe critic is robust to generator lag. Figure 2 reports WebShop test success at $\eta \in \{0, 4, 16\}$ for both probe initializations and for the policy-sized critic. The clip fraction grows by an order of magnitude from η=0 to η=16, and the probe critic's task success does not degrade. The policy-sized critic also scores about the same at every lag, and the two critics stay in parity throughout. The probe critic does so without a second copy of the model.

5 A PROBE CRITIC FIT ONCE TRAINS THE POLICY

5.1 HOLDING THE PROBE FIXED

Hou et al. (2026) identify critic staleness as a central difficulty and update the critic more often than the policy to contain it. We test the opposite extreme, a probe critic that is never updated. We take the warm-start probe, fit once on the untrained policy's rollouts, and switch off its updates for the whole run, with no value optimizer and no value gradient. Its weights stay fixed, while the running return mean and scale that denormalize its output keep updating, as in every run. This fixed probe trains the policy in every setting (Figure 3). On WebShop the fixed probe trains the policy from 18.70% untrained to 78.67% on Qwen3-4B and from 12.30% to 73.22% on granite-4.1-3b. The fixed probe does not separate from either co-learned probe on Qwen3-4B and trails both on granite-4.1-3b, by 6 to 8 points on WebShop and 3 to 5 on BFCL.

5.2 HOW CO-LEARNED AND FIXED PROBES DIFFER

We examine which part of the hidden state the co-learned probe reads and whether it moves away from the fixed probe as the policy changes. The fixed probe responds only to the 64-dimensional subspace of the hidden state spanned by the rows of its first layer. The co-learned probe starts from the same weights and is refit on every batch. We measure how much of the co-learned probe's first layer W_1 stays in the fixed probe's subspace by the subspace overlap, the share of its squared norm there, $S(W_1) = \frac{\|W_1 Q\|_F^2}{\|W_1\|_F^2}$, with Q an orthonormal basis of the fixed probe's first-layer row space. As a control, a random subspace of the same size gives $S = 64/2560 = 2.5\%$, and a random first layer drawn from the covariance of the hidden states gives 3% to 4% on WebShop and 1.5% to 3% on BFCL. We also count the directions of the hidden state along which the co-learned probe's value changes, by the participation ratio (PR) of the Gram matrix G of its unit value gradients $u = (\partial V_\phi/\partial h)/\|\partial V_\phi/\partial h\|$ over 512 token states per update, $\mathrm{PR} = (\operatorname{tr} G)^2/\|G\|_F^2$ with $G = \mathbb{E}\left[uu^\top\right]$. Because each gradient is normalized, PR depends only on which directions of the hidden state the value gradients point along, not on their size. A linear readout has a participation ratio of exactly one, and a constant has no gradient at all.

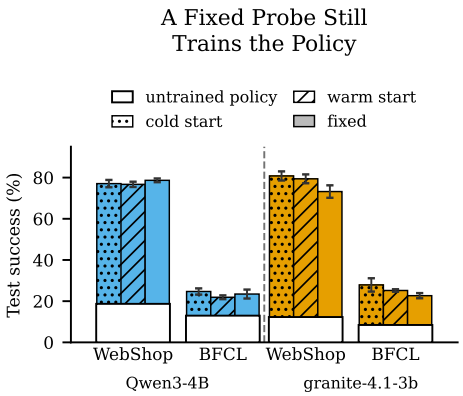

Figure 3: Test success for a probe fit once on the untrained policy and never updated (fixed), against the co-learned cold-start and warm-start probes, on both environments and both models at three seeds.

Figure 4 reports both over training. On Qwen3-4B the co-learned warm-start probe's first layer barely moves, changing by 5% to 8% in norm, and stays inside the fixed probe's subspace. On granite-4.1-3b it changes by 43% to 73% and drifts partly out of that subspace. On both models the overlap is far above the covariance-matched control. The fixed probe does not separate from the co-learned warm-start probe on Qwen3-4B and trails it on granite-4.1-3b.

The warm-start probe's value changes along one or two directions in both models on both environments, close to one on Qwen3-4B WebShop and about two elsewhere. The cold-start probe starts from a random first layer, and its value changes along more directions. In some settings its PR never settles and keeps fluctuating until the end of training. We track the top direction of the warm-start probe's value gradients on a fixed set of hidden states. On Qwen3-4B it keeps an absolute cosine above 0.98 with its direction early in training throughout the run on both environments. On granite-4.1-3b it ends at 0.81 on WebShop and 0.50 on BFCL, as the probe leaves the fixed probe's subspace. Two random directions in the hidden space have an expected absolute cosine of 0.016. We discuss interpretation of the direction in Appendix I.

