DISAGREEMENT-GUIDED WEAK-TO-STRONG ROUTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Small language models are increasingly deployed on edge devices and robotic systems to meet cost and latency constraints, yet their limited capabilities can make reliable deployment challenging. Stronger models can provide assistance when needed, but deciding when to route a weak-model response to a stronger model often relies on signals of response correctness. In open-ended and agentic tasks, however, obtaining such signals can require costly ground-truth supervision that is difficult to obtain at scale. We revisit the foundations of weak-to-strong routing and ask whether routing decisions can instead be guided by signals that do not require ground-truth supervision. We first characterize the optimal weak-to-strong routing policy, while allowing for multiple weak-model responses generated through local inference. Building on this characterization, we derive a principled approximation to the optimal routing based on model-to-model comparison. In particular, we introduce a notion of model disagreement that can be measured directly from model-generated responses, without ground-truth supervision. Importantly, the approximation becomes increasingly accurate as the strong model becomes a better proxy for the ground-truth distribution. Across agentic and single-turn tasks, our approach effectively navigates the trade-off between routing cost and response accuracy without ground-truth supervision. Moreover, even when ground-truth correctness signals are available for training, disagreement-based routing can outperform methods trained directly to predict correctness.

1 INTRODUCTION

Small language models offer a practical path to deploying AI agents on edge devices, robots, and other systems constrained by cost, latency, connectivity, or privacy. Yet their limited capabilities make it difficult to rely on them across the full range of tasks they may encounter (Lu et al., 2025; Chen et al., 2024). This has motivated growing interest in systems that primarily rely on a weak LLM, but occasionally route difficult instances to a stronger model when additional capability is needed (Gupta et al., 2024; Yue et al., 2024; Chen et al., 2023).

The possibility of such routing changes how we should think about inference with weak LLMs. Rather than simply generating a response and returning it, inference now involves three decisions: whether to *accept* the current response, *spend additional local computation* on further reasoning or sampling, or *seek assistance* from a stronger model at substantially higher cost. These choices introduce a new trade-off: *Stopping local inference too early* miss improvements achievable through inexpensive additional computation; *persisting locally for too long* waste time and resources when stronger assistance is ultimately needed; and *routing too readily* undermines the cost and latency advantages of the weak model.

A natural way to navigate this trade-off is to assess response correctness through learned confidence scores, verifiers, or reward models (Lightman et al., 2023; Wang et al., 2024). In practical agentic applications, however, correctness is often difficult to evaluate. For a robotic agent, assessing a proposed next action may require knowledge of the physical environment and consequences that become apparent after execution. For a computer-use agent, whether an action is appropriate may depend on the state of an external application, user preferences, or the downstream effects of modifying a file or sending a message. More broadly, in long-horizon agentic workflows, a locally plausible action may undermine the eventual outcome, violate a user constraint, or introduce an error that be-

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.

comes apparent several steps later (Shinn et al., 2023; Fan et al., 2026; Lù et al., 2025). Obtaining correctness labels in such settings may require subjective or expert judgment, interaction with the environment, or evaluation of entire trajectories. Such supervision is costly and does not readily scale across tasks and deployment conditions. These challenges motivate our central question:

Can we learn when to rely on weak-model inference and when to seek stronger assistance—without ground-truth correctness labels?

To answer this question, we take a step back and shift the perspective from asking when the weak model needs help to asking about the *value of using a stronger model*. From this perspective, routing is fundamentally a comparative decision: what matters is not simply whether the weak model's response is incorrect, but whether using the strong model instead is likely to improve the outcome. If we had access to the true likelihood that each response is correct, the value of using the strong model instead would naturally be captured by the difference

$$\text{strong-model correctness likelihood} - \text{weak-response correctness likelihood}. \tag{1}$$

This difference represents the *gain* in correctness likelihood from going from the current weak-model response to a strong-model response. In Section3, we formalize a generalized version of equation 1 and characterize optimal routing within a broader framework that jointly accounts for weak-model test-time inference allocation.

Both correctness quantities in (1), however, are difficult to access without ground-truth supervision. This motivates us to ask how (1) can be approximated using only quantities available from the models themselves. A key observation is that (1) depends on the *difference* between the two correctness likelihoods, rather than requiring either likelihood individually. This naturally shifts our attention from model-to-ground-truth comparison to *model-to-model comparison*, and in particular to how much the models *disagree*, which can be measured directly from model-generated responses.

In Section 4, we derive a principled, ground-truth-free approximation to (1) using only model-generated responses. At a high level, the resulting score takes a simple form:

$$\underbrace{\text{weak–strong disagreement}}_{\text{difference across models}} - \underbrace{\text{strong–strong disagreement}}_{\text{uncertainty within the strong model}}. \tag{2}$$

The first term asks whether the strong model is likely to provide something different from the weak response. The second then accounts for how uncertain the strong model itself is, since a different response is less likely to represent a useful correction when the strong model is itself uncertain. This approximation is principled in a natural sense: it becomes increasingly accurate as the strong model's predictive distribution approaches the ground-truth distribution, and is exact when the two coincide. To this end, our contributions are summarized as follows:

Our contributions are summarized as follows:

- **Fundamentals of weak-to-strong routing.** In Section 3, we characterize optimal routing within a broader framework that jointly accounts for weak-model test-time inference allocation, identifying the population quantities governing when to continue locally and when to seek stronger assistance.
- **Routing through model disagreement.** In Section 4, we derive a novel, ground-truth-free approximation to the optimal routing signal based on two disagreement signals: how much the weak and strong models disagree, and how much the strong model disagrees with itself. Importantly, (i) the approximation gap shrinks as the strong model becomes more accurate, and (ii) both signals can be *efficiently learned from scalable model-generated data, without ground-truth supervision.*
- **Empirical evaluation in agentic and reasoning tasks.** In Section 6, we evaluate disagreement-based routing across *multi-step agentic environments, where step-level correctness is difficult to define or supervise*, and controlled single-turn reasoning tasks. Across both regimes, our learned disagreement-based signals consistently outperform label-free baselines, validating that disagreement-based approximation provides a reliable, scalable learning signal when ground-truth supervision is unavailable.

2 Related Work

We briefly discuss the works most closely related to our evaluation setting and baselines, and defer a discussion to Appendix A.1.

Learning to defer and supervised routing. The fundamental problem of knowing when a model should abstain or seek help has been extensively studied under the umbrella of learning-to-defer and selective prediction (Mozannar & Sontag, 2021; Jitkrittum et al., 2024). With the advent of large language models, this challenge translates into determining when a weak model should autonomously escalate to a stronger one. Many current cascade methods rely on ground-truth correctness labels to train quality scorers, process verifiers, or token-level confidence predictors (Chen et al., 2023; Gupta et al., 2024; Aggarwal et al., 2025; Chuang et al., 2025; Rabanser et al., 2025). In our experiments, we represent this paradigm using a ground-truth supervised baseline highlighting that our proposed framework fundamentally eliminates the need for such expensive, task-specific supervision.

Label-free routing signals. To avoid human annotations, recent work extracts reliability signals directly from a single unlabeled generation using internal uncertainty metrics or hidden-state probes (Gupta et al., 2024; Kossen et al., 2024; Du et al., 2024). A prevalent alternative assesses reliability by generating multiple responses to measure internal agreement or semantic overlap (Wang et al., 2023; Yue et al., 2024; Kuhn et al., 2023; Farquhar et al., 2024). These two paradigms motivate our label-free baselines in both settings to demonstrate that our single-pass, disagreement-trained probe yields a better routing signal than standard internal uncertainty metrics.

Step-level routing in agentic tasks. In sequential environments, determining when to escalate an action to a stronger model is typically guided by process-level verifiers or trajectory-level rewards (Kapoor et al., 2026; Hemadri et al., 2026; Sarukkai et al., 2026). Because these approaches rely on explicit step-level correctness labels or complex episode-success evaluations, they are difficult to scale in open-ended domains. We contrast these paradigms with our agentic experiments, demonstrating that our framework can route actions autonomously at each step by predicting offline action disagreement, entirely bypassing the need for process verifiers or trajectory rewards.

3 Fundamentals of Weak-to-Strong Routing

In this section, we study the problem of weak-to-strong routing jointly with the allocation of local inference computation.[1] We first study this problem at the population level, assuming access to the underlying probabilistic quantities, which allows us to characterize the optimal solution and identify the fundamental principles governing sampling, selection, and routing. In the next sections, we use these principles to develop practical methods that do not require access to such population quantities.

Setup and sampling. Let (P, Y, Y_w, Y_s) follow a joint distribution, where $P \in \mathcal{P}$ is a prompt, $Y \in \mathcal{Y}$ its correct answer, and $Y_w, Y_s \in \mathcal{Y}$ are the weak- and strong-model responses, respectively.[2]

A sampling policy $\sigma : \mathcal{P} \to \mathbb{N}_+$ specifies the number of weak-model responses generated for each prompt. Given P, let $Y_{w,1}, \ldots, Y_{w,\sigma(P)}$ denote these responses, and define the resulting history as

$$H_\sigma = (P, Y_{w,1}, \ldots, Y_{w,\sigma(P)}).$$

Selection and routing. Given the sampled weak responses, we define two policies. The routing policy decides whether to settle for one of the weak-model responses or query the strong model, while the selection policy determines which weak-model response to return if we settle for the weak model. Formally,

$$\pi(H_\sigma) \in [0, 1], \qquad \phi(H_\sigma) \in \{1, \ldots, \sigma(P)\},$$

where $\pi(H_\sigma)$ is the probability of querying the strong model and $\phi(H_\sigma)$ is the index of the selected weak response.

Together, (σ, ϕ, π) determine how much local inference to use, which weak response to return, and when to seek strong-model assistance. Our goal is to maximize the accuracy of the final response while controlling the use of both weak- and strong-model inference. We impose separate budgets: B_w on the expected number of weak-model samples and B_s on the expected frequency of strong-model queries. This gives the following optimization problem:

[1] For simplicity, we model local test-time computation through repeated weak-model sampling (e.g., best-of-n), although more general forms of test-time inference are possible.

