Learning Structured Trust Policies from Uncertainty, Advisor Signals, and Agreement

Extended Abstract

Gia Ancone Jaden Chen (gancone@stanford.edu, jaden1@stanford.edu)

Motivation. When should an agent act on advice of unknown reliability? Reinforcement learning (RL) agents that consult peers (such as ensembles of language-model advisors) and humans weighing a colleague's opinion both face this problem: the advisor's competence is hidden, varies, and may be correlated with superficial cues like confidence, fluent reasoning, or an agreeable answer. Hence, our project's goal is to explore methods for training a *structured trust policy* in the spirit of the learning-to-defer framework, built from the same cues a person uses when deciding whether to take advice: *self-uncertainty*, the *advisor's ethos* (tone and perceived reasoning quality), and *agreement* between the advisor and the agent's own answer. Prior work either assumes reliability away, places a prior over it, or hardcodes single-signal trust rules that break when the signal is correlated or adversarially manipulated. We ask whether an agent can instead learn to combine these imperfect cues and infer an advisor's reliability online, from the outcomes of its own advisor consultations.

Method. We frame trust as a multistep RL problem. Each episode is a 10-step interaction with an advisor whose hidden reliability $r \in \{0.2, 0.5, 0.8\}$ is sampled at episode start. At each step, the agent observes an 8-dimensional state: four per-task signals (a learned self-uncertainty probe $\hat{p}$, advisor tone, advisor reasoning quality, and agent-advisor agreement) and four running episode statistics (step progress, consult rate, observed advisor accuracy, agreement rate). Then, the agent chooses to trust its own answer or the advisor's, and is rewarded for a correct selection. We train a PPO actor-critic and compare two variants under an identical back-loaded reliability curriculum: *Heuristic* (vanilla PPO with mild early-consult shaping) and *Adaptive*, which adds forced first disagreement-consults and an "advisor-save" reward bonus to make the reliability signal learnable.

Implementation. We built a synthetic environment of 50,000 four-choice questions across three reasoning families (arithmetic, probability, Pearl-style causal/counterfactual) and three difficulties, with simulated per-question agent answer distributions, simulated advisor (answer, tone, reasoning-quality cue), and a frozen Kadavath-style MLP uncertainty probe. We implemented PPO (clipped surrogate, GAE, entropy bonus) with two-layer $\tanh$ MLPs, trained for 2500 updates over 3 seeds, and evaluated against three fixed-action baselines and adversarial / out-of-distribution conditions.

Results. No learned policy beats the trivial `always-trust-own` baseline on raw per-step accuracy, a consequence of the joint distribution, not of poor learning. The interesting result lives in metrics that ignore the dataset's statistical bias. On a balanced *composite* metric, Adaptive wins at every reliability condition by 8–10 pp; it defers 2–18$\times$ more often than Heuristic when deferral is correct; and its reliability-sensitivity slope is always positive ($+0.069 \pm 0.011$) where Heuristic's is ~ 0 and seed dependent. Adaptive's per-metric seed variance is 5–8$\times$ lower than Heuristic's. Ablating the uncertainty probe drops accuracy 6.1 pp and inflates seed variance 5–10$\times$, revealing the probe as a training time regularizer. Under OOD, Adaptive extrapolates cleanly to unseen reliabilities (beating `always-trust-own` by $+9.6$ pp at $r{=}0.95$) but correctly collapses below random under a *directional* attack that inverts its dominant signal.

Discussion. Our takeaways are methodological. First, the shaping ingredients are what make learning work: forced first disagreement-consults and an advisor-save reward bonus turn an otherwise dead reliability signal into a learnable one, and without them the policy collapses to a near `always-trust-own` rule. Second, aggregate accuracy is the wrong objective; balanced composites, multiseed variance, ablations, and directional adversarial tests separate a genuine trust mechanism from a lucky heuristic. Finally, an ablation reveals that the self-uncertainty probe acts as a training time stabilizer rather than an inference signal, a role that basic linear feature analysis would miss.

Conclusion. Combining step-level signals with running statistics, forced exploration, and targeted reward shaping produces a reproducible multisignal trust policy that genuinely infers reliability online. The shaping ingredients, not raw reward, are the contribution, and the right evaluation lens is decision quality under shift, not accuracy on the training distribution.

Learning Structured Trust Policies from Uncertainty, Advisor Signals, and Agreement

Gia Ancone
Department of Computer Science
Stanford University
gancone@stanford.edu

Jaden Chen
Department of Computer Science
Stanford University
jaden1@stanford.edu

Abstract

We study whether a reinforcement learning (RL) agent can learn a *structured trust policy*: a per-step decision rule that integrates multiple imperfect advisor cues and infers an unknown advisor's reliability online. We frame trust as a multistep RL problem over a synthetic 50,000-question environment in which an advisor's reliability is hidden and re-sampled each episode. A PPO actor-critic observes a self-uncertainty probe, advisor tone, advisor reasoning quality, agreement, and running consultation statistics, and chooses whom to trust at each of ten steps. We show that raw per-step accuracy is a misleading objective, it is dominated by the trivial `always-trust-own` baseline because of the task's joint distribution, and that the real signal appears only in bias-free decision-quality metrics. Our *Adaptive* variant, which adds forced exploration and an advisor-save reward bonus, produces a reproducible multisignal mechanism: it wins the balanced composite at every reliability condition (+8–10 pp), defers 2–18$\times$ more when deferral is correct, has a multiseed-robust positive reliability slope, and exhibits 5–8$\times$ lower seed variance than an unshaped baseline. An ablation reveals the uncertainty probe is a regularizer at training time rather than a deployment-time channel, and adversarial tests confirm that the policy genuinely relies on advisor reasoning quality: inverting that cue is 3–5$\times$ more damaging than randomizing it.