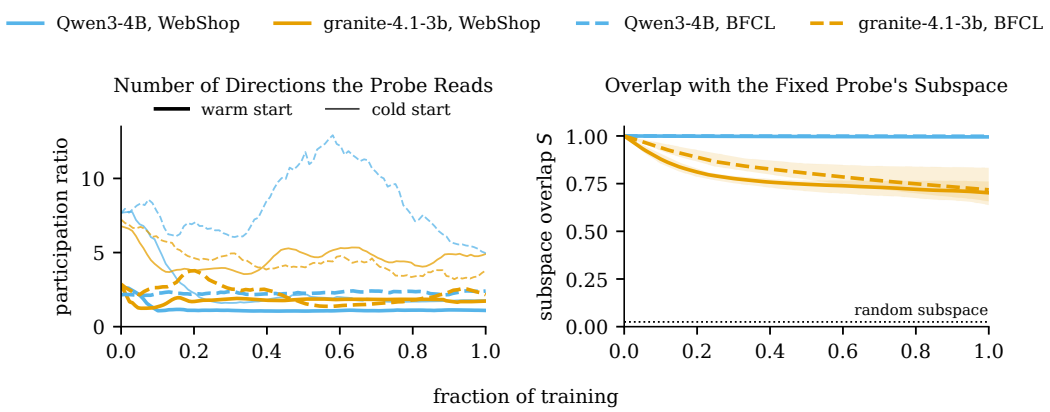

Figure 4: The co-learned probes on WebShop and BFCL. **Left:** participation ratio of the value gradients of the warm-start (thick) and cold-start (thin) probes, showing the warm-start probe's value changes along few directions and the cold-start probe's along more. **Right:** subspace overlap S of the warm-start probe's first layer with the fixed probe's, mean and range over three seeds. The cold-start probe has no fixed probe to compare against. The two Qwen3-4B curves coincide at 1 while granite's critics slowly drift downward.

6 Related Work

PPO often utilizes a value network the size of the policy (Stiennon et al., 2022; Ouyang et al., 2022; Zheng et al., 2023). PPO with a value network is often used on long reasoning traces (Yue et al., 2025) and in agentic training (Hou et al., 2026; Z.ai, 2026). VinePPO replaces the value network with Monte Carlo estimates from resampled intermediate states, arguing that learned critics assign credit poorly (Kazemnejad et al., 2025). Other work reduces the critic's cost with a shared backbone or an adapter (Mnih et al., 2016; Jin et al., 2023). Critic-free methods subtract a leave-one-out or batch-normalized baseline (Ahmadian et al., 2024; Hu et al., 2025), and GRPO and its variants normalize rewards within a group of rollouts drawn from the same prompt (Shao et al., 2024; Guo et al., 2025; Zheng et al., 2025; Yang et al., 2025b; Gao et al., 2025). Turn-level variants extend the group to intermediate trajectory states (Feng et al., 2025), and single-rollout methods track a baseline across updates (Xu & Ding, 2025).

Probes are small classifiers trained on a network's intermediate representations to measure what each layer encodes (Alain & Bengio, 2018; Tenney et al., 2019). Language models encode many high-level concepts as directions in activation space (Elhage et al., 2022; Park et al., 2024), often only a few of them (Hernandez & Andreas, 2021; Siu et al., 2025b), and steering methods edit such directions to change behavior (Zou et al., 2023; Turner et al., 2023; Siu et al., 2025a). Probes recover whether the model treats a statement as true (Burns et al., 2023), detect hallucination (Chen et al., 2024), and predict the model's own uncertainty (Ji et al., 2025) and whether its answer will be correct (Kadavath et al., 2022; Orgad et al., 2025).

7 Conclusion

We present a probe critic for agentic PPO, a small multilayer perceptron on the policy's frozen final hidden state in place of a value network the size of the policy. The probe critic trains policies matching those of a critic four orders of magnitude larger on both models and both environments. It can be trained from a cold start alongside the policy, and holds its performance when trajectories trail the learner by up to sixteen updates. We also find that the probe critic's value changes along as few as one or two directions of the policy's own representation. A probe fit on the untrained policy and never updated still trains the policy in every setting tested.

AI USE STATEMENT

In this work, we used generative AI tools to implement methods, clean and reformat datasets, and design ablation experiments. We have not used generative AI tools to generate synthetic datasets, formulate hypotheses, or interpret results, and theoretical models, proofs, translation, and qualitative or thematic data analysis are not applicable to this work. Additionally, we used generative AI tools to create and edit software code, create and modify figures, search for information and identify relevant literature, format references, and draft and edit parts of the paper. We have reviewed all AI-assisted work. The authors checked AI-generated code, figures, and text against experimental outputs and logs, and checked every cited reference against its arXiv or venue record. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We will release code and configurations upon acceptance. Appendix A gives the training configuration, splits and evaluation protocol, Section 2.4 and Appendix B the probe corpus and selection, and Appendix C the policy-sized critic.

REFERENCES

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms, 2024. URL https://arxiv.org/abs/2402.14740.

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes, 2018. URL https://arxiv.org/abs/1610.01644.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022. URL https://arxiv.org/abs/2204.05862.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-bench: Evaluating conversational agents in a dual-control environment, 2025. URL https://arxiv.org/abs/2506.07982.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=ETKGuby0hcs.

Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. INSIDE: llms' internal states retain the power of hallucination detection. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=Zj12nzlQbz.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition, 2022. URL https://arxiv.org/abs/2209.10652.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training, 2025. URL https://arxiv.org/abs/2505.10978.

Chang Gao, Chujie Zheng, Xiong-Hui Chen, Kai Dang, Shixuan Liu, Bowen Yu, An Yang, Shuai Bai, Jingren Zhou, and Junyang Lin. Soft adaptive policy optimization, 2025. URL https://arxiv.org/abs/2511.20347.

Evan Greensmith, Peter L. Bartlett, and Jonathan Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. *J. Mach. Learn. Res.*, 5:1471–1530, December 2004. ISSN 1532-4435.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645 (8081):633–638, September 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z. URL http://dx.doi.org/10.1038/s41586-025-09422-z.

Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. Finding neurons in a haystack: Case studies with sparse probing, 2023. URL https://arxiv.org/abs/2305.01610.

Evan Hernandez and Jacob Andreas. The low-dimensional linear geometry of contextualized word representations. In Arianna Bisazza and Omri Abend (eds.), *Proceedings of the 25th Conference on Computational Natural Language Learning*, pp. 82–93, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.conll-1.7. URL https://aclanthology.org/2021.conll-1.7.

Matteo Hessel, Hubert Soyer, Lasse Espeholt, Wojciech Czarnecki, Simon Schmitt, and Hado van Hasselt. Multi-task deep reinforcement learning with popart, 2018. URL https://arxiv.org/abs/1809.04474.

Zhenyu Hou, Yujiang Li, Jie Tang, and Yuxiao Dong. Single-rollout asynchronous optimization for agentic reinforcement learning, 2026. URL https://arxiv.org/abs/2607.07508.

Jian Hu, Jason Klein Liu, Haotian Xu, and Wei Shen. Reinforce++: Stabilizing critic-free policy optimization with global advantage normalization, 2025. URL https://arxiv.org/abs/2501.03262.

Robert Huben, Hoagy Cunningham, Logan Riggs, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=F76bwRSLeK.

IBM Granite Team. Granite 4.1, 2026. URL https://huggingface.co/ibm-granite/granite-4.1-3b.

Ziwei Ji, Lei Yu, Yeskendir Koishekenov, Yejin Bang, Anthony Hartshorn, Alan Schelten, Cheng Zhang, Pascale Fung, and Nicola Cancedda. Calibrating verbal uncertainty as a linear feature to reduce hallucinations, 2025. URL https://arxiv.org/abs/2503.14477.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. Swe-bench: Can language models resolve real-world github issues?, 2024. URL https://arxiv.org/abs/2310.06770.

Yizhao Jin, Greg Slabaugh, and Simon Lucas. Adapter-rl: Adaptation of any agent using reinforcement learning, 2023. URL https://arxiv.org/abs/2311.11537.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. Language models (mostly) know what they know, 2022. URL https://arxiv.org/abs/2207.05221.

Subhash Kantamneni, Joshua Engels, Senthooran Rajamanoharan, Max Tegmark, and Neel Nanda. Are sparse autoencoders useful? a case study in sparse probing, 2025. URL https://arxiv.org/abs/2502.16681.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. Vineppo: Refining credit assignment in rl training of llms, 2025. URL https://arxiv.org/abs/2410.01679.

Kenneth Li, Aspen K. Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Emergent world representations: Exploring a sequence model trained on a synthetic task, 2024. URL https://arxiv.org/abs/2210.13382.

Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning, 2016. URL https://arxiv.org/abs/1602.01783.

Neel Nanda, Andrew Lee, and Martin Wattenberg. Emergent linear representations in world models of self-supervised sequence models. In Yonatan Belinkov, Sophie Hao, Jaap Jumelet, Najoung Kim, Arya McCarthy, and Hosein Mohebbi (eds.), *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 16–30, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.blackboxnlp-1.2. URL https://aclanthology.org/2023.blackboxnlp-1.2.

OpenAI, :, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, Che Chang, Kai Chen, Mark Chen, Enoch Cheung, Aidan Clark, Dan Cook, Marat Dukhan, Casey Dvorak, Kevin Fives, Vlad Fomenko, Timur Garipov, Kristian Georgiev, Mia Glaese, Tarun Gogineni, Adam Goucher, Lukas Gross, Katia Gil Guzman, John Hallman, Jackie Hehir, Johannes Heidecke, Alec Helyar, Haitang Hu, Romain Huet, Jacob Huh, Saachi Jain, Zach Johnson, Chris Koch, Irina Kofman, Dominik Kundel, Jason Kwon, Volodymyr Kyrylov, Elaine Ya Le, Guillaume Leclerc, James Park Lennon, Scott Lessans, Mario Lezcano-Casado, Yuanzhi Li, Zhuohan Li, Ji Lin, Jordan Liss, Lily, Liu, Jiancheng Liu, Kevin Lu, Chris Lu, Zoran Martinovic, Lindsay McCallum, Josh McGrath, Scott McKinney, Aidan McLaughlin, Song Mei, Steve Mostovoy, Tong Mu, Gideon Myles, Alexander Neitz, Alex Nichol, Jakub Pachocki, Alex Paino, Dana Palmie, Ashley Pantuliano, Giambattista Parascandolo, Jongsoo Park, Leher Pathak, Carolina Paz, Ludovic Peran, Dmitry Pimenov, Michelle Pokrass, Elizabeth Proehl, Huida Qiu, Gaby Raila, Filippo Raso, Hongyu Ren, Kimmy Richardson, David Robinson, Bob Rotsted, Hadi Salman, Suvansh Sanjeev, Max Schwarzer, D. Sculley, Harshit Sikchi, Kendal Simon, Karan Singhal, Yang Song,

Dane Stuckey, Zhiqing Sun, Philippe Tillet, Sam Toizer, Foivos Tsimpourlas, Nikhil Vyas, Eric Wallace, Xin Wang, Miles Wang, Olivia Watkins, Kevin Weil, Amy Wendling, Kevin Whinnery, Cedric Whitney, Hannah Wong, Lin Yang, Yu Yang, Michihiro Yasunaga, Kristen Ying, Wojciech Zaremba, Wenting Zhan, Cyril Zhang, Brian Zhang, Eddie Zhang, and Shengjia Zhao. gpt-oss-120b & gpt-oss-20b model card, 2025. URL https://arxiv.org/abs/2508.10925.