[2] For simplicity, we treat $\mathcal{Y}$ as a discrete response space, so equality is well defined. For LLM outputs, this can be understood as equality up to semantic equivalence.

Budgeted Weak-to-Strong Routing

$$\begin{aligned} \max_{\sigma,\phi,\pi} \quad & \mathbb{E}\big[(1-\pi(H_\sigma))\mathbf{1}\{Y_{w,\phi(H_\sigma)} = Y\} + \pi(H_\sigma)\mathbf{1}\{Y_s = Y\}\big] \\ \text{subject to} \quad & \mathbb{E}[\sigma(P)] \le B_w, \qquad \mathbb{E}[\pi(H_\sigma)] \le B_s. \end{aligned} \tag{3}$$

The expectations are over the prompt–answer distribution and model randomness. The two constraints allow weak-model inference and strong-model queries to be allocated across prompts according to their respective benefits.

Population quantities. To characterize the optimal policy, define

$$q_p(y) := \Pr(Y = y \mid P = p), \qquad t_p := \Pr(Y_s = Y \mid P = p). \tag{4}$$

Here, y is a fixed candidate response, Y is the random ground-truth answer, and Y_s is a response drawn from the strong model. Thus, $q_p(y)$ is the probability that y is correct, while t_p is the probability that the strong-model response is correct.

Theorem 1. *Fix any sampling policy σ satisfying $\mathbb{E}[\sigma(P)] \le B_w$. For a realized history $h_\sigma = (p, y_{w,1}, \ldots, y_{w,\sigma(p)})$, define*

$$\Delta(h_\sigma) := t_p - \max_{i \le \sigma(p)} q_p(y_{w,i}). \tag{5}$$

In words, $\Delta(h_\sigma)$ is the gain in correctness probability from querying the strong model rather than returning the best available weak response. The optimal selection and routing policies are

$$\phi^*(h_\sigma) \in \arg\max_{i \le \sigma(p)} q_p(y_{w,i}), \qquad \pi^*(h_\sigma) = \begin{cases} 1, & \Delta(h_\sigma) > \tau, \\ 0, & \Delta(h_\sigma) < \tau, \end{cases} \tag{6}$$

where $\tau \ge 0$ is a global threshold chosen to satisfy the strong-model budget.[3]

Interpreting the optimal policies. From equation 6, the optimal selection rule is straightforward: among the sampled weak responses, it selects the one with the highest probability of being correct. However, the optimal routing policy is more involved: it is a threshold rule over the score $\Delta(h_\sigma)$ defined in equation 5. This score measures the improvement in correctness probability from querying the strong model, t_p, instead of returning the best weak response, $\max_{i \le \sigma(p)} q_p(y_{w,i})$. The policy routes whenever this improvement exceeds the global threshold τ, which controls the routing budget. Thus, routing depends not only on how reliable the weak response is, but on how much the strong model is expected to improve upon it.

One-shot sampling as a special case. When $\sigma(P) \equiv 1$, selection is trivial and routing is the only remaining decision. In this case, (5) reduces to $\Delta(h_\sigma) = t_p - q_p(y_{w,1})$. This formalizes the accuracy-gain intuition introduced in (1), also recovering the known one-shot characterization of Mozannar & Sontag (2021); Jitkrittum et al. (2024).

4 Approximations

The optimal routing rule derived in Section 3, particularly equations 5 and 6, depends on the conditional correctness probabilities q_p and t_p defined in equation 4. In many practical tasks, these population quantities are unavailable without expensive human labels or exhaustive ground-truth verifiers. In this section, we derive a principled approximation to the optimal routing score in equation 5 that eliminates this dependence on ground-truth correctness. We first present the resulting routing score and its practical interpretation, and then derive it from the population-optimal rule.

Our approximation, which we call *corrected disagreement*, takes a simple form. Recall that $h_\sigma = (p, y_{w,1}, \ldots, y_{w,\sigma(p)})$ contains the prompt and its sampled weak-model responses, and that Y_s and Y_s' are independent responses drawn from the strong model for the same prompt. We define the

[3] A more detailed version of the theorem in Appendix B also characterizes the optimal sampling policy σ.

corrected-disagreement routing score as

$$S_{\mathrm{CD}}(h_\sigma) = \underbrace{\min_{i \le \sigma(p)} \Pr(Y_s \neq y_{w,i} \mid P = p)}_{\text{weak–strong disagreement}} - \underbrace{\Pr(Y_s \neq Y_s' \mid P = p)}_{\text{strong-model self-disagreement}}. \tag{7}$$

The first term measures the disagreement of the strong model with the collection of sampled weak responses. It corresponds to selecting the weak response that the strong model is least likely to disagree with. The second term corrects for disagreement arising from uncertainty within the strong model itself. Importantly, both terms can be learned from model-generated data without ground-truth labels. We discuss how to learn these quantities in practice in Section 5.

We next show S_{CD} arises as a principled approximation to the population-optimal routing rule.

4.1 Deriving Corrected Disagreement

The population-optimal routing gain in equation 5 is $\Delta(h_\sigma) = t_p - \max_{i \le \sigma(p)} q_p(y_{w,i})$, where q_p and t_p are defined in equation 4.

Let $s_p(y) := \Pr(Y_s = y \mid P = p)$ denote the strong model's predictive distribution. Then

$$\begin{aligned}\Delta(h_\sigma) &\overset{(a)}{=} \sum_{y \in \mathcal{Y}} q_p(y) s_p(y) - \max_{i \le \sigma(p)} q_p(y_{w,i}) \overset{(b)}{\approx} \sum_{y \in \mathcal{Y}} s_p(y)^2 - \max_{i \le \sigma(p)} s_p(y_{w,i}) \\ &\overset{(c)}{=} \min_{i \le \sigma(p)} \Pr(Y_s \neq y_{w,i} \mid P = p) - \Pr(Y_s \neq Y_s' \mid P = p) = S_{\mathrm{CD}}(h_\sigma). \end{aligned} \tag{8}$$

Here, (a) uses the conditional independence $Y_s \perp\!\!\!\perp Y \mid P$ to write $t_p = \sum_y q_p(y) s_p(y)$. In (b), we replace the unknown ground-truth distribution q_p by the strong model's predictive distribution s_p. Finally, (c) follows from $\Pr(Y_s \neq y \mid P = p) = 1 - s_p(y)$ and $\Pr(Y_s \neq Y_s' \mid P = p) = 1 - \sum_y s_p(y)^2$. Thus, corrected disagreement approximates the population-optimal routing gain using only comparisons between model-generated responses.

Approximation quality. The only approximation in equation 8 occurs in step (b), where the unknown ground-truth distribution q_p is replaced by the strong model's predictive distribution s_p. As the strong model becomes a better proxy for the ground-truth distribution, this approximation becomes more accurate. In the ideal case $s_p = q_p$, step (b) is exact, and corrected disagreement recovers the population-optimal routing gain: $S_{\mathrm{CD}}(h_\sigma) = \Delta(h_\sigma)$. Thus, the approximation gap naturally shrinks as s_p approaches q_p.

4.2 Population Comparison of Routing Scores

In this section, we compare the quality of different routing scores at the population level. We use extensive Monte Carlo sampling to accurately estimate the probabilities defining each score, allowing us to compare the underlying routing signals independently of how they are learned in practice. For simplicity, we consider the one-shot setting $\sigma(P) \equiv 1$. We compare four routing signals:

- **Optimal routing gain:** $\Delta(p, y_w) = \Pr(Y_s = Y \mid P = p) - \Pr(Y = y_w \mid P = p)$. This is the exact gain defined in Theorem 1. It serves as an oracle upper bound representing full knowledge of both models' correctness, and requires ground-truth supervision.
- **Weak-model correctness:** $S_{\mathrm{cor}}(p, y_w) = 1 - \Pr(Y = y_w \mid P = p)$. This represents optimal confidence-based routing, routing when the weak model is likely incorrect. It serves as the population oracle for confidence- and consistency-based methods and requires ground-truth labels.
- **Weak-strong disagreement:** $S_{\mathrm{disagree}}(p, y_w) = \Pr(Y_s \neq y_w \mid P = p)$. This baseline routes purely based on raw cross-model disagreement, without using ground-truth labels.
- **Corrected disagreement:** $S_{\mathrm{CD}}(p, y_w)$ (Equation 7). This represents our proposed approximation and requires no ground-truth supervision.

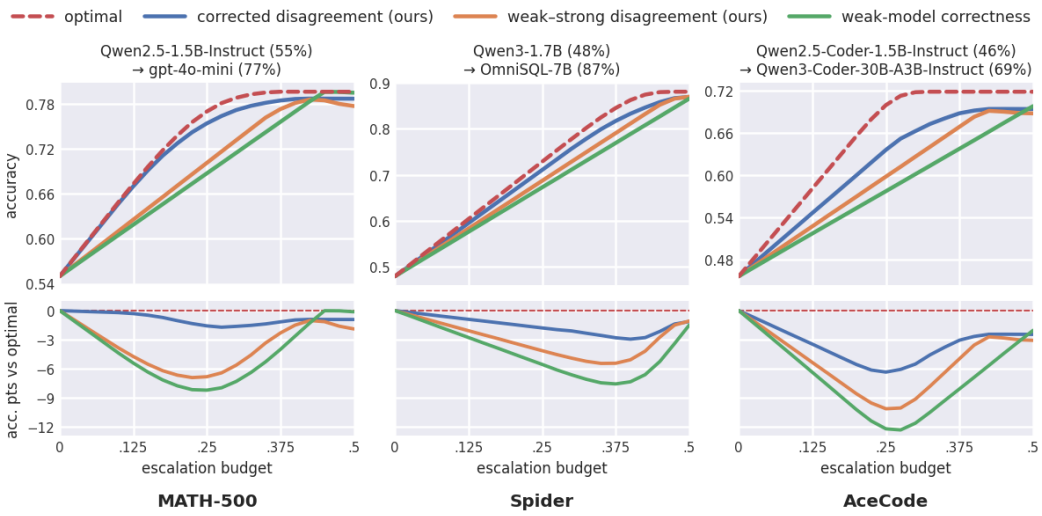


Figure 1: Corrected disagreement better approximates optimal routing than alternatives.