1 Introduction

Just as a person weighs a colleague's recommendation against their own judgment, multiagent systems are slowly evolving to match human interactions: an RL agent may consult an ensemble of tools or language-model advisors of heterogeneous and hidden competence. Hence, agents are increasingly acting on advice from individuals whose reliability they cannot directly observe. In real world interactions, humans decide whether to trust someone based on numerous cues, including how confident they are in their own answer, the advisor's ethos (how it sounds and whether its reasoning seems sound), and whether the advisor agrees with them. We decided to implement a learned trust policy based on the same signals.

Existing approaches do not directly employ a multisignal, interaction-history-based policy inspired by real-life social navigation. Self-uncertainty estimation supports introspection, but typically stops short of a downstream action rule; studies of advisor tone and sycophancy characterize speaker-side behavior, not the listener's trust decision; learning-to-defer is supervised and typically binary, without per-instance advisor signals at inference; and action-advising RL decides when to advise but leaves the student's trust rule hand-coded. We review this past work in Section 2.

We instead ask: can an agent learn, by reinforcement, a structured trust policy that fuses multiple cues and infers reliability online? We make the problem a genuine multistep RL task. Reliability is hidden and constant within an episode, so the agent must infer it from the cumulative past outcomes

of its own consultations, and we study not just whether such a policy can be trained, but how to tell whether it is reasoning or merely exploiting the statistical shortcuts of our environment.

Our contributions are: (1) a multistep trust-policy formulation and a controlled, LLM-agnostic synthetic environment with a learned self-uncertainty probe; (2) a set of shaping ingredients (including forced first disagreement-consults and an advisor-save bonus) that turn an otherwise dead reliability signal into a learnable one and yield a reproducible multisignal policy; (3) an environment-appropriate evaluation methodology: bias-free composite accuracy scores, multiseed variance, feature ablations, and an adversarial / OOD suite that distinguishes genuine trust from shortcut learning; and (4) the discovery that features can be supportive in unintentional and contradictory ways, namely the uncertainty probe was crucial as a training-time regularizer but did not necessarily improve composite accuracy.

2 Related Work

Gauging self-certainty. Kadavath et al. [2022] train a separate $P(\text{IK})$ head that predicts the probability a model *knows* the answer to a question, without conditioning on any specific proposed answer, supervised on whether the base model actually answered correctly across tasks such as TriviaQA, arithmetic, and code synthesis. They find that large models are well calibrated on multiple-choice and true/false formats, that $P(\text{IK})$ partially generalizes to out-of-distribution tasks, and that it responds sensibly to evidence, rising when relevant source material is in context. This directly motivates our frozen self-uncertainty probe, which we train Kadavath-style to estimate own-answer correctness. However, while their work stops at introspection, our question is what an agent should *do* with such an estimate, especially in combination with several other cues. Because Guo et al. [2017] show that modern networks are systematically overconfident and that simple post-hoc temperature scaling restores calibration, we measure our probe by Brier score and calibration rather than accuracy alone, and treat its raw output as a signal for calibration, rather than taken at face value.

Advisor cues and sycophancy. Mielke et al. [2022] study whether a chatbot's verbalized confidence (hedged "I'm not sure" vs. assertive wording) matches its factual accuracy in closed-book QA. Their pipeline first predicts answer correctness from the model's own internal representations, and then it routes that estimate through a controllable-generation model that rerenders the reply with matching linguistic confidence. Crucially, the base model's out-of-the-box tone does *not* track correctness. Sharma et al. [2023] probe five production assistants with SycophancyEval and, analyzing the human-preference data behind RLHF with a Bayesian logistic model, show that "matching the user's stated beliefs" is among the most predictive features of which response humans prefer. This means that agreement and confident phrasing are rewarded largely independently of truth. While these works calibrate or diagnose speaker-side behavior and are proof-of-concept, neither feeds tone or agreement into a listener's trust decision. We encode their lesson directly into the environment: our scripted advisor's tone is independent of correctness and agreement carries no privileged evidence, while its reasoning-quality cue is only imperfectly coupled to correctness. This means that a policy which leans on persuasive-but-hollow cues is exposed under distribution shift, exactly the sycophantic failure mode our agent must learn to resist.

Learning to defer. Madras et al. [2018] cast trust as adaptive rejection: rather than a constant reject penalty, their model learns end-to-end when to "pass" a case to a fixed downstream decision-maker, with the deferral coupled to that decision-maker's actual loss, motivated by building fairer composite pipelines. Mozannar and Sontag [2020] formalize the same problem as a single augmented $(K+1)$-class task (K labels plus a "defer" action) and derive the first consistent surrogate loss (a generalization of softmax cross-entropy) for jointly learning the classifier and rejector from samples of a fixed expert's decisions. In both, deferral is supervised, single-step, and conditioned only on the agent's own input: per-instance advisor signals such as tone, expressed confidence, or reasoning never enter the rejector at inference, and the expert's reliability is static and known offline. Our setting keeps their loyalty-versus-deferral framing, which directly inspires our bias-free composite metric. However, our pipeline is sequential and reward-driven, with a hidden, episode-level advisor reliability the agent must infer online from the outcomes of its own consultations.

Advice in RL. Torrey and Taylor [2013] introduce the budgeted teacher-student paradigm: a trained teacher may suggest an action on a limited number of steps, and the contribution is a family of

teacher-side heuristics for spending that budget. One of the key principles is "importance advising", which advises when the gap $\max_a Q(s,a) - \min_a Q(s,a)$ is large. They also employ "mistake correcting", which advises only when the student is about to deviate from the teacher's policy. When advice is given, the student simply executes it; its "trust rule" is hardcoded obedience conditioned on no advisor cues. Subramanian et al. [2023] extend this to multiple independent advisors of unknown, heterogeneous quality with a two-level Q-learning architecture: a high-level value estimates each advisor's per-state reliability and selects among them, while a low-level value learns action quality directly, letting the agent learn to ignore bad advice. Even here, trust is collapsed to a scalar estimate of advisor value, and the choice is *which advisor* to follow rather than whether to trust an advisor over the agent's *own* answer given rich cues. We make that cue-conditioned incorporation the learned object: the student's structured trust policy, not the teacher's budget or an advisor-selection value, is what we optimize. Our environment's causal-reasoning family draws on Pearl and Mackenzie [2018] and the CLadder benchmark [Jin et al., 2023], our distribution-shift tests follow the spirit of contrast sets [Gardner et al., 2020], and we optimize with PPO [Schulman et al., 2017] and GAE [Schulman et al., 2016].