Hadas Orgad, Michael Toker, Zorik Gekhman, Roi Reichart, Idan Szpektor, Hadas Kotek, and Yonatan Belinkov. Llms know more than they show: On the intrinsic representation of llm hallucinations, 2025. URL https://arxiv.org/abs/2410.02707.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=UGpGkLzwpP.

Shishir G. Patil, Huanzhi Mao, Charlie Cheng-Jie Ji, Fanjia Yan, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. The berkeley function calling leaderboard (bfcl): From tool use to agentic evaluation of large language models. In *Forty-second International Conference on Machine Learning*, 2025.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation, 2018. URL https://arxiv.org/abs/1506.02438.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Vincent Siu, Nicholas Crispino, David Park, Nathan W. Henry, Zhun Wang, Yang Liu, Dawn Song, and Chenguang Wang. Steeringsafety: A systematic safety evaluation framework of representation steering in llms, 2025a. URL https://arxiv.org/abs/2509.13450.

Vincent Siu, Nathan W. Henry, Nicholas Crispino, Yang Liu, Dawn Song, and Chenguang Wang. Repit: Steering language models with concept-specific refusal vectors, 2025b. URL https://arxiv.org/abs/2509.13281.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback, 2022. URL https://arxiv.org/abs/2009.01325.

Yiyou Sun, Xinyang Han, Weichen Zhang, Yuanbo Pang, Tianyu Wang, Yuhan Cao, Yixiao Huang, Chris Duroiu, Haoyun Zhang, Jeffrey Lin, Weishu Zhang, Tyler Zeng, Ying Yan, Bo Liu, Hanson Wen, Mingyang Xu, Xiaoyuan Liu, Zimeng Chen, Weiyan Shi, Amanda Dsouza, Vincent Sunn Chen, Patrick Bryant, Carl Boettiger, Yamini Rangan, Bradley Rothenberg, Kyle Steinfeld, Arvind Rao, Tapio Schneider, Georgios Yannakakis, Laure Zanna, Kaan Ozbay, Ida Sim, Tarek Zohdi, George Em Karniadakis, Jack Gallant, Teresa Head-Gordon, Yushan Li, Wenxi Deng, Tao Sun, Huiqi Wang, Zhun Wang, Justin Xu, Chris Yuhao Liu, Yafei Cheng, Rongwang Hu, Aras Bacho, Shengcao Cao, Zengyi Qin, Yixiong Chen, Hengduan Fan, Hao Liu, Lin Zeng,