Figure 1 reports results across the three varying benchmarks. In each panel, the top row displays weak-to-strong routing accuracy across informative escalation budget regime $B \in [0, 0.5]$, and the bottom row illustrates the gap relative to optimal routing. The results reveal three central insights:

(i) Corrected disagreement, S_{CD}, consistently outperforms uncorrected weak–strong disagreement, S_{disagree}, across all budgets. **(ii)** In operational regimes where the escalation budget is constrained ($B \leq 0.4$), S_{CD} and S_{disagree} outperform the weak-model correctness baseline. The reason is, while correctness indicates when the weak model is wrong, it cannot anticipate whether the strong model will succeed. **(iii)** Because S_{CD} replaces ground truth with the strong model's predictive distribution, the approximation gap of S_{CD} is smaller for more accurate strong models.

5 Finite-Sample Routing Algorithms

We now turn the corrected-disagreement score into a practical routing algorithm. We describe two components: first, how the learned score is used at test time to route between the weak and strong models. Second, how this score can be learned using only weak- and strong-model generations.

Routing algorithm. Following Theorem 1, we fix a sampling policy σ and a selection policy ϕ. Recall from Section 3 that $\sigma(p)$ specifies the number of weak responses generated for prompt p, $\phi(h_\sigma)$ selects one of these responses, and $h_\sigma = (p, y_{w,1}, \ldots, y_{w,\sigma(p)})$ denotes the resulting history.

Given a routing score $\widehat{S}(h_\sigma) \in \mathbb{R}$, test-time routing proceeds as follows:

1. Generate $\sigma(p)$ weak-model responses, forming h_σ.
2. Compute the routing score $\widehat{S}(h_\sigma)$.
3. If $\widehat{S}(h_\sigma) > \tau$, query the strong model; otherwise, return the weak response selected by $\phi(h_\sigma)$.

The threshold τ is calibrated on held-out data to satisfy the desired strong-model budget B_s.

To recap, Theorem 1 shows that the optimal routing rule is obtained by setting the routing score to the accuracy gain $\Delta(h_\sigma)$ in equation 5. In Section 4.1, we approximated this optimal score by $S_{\mathrm{CD}}(h_\sigma)$ in equation 7, which, unlike $\Delta(h_\sigma)$, depends only on model disagreement and does not require ground-truth supervision. We now complete the framework by showing how S_{CD} can be learned in practice using only model-generated data.

Learning corrected disagreement. The corrected-disagreement score in equation 7 consists of two quantities: weak–strong disagreement and strong-model self-disagreement. We learn both offline using only model-generated responses, without ground-truth labels.

For each training prompt p, we generate K_w independent responses from the weak model, $y_{w,1}, \ldots, y_{w,K_w}$, and K_s independent responses from the strong model, $y_{s,1}, \ldots, y_{s,K_s}$. Let $A(y, y') \in \{0, 1\}$ indicate whether two responses y and y' disagree. These generations allow us to construct empirical training targets for the two terms appearing in S_{CD}.

Weak–strong disagreement. For each weak response $y_{w,i}$, we define

$$\widehat{d}(p, y_{w,i}) := \frac{1}{K_s} \sum_{k=1}^{K_s} A(y_{w,i}, y_{s,k}), \qquad i = 1, \ldots, K_w. \tag{9}$$

This is the fraction of strong-model responses that disagree with $y_{w,i}$, and hence provides an empirical estimate of the weak–strong disagreement probability $\Pr(Y_s \neq y_{w,i} \mid P = p)$.

Strong-model self-disagreement. We similarly define

$$\widehat{g}(p) := \frac{1}{K_s(K_s - 1)} \sum_{k \neq k'} A(y_{s,k}, y_{s,k'}). \tag{10}$$

This is the fraction of pairs of strong responses that disagree, and hence provides an empirical estimate of the strong-model self-disagreement probability $\Pr(Y_s \neq Y_s' \mid P = p)$.

The disagreement indicator A depends on the task and only needs to determine whether two responses are equivalent, rather than whether either response is correct. For example, it can be implemented using exact matching, task-specific parsing or execution, or an LLM judge. Importantly, A is used only offline to construct the training targets and is not required at test time.

We train two predictors, $\widehat{f}_d(p, y_w)$ and $\widehat{f}_g(p)$, to predict the targets in equation 9 and equation 10, respectively. The first predicts the probability that the strong model disagrees with a given weak response, while the second predicts the strong model's self-disagreement from the prompt alone. In Appendix A.5, we provide implementation details for the predictors used in our experiments.

At test time, we combine the two predictors according to the corrected-disagreement score:

$$\widehat{S}_{\mathrm{CD}}(h_\sigma) = \min_{i \leq \sigma(p)} \widehat{f}_d(p, y_{w,i}) - \widehat{f}_g(p). \tag{11}$$

Thus, $\widehat{S}_{\mathrm{CD}}$ can be computed using only the prompt and the sampled weak-model responses. In particular, neither the strong model nor ground-truth labels are required at test time.

6 Experiments

In this section, we evaluate our disagreement-based routing framework across two settings. Section 6.1 first examines multi-step agentic environments, where step-level correctness is notoriously difficult to define. Section 6.2 considers single-turn reasoning tasks, where gold answers provide a rigorous testbed to directly compare our approach with ground-truth-supervised baselines.

Baselines. We benchmark against established routing signals, categorized by their test-time inference cost and reliance on ground-truth labels:

Single-response, label-free signals ($N = 1$): Evaluating a single weak response adds zero test-time generation overhead. We compare Corrected disagreement (ours) (Equation 7); Weak–strong disagreement (ours), isolating the value of the self-disagreement penalty; and Token entropy (Gupta et al., 2024), averagin internal predictive uncertainty over the generated sequence.

Multi-response, label-free signals ($N > 1$): These assess confidence via weak-model self-agreement, a standard approach for cascade deferral (Yue et al., 2024). Though label-free, generating and clustering N responses significantly increases inference overhead. We compare against Self-consistency (Wang et al., 2023), routing via the plurality cluster's vote share, and Semantic entropy (Kuhn et al., 2023), routing via the cluster distribution's entropy.

Supervised reference: For benchmark with ground-truth labels, we train a PROXY REWARD linear probe $\widehat{f}_c$. Requiring privileged supervision unavailable in open-ended tasks, it serves strictly as a supervised upper-bound reference rather than a fair label-free competitor.

Evaluation metrics. We evaluate cascade performance by plotting task success (final-answer accuracy for single-turn benchmarks, and end-to-end trajectory success for agentic environments) against the realized routing budget (the fraction of escalated queries or actions). To isolate the routing signal, we fix the selection policy across all baselines, deferring selection ablations to Appendix A.3.

6.1 AGENTIC ENVIRONMENTS: ROUTING WITHOUT GROUND TRUTH

In multi-step environments, an agent completes a goal through a sequence of actions. Let $H_t = (p, y^1, o^1, \ldots, y^{t-1}, o^{t-1})$ denote the trajectory history at step t, comprising the prompt p, past actions y_i, and observations o_i. Rather than escalating entire trajectories, we formulate routing at the *action level*: given H_t, the router decides whether to execute the weak model's action locally (y_w^t) or defer that specific step to the strong model (y_s^t). Defining step-level correctness in sequential environments is fundamentally difficult. Because tasks admit multiple valid paths, and expert evaluation is costly to scale(Lù et al., 2025). Furthermore, locally plausible actions may cause delayed downstream failures, making long-horizon credit assignment ambiguous (Fan et al., 2026). Consequently, correctness-supervised routers and automated LLM judges remain notoriously unreliable in agentic domains (Shinn et al., 2023; Zheng et al., 2023).

While correctness is ambiguous, step-level *disagreement* remains objective and easily verifiable. Let $\mathcal{T}(H_t, y)$ denote the transition triggered by action y (e.g., a parsed tool call or state change). Two actions are equivalent if they induce identical transitions: $A(y_w, y_s) = \mathbb{1}\{\mathcal{T}(H_t, y_w) = \mathcal{T}(H_t, y_s)\}$. Relying on observable mechanics rather than subjective evaluation enables effortless data collection from offline rollouts, entirely bypassing human labels and complex reward engineering.

Datasets and models. We evaluate two interactive environments, generating $K = 10$ samples per step for training targets and multi-sample baselines. In BFCL MULTI-TURN (Patil et al., 2025) (160 train / 40 test tasks), Qwen3-8B routes to GPT-5-mini; action agreement requires identical function calls and arguments. In WEBSHOP (Yao et al., 2023) (2,000 train / 500 test), Qwen3-8B routes to a task-specialized RL policy. To prevent distribution shifts by test-time strong-model interventions, we collect training data via an iterative, DAgger-style schedule (Li et al., 2026) (Appendix A.6).

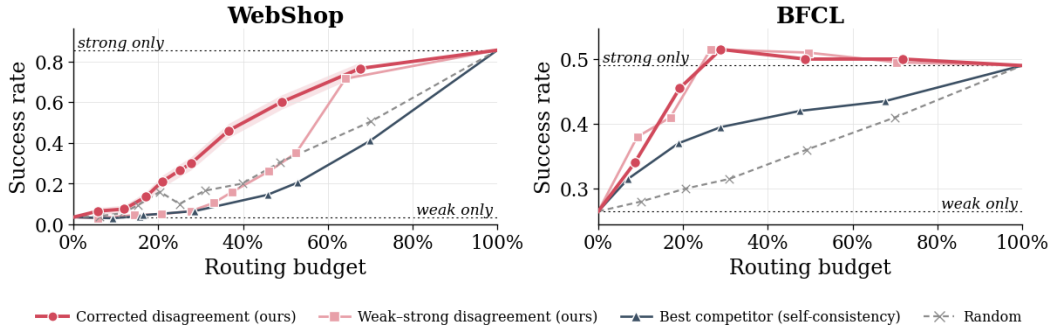


Figure 2: Action-level routing performance in agentic environments. For clarity, we plot our proposed signals against only the best-performing label-free baseline. Results are averaged over m=5 runs; comprehensive baselines, standard deviations, and tabular data are provided in Appendix A.4.