3 Method

3.1 Pipeline overview

The agent interacts with the environment through a fixed per-step loop, repeated for $T{=}10$ steps per episode; experience from 16 episodes then forms one PPO update. Each episode samples 10 question tasks under a single hidden advisor reliability r, and a single step proceeds as follows.

1. **Self-uncertainty probe ($\hat{p}$).** A frozen MLP which we trained maps the agent's own 4-way answer distribution $p_{ag} \in \mathbb{R}^4$, together with $\max p_{ag}$, the entropy $H(p_{ag})$, and a task-family one-hot, to a calibrated personal-correctness estimate $\hat{p} \in [0,1]$. The probe is trained offline (Kadavath-style) and held fixed during RL.
2. **Advisor cues.** A scripted advisor emits an answer that is correct with probability governed by the episode's hidden reliability r, a tone (confident vs. hedged) independent of correctness, and a reasoning-quality flag that is positively, but imperfectly coupled to correctness.
3. **State assembly ($s_t \in \mathbb{R}^8$).** Our four per-task signals are $\hat{p}$, advisor tone, advisor reasoning quality, and agent-advisor agreement. They are concatenated with four episode-level running statistics, step progress, consult rate, observed advisor accuracy, and agreement rate.
4. **Decision and reward.** The PPO actor outputs the trust action $a_t \in \{\text{trust own}, \text{trust advisor}\}$ and the critic outputs the value $V(s_t)$. The selected answer is scored, giving reward $R_t = \mathbb{I}[\text{final answer correct}]$, after which the running statistics are updated to form s_{t+1}.

Every 16 episodes (160 transitions), the collected tuples $(s_t, a_t, R_t, V(s_t), \log \pi_\theta(a_t \mid s_t))$ form a rollout buffer over which we run a clipped-surrogate PPO update with generalized advantage estimation. The remainder of this section formalizes the MDP, details each of the eight state features (including a subtlety in observed advisor accuracy), describes the scripted components, and specifies the optimization.

3.2 Problem formulation

We model trust as a finite-horizon MDP. An episode is a sequence of $T{=}10$ multiple choice tasks drawn without replacement from the corpus. At episode start, an advisor reliability r is sampled (uniformly from $\{0.2, 0.5, 0.8\}$ during training) and held fixed but hidden from the agent. At step t the agent observes state s_t and chooses $a_t \in \{\text{trust own}, \text{trust advisor}\}$; the corresponding answer is scored, giving reward $R_t = \mathbb{I}[\text{final answer correct}]$, and the running statistics in s_{t+1} update. Because an episode's reliability is hidden and constant, the agent must infer r from the outcomes of its consultations, meaning that this is an exploration/exploitation problem with the flavor of a contextual bandit with state. This sequential framing is what makes the task genuine RL rather than one-shot classification: the agent's choice to consult is the only way to gather reliability evidence, and that evidence changes future states.

3.3 State representation

The 8-dimensional state s_t introduced above splits into four per-task signals and four episode-wide running statistics:

- **Per-task:** $\hat{p}$ (self-uncertainty probe estimate of agent correctness, see Subsection 3.4), `tone_confident`, `reasoning_seems_solid` (how sound the advisor's argument appears), and `agreement` (between agent choice and advisor choice).
- **Running:** `step_normalized` (how far into episode), `advisor_consult_rate` (proportion of prior steps for which the agent trusted the advisor), `observed_advisor_accuracy`, and `agreement_rate`.

We artificially crafted the agent's own 4-way answer distribution p_{ag} with per-cell accuracy and a trap bias, designed so that $\max p_{ag}$ alone is not predictive of correctness. The advisor emits an answer (reliability-controlled), a tone independent of correctness, and a reasoning-quality flag positively but imperfectly related to correctness (a fixed conditional gap).

A critical design choice concerns observed advisor accuracy (OAA). Naively, OAA is the success rate of consults so far; but the agent consults freely on agreement steps (where both answers match), and on those $P(\text{advisor right}) \approx P(\text{own right})$ independent of r. This swamps OAA with reliability-uninformative observations, leaving it nearly constant across conditions. We therefore redefine OAA to count only disagreement-consults, defaulting to 0.5 before any such consult. This change converts OAA from a dead feature into a usable reliability proxy and is methodologically clean, as it uses only outcomes the agent actually observed.

3.4 Uncertainty probe

The uncertainty probe is a frozen MLP, trained Kadavath-style [Kadavath et al., 2022] to map $(p_{ag}, \max p_{ag}, H(p_{ag})$, and a task-family one-hot) to a calibrated $\hat{p} \in [0, 1]$. Entropy $H(p_{ag})$ is quite an informative input, as a more spread out answer distribution signals a less decisive prediction. But, because the trap bias makes raw confidence non-predictive, the probe must learn to combine it with the rest of the answer distribution and task-family identity rather than relying on $\max p_{ag}$.

We train the probe offline on a separate corpus of $\sim$10,000 tasks (8,000 train / 2,000 held-out) drawn from the same generator as the PPO environment, minimizing binary cross-entropy against whether the agent's answer was correct, with a two-layer MLP (hidden 32, $\tanh$) and Adam. We judge it by its Brier score, which is the mean squared error between $\hat{p}$ and the binary correctness outcome (lower is better and 0 is perfect), rather than by accuracy. The trained probe reaches a held-out Brier of 0.172, near the ≈ 0.165 bound set by how much the available features can reveal about correctness; scaling the probe's training data $5\times$ (to the full 50,000 task corpus) does not lower it, indicating the probe is information-limited rather than data-limited. It is therefore better calibrated than a stub yet deliberately imperfect: a near perfect probe would trivialize the trust problem.