Shashank Muralidhar Bharadwaj, Litian Gong, Yingxuan Yang, Maojia Song, Ruheng Wang, Zongzheng Zhang, Honglin Bao, Shuo Lu, Jianhong Tu, Zhonghua Wang, Zheng Zhang, Zijiao Chen, Yanqiong Jiang, Zhendong Li, Bohan Lyu, Chang Ma, Peiran Xu, Benran Zhang, Shangding Gu, Haoyue Hua, Haoyang Li, Wanzhe Liao, Chengzhi Liu, Junbo Peng, Haoran Sun, Zechen Xu, Bo Chen, Jiayi Cheng, Yi Jiang, Keying Kuang, Yuan Li, Youbang Pan, Ziyan Rao, Alexander Schubert, Yifan Shen, Vincent Siu, Xiatao Sun, Kangqi Zhang, Xiaopan Zhang, Yuchen Zhu, Ishaan Singh Chandok, Lei Ding, Jingxuan Fan, Andrew Glover, Jiaming Hu, Yiran Hu, Wenbo Huang, Zixin Jiang, Haoran Jin, Lukas Kim, Ming Liu, Yang Liu, Alireza Rafiei, Xuhuan Shen, Kunyang Sun, Sophia Sun, Ting Sun, Eric Wang, Yixin Wang, Hanwen Xing, Sihan Xu, Yuzheng Xu, Zhongxing Xu, Zhiling Yan, Boqin Yuan, Ruiqi Zhang, Yifan Zhang, Zibo Zhao, Liana, Santanu Bosu Antu, Haoyue Bai, Carlo Bosio, Joseph Cavanagh, Patricia Cavazos-Rehg, Tianxing Chen, Xuewen Chen, Yipu Chen, Chenyu Zhu, Chen Dai, Stefano De Castro, Yunfu Deng, Kaustubh Dhole, Jiayuan Ding, Chenchen Du, Zhehang Du, Hao Fan, Run-Ze Fan, Hengyu Fu, Shi Gu, Yifan Gu, Charlie Guo, Baihe Huang, Baixiang Huang, Rimika Jaiswal, Zhihan Jiang, Ran Jin, Erin Kasson, Xin Lan, Joseph Lee, Deren Lei, Chenyu Li, Daofeng Li, Haitao Li, Hongwei Li, Jingyan Li, Xiao Li, Yi Li, Yinsheng Li, Yuangang Li, Zhixu Li, Wenyu Liang, Longtai Liao, Kevin Qinghong Lin, Andy Zeyi Liu, Che Liu, Jiaming Liu, Kaiyuan Liu, Xuan Liu, Pan Lu, Wenbo Lv, Yicheng Lyu, Qiuyang Mang, Kyle Montgomery, Yuzhou Nie, Ruoxi Ning, Jorin Overwiening, Xu Pan, Layna Paraboschi, Core Francisco Park, Justin Purnomo, Swati Rajwal, Scott Rankin, Bixuan Ren, Yiren Rong, HaoYang Shang, Ventus Shaw, Fiona Shen, Jiawei Shen, Minqi Shi, Shi Qiu, Huaxiu Yao, Tianneng Shi, Jonah So, Vladislav Susoy, Hannah Szlyk, Haocheng Wang, Jialu Wang, Wei Wang, Xinyu Wang, Zehao Wang, Dowling Wong, Angela Wu, Dehao Wu, Fangyu Wu, Mengyuan "Millie" Wu, Yu Wu, Yuchen Wu, Yuhao Wu, Qingpo Wuwu, Weihang Xiao, Yongyi Xiong, Fan Xu, Ruiling Xu, Mingxuan Yan, Benjamin Yang, Jirong Yang, Sen Yang, Xiaoli Yang, Yushi Yang, Haoran Ye, Xiaohu Yu, Zhengming Yu, Chenlong Zhang, Chi Zhang, Hanning Zhang, Hanwen Zhang, Junge Zhang, Kunpeng Zhang, Song Zhang, Wenjin Zhang, Wenshuo Zhang, Ying Zhang, Yizhi Zhang, Brian Zhao, Qijian Zhao, Yimin Zhao, Yuhaohua Zheng, Liwei Zhou, Tianyue Zhou, Sichen Zhu, Siqi Zhu, Yan Zhu, Yishu Zhu, Jierui Zuo, Chonghao Cai, Helena Casademunt, Wenjia Chen, Cheng Cheng, Nawen Deng, Rao Fu, Tianfu Fu, Yifan Han, He Ren, Zhenyu He, Qiao Jin, Langlang Li, Yuetai Li, Sylvia Liu, Lu Lu, Luqing Zhou, Subhabrata Mukherjee, Yunqi Ouyang, Yin Ren, Dawei Shi, Haoran Wu, Zhiyue Wu, Hannah Yao, Zhuoran Yi, Jenny Yu, Rhea Zhan, Hang Zhou, Blake Zhu, Junfan Zhu, Alan Yuille, Yang Liu, Russell Alan Poldrack, Jiachen Li, Zhenglu Li, Molei Tao, Jing Huang, Wenqi Shi, Costas Spanos, Lichao Sun, Chenguang Wang, Orson Xu, Zhen Dong, Hector Gomez, Aylin Caliskan, Ali Emami, Haimin Hu, Zhi Li, Lihui Liu, Murphy Niu, Yi Shao, Jianxin Sun, Mikko Tolonen, Ting Wang, Sanjiv Das, Yanjun Gao, Wenbo Guo, Erika J Schneider, Zhiyong Lu, Yian Ma, Mark Mueller, Radha Poovendran, Somayeh Sojoudi, Yinglun Zhu, and Dawn Song. Agents' last exam, 2026. URL https://arxiv.org/abs/2606.05405.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. Bert rediscovers the classical nlp pipeline, 2019. URL https://arxiv.org/abs/1905.05950.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering, 2023. URL https://arxiv.org/abs/2308.10248.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8(3–4):229–256, May 1992. ISSN 0885-6125. doi: 10.1007/BF00992696. URL https://doi.org/10.1007/BF00992696.

Zhongwen Xu and Zihan Ding. Single-stream policy optimization, 2025. URL https://arxiv.org/abs/2509.13232.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger

Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025a. URL https://arxiv.org/abs/2505.09388.

Shihui Yang, Chengfeng Dou, Peidong Guo, Kai Lu, Qiang Ju, Fei Deng, and Rihui Xin. Dcpo: Dynamic clipping policy optimization, 2025b. URL https://arxiv.org/abs/2509.02333.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents, 2023. URL https://arxiv.org/abs/2207.01206.

Yu Yue, Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, TianTian Fan, Zhengyin Du, Xiangpeng Wei, Xiangyu Yu, Gaohong Liu, Juncai Liu, Lingjun Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Ru Zhang, Xin Liu, Mingxuan Wang, Yonghui Wu, and Lin Yan. Vapo: Efficient and reliable reinforcement learning for advanced reasoning tasks, 2025. URL https://arxiv.org/abs/2504.05118.

Z.ai. GLM-5.3: Frontier coding with emergent cyber capabilities. https://z.ai/blog/glm-5.3, 2026. Accessed 2026-08-14.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization, 2025. URL https://arxiv.org/abs/2507.18071.

Rui Zheng, Shihan Dou, Songyang Gao, Yuan Hua, Wei Shen, Binghai Wang, Yan Liu, Senjie Jin, Qin Liu, Yuhao Zhou, Limao Xiong, Lu Chen, Zhiheng Xi, Nuo Xu, Wenbin Lai, Minghao Zhu, Cheng Chang, Zhangyue Yin, Rongxiang Weng, Wensen Cheng, Haoran Huang, Tianxiang Sun, Hang Yan, Tao Gui, Qi Zhang, Xipeng Qiu, and Xuanjing Huang. Secrets of rlhf in large language models part i: Ppo, 2023. URL https://arxiv.org/abs/2307.04964.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to ai transparency, 2023. URL https://arxiv.org/abs/2310.01405.