.

Figure 2 shows both disagreement-based signals consistently outperform the best label-free baseline. Crucially, contrasting the two environments reveals how teacher capability dictates the value of the self-disagreement correction. In WEBSHOP, where the strong model is highly capable, corrected disagreement significantly outperforms raw weak–strong disagreement. This reinforces our population-level finding: under a stronger teacher, the self-disagreement penalty prevents wasteful queries and the approximation approaches optimal routing. Conversely, in BFCL, the strong model

itself struggles; here, the self-disagreement correction provides negligible benefit, rendering the gap between the two scores statistically insignificant. Beyond performance gains, the results highlight a practical advantage for agentic workflows. Because step-level correctness is inherently vague, determining exactly when an action requires escalation remains a persistent challenge. Relying instead on observable cross-model disagreement provides practitioners with a reliable, objective signal to safely guide routing decisions.

6.2 Controlled Ablations on Single-Turn Benchmarks

Single-turn reasoning benchmarks provide an ideal testbed for controlled ablations that are difficult to isolate in multi-step settings. Because these tasks feature canonical gold answers, they allow us to directly compare our label-free signals against a supervised proxy reward and study the impact of the weak-model sampling budget.

Datasets and models. We use three diverse datasets spanning a spectrum of evaluation complexity: MATH-500 (Hendrycks et al., 2021) (Phi-3.5-mini-instruct → GPT-4o-mini; Disagreement indicator A is based on exact final-answer match), SPIDER (Yu et al., 2019) (Qwen3-1.7B → OmniSQL-7B; A is identical SQL execution), and ACECODE (Zeng et al., 2025) (Qwen2.5-Coder-1.5B-Instruct → Qwen3-Coder-30B-A3B-Instruct; A is identical hidden unit test behavior). These tasks progress from closed-form verification to open-ended generation, where structurally distinct outputs can yield identical functional results. This spectrum allows us to observe how routing methods scale as as simple answer-matching becomes insufficient, reflecting the practical challenges of open-ended generations. Finally, to confirm that our findings hold across different capability gaps, we evaluate additional Weak–strong model pairings in Appendix A.2.

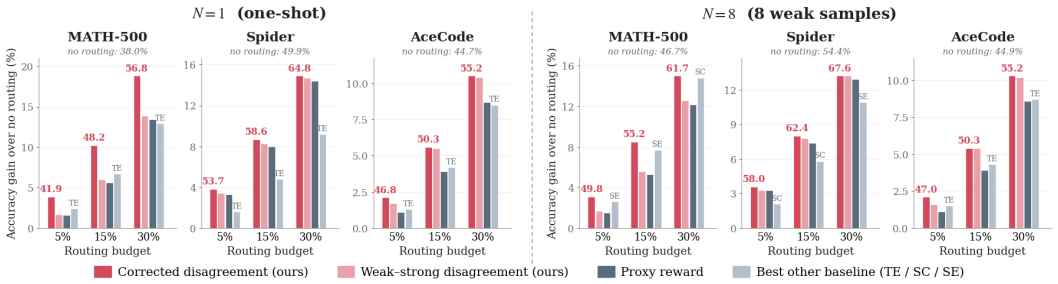


Figure 3: Finite-sample routing performance on single-turn benchmarks for one-shot ($N = 1$) and multi-sample ($N = 8$) regimes. Corrected disagreement consistently outperforms raw disagreement and the best label-free baselines. Multi-sample baselines only close the gap on closed-form tasks (MATH) where clustering is trivial, but break down on open-ended tasks (Spider, AceCode).

Results and discussion.

Figure 3 shows cascade accuracy at representative representative escalation budgets (further budgets in Appendix A.2). We compare the one-shot regime ($N = 1$), where consistency scores are undefined, against a multi-sample regime using uniform sampling ($N = 8$). For clarity, we plot our scores against the supervised proxy reward and only the strongest label-free baseline (exhaustive comparisons and additional budget values and sample sizes are deferred to Appendix A.2).

The empirical results validate our population-level analysis from Section 4.2. First, at these practical budget constraints, corrected disagreement consistently outperforms the supervised proxy reward. Second, across all three benchmarks, corrected disagreement yields higher accuracy than raw weak–strong disagreement, confirming that penalizing strong-model self-inconsistency successfully prevents the cascade from wasting queries on ambiguous prompts.

Finally, the multi-sample ($N = 8$) results expose the brittleness of consistency-based routing. On MATH-500, where answers are easily grouped, self-consistency leverages additional samples to narrow the gap with our approach. However, on SPIDER and ACECODE, where semantic clustering is difficult, consistency baselines fail to scale. In these domains, our learned probe maintains superior performance without additional inference overhead of generating and clustering multiple responses.

AI USE STATEMENT

We used generative AI tools to check grammar, suggest alternative wording, and assist with figure preparation and coding. We also used these tools to help discover relevant literature. The authors independently obtained and read the cited papers and prepared the references from the original sources. Additionally, we used AI tools to discuss and refine mathematical formulations and experimental choices. We reviewed all AI-assisted work and take responsibility for the final content, including the text, references, mathematical claims, code, and experimental results.

ETHICS STATEMENT

This work studies routing between weak and strong language models without ground-truth correctness labels for learning the proposed routing signals. We do not anticipate ethical concerns specific to the proposed method beyond those associated with the underlying models and their applications. In particular, agreement between models does not guarantee correctness or eliminate shared biases, and our benchmark results do not establish suitability for high-stakes deployment.

REPRODUCIBILITY STATEMENT

We plan to release all code upon acceptance. Section 3 states the problem formulation and theoretical assumptions, with proofs provided in the appendix. The main text describes the routing methods and experimental setup, while Appendix A.6 details on-policy data collection and probe training for the agentic experiments. Appendices A.2 and A.3 provide additional results across routing budgets, sample counts, and selection policies.

REFERENCES

Pranjal Aggarwal, Aman Madaan, Ankit Anand, Srividya Pranavi Potharaju, Swaroop Mishra, Pei Zhou, Aditya Gupta, Dheeraj Rajagopal, Karthik Kappaganthu, Yiming Yang, Shyam Upadhyay, Manaal Faruqui, and Mausam. Automix: Automatically mixing language models, 2025. URL https://arxiv.org/abs/2310.12963.

Marija akota, Maxime Peyrard, and Robert West. Fly-swat or cannon? cost-effective language model choice via meta-modeling. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pp. 606–615. ACM, March 2024. doi: 10.1145/3616855.3635825. URL http://dx.doi.org/10.1145/3616855.3635825.

Jianhao Chen, Zishuo Xun, Bocheng Zhou, Han Qi, Hangfan Zhang, Qiaosheng Zhang, Yang Chen, Wei Hu, Yuzhong Qu, Wanli Ouyang, and Shuyue Hu. Do we truly need so many samples? multi-llm repeated sampling efficiently scales test-time compute, 2025. URL https://arxiv.org/abs/2504.00762.

Lijie Chen, Binghui Peng, and Hongxun Wu. Theoretical limitations of multi-layer transformer, 2024. URL https://arxiv.org/abs/2412.02975.

Lingjiao Chen, Matei Zaharia, and James Zou. Frugalgpt: How to use large language models while reducing cost and improving performance, 2023. URL https://arxiv.org/abs/2305.05176.

Yu-Neng Chuang, Prathusha Kameswara Sarma, Parikshit Gopalan, John Boccio, Sara Bolouki, Xia Hu, and Helen Zhou. Learning to route llms with confidence tokens, 2025. URL https://arxiv.org/abs/2410.13284.

Jasper Dekoninck, Maximilian Baader, and Martin Vechev. A unified approach to routing and cascading for llms, 2025. URL https://arxiv.org/abs/2410.10347.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Ruhle, Laks V. S. Lakshmanan, and Ahmed Hassan Awadallah. Hybrid llm: Cost-efficient and quality-aware query routing, 2024. URL https://arxiv.org/abs/2404.14618.

Dujian Ding, Ankur Mallick, Shaokun Zhang, Chi Wang, Daniel Madrigal, Mirian Del Carmen Hipolito Garcia, Menglin Xia, Laks V. S. Lakshmanan, Qingyun Wu, and Victor Rühle. Best-route: Adaptive llm routing with test-time optimal compute, 2025. URL https://arxiv.org/abs/2506.22716.

Xuefeng Du, Chaowei Xiao, and Yixuan Li. Haloscope: Harnessing unlabeled llm generations for hallucination detection, 2024. URL https://arxiv.org/abs/2409.17504.

Shengda Fan, Xuyan Ye, Yupeng Huo, Zhi-Yuan Chen, Yiju Guo, Shenzhi Yang, Wenkai Yang, Shuqi Ye, Jingwen Chen, Haotian Chen, Xin Cong, and Yankai Lin. Agentprocessbench: Diagnosing step-level process quality in tool-using agents, 2026. URL https://arxiv.org/abs/2603.14465.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630:625–630, 06 2024. doi: 10.1038/s41586-024-07421-0.

Matt Gorbett and Suman Jana. Cross-model disagreement as a label-free correctness signal, 2026. URL https://arxiv.org/abs/2603.25450.

Neel Guha, Mayee F. Chen, Trevor Chow, Ishan S. Khare, and Christopher Ré. Smoothie: Label free language model routing, 2024. URL https://arxiv.org/abs/2412.04692.

Neha Gupta, Harikrishna Narasimhan, Wittawat Jitkrittum, Ankit Singh Rawat, Aditya Krishna Menon, and Sanjiv Kumar. Language model cascades: Token-level uncertainty and beyond, 2024. URL https://arxiv.org/abs/2404.10136.

Kimia Hamidieh, Veronika Thost, Walter Gerych, Mikhail Yurochkin, and Marzyeh Ghassemi. Complementing self-consistency with cross-model disagreement for uncertainty quantification, 2026. URL https://arxiv.org/abs/2604.17112.