3.5 Policy optimization and variants

We optimize a two-layer $\tanh$ MLP actor and critic (hidden dim 64) with PPO [Schulman et al., 2017]: clipped surrogate loss

$$L^{\text{CLIP}}(\theta) = \hat{\mathbb{E}}_t\big[\min(w_t\hat{A}_t,\ \text{clip}(w_t, 1{-}\epsilon, 1{+}\epsilon)\hat{A}_t)\big],$$

GAE(λ=0.95) [Schulman et al., 2016], and an entropy bonus. Both variants share a back-loaded curriculum that introduces full reliability diversity only late (10%/10%/80% of updates at r=$0.8 \rightarrow \{0.5, 0.8\} \rightarrow \{0.2, 0.5, 0.8\}$), which stabilizes early learning by first acquainting the model with a more consistent reliability level, and then gradually teaching it to handle a wider r distribution.

The *Heuristic* variant is vanilla PPO with mild early-consult shaping only: a small fixed bonus ($+0.1$) is added to the reward whenever the agent trusts the advisor during the first three steps of an episode, gently encouraging it to gather a few advisor observations early rather than immediately committing to trust-own. The shaping is deliberately weak, it barely nudges early exploration, and the Heuristic still tends to converge to a near `always-trust-own` rule, which is rewarding given the data statistics. The *Adaptive* variant adds two training only ingredients that make reliability learnable: (i) `force_first_disagree`, which overrides the action to "trust advisor" on the first disagreement

step of each training episode, guaranteeing at least one informative OAA observation and breaking the chicken-and-egg nature of needing to consult to learn that OAA is useful and needing OAA to be useful before consulting; and (ii) an `advisor_save_bonus` of 0.7, an information-dense bonus that's applied only when trusting the advisor flips a wrong own-answer into a correct one. Neither intervention modifies evaluation.

4 Experimental Setup

This section describes our experimental methodology: data, training, metrics, baselines, and two probing experiments (feature ablations and an adversarial/OOD suite). The corresponding results are reported in Section 5, and we give a forward reference to the relevant figure alongside each experiment below.

Data. 50,000 procedurally generated four-choice questions span three reasoning families (arithmetic, probability, Pearl-style causal/counterfactual) [Pearl and Mackenzie, 2018, Jin et al., 2023] $\times$ three difficulties, manually inspected for template quality, split 40K train / 10K test.

Training. PPO with 16 episodes $\times$ 10 steps per rollout, 4 epochs per update, 2500 updates, γ=0.95, λ=0.95, clip ϵ=0.2, entropy coef 0.01; actor LR $1\mathrm{e}{-4}$ (Adaptive) / $3\mathrm{e}{-4}$ (Heuristic). Each variant is trained for seeds $\{42, 43, 44\}$.

Evaluation. For each (policy, seed, condition) we run 400 episodes (4000 steps) at a fixed master seed (2000). We report three families of metrics: per-step accuracy, disagreement accuracy, and a bias-free composite accuracy. We also used the reliability-sensitivity slope (composite at r=0.8 minus at r=0.2) as an indicator of performance. For more detail regarding our evaluation metrics, please see Table 1. We compare to three fixed-action baselines (always-trust-own, always-trust-advisor, random) and to an adversarial / OOD suite applied at evaluation time only. The headline policy comparison across conditions is reported in Table 2, with per-metric breakdowns in Figs. 1–8 (Section 5).

Table 1: Experiment Evaluation Metrics.

Metric	Description	Calculation
Per-step accuracy	Fraction of all steps whose final chosen answer is correct. Weighted by frequency in the dataset, so it is dominated by whichever side is most often right at a given reliability.	$P(\text{chosen} = \text{correct})$
Disagreement accuracy	Among steps where the own and advisor answers differ (so the trust decision actually matters), the fraction on which the actor picks the correct side. With four answer options, these steps also include cases where *both* answers are wrong (and differ), on which neither choice can be correct. Still frequency weighted.	$P(\text{correct} \mid \text{own} \neq \text{adv})$
Composite (bias-free)	Equally-weighted mean of the success rates within the two informative disagreement cells: *loyalty* $= P(\text{trust own} \mid \text{own right, adv. wrong})$ *deferral* $= P(\text{trust adv} \mid \text{own wrong, adv. right})$ Frequency-independent, and exactly 0.5 for any fixed-action policy.	$\frac{1}{2}\big(\text{loyalty} + \text{deferral}\big)$
Derived sensitivity measure (robust learning indicator):		
Reliability slope	Change in the composite as the advisor becomes more reliable; a positive value means the policy defers more when the advisor is better, the signature of online reliability inference.	$\text{composite}\big\vert_{r=0.8} - \text{composite}\big\vert_{r=0.2}$

Feature ablations. To isolate what each signal contributes, we retrain the Adaptive policy with one state feature masked at a time and measure the resulting change in composite and in seed variance. Separately, on the unablated policy we compute the Pearson correlation of each feature with $P(\text{trust advisor})$ at deployment. The two views answer different questions: the correlation measures

how much the deployed policy reads a feature, whereas the retrain-and-mask ablation measures how necessary the feature is for learning a stable mechanism in the first place. We report this analysis in Section 5 (Figs. 6 and 7).

Adversarial and OOD shifts. We stress-test each trained policy with evaluation-time shifts applied to the advisor or environment after training: out-of-range reliability ($r{=}0.05, 0.95$, outside the training set $\{0.2, 0.5, 0.8\}$); random reasoning, which decouples the reasoning-quality cue from correctness; and inverted reasoning, a directional attack that flips the reasoning-correctness coupling. Unlike the feature ablations above, these never touch training; instead they probe how the trained policy degrades under shift. Results are reported in Section 5 (Fig. 8).

5 Results

5.1 Training dynamics

Both variants converge stably across seeds (Fig. 1). Reward dips align with curriculum transitions as harder (lower-r) advisors are introduced and our policies have to edit their trust rules. Heuristic's entropy collapses earlier (it commits quickly to trust-own), whereas Adaptive maintains exploration longer before settling.