A Implementation details

The probe reads the policy's final-layer post-normalization hidden state, the tensor consumed by the language-model head. Both models have hidden width 2560. The staleness sweep covers $\eta \in \{4, 16\}$ against the synchronous η=0 anchor, enforced by bounding how many policy versions a consumed trajectory may trail the learner. **Splits and episode handling.** WebShop tasks are partitioned disjointly by index into 3072 train, 128 validation and 500 test instructions. BFCL v4 takes the 600 entries of `multi_turn_base`, `multi_turn_miss_func` and `multi_turn_miss_param`, groups them into connected components of the graph joining entries by initial configuration and task index, and assigns whole components to 240 train, 120 validation and 240 test entries. An initial configuration seen in training never reappears at test. This is a held-out split of our own construction and its scores are not comparable to leaderboard reports. An episode ends when the verifier reports completion, when the agent emits a terminating action, or at the step budget. The budget is 15 turns on WebShop and 64 during BFCL training, raised to the harness default of 100 at evaluation. Truncated episodes are kept and scored as they stand. Dropping them would bias the return distribution toward short successes.

Policy optimization. PPO with $\gamma = 1$ and generalized advantage estimation, with the policy's λ set per trajectory to $1 - 1/(1.5\,\ell)$ for a trajectory of ℓ generated tokens, the critic's target held at $\lambda = 1$, and observation tokens skipped in the advantage recursion. The clip range is symmetric at 0.2, one PPO epoch per batch, train batch 64 trajectories on WebShop and 80 on BFCL with minibatches of 32 and 40, and gradients clipped to norm 1.0. The entropy bonus is 2×10^{-3} for every method. Every method on every environment applies a token-level KL penalty against the

initial policy at coefficient 10^{-2}. Runs are 200 steps on WebShop and 100 on BFCL. Each trained entry of Table 2 is the mean of three seeds, as is the probe sweep of Appendix B. Training rollouts sample at temperature 1.0. Every 10 updates on WebShop and every 5 on BFCL the current policy is scored on the validation split and the highest-scoring checkpoint is kept. The reported test number is sampled at temperature 0.7, with $K{=}4$ samples per task on WebShop and $K{=}16$ on BFCL. Responses are budgeted at 16,384 tokens, with prompts at 4,096. The τ^2-bench transfer evaluation runs the published retail protocol, with four trials per task, a 200-step budget, a `gpt-oss-120b` user simulator, and scoring on the final database state with no model judge.

Optimizers and learning rates. AdamW throughout, weight decay 10^{-2}, no warmup. The policy trains at 10^{-6}, the probe critic at 3×10^{-4}, and the policy-sized critic at 10^{-5}. The gap between the policy and the probe critic is deliberate and is derived in Appendix E: a probe whose output layer starts at zero has no gradient signal until that layer leaves zero, and at a critic learning rate of 5×10^{-6} it barely leaves zero within a run.

The probe and its fit. The deployed probe is $\mathrm{Linear}(2560, 64)$, GELU, dropout 0.1, $\mathrm{Linear}(64, 32)$, GELU, dropout 0.1, $\mathrm{Linear}(32, 1)$: 166,017 parameters. For each generated token it reads the final layer post-normalization hidden state from which the policy sampled that token, with observation tokens excluded. Online it is fit by mean squared error against the $\lambda = 1$ return under a stop-gradient, two updates per policy update, on a standardized target that is denormalized at use. The offline sweep of Section 2.4 fits the same architectures on 6,371 frozen-policy rollouts for 30 epochs under three seeds.

The asynchronous pipeline. Asynchrony is simulated with a FIFO of policy weight versions. At staleness bound η the rollouts consumed at step t are generated by $\theta_{t-\eta}$. At $\eta = 0$ nothing is installed and the run is byte-identical to the synchronous one at the same advantage settings. The $\eta = 0$ points of Figure 2 are the synchronous runs of Table 2.

B PROBE ARCHITECTURE SELECTION

		WebShop				BFCL			
		Qwen3-4B		granite-4.1-3b		Qwen3-4B		granite-4.1-3b	
Critic	Params	EV	Spearman	EV	Spearman	EV	Spearman	EV	Spearman
Linear readout	2.6K	0.091 ± 0.002	0.295 ± 0.004	0.183 ± 0.001	0.348 ± 0.004	0.268 ± 0.009	0.521 ± 0.010	0.332 ± 0.003	0.537 ± 0.001
[32]	82K	0.201 ± 0.009	0.366 ± 0.004	0.281 ± 0.004	0.388 ± 0.003	0.546 ± 0.010	0.699 ± 0.009	0.521 ± 0.011	0.590 ± 0.004
[64,32] *(deployed)*	166K	0.239 ± 0.015	0.376 ± 0.007	0.288 ± 0.010	0.389 ± 0.005	0.617 ± 0.014	0.717 ± 0.012	0.540 ± 0.004	0.565 ± 0.005
[128,64]	336K	0.217 ± 0.008	0.366 ± 0.005	0.291 ± 0.002	0.390 ± 0.002	0.629 ± 0.011	0.723 ± 0.008	0.535 ± 0.013	0.585 ± 0.003
[256,128]	689K	0.222 ± 0.024	0.371 ± 0.011	0.290 ± 0.014	0.391 ± 0.006	0.639 ± 0.014	0.727 ± 0.009	0.563 ± 0.016	0.578 ± 0.009
Causal attention	214K	0.196 ± 0.040	0.368 ± 0.013	0.334 ± 0.005	0.399 ± 0.003	0.644 ± 0.031	0.722 ± 0.018	0.483 ± 0.022	0.571 ± 0.011
no sequence mixing	197K	0.127 ± 0.007	0.329 ± 0.005	0.263 ± 0.007	0.387 ± 0.006	0.393 ± 0.047	0.613 ± 0.035	0.385 ± 0.033	0.549 ± 0.012
Policy-sized critic	3–4B	0.445	0.470	0.270	0.403	0.446	0.556	0.158	0.363