Raghu Vamshi Hemadri, Humaira Firdowse Mohammed, Rishabh Maheshwary, Srivatsava Daruru, Sagar Davasam, Vikas Yadav, Srinivas Sunkara, and Sai Rajeswar. R2v agent: Teaching slms when to ask for help, 2026. URL https://arxiv.org/abs/2605.16604.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset, 2021. URL https://arxiv.org/abs/2103.03874.

Wittawat Jitkrittum, Neha Gupta, Aditya Krishna Menon, Harikrishna Narasimhan, Ankit Singh Rawat, and Sanjiv Kumar. When does confidence-based cascade deferral suffice?, 2024. URL https://arxiv.org/abs/2307.02764.

Vansh Kapoor, Aman Gupta, Hao Chen, Anurag Beniwal, Jing Huang, and Aviral Kumar. Trim: Hybrid inference via targeted stepwise routing in multi-step reasoning tasks, 2026. URL https://arxiv.org/abs/2601.10245.

Steven Kolawole, Don Dennis, Ameet Talwalkar, and Virginia Smith. Agreement-based cascading for efficient inference, 2025. URL https://arxiv.org/abs/2407.02348.

Jannik Kossen, Jiatong Han, Muhammed Razzak, Lisa Schut, Shreshth Malik, and Yarin Gal. Semantic entropy probes: Robust and cheap hallucination detection in llms, 2024. URL https://arxiv.org/abs/2406.15927.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation, 2023. URL https://arxiv.org/abs/2302.09664.

Changhao Li, Rushi Qiang, Jiawei Huang, Chenxiao Gao, Chao Zhang, Niao He, and Bo Dai. Revisiting dagger in the era of llm-agents, 2026. URL https://arxiv.org/abs/2605.12913.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step, 2023. URL https://arxiv.org/abs/2305.20050.

Zechun Liu, Changsheng Zhao, Forrest Iandola, Chen Lai, Yuandong Tian, Igor Fedorov, Yunyang Xiong, Ernie Chang, Yangyang Shi, Raghuraman Krishnamoorthi, Liangzhen Lai, and Vikas Chandra. Mobilellm: Optimizing sub-billion parameter language models for on-device use cases, 2024. URL https://arxiv.org/abs/2402.14905.

Keming Lu, Hongyi Yuan, Runji Lin, Junyang Lin, Zheng Yuan, Chang Zhou, and Jingren Zhou. Routing to the expert: Efficient reward-guided ensemble of large language models, 2023. URL https://arxiv.org/abs/2311.08692.

Zhenyan Lu, Xiang Li, Dongqi Cai, Rongjie Yi, Fangming Liu, Wei Liu, Jian Luan, Xiwen Zhang, Nicholas D. Lane, and Mengwei Xu. Demystifying small language models for edge deployment. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14747–14764, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.718. URL https://aclanthology.org/2025.acl-long.718/.

Yingfeng Luo, Hongyu Liu, Dingyang Lin, Kaiyan Chang, Chenglong Wang, Bei Li, Quan Du, Tong Xiao, and Jingbo Zhu. Routelmt: Learned sample routing for hybrid llm translation deployment, 2026. URL https://arxiv.org/abs/2604.22520.

Xing Han Lù, Amirhossein Kazemnejad, Nicholas Meade, Arkil Patel, Dongchan Shin, Alejandra Zambrano, Karolina Stańczak, Peter Shaw, Christopher J. Pal, and Siva Reddy. Agentrewardbench: Evaluating automatic evaluations of web agent trajectories, 2025. URL https://arxiv.org/abs/2504.08942.

Hussein Mozannar and David Sontag. Consistent estimators for learning to defer to an expert, 2021. URL https://arxiv.org/abs/2006.01862.

Tianyi Niu, Justin Chih-Yao Chen, Genta Indra Winata, Shi-Xiong Zhang, Supriyo Chakraborty, Sambit Sahu, Yue Zhang, Elias Stengel-Eskin, and Mohit Bansal. Routing with generated data: Annotation-free llm skill estimation and expert selection, 2026. URL https://arxiv.org/abs/2601.09692.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M Waleed Kadous, and Ion Stoica. Routellm: Learning to route llms with preference data, 2025. URL https://arxiv.org/abs/2406.18665.

Shishir G Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, 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. URL https://openreview.net/forum?id=2GmDdhBdDk.

Stephan Rabanser, Nathalie Rauschmayr, Achin Kulshrestha, Petra Poklukar, Wittawat Jitkrittum, Sean Augenstein, Congchao Wang, and Federico Tombari. Gatekeeper: Improving model cascades through confidence tuning, 2025. URL https://arxiv.org/abs/2502.19335.

Zachary Ravichandran, Ignacio Hounie, Fernando Cladera, Alejandro Ribeiro, George J. Pappas, and Vijay Kumar. Distilling on-device language models for robot planning with minimal human intervention, 2025. URL https://arxiv.org/abs/2506.17486.

Stephane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning, 2011. URL https://arxiv.org/abs/1011.0686.

Vishnu Sarukkai, Asanshay Gupta, James Hong, Michaël Gharbi, and Kayvon Fatahalian. Inference-time distillation: Cost-efficient agents without fine-tuning or manual prompt engineering, 2026. URL https://arxiv.org/abs/2512.02543.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning, 2023. URL https://arxiv.org/abs/2303.11366.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters, 2024. URL https://arxiv.org/abs/2408.03314.

KV Aditya Srivatsa, Kaushal Kumar Maurya, and Ekaterina Kochmar. Harnessing the power of multiple minds: Lessons learned from llm routing, 2024. URL https://arxiv.org/abs/2405.00467.

Peiyi Wang, Lei Li, Zhihong Shao, R. X. Xu, Damai Dai, Yifei Li, Deli Chen, Y. Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations, 2024. URL https://arxiv.org/abs/2312.08935.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models, 2023. URL https://arxiv.org/abs/2203.11171.

Yihao Xue, Kristjan Greenewald, Youssef Mroueh, and Baharan Mirzasoleiman. Verify when uncertain: Beyond self-consistency in black box hallucination detection, 2026. URL https://arxiv.org/abs/2502.15845.

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.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task, 2019. URL https://arxiv.org/abs/1809.08887.

Zhongzhi Yu, Zheng Wang, Yuhan Li, Haoran You, Ruijie Gao, Xiaoya Zhou, Sreenidhi Reedy Bommu, Yang Katie Zhao, and Yingyan Celine Lin. Edge-llm: Enabling efficient large language model adaptation on edge devices via layerwise unified compression and adaptive layer tuning and voting, 2024. URL https://arxiv.org/abs/2406.15758.

Murong Yue, Jie Zhao, Min Zhang, Liang Du, and Ziyu Yao. Large language model cascades with mixture of thoughts representations for cost-efficient reasoning, 2024. URL https://arxiv.org/abs/2310.03094.

Huaye Zeng, Dongfu Jiang, Haozhe Wang, Ping Nie, Xiaotong Chen, and Wenhu Chen. Acecoder: Acing coder rl via automated test-case synthesis. *ArXiv*, abs/2207.01780, 2025.

Tuo Zhang, Asal Mehradfar, Dimitrios Dimitriadis, and Salman Avestimehr. Leveraging uncertainty estimation for efficient llm routing, 2025. URL https://arxiv.org/abs/2502.11021.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023. URL https://arxiv.org/abs/2306.05685.

A Appendix

A.1 Extended Related Works

As large language models are increasingly deployed, a broad literature has emerged around adaptive inference to balance response quality and computational cost. This design space generally splits into two paradigms: prompt-based routing, which selects a model before any text is generated (Ong et al., 2025; Ding et al., 2024), and sequential cascades, which evaluate a cheaper model's output before

deciding whether to escalate (Chen et al., 2023; Yue et al., 2024; Jitkrittum et al., 2024). While recent works such as Dekoninck et al. (2025) develop a unified formulation bridging both approaches, our work is specifically grounded in the cascade paradigm. Motivated by the growing need to deploy LLMs locally on resource-constrained edge devices (Liu et al., 2024; Yu et al., 2024; Ravichandran et al., 2025), we study a two-model cascade with a significant capability gap. In this setting, a weak local model first generates candidate responses, and the system subsequently decides whether to invoke a more capable, expensive strong model based entirely on the realized generation history. To contextualize our contributions, the remainder of this section categorizes prior work along three critical axes: when the routing decision occurs, the required test-time inference compute, and the reliance on ground-truth supervision.

Prompt-Based Routing and Learning to Defer. Prompt-based routers learn which model should answer an input without first observing its generated response. Building on classical literature regarding optimal deferral rules (Mozannar & Sontag, 2021; Jitkrittum et al., 2024), methods like Hybrid LLM (Ding et al., 2024) and Route LLM (Ong et al., 2025) allocate queries between models using learned predictions of quality. Other approaches focus on selecting a suitable expert from a larger pool of LLMs by predicting model performance through meta-models (akota et al., 2024), distilling reward assessments to identify expertise (Lu et al., 2023), using input-dependent routing (Srivatsa et al., 2024), or estimating translation improvements using prompt representations (Luo et al., 2026). These methods fundamentally differ from our approach in two ways: they make routing decisions *before* observing the generated weak response, and they heavily rely on ground-truth correctness or reward signals for training. In contrast, our cascade evaluates the realized weak response before routing, strictly in an unannotated two-model setting.

Supervised Confidence and Deferral ($N = 1$). Cascade methods commonly assess whether a single generated weak response is reliable enough to retain. FrugalGPT (Chen et al., 2023) trains a response-quality scorer to escalate when its confidence is insufficient. Gupta et al. (2024) study token-level uncertainty and learned deferral predictors supervised directly by whether the strong model corrects a weak-model error. Automix (Aggarwal et al., 2025) utilizes a POMDP routing policy trained on self-verification judgments and ground-truth signals. Finally, SELF-REF (Chuang et al., 2025) and Gatekeeper (Rabanser et al., 2025) train the weak model to express verbalized confidence or produce more useful confidence scores. All of these methods rely on task-specific correctness labels for supervision, and cannot be deployed in our purely label-free setting. Motivated by these works, our supervised weak-correctness probe PROXY REWARD serves as a controlled oracle reference to represent this category of quality-based deferral. Our proposed framework assumes no access to ground-truth correctness or improvement signals.