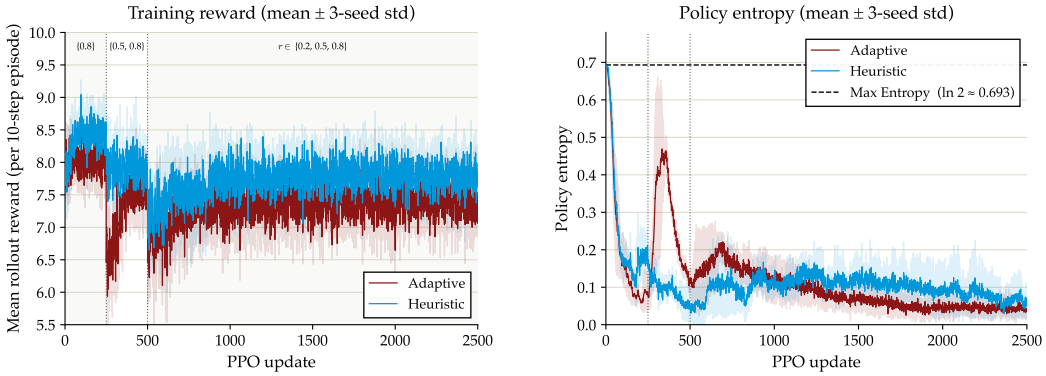

Figure 1: Training reward (left) and policy entropy (right), 3-seed mean ± std.

5.2 Decision quality and why raw accuracy misleads

Per-step accuracy (Table 1) is the fraction of steps whose final chosen answer is correct. Averaged across $r \in \{0.05, 0.2, 0.5, 0.8, 0.95\}$, Heuristic edges Adaptive (0.792 vs. 0.773) and neither consistently beats `always-trust-own` (0.780) (Table 2, Fig. 2). This is not a learning failure: at low reliability, ∼80–87% of disagreement steps are (own-right, advisor-wrong) cells, so blanket loyalty to one's own answer is statistically correct most of the time. A policy "wins" raw accuracy by exploiting this joint distribution, not by learning to trust well, and the same behavior loses at high reliability where deferring would pay off, hence Adaptive's crossover above the baseline at $r{=}0.95$.

Disagreement step accuracy (Table 1) restricts to steps where own and advisor answers differ, so the trust decision actually matters; it tells the same cautionary story (Fig. 3). Because both per-step and disagreement step accuracy are weighted by how often each disagreement cell occurs, they favor whichever policy aligns with the majority cell at a given reliability: Heuristic at low r, Adaptive at high r. Even then, the advantage each policy holds is modest and changes sign near $r{=}0.5$: Heuristic leads disagreement step accuracy by roughly ten points at $r{=}0.05$, the two are within a few points at intermediate reliabilities, and Adaptive leads by a similar margin from $r{=}0.8$ onward. The more informative difference is in dispersion rather than mean. Heuristic's accuracy is tightly clustered across seeds where it leads, but its seed standard deviation grows by an order of magnitude at high r (from about 0.01 to 0.13), while Adaptive's stays near 0.01 across all conditions. So even on this frequency-weighted metric which, if anything, is biased in Heuristic's favor at low r, the growth in Heuristic's across-seed variance is enough to show that its learned policy is not robust. The same training procedure yields materially different high-r behavior from one seed to the next.

The bias-free composite (Table 1) weights the two disagreement cells equally, so unlike the accuracy metrics above it cannot be won by simply matching the more common cell. On this metric the ranking is stable across reliabilities: Adaptive averages 0.637, Heuristic 0.549, and every fixed-action baseline exactly 0.500 (Table 2, Fig. 4). Adaptive clears the 0.5 no-skill floor at every condition, by 8–10 pp, significant for $r \leq 0.5$ (Welch's t, 3 vs. 3 seeds), while Heuristic sits only marginally above it. Since clearing 0.5 requires being right in both disagreement cells at once, this is direct evidence that Adaptive makes genuine per-step trust decisions rather than defaulting to a statistical hack of the RL environment.

Two key findings account for this difference. First, Adaptive defers when it should: its correct-deferral rate is 2–$18\times$ Heuristic's, and its composite rises with reliability (slope $+0.069 \pm 0.011$ on all three seeds), whereas Heuristic's slope is flat and seed-dependent ($+0.057 \pm 0.050$, with one seed collapsing to a fixed action). Second, Adaptive is reproducible: its per-seed variance is 5–$8\times$ lower than Heuristic's on every metric, and up to $17\times$ lower at r=0.95. Together, these indicate that our design choices properly steer PPO to the same multisignal trust rule across seeds, instead of the seed-dependent shortcuts (agreement-only, defer-everything, or fixed-action) that Heuristic otherwise falls into.

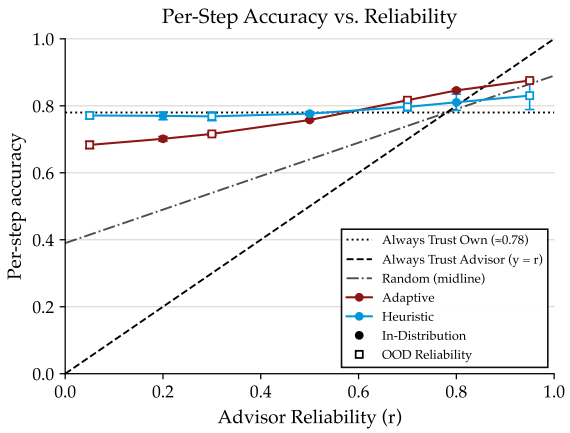

Figure 2: Per-step accuracy vs. advisor reliability r (3-seed mean $\pm$ std).