Table 3: Explained variance and Spearman correlation of candidate probe critics and the pretrained policy-sized critic, fit on the frozen policy's rollouts. EV is the fraction of return variance the critic explains, and Spearman correlation measures only whether it orders states correctly. Every score is taken on eight evenly spaced tokens per held-out rollout, so every rollout counts equally. Probes are scored at the best held-out epoch, with one standard error over 3 seeds. The policy-sized critic is one pretrained build, scored on its own held-out half of the same corpus.

The sweep reported in Section 2.4 fits every candidate probe on one fixed corpus of the policy's own rollouts, harvested with the policy frozen, and scores each on a held-out half of that corpus. The corpus is 6,371 WebShop rollouts over 3,072 tasks, hidden width 2560, regression target the $\lambda{=}1$ return, no centering, thirty epochs, three seeds. Each fit is scored at its best held-out epoch. The deployed probe has neither a held-out fold nor a stopping rule and is refit continuously against a moving policy.

What the selection scores measure. With a terminal reward and $\gamma{=}1$ the target at every token of an episode is the return $R(\tau)$, so explained variance measures a critic's pointwise error against $V^*(s_t) = \mathbb{E}[R \mid s_t]$, above a floor set by the variance of the outcome given s_t. In Equation 3 this is the error that enters the first term. The second term depends only on changes in the critic's value within an episode, so an error shared by every token of an episode lowers both scores but cancels

there. A score aligned with the second term, such as agreement with Monte Carlo estimates from resampled intermediate states (Kazemnejad et al., 2025), is left to future work.

Anisotropy. Hidden states are strongly anisotropic. The top principal component of the mean-centered features carries 15% of their variance, and the top eight carry 51%.

C THE POLICY-SIZED CRITIC

The policy-sized critic is a separate network the size of the policy, initialized from it, warm-started by regression on the policy's own rollouts, and then trained online for the run. Its pretraining corpus is drawn from the same rollouts as the probe's. That corpus is bounded by the task set rather than by rollout budget. BFCL has 240 training tasks, so further rollouts add return samples of the same tasks rather than new tasks. A critic the size of the policy is fit to 960 rollouts there, and a probe with four orders of magnitude fewer parameters is fit to the same rollouts. The two critics are fit on identical data and differ only in what they read. The policy-sized critic adapts its own backbone, while the probe reads the policy's features without adapting them.

Building this critic correctly costs more than its parameter count suggests. Under AdamW at this recipe's learning rate, a bfloat16 parameter receives updates below half a unit in the last place and rounds back to its previous value. An earlier build of this critic was, tensor by tensor, 91.8% bit-identical to the policy it was initialized from. It was in effect a frozen backbone with a trained value probe, and its training loss did not reveal it. The fix is fp32 master weights, which raise its weight memory to roughly three times that of bfloat16 weights. Table 5 counts the full training state, including gradients and optimizer moments, at 16 bytes per parameter. Every policy-sized critic result in Table 2 uses the fp32 build.

The closest prior single-rollout asynchronous system initializes both its policy and its value model from a model fine-tuned for tool-integrated reasoning on data generated by a 120B model (Hou et al., 2026). Its results incorporate a much larger teacher. Our critic and policy both initialize from the same model, with no larger teacher.

D REINFORCE++

REINFORCE++ (Hu et al., 2025) replaces the value function with a scalar baseline subtracted from the return. We run it with the configuration of Table 2 and change only the advantage estimator. Table 4 reports BFCL test success. On both models REINFORCE++ scores below every individual run in Table 2, and on granite-4.1-3b it stays at the success of the untrained policy. Its validation score stops improving within 10 steps on Qwen3-4B and never exceeds its starting value on granite-4.1-3b, and its policy entropy rises about threefold on both. We stop each BFCL run once its validation score has plateaued and evaluate its best checkpoint.

Table 4: BFCL test success (%) of REINFORCE++, one run per model, against the untrained policy and the cold-start probe critic of Table 2.

	Untrained	REINFORCE++	Cold-start probe
Qwen3-4B	12.97	17.66	24.70 ± 1.51
granite-4.1-3b	8.44	8.52	27.89 ± 3.24

On WebShop, REINFORCE++ collapses on Qwen3-4B. Validation reward, out of 10, falls from 2.27 before training to 0.47 after 10 steps. Over 20 steps the mean response length grows by up to $1.5\times$, the training score halves and policy entropy rises $2.4\times$. The collapse does not come from the entropy bonus. A run with the entropy coefficient set to zero shows the same pattern, with response length growing by up to $1.7\times$ and the training score falling from 1.72 to 0.47 within 13 steps.

We report one run per model and environment rather than three seeds. The collapse lies far outside the seed spread of Table 2, and a collapsing policy generates long responses that make every further run and its evaluation slow.