Consistency and Ensemble-Based Cascades ($N > 1$ Label-Free). Consistency-based approaches assess reliability without labels by comparing multiple generated responses. Yue et al. (2024) use self-consistency((Wang et al., 2023)) agreement among reasoning paths for deferral. Semantic Entropy (Kuhn et al., 2023) and Kernel Language Entropy (Farquhar et al., 2024) measure uncertainty over clusters of equivalent answers which can be used as deferral rules in cascades. Similarly, Zhang et al. (2025) rely on clustering to compute the semantic entropies of both student and teacher models to construct routing preferences. Other methods evaluate consensus across an ensemble of multiple models: ABC (Kolawole et al., 2025) uses pairwise similarities across inexpensive models, SMOOTHIE (Guha et al., 2024) estimates quality from distances between embedded outputs (which requires generator triplets), and Niu et al. (2026) groups queries into skill clusters via confidence-weighted ensemble consensus. These within-model consistency methods (particularly self-consistency and semantic entropy) directly motivate our multi-sample ($N > 1$) baselines. However, as a broader category, these approaches fundamentally differ from our framework in their operational assumptions and mechanics. First, ensemble routers ((Guha et al., 2024; Niu et al., 2026)) are designed to select among a diverse pool of multiple models, whereas we study a strict cascade between a single local weak model and a more capable strong model. Consequently, our framework only requires the weak model's internal representations; the strong model can remain entirely black-box, accessed only for its text outputs during offline training. Second, consistency methods require generating multiple responses at inference time and rely heavily on semantic clustering. As we demonstrate in Section 6.2, grouping responses into equivalence classes is notoriously brittle for open-ended tasks.

Learned Label-Free Reliability and Cross-Model Verification. Recent work has explored learning reliability predictors from unlabeled outputs or cross-model comparisons. Semantic Entropy

Probes (Kossen et al., 2024) predict semantic-entropy targets directly from model representations to enable single-response uncertainty estimation. HaloScope (Du et al., 2024) derives pseudo-labels from unlabeled response representations to train a truthfulness classifier. Additionally, Xue et al. (2026), Hamidieh et al. (2026), and Gorbett & Jana (2026) explore cross-model disagreement and verifier surprise to quantify uncertainty. These works provide strong conceptual motivation for comparing models and utilizing hidden-state probes. However, rather than detecting hallucinations or requiring test-time queries to a second model, our approach uniquely trains a probe on offline strong-model disagreement to directly approximate the optimal cascade routing gain, evaluating it autonomously at test time.

Allocating Test-Time Computation. Additional inference computation can significantly improve response quality by generating and selecting among multiple candidates. For example, self-consistency (Wang et al., 2023) aggregates sampled reasoning paths, while broader test-time compute strategies (Snell et al., 2024) scale sampling and refinement based on input difficulty. For routing specifically, BEST-Route (Ding et al., 2025) pre-determines a sampling budget combined with model selection, and ModelSwitch (Chen et al., 2025) evaluates consistency across multi-model samples to adaptively halt generation. Our work studies when to escalate after observing the weak model's candidate responses. We characterize the expected improvement from replacing the selected weak response with a strong-model response and develop a disagreement-based approximation to guide this decision without ground-truth correctness labels.

Sequential Tasks and Step-Level Routing. In sequential tasks, escalation decisions must be made repeatedly as the agent's state evolves. TRIM (Kapoor et al., 2026) uses process-level feedback to decide when to replace a weak reasoning step with a strong-model step. R2V-Agent (Hemadri et al., 2026) learns step-level escalation from weak-policy episode outcomes by combining uncertainty statistics with task-specific verifier features trained via verifier-guided DPO. Alternatively, Inference-Time Distillation (Sarukkai et al., 2026) determines escalation based on consistency among candidate actions to guide a weak agent. These methods typically rely on explicit step-level correctness labels, task-specific process verifiers, or expensive test-time consistency checks (which we benchmark via our multi-sample baselines). Our agentic experiments (Section 6.1) show that by learning from objective action disagreement, we can outperform consistency based alternatives and effectively route at the step level without ever requiring episode-success labels or step-correctness annotations.

A.1.1 Population estimates of routing signals across varying weak-strong model pairs

We extend the population-level comparison in Section 4.2 to the full routing-budget range $B \in [0, 1]$ for completeness and additional weak–strong model pairs and examine how the findings vary across models, datasets, and numbers of weak-model samples. Figure 4 reports the one-shot setting, while Figures 5 and 6 report results with $N = 4$ and $N = 8$ weak-model samples, respectively.

The results support the main observations: corrected disagreement generally improves on raw weak–strong disagreement and closely approximates optimal routing when the strong model is accurate. Its advantage over weak-model correctness is most apparent at limited routing budgets; the ordering can change at larger budgets. We first report the one-shot setting ($N = 1$), then vary the number of weak-model samples N. While correctness flags when the weak model is wrong, it cannot anticipate whether the strong model will actually succeed. In practice, weak-model correctness is often viewed as a highly accessible baseline because the strong model can act as an automated judge to provide those labels. Conversely, obtaining ground-truth labels for the strong model's own correctness is notoriously difficult because it typically surpasses available automated judges. Our disagreement scores bypass this labeling bottleneck entirely while explicitly capturing potential accuracy gains. At very high escalation budgets, weak-model correctness slightly surpasses corrected disagreement because it strictly avoids routing those rare instances where the weak model is correct but the strong model fails. However, in typical weak-to-strong deployments, these failure modes are infrequent, making low-budget efficiency the governing practical metric.

A.2 Extended Finite-Sample Results Across Routing and Sampling Budgets

We supplement Section 6.2 with detailed results on MATH-500, SPIDER, and ACECODE in Tables 1, 3, and 5, respectively. The tables include baselines omitted from the main-text plots, additional routing budgets, and $N \in \{1, 4, 8, 16\}$ weak-model samples per prompt.

The results support the findings in the main text. On MATH-500, self-consistency and semantic entropy approach corrected disagreement as the routing budget and sample count increase, and slightly outperform it at the 50% budget for $N > 1$. These consistency-based scores are less effective on SPIDER and ACECODE, where grouping equivalent responses is more difficult. On ACECODE, increasing N provides little benefit to either the selected weak response or consistency-based routing. Disagreement-based scores remain competitive across all three datasets, with corrected disagreement generally matching or improving on raw weak–strong disagreement.

Table 1: MATH-500 routing results for the weak–strong pair **Phi-3.5-mini-instruct → GPT-4o-mini**. The student answer is kept unless routed to the strong model. Each panel reports accuracy at different routing budgets. The $b = 0$ value is the accuracy of the kept weak answer before routing.

Setting	Routing score	5%	10%	15%	20%	25%	30%	50%
$N = 1$ **weak sample per prompt** ($b = 0$: 0.380)								
Corrected disagreement	ours	**0.419**	**0.452**	**0.482**	**0.513**	**0.540**	**0.568**	**0.659**
Weak–strong disagreement	ours	0.397	0.418	0.440	0.464	0.491	0.519	0.631
Token entropy	baseline	0.404	0.427	0.447	0.468	0.489	0.509	0.584
Proxy reward†	baseline	0.396	0.415	0.436	0.460	0.487	0.514	0.629
Oracle routing gain	oracle	0.429	0.479	0.529	0.578	0.626	0.670	0.776
$N = 4$ **weak samples per prompt** ($b = 0$: 0.430)								
Corrected disagreement	ours	**0.465**	**0.495**	**0.524**	**0.551**	**0.576**	**0.599**	0.677
Weak–strong disagreement	ours	0.447	0.467	0.489	0.512	0.539	0.563	0.664
Self-consistency	baseline	0.456	0.482	0.509	0.536	0.564	0.590	**0.679**
Semantic entropy	baseline	0.456	0.483	0.509	0.536	0.563	0.590	0.676
Token entropy	baseline	0.449	0.469	0.486	0.505	0.524	0.542	0.607
Proxy reward†	baseline	0.445	0.463	0.484	0.507	0.533	0.558	0.661
Oracle routing gain	oracle	0.480	0.529	0.579	0.627	0.672	0.712	0.783
$N = 8$ **weak samples per prompt** ($b = 0$: 0.467)								
Corrected disagreement	ours	**0.498**	**0.524**	**0.552**	**0.575**	**0.596**	**0.617**	0.686
Weak–strong disagreement	ours	0.484	0.503	0.523	0.547	0.571	0.593	0.680
Self-consistency	baseline	0.492	0.518	0.544	0.568	0.592	0.615	**0.694**
Semantic entropy	baseline	0.493	0.518	0.544	0.567	0.589	0.612	0.690
Token entropy	baseline	0.483	0.501	0.516	0.532	0.549	0.564	0.623
Proxy reward†	baseline	0.482	0.499	0.520	0.543	0.566	0.589	0.680
Oracle routing gain	oracle	0.516	0.566	0.615	0.662	0.704	0.741	0.787
$N = 16$ **weak samples per prompt** ($b = 0$: 0.490)								
Corrected disagreement	ours	**0.518**	**0.541**	**0.567**	**0.588**	**0.609**	**0.628**	0.691
Weak–strong disagreement	ours	0.507	0.525	0.544	0.566	0.588	0.609	0.687
Self-consistency	baseline	0.514	0.538	0.562	0.584	0.606	0.627	**0.700**
Semantic entropy	baseline	0.514	0.538	0.561	0.582	0.602	0.624	0.697
Token entropy	baseline	0.505	0.519	0.533	0.548	0.564	0.577	0.634
Proxy reward†	baseline	0.505	0.522	0.542	0.562	0.585	0.606	0.687
Oracle routing gain	oracle	0.539	0.588	0.637	0.682	0.723	0.754	0.789

Table 2: *
Consistency-based scores are undefined for $N = 1$ and are therefore omitted in that panel. Bold numbers mark the best non-oracle routing score at each budget within a fixed N. †Proxy reward denotes the reward-model-based routing baseline.