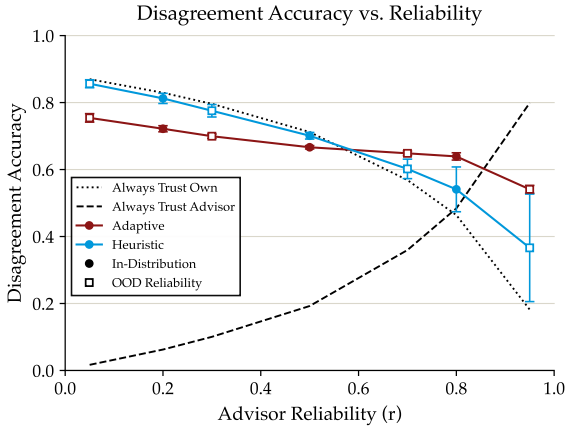

Figure 3: Disagreement accuracy vs. advisor reliability r (3-seed mean $\pm$ std).

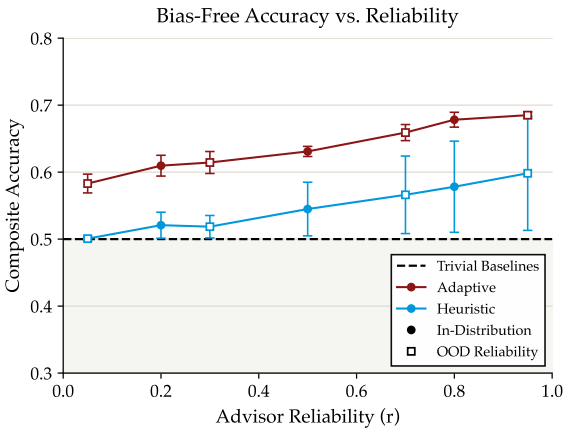

Figure 4: Bias-free composite accuracy vs. advisor reliability r (3-seed mean $\pm$ std).

Table 2: Headline comparison (3-seed mean). Per-step, disagreement, and composite accuracy are averaged over $r \in \{0.05, 0.2, 0.5, 0.8, 0.95\}$; *composite accuracy* at the unseen, out-of-distribution reliabilities r=0.05 and r=0.95. Bold marks the better of the two learned policies.

Policy	Per-step	Disagree.	Composite	Comp. @ r=0.05	Comp. @ r=0.95	Slope ($r_{.8}-r_{.2}$)
always-trust-own	0.780	0.611	0.500	0.500	0.500	0.000
always-trust-advisor	0.500	0.311	0.500	0.500	0.500	0.000
Heuristic	**0.792**	0.655	0.549	0.501	0.598	$+0.057 \pm 0.050$
Adaptive	0.773	**0.665**	**0.637**	**0.583**	**0.685**	$\mathbf{+0.069 \pm 0.011}$

5.3 Qualitative results: example trajectories

Rolling out the final Adaptive policy (seed 42, no further training) shows proper trust policy at the episode level (Fig. 5). With a reliable advisor it trusts the advisor on most steps; with an unreliable one it tries the advisor early, finds it wrong, and settles into personal loyalty.

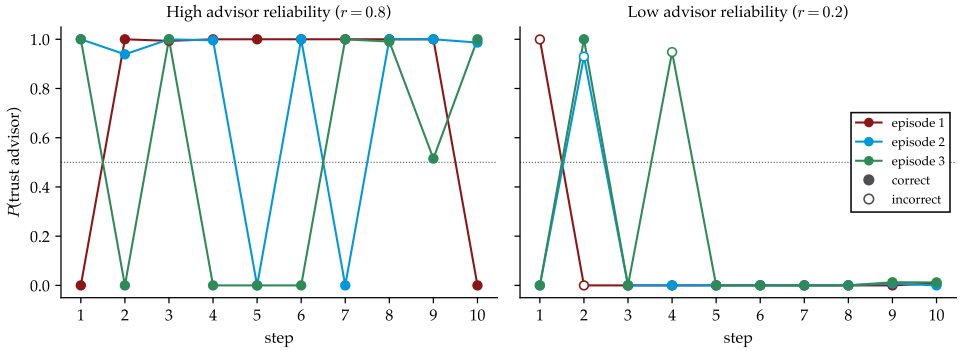

Figure 5: Three example rollouts of the final Adaptive policy at each reliability (one line per episode): per-step P(trust advisor); filled markers are correct selections, hollow incorrect.

5.4 Feature evaluation

We examine what each signal contributes in two complementary ways (Section 4): the deployment-time correlation of each feature with the trust action (Fig. 6), and a retrain-and-mask ablation that drops one feature at a time (Fig. 7).

The correlations confirm that Adaptive learns a genuine multisignal rule rather than an agreement-only shortcut. Its probability of trusting the advisor is driven mainly by advisor reasoning quality ($+0.85$), then by agreement ($+0.40$) and observed advisor accuracy ($+0.16$), with small inhibitory contributions from self-uncertainty $\hat{p}$ (-0.06) and tone (-0.08). That the policy effectively ignores tone is reassuring rather than disappointing: tone is exactly the kind of superficial cue a human listener is liable to be swayed by even though it carries no information about correctness, so a trust rule that learns to look past it is behaving the way we would want.

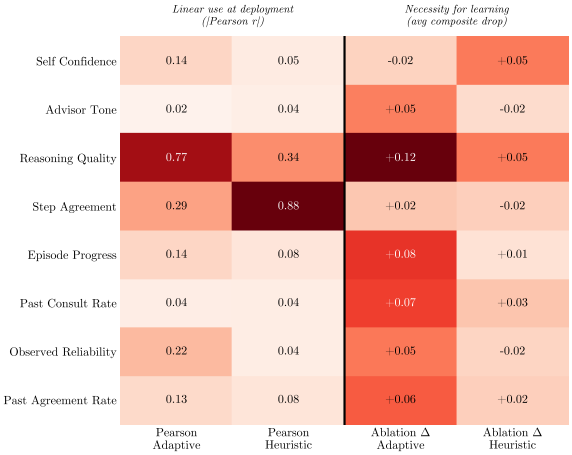

Figure 6: Feature importance: deployment-time linear use (|Pearson r| with $P(\text{trust advisor})$) vs. necessity for learning (composite drop under ablation).