Table 5: Training cost of the two critics on WebShop. Parameters and training state are computed from the parameter count, with 16 bytes per parameter for fp32 weights, gradients and Adam moments. Step time and GPU-hours are measured on one node of 8 GPUs over 200 updates, averaged over the six probe runs (both initializations) and the three policy-sized critic runs of Table 2. Our implementation recomputes the policy's hidden states for the probe, and the probe's step time includes that forward pass.

	Probe critic	Policy-sized critic
Critic parameters	166k	4.02B
Critic training state	2.7 MB	64 GB
Critic pretraining	under 1 CPU-minute	160 GPU-minutes
Time per step, Qwen3-4B	50.9 s	55.4 s
Time per step, granite-4.1-3b	44.4 s	48.0 s
GPU-hours per run, Qwen3-4B	23.6	26.5
GPU-hours per run, granite-4.1-3b	21.1	22.3

E WHY THE CRITIC LEARNING RATE IS 3×10^{-4}

The cold-start probe initializes the hidden layers of f_ϕ under Kaiming and sets the output layer and all biases to zero. $V_{\phi_0} \equiv 0$. Writing the probe as $V_\phi(s) = w^\top g(s) + b$ with g the penultimate activation, the gradient of the squared error with respect to the hidden parameters is proportional to w. At $w = 0$ that gradient vanishes. The hidden layers cannot move until the output layer does.

AdamW normalizes each coordinate by a running estimate of its own gradient magnitude. A single step moves any parameter by at most about the learning rate, and over n updates the output layer travels at most $O(n\alpha_\phi)$ from its initialization, writing α_ϕ for the critic learning rate to keep it distinct from the staleness η. At a critic learning rate of 5×10^{-6}, that budget over a full run is a small fraction of the probe's own weight scale. A probe starting at a fitted solution is unaffected. A probe starting at exactly zero changes its output only slightly within the run. The cold-start probe needs a larger critic learning rate, one that moves w off zero within the first few updates.

Both initializations run at the same critic learning rate, 3×10^{-4}. We find 5×10^{-6} creates a degenerate cold-start probe, since the output layer never meaningfully leaves zero within the run. Raising the rate to 3×10^{-4} leaves the warm-start probe stable. We did not tune the critic learning rate for either initialization. The cold-start probe's rate was chosen only to satisfy the constraint above.

F TRAINING COST

Our implementation is a research prototype built on verl and is not optimized for speed. The probe critic runs as a patch inside the trainer and recomputes the policy's hidden states in a separate forward pass rather than reusing the policy's own. The step times in Table 5 compare two critics within one unoptimized stack and are not a measure of achievable throughput. Within that stack, a training run with the probe critic takes 5% to 11% fewer GPU-hours than one with the policy-sized critic.

G POOLED COMPARISON WITH THE POLICY-SIZED CRITIC

Three seeds per cell are too few to separate methods cell by cell. We pool the runs within each environment, normalizing each run's success by its setting's gain over the untrained policy, and form 95% intervals by a bootstrap that resamples runs within each setting. The cold-start probe attains 97% of the policy-sized critic's gain on WebShop, with an interval of 94% to 101%, and 120% on BFCL, with an interval of 97% to 148%. Both intervals exclude a shortfall of more than about 6% of the critic's gain, against critic gains of 10 to 70 points.

H LEARNING CURVES

Figure 5 shows validation reward over training for the three critics of Table 2.

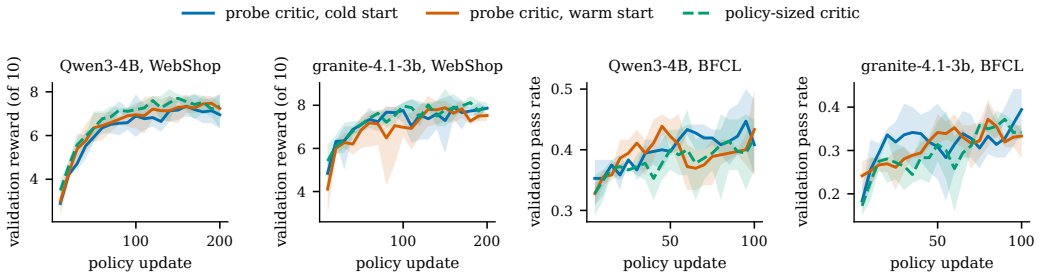

Figure 5: Validation reward over synchronous training for the cold-start and warm-start probe critics and the policy-sized critic, mean of three seeds with the range shaded. WebShop reward is out of 10 and BFCL is the pass rate. The three critics learn at about the same rate.

I LIMITATIONS

The comparison is anchored at two model families, two training environments with a third for transfer, and one policy-sized critic, the policy-initialized network of Appendix C. The probe critic and that critic are not separable on either model or environment. The staleness sweep runs on WebShop only. τ^2-bench retail is a transfer evaluation of the BFCL-trained policies. Its row measures transfer to an unseen domain. We choose the probe's architecture with coarse scores of between-episode prediction (Section 2.4), so a smaller probe may suffice. Section 5 is consistent with this, since a probe fit once and never updated still trains the policy. Semantic interpretation of the directions along which the co-learned probe's value changes (Section 5.2) is difficult. Much interpretability work studies middle layers rather than the last one, and the read direction rotates during training on granite-4.1-3b. We therefore leave this avenue of research to future work.