Table 3: SPIDER routing results for the weak–strong pair **Qwen3-1.7B → OmniSQL-7B**. The weak query is kept unless routed to the strong model. Each panel reports accuracy at different routing budgets. The $b = 0$ value is the accuracy of the kept weak query before routing.

Setting	Routing score	5%	10%	15%	20%	25%	30%	50%
$N = 1$ **weak sample per prompt** ($b = 0$: 0.499)								
Corrected disagreement	ours	**0.537**	**0.563**	**0.586**	**0.608**	**0.628**	**0.648**	**0.716**
Weak–strong disagreement	ours	0.533	0.559	0.582	0.605	0.626	0.646	**0.716**
Token entropy	baseline	0.515	0.530	0.547	0.562	0.577	0.591	0.647
Proxy reward†	baseline	0.532	0.557	0.579	0.601	0.623	0.643	0.714
Oracle routing gain	oracle	0.549	0.598	0.648	0.698	0.744	0.783	0.818
$N = 4$ **weak samples per prompt** ($b = 0$: 0.529)								
Corrected disagreement	ours	**0.565**	**0.591**	**0.612**	**0.632**	**0.650**	**0.668**	0.728
Weak–strong disagreement	ours	0.561	0.586	0.608	0.629	0.647	0.666	**0.729**
Self-consistency	baseline	0.552	0.573	0.590	0.608	0.625	0.643	0.700
Semantic entropy	baseline	0.552	0.573	0.591	0.609	0.626	0.644	0.701
Token entropy	baseline	0.542	0.555	0.569	0.582	0.595	0.609	0.659
Proxy reward†	baseline	0.561	0.585	0.605	0.625	0.645	0.663	0.727
Oracle routing gain	oracle	0.579	0.628	0.678	0.728	0.771	0.802	0.820
$N = 8$ **weak samples per prompt** ($b = 0$: 0.544)								
Corrected disagreement	ours	**0.580**	**0.604**	**0.624**	**0.642**	**0.660**	**0.676**	0.734
Weak–strong disagreement	ours	0.577	0.601	0.622	**0.642**	0.659	**0.676**	**0.735**
Self-consistency	baseline	0.565	0.583	0.602	0.620	0.637	0.653	0.708
Semantic entropy	baseline	0.564	0.583	0.601	0.620	0.637	0.653	0.709
Token entropy	baseline	0.557	0.569	0.582	0.594	0.607	0.619	0.666
Proxy reward†	baseline	0.577	0.600	0.618	0.636	0.655	0.673	0.732
Oracle routing gain	oracle	0.594	0.643	0.693	0.742	0.782	0.809	0.820
$N = 16$ **weak samples per prompt** ($b = 0$: 0.552)								
Corrected disagreement	ours	**0.588**	**0.612**	**0.631**	**0.649**	**0.666**	**0.682**	**0.737**
Weak–strong disagreement	ours	0.585	0.608	0.629	0.647	0.664	0.681	**0.737**
Self-consistency	baseline	0.570	0.588	0.606	0.625	0.642	0.658	0.712
Semantic entropy	baseline	0.570	0.589	0.605	0.622	0.640	0.658	0.714
Token entropy	baseline	0.564	0.575	0.587	0.600	0.611	0.624	0.668
Proxy reward†	baseline	0.584	0.607	0.625	0.643	0.661	0.677	0.734
Oracle routing gain	oracle	0.602	0.651	0.701	0.749	0.788	0.811	0.821

Table 4: *

Consistency-based scores are undefined for $N = 1$ and are therefore omitted in that panel. Bold numbers mark the best non-oracle routing score at each budget within a fixed N. †Proxy reward denotes the reward-model-based routing baseline.

Table 5: ACECODE routing results for the weak–strong pair **Qwen2.5-Coder-1.5B-Instruct** → **Qwen3-Coder-10B-A3B-Instruct**. The weak program is kept unless routed to the strong model. Each panel reports accuracy at different routing budgets. The $b = 0$ value is the accuracy of the kept weak program before routing.

Routing score	Type	5%	10%	15%	20%	25%	30%	50%
$N = 1$ **weak sample per prompt** ($b = 0$: 0.447)								
Corrected disagreement	ours	**0.468**	**0.486**	**0.503**	**0.519**	**0.535**	**0.552**	**0.608**
Weak–strong disagreement	ours	0.464	0.484	0.502	0.518	**0.535**	0.551	0.606
Token entropy	baseline	0.460	0.475	0.489	0.504	0.518	0.532	0.591
Proxy reward†	baseline	0.458	0.471	0.486	0.501	0.517	0.534	0.594
Oracle routing gain	oracle	0.497	0.547	0.596	0.646	0.690	0.708	0.709
$N = 4$ **weak samples per prompt** ($b = 0$: 0.449)								
Corrected disagreement	ours	**0.469**	**0.487**	**0.503**	**0.520**	0.535	**0.552**	**0.608**
Weak–strong disagreement	ours	0.465	0.485	**0.503**	0.519	**0.536**	**0.552**	0.607
Self-consistency	baseline	0.460	0.472	0.484	0.495	0.507	0.519	0.565
Semantic entropy	baseline	0.459	0.470	0.482	0.494	0.506	0.518	0.564
Token entropy	baseline	0.461	0.476	0.490	0.505	0.520	0.535	0.592
Proxy reward†	baseline	0.460	0.472	0.488	0.503	0.519	0.535	0.594
Oracle routing gain	oracle	0.500	0.550	0.599	0.649	0.692	0.709	0.710
$N = 8$ **weak samples per prompt** ($b = 0$: 0.449)								
Corrected disagreement	ours	**0.470**	**0.487**	**0.503**	**0.520**	**0.536**	**0.552**	**0.608**
Weak–strong disagreement	ours	0.465	0.485	**0.503**	0.519	0.535	0.551	0.607
Self-consistency	baseline	0.459	0.471	0.482	0.494	0.505	0.517	0.563
Semantic entropy	baseline	0.459	0.470	0.482	0.493	0.506	0.517	0.564
Token entropy	baseline	0.464	0.478	0.492	0.507	0.521	0.536	0.592
Proxy reward†	baseline	0.460	0.472	0.488	0.503	0.519	0.535	0.594
Oracle routing gain	oracle	0.499	0.549	0.598	0.648	0.691	0.708	0.709
$N = 16$ **weak samples per prompt** ($b = 0$: 0.448)								
Corrected disagreement	ours	**0.468**	**0.486**	**0.503**	**0.519**	**0.536**	**0.551**	**0.608**
Weak–strong disagreement	ours	0.464	0.484	0.502	0.518	0.534	0.550	0.606
Self-consistency	baseline	0.459	0.471	0.483	0.495	0.507	0.518	0.565
Semantic entropy	baseline	0.458	0.470	0.481	0.493	0.504	0.516	0.563
Token entropy	baseline	0.463	0.477	0.491	0.506	0.521	0.535	0.592
Proxy reward†	baseline	0.460	0.473	0.488	0.504	0.519	0.535	0.594
Oracle routing gain	oracle	0.499	0.549	0.599	0.648	0.691	0.708	0.709

Table 6: *

Consistency-based scores are undefined for $N = 1$ and are therefore omitted in that panel. For $N > 1$, judge-based clusters do not improve the kept-program accuracy, so gains come from improved routing signals rather than a better plurality answer. Bold numbers mark the best non-oracle routing score at each budget within a fixed N. †Proxy reward denotes the reward-model-based routing baseline.

A.3 Ablations on Selection Policy

In the main experiments, we use the same selection policy across routing methods to isolate the effect of the routing score. Here, we examine how the choice of weak response affects cascade accuracy.

Recall from Section 3 that, given $h_\sigma = (p, y_{w,1}, \ldots, y_{w,N})$, the optimal selection policy is

$$\phi^*(h_\sigma) \in \arg\max_{i \le N} \Pr(Y = y_{w,i} \mid P = p).$$

Thus, the selection policy should choose the weak response most likely to be correct. Learning this probability directly requires correctness labels. We compare several selection rules that do not require ground-truth supervision, together with a supervised correctness probe:

- **Majority vote.** Group equivalent responses and select a response from the largest group.
- **Token entropy.** Select the response with the lowest mean token entropy under the weak model.
- **Disagreement (ours).** Select the response with the lowest predicted weak–strong disagreement, $\phi(h_\sigma) \in \arg\min_{i \le N} \widehat{f}_d(\mathbf{h}(p, y_{w,i}))$. This favors responses that the strong model is more likely to agree with.
- **Correctness probe.** Select the response with the highest predicted probability of correctness. This probe uses ground-truth correctness labels during training and serves as a supervised comparison.

Selection across routing scores. Figure 7 compares these selection policies on MATH-500 with $N = 8$ weak responses per prompt. Each panel fixes the routing score and varies the selection policy and routing budget. Majority vote achieves the highest accuracy at every reported budget for all four routing scores. This is consistent with the effectiveness of grouping equivalent final answers in mathematical reasoning. Disagreement-based selection performs similarly to the supervised correctness probe, and both improve on random selection and token entropy.

Selection across sample counts. Figures 8 and 9 extend the comparison to SPIDER and ACECODE. We fix the routing score to corrected disagreement and vary both the routing budget and the number of weak responses N. These experiments examine whether selection policies can benefit from additional candidates in tasks where identifying equivalent responses is more difficult.

A.4 Additional Results for the Agentic Experiments

In this section, we provide complementary results for the multi-step agentic experiments presented in Section 6.1. Table 7 reports additional results, including the token-entropy baseline omitted from the main text. Alongside full-trajectory success, we report the mean task score for completeness.

Table 7: WebShop routing performance (200 evaluation tasks) comparing realized routing budget, end-to-end success rate, and average task score across multiple label-free baselines.

Routing Strategy	**Success Rate (SE)**			**Average Task Score**		
Realized Budget (≈)	**20%**	**40%**	**60%**	**20%**	**40%**	**60%**
Corrected disagreement (ours)	**0.210** (0.029)	**0.460** (0.035)	**0.765** (0.030)	**0.665**	**0.748**	**0.828**
Weak–strong disagreement (ours)	0.065 (0.017)	0.260 (0.031)	0.715 (0.032)	0.466	0.716	0.880
Self-consistency	0.045 (0.015)	0.145 (0.025)	0.410 (0.035)	0.448	0.536	0.755
Token entropy	0.040 (0.014)	0.150 (0.025)	0.405 (0.035)	0.448	0.535	0.698
Random baseline	0.160 (0.026)	0.200 (0.028)	0.505 (0.035)	0.543	0.590	0.776

A.5 Probe Representations and Training

We provide additional details on the probes used to estimate the routing signals in Section 5. The probes are trained on empirical targets computed from offline model generations, while the weak

model remains frozen. We extract both probe inputs from the weak model's hidden states. For the strong-model self-disagreement probe, we use the hidden state at the final prompt token, denoted by $\mathbf{h}(p)$. For the weak–strong disagreement probe, we use the mean hidden state over the generated response tokens, denoted by $\mathbf{h}(p, y_w)$. The second representation depends on both the prompt and the generated response from the weak model. In the agentic environments, the trajectory history H_t serves as the prompt and the proposed action serves as the response.

We train each probe:

$$\widehat{f} \in \arg\min_{f \in \mathcal{F}} \frac{1}{|\mathcal{D}|} \sum_{(z,\widehat{u}) \in \mathcal{D}} \ell\big(f(\mathbf{h}(z)), \widehat{u}\big) \tag{12}$$

by minimizing binary cross-entropy with empirical probabilities as soft targets. For $\widehat{f}_d$, each example pairs a prompt–response representation $\mathbf{h}(p, y_w)$ with the target $\widehat{d}(p, y_w)$ defined in Equation 9. For $\widehat{f}_g$, each example pairs a prompt representation $\mathbf{h}(p)$ with the target $\widehat{g}(p)$ defined in Equation 10. Probe outputs are restricted to $[0, 1]$. For brevity, the main text writes these predictors as $\widehat{f}_d(p, y_w)$ and $\widehat{f}_g(p)$, leaving the representation extraction implicit.

Additionally, when ground-truth answers are available, we also train the proxy-reward probe $\widehat{f}_c$ using the same prompt–response representation and binary cross-entropy loss, with correctness target $1 - A(y_w, Y)$. This probe provides a supervised comparison; the two disagreement probes require no ground-truth correctness labels. At inference, computing the learned routing scores requires only weak-model representations; the strong model is queried only when the router chooses to escalate. We will release the training code and experiment configurations upon acceptance, which will including feature-extraction layers used for each model and hyperparameter details.

A.6 On-Policy Data Collection for Agentic Probes

To ensure our learned probes generalize to test-time deployments where strong-model interventions alter the trajectory state, we collect training data using an iterative, on-policy schedule inspired by DAgger (Ross et al., 2011; Li et al., 2026).

During test-time routing, executing strong-model actions steers the agent into states that the weak model would not visit autonomously. Fitting probes exclusively on purely weak-model trajectories results in a distribution mismatch (the classic compounding-shift problem of behavioral cloning (Ross et al., 2011).) We mitigate this by collecting training data across iterative phases: each phase rolls out the current routed policy, labels every visited state, and trains subsequent probes on the aggregated dataset from all preceding phases.

Dataset aggregation. In multi-step environments, let $H_t = (p, y^1, o^1, \ldots, y^{t-1}, o^{t-1})$ denote the trajectory history at step t, comprising the instruction prompt p, past actions y_i, and observations o_i. At each step, both models propose actions: the weak model generates K sampled actions $\{y_1^w, \ldots, y_K^w\}$ and K sampled actions $\{y_1^s, \ldots, y_K^s\}$ from the strong model. ($K = 10$ across all environments). For simplicity and consistency across agentic baselines, we fix the weak-model candidate selection policy $\phi(H_t)$ to execute the first sampled action (y_1^w). Using the task-specific transition agreement indicator A defined in Section 5, we compute empirical disagreement targets $\widehat{d}(H_t, y_i^w)$ and $\widehat{g}(H_t)$. We record the weak model's hidden states: the final history token representation $\mathbf{h}(H_t)$ for the self-disagreement probe $\widehat{f}_g$, and the mean hidden-state representation over action tokens $\mathbf{h}(H_t, y_i^w)$ for the weak–strong disagreement probe $\widehat{f}_d$.

Stochastic routing during collection. In the first phase, the weak model acts at every step. In subsequent phases, we fit the probes on the accumulated training data and keep them fixed throughout collection. At each visited history H_t, we map the corrected disagreement score to a routing probability:

$$p_{\text{route}}^{(m)}(H_t, y_1^w) = \begin{cases} 0, & m = 1, \\ \dfrac{1 + \widehat{f}_d^{(m)}(\mathbf{h}(H_t, y_1^w)) - \widehat{f}_g^{(m)}(\mathbf{h}(H_t))}{2}, & m \geq 2. \end{cases} \tag{13}$$

Both probes take values in $[0, 1]$, so the corrected disagreement score lies in $[-1, 1]$ and the routing probability lies in $[0, 1]$. We sample a routing decision from this probability and execute either the strong model's greedy action or the first weak sample.

Algorithm 1 On-Policy Collection of Disagreement Data

Require: Tasks $\mathcal{T}_{\text{train}}, \mathcal{T}_{\text{test}}$; weak and strong models; selection policy ϕ; samples K; phases M

$\mathcal{D} \leftarrow \emptyset$
for $m = 1, \ldots, M$ **do**
  if $m \geq 2$ **then**
    Fit $\widehat{f}_d^{(m)}, \widehat{f}_g^{(m)}$ on training-task records in $\mathcal{D}$
  end if
  for each task $u \in \mathcal{T}_{\text{train}} \cup \mathcal{T}_{\text{test}}$ **do**
    Initialize trajectory history H_1; set $t \leftarrow 1$
    while task u is not finished **do**
      Draw $y_{w,1}, \ldots, y_{w,K}$ independently from the weak model given H_t
      Draw $y_{s,1}, \ldots, y_{s,K}$ independently from the strong model given H_t
      $h_{t,K} \leftarrow (H_t, y_{w,1}, \ldots, y_{w,K})$; $i_t \leftarrow \phi(h_{t,K})$
      Compute $\widehat{g}(H_t)$ and $\{\widehat{d}(H_t, y_{w,i})\}_{i=1}^{K}$ from the sampled actions
      Add these targets and their weak-model representations to $\mathcal{D}$, tagged by task u
      $p_t \leftarrow 0$
      if $m \geq 2$ **then**
        $p_t \leftarrow \dfrac{1 + \widehat{f}_d^{(m)}\big(\mathbf{h}(H_t, y_{w,i_t})\big) - \widehat{f}_g^{(m)}\big(\mathbf{h}(H_t)\big)}{2}$
      end if
      Draw $b_t \sim \text{Bernoulli}(p_t)$
      if $b_t = 1$ **then**
        Draw y_s from the strong model given H_t
        $y^t \leftarrow y_s$
      else
        $y^t \leftarrow y_{w,i_t}$
      end if
      Execute y^t and observe o^t
      $H_{t+1} \leftarrow (H_t, y^t, o^t)$; $t \leftarrow t + 1$
    end while
  end for
end for
return $\mathcal{D}$

B PROOF OF THEOREM 1

We first give a more detailed statement of the result, including the optimal sampling policy. The proof has a simple structure. For a fixed number of weak samples, we condition on the sampled responses: the best weak response is the one with the largest correctness probability, and routing is optimal when the improvement offered by the strong model is large enough. To optimize sampling, we introduce prices for weak-model samples and strong-model queries. For each prompt, the optimal number of weak samples then maximizes its expected value net of the sampling price.

For the sampling characterization, assume that at most $K < \infty$ weak responses may be generated. Conditional on $P = p$, the weak responses are i.i.d. and independent of (Y, Y_s). Recall

$$q_p(y) := \Pr(Y = y \mid P = p), \qquad t_p := \Pr(Y_s = Y \mid P = p).$$

Theorem 2 (Full version of Theorem 1)**.** *Let $1 \le B_w \le K$ and $0 \le B_s \le 1$.*

(i) Fixed sampling. *Fix any sampling policy σ satisfying $\mathbb{E}[\sigma(P)] \le B_w$. For a realized history*

$$h_\sigma = (p, y_{w,1}, \dots, y_{w,\sigma(p)}),$$

define

$$\Delta(h_\sigma) := t_p - \max_{i \le \sigma(p)} q_p(y_{w,i}). \tag{14}$$

Then an optimal selection and routing policy is

$$\phi^*(h_\sigma) \in \arg\max_{i \le \sigma(p)} q_p(y_{w,i}), \qquad \pi^*(h_\sigma) = \begin{cases} 1, & \Delta(h_\sigma) > \tau, \\ 0, & \Delta(h_\sigma) < \tau, \\ \rho(h_\sigma), & \Delta(h_\sigma) = \tau, \end{cases} \tag{15}$$

where $\tau \ge 0$ and $\rho(h_\sigma) \in [0, 1]$ can be chosen so that

$$\mathbb{E}[\pi^*(H_\sigma)] \le B_s, \qquad \tau\big(\mathbb{E}[\pi^*(H_\sigma)] - B_s\big) = 0. \tag{16}$$

(ii) Optimal sampling. *For $n \in \{1, \dots, K\}$, define*

$$M_{p,n} := \max_{i \le n} q_p(Y_{w,i})$$

and

$$U_p(n; \tau) := \mathbb{E}[\max\{M_{p,n},\, t_p - \tau\} \mid P = p]\,. \tag{17}$$

Then, for some global weak-sampling price $\beta \ge 0$, an optimal sampling policy satisfies

$$\sigma^*(p) \in \arg\max_{1 \le n \le K} \{U_p(n; \tau) - \beta n\}\,. \tag{18}$$

Together with the selection and routing rules in equation 15, the prices can be chosen so that

$$\mathbb{E}[\sigma^*(P)] \le B_w, \qquad \beta\big(\mathbb{E}[\sigma^*(P)] - B_w\big) = 0, \tag{19}$$

and equation 16 holds.

If needed at a budget boundary, the policy may randomize between tied maximizers in equation 18 or at $\Delta(h_\sigma