The ablation enabled a more nuanced analysis about the importance of our trained uncertainty probe, $\hat{p}$. By correlation it looks almost unused, and removing it actually *raises* the mean composite slightly (by about 2 pp), so its direct contribution is at best marginal. What it does instead is stabilize training: without it, the per-seed composite variance inflates by roughly an order of magnitude and the reliability slope collapses from $+0.069$ to $+0.003$, with two of three seeds drifting to a high-deferral equilibrium and the third to a conservative one. The probe is therefore best understood as a training-time regularizer on the PPO landscape rather than a signal the deployed policy reads: a role invisible to correlation analysis and to mean-impact ablation alone. The same ablation is far harsher on Heuristic, which collapses to a fixed-action policy (composite 0.5) when either $\hat{p}$ or reasoning is removed, underscoring how narrow and fragile its apparent competence is.

5.5 Robustness under distribution shift

We probe robustness with evaluation-time shifts (Section 4) applied to the already-trained policies. Adaptive's clearest strength is extrapolation to unseen reliabilities: pushed to $r{=}0.05$ and $r{=}0.95$, well outside the training range, its behavior changes smoothly and monotonically, and at $r{=}0.95$ its per-step accuracy exceeds `always-trust-own` by 9.6 pp, whereas Heuristic swings widely from seed to seed, the same instability we saw on the in-distribution metrics.

The most informative stress test targets the cue Adaptive relies on most, advisor reasoning quality (Fig. 8). Randomizing the cue degrades both policies only modestly, but *inverting* it such that solid-looking reasoning now signals a wrong answer is $3-5\times$ more damaging, driving Adaptive's composite below the 0.5 no-skill floor (to 0.187 at $r{=}0.8$), while Heuristic, which barely used the cue, is hurt far less. We read this as confirmation rather than weakness: a policy can only be badly hurt by inverting a signal it genuinely depends on, so the collapse is direct evidence that Adaptive properly interprets reasoning quality's relationship to valid advice. And that is a sensible thing to have learned. In real interactions, the soundness of someone's reasoning is among the better available indicators of

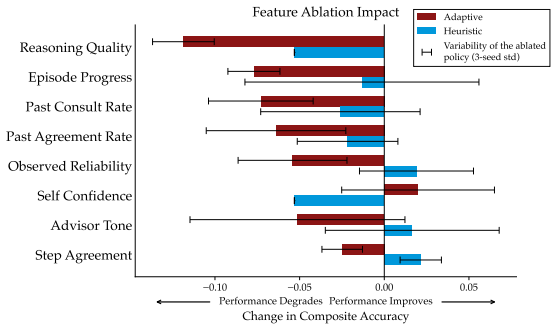

Figure 7: Change in composite under training-time feature masking, one feature removed at a time (positive = removal helped; 3-seed std).

whether their answer can be trusted. Hence, a policy that keys on it, and is correspondingly vulnerable when that cue is deliberately corrupted, indicates a more useful trust policy.

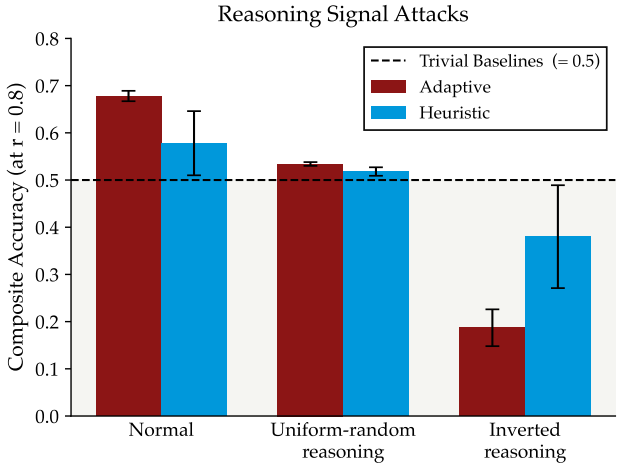

Figure 8: Composite under normal / random / inverted reasoning at r=0.8 (3-seed mean $\pm$ std).

6 Discussion

Key takeaways. Several deliberate design choices let us train a robust, multisignal trust policy. The most important is the state itself, which combines instantaneous per-step cues with running statistics that summarize the episode so far. We chose this because the RL formulation is episodic and because people, too, build trust over repeated interaction. The running advisor-accuracy statistic is what lets the policy infer reliability during an episode, and without it the policy falls back on a fixed rule. This idea is not specific to trust. Pairing an immediate observation with a running summary of history is the standard way to act under partial observability, as in Kalman-filter state estimation or recurrent POMDP policies. Two training choices then made the reliability signal learnable. We forced the agent to consult on the first disagreement of each episode, so that it saw at least one informative outcome, and we rewarded it when consulting turned a wrong answer into a right one. Bootstrapping a rarely taken but informative action this way is a common need in sparse-reward problems. Evaluation mattered just as much as training. Because most low-reliability disagreements favor the agent's own answer, plain accuracy rewards a policy that always trusts itself, so we relied instead on the bias-free composite, the variance across seeds, and the targeted reasoning attack to tell whether a policy was genuinely reasoning about trust. This caution applies to any imbalanced decision problem, such as deferral, triage, or fraud detection. Finally, our ablations showed that the self-uncertainty probe barely changed mean accuracy but sharply stabilized training, which is a reminder that a feature can

matter even when its average effect looks small, the kind of contribution a one-dimensional impact analysis would miss.

Limitations. A few caveats temper these results. With only three seeds, our statistical power on small mean differences is limited, although the large variance differences and OOD effects are robust, however we were constrained by compute and time. While it was an intentional choice to use simulated agent and advisor data, as it keeps our findings LLM agnostic, it leaves the behavior of real language-model advisors untested, and this would be an important stress-test and confirmation of real-world applicability. Separately, the uncertainty probe currently does not take into account the raw question text, which could provide valuable and more nuanced insight into the challenging nature of the question. Furthermore, our reliability levels are drawn from a small discrete set, so we did not evaluate continuous or drifting reliability, which would more accurately represent the variety of real-world agent interactions.

Future work. These limitations point to the natural next steps. The most important is to replace the scripted advisor with a range of real language-model advisors and test whether the learned trust cues survive genuine model behavior. Giving the probe access to the raw question text could also sharpen its uncertainty estimates. Finally, letting reliability vary continuously and drift within an episode would turn the agent's one-shot inference into ongoing tracking, and sweeping the strength of the adversarial attacks instead of testing a single inversion would show how gracefully the trust mechanism degrades.

7 Conclusion

We set out to ask a deceptively simple question, "when should an agent act on advice it cannot verify?", and framed it as a multistep reinforcement learning problem in which an advisor's reliability is hidden and must be inferred from experience. Our agent learned to do this, demonstrating that trust can itself be learned as a policy: from instantaneous and running signals of self-uncertainty, advisor confidence, reasoning quality, and agreement, the agent learned when to rely on external advice and when to reject it. Just as important was knowing how to recognize that learning in the first place, since raw accuracy was actively misleading on this task: it rewarded a policy that merely exploited the data's statistical skew. Only intentionally bias-free, decision-quality evaluation revealed whether the agent was genuinely reasoning or simply getting lucky. That gap between *looking* competent and *being* competent is a durable contribution, and it is precisely the gap that widens as advisors themselves grow more fluent and more persuasive. As autonomous systems increasingly act on the word of conversational proxy LLMs and ensembles of model advisors, each confident, but of genuinely uneven and hidden competence, deciding how much to trust each source becomes a daily, high-stakes problem. Meeting it will demand exactly this kind of patient, mechanism-level scrutiny of trust policies. Treating trust as something learned from the outcomes of one's own interactions, the way a person calibrates their confidence in a colleague over time, strikes us as a more elegant and realistic foundation than assuming reliability or hardcoding it, and we hope this work is an important initial step in that direction.

8 Team Contributions

- **Gia Ancone:** Led the project framing and the multistep RL formulation. Implemented the PPO actor-critic, buffer, GAE, and reward shaping, designed the environment, scripted agent, and advisor. Designed and ran the curriculum, shaping, ablation, multiseed, and adversarial experiments.
- **Jaden Chen:** Led the Kadavath-style uncertainty probe (interface, feature selection, training, and calibration evaluation), contributed to the evaluation-metric design and the OOD analysis.
- **Both:** Collaborated to improve and iterate on the pipeline; Conducted hyperparameter optimization; ran evaluations; produced the analysis and figures; constructed and manually inspected the task dataset; synthesized the related work; wrote, revised, and presented the report and poster.

Changes from Proposal. The division of labor stayed essentially as planned from the proposal and milestone through to the final report: Jaden owned the uncertainty probe as a self-contained module, Gia owned the PPO core and reward shaping, and the experiments, analysis, and write-up were shared, as anticipated.

References

Matt Gardner, Yoav Artzi, et al. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

Zhijing Jin, Yuen Chen, Felix Leeb, Luigi Gresele, Ojasv Kamal, Zhiheng Lyu, Kevin Blin, Fernando Gonzalez Adauto, Max Kleiman-Weiner, Mrinmaya Sachan, and Bernhard Schölkopf. Cladder: Assessing causal reasoning in language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

David Madras, Toniann Pitassi, and Richard Zemel. Predict responsibly: Improving fairness and accuracy by learning to defer. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Sabrina J. Mielke, Arthur Szlam, Emily Dinan, and Y-Lan Boureau. Reducing conversational agents' overconfidence through linguistic calibration. *Transactions of the Association for Computational Linguistics (TACL)*, 2022.

Hussein Mozannar and David Sontag. Consistent estimators for learning to defer to an expert. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.

Judea Pearl and Dana Mackenzie. *The Book of Why: The New Science of Cause and Effect*. Basic Books, 2018.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations (ICLR)*, 2016.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R. Bowman, et al. Towards understanding sycophancy in language models. *arXiv preprint arXiv:2310.13548*, 2023.

Sriram Ganapathi Subramanian, Matthew E. Taylor, Kate Larson, and Mark Crowley. Learning from multiple independent advisors in multi-agent reinforcement learning. In *Proceedings of the 22nd International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2023.

Lisa Torrey and Matthew E. Taylor. Teaching on a budget: Agents advising agents in reinforcement learning. In *Proceedings of the 12th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 1053–1060, 2013.

A Implementation Details

Hyperparameters. Actor/critic: 2-layer MLPs, hidden 64, $\tanh$. PPO: 16 episodes $\times$ 10 steps per rollout, 4 epochs, minibatch 64, γ=0.95, λ=0.95, clip ϵ=0.2, value coef 0.5, entropy coef 0.01, max grad norm 0.5, 2500 updates. Adaptive: actor LR $1\mathrm{e}{-4}$, `force_first_disagree` on (training only), `advisor_save_bonus`$= 0.7$ (training only), early-consult shaping. Heuristic: actor LR $3\mathrm{e}{-4}$, early-consult shaping only. Curriculum (both): $[(0, (0.8,)), (250, (0.5, 0.8)), (500, (0.2, 0.5, 0.8))]$. Seeds $\{42, 43, 44\}$; evaluation master seed 2000, 400 episodes per condition.

Reproducibility. All training and evaluation use fixed seeds; the disagreement-only OAA semantics and all adversarial conditions are env/advisor flags that default off, so the control behavior is recovered when they are unset.

B AI Tools Disclosure

Per CS224R Custom Project Guidelines §2.6, we used Claude (Anthropic, via the Claude Code CLI) as an aid for boilerplate code in the synthetic task and agent/advisor data generator, debugging portions of the evaluation and analysis scripts (e.g., Pearson-correlation analyses debugging), and the plotting code used to generate figures. However, the core intellectual and technical contributions were developed independently by the team, including the PPO implementation (clipped surrogate objective, GAE, policy/value updates, and advantage estimation), the trust-decision logic and reward shaping, the project's framing and hypotheses, the design of the environment and evaluation metrics, and the synthesis of related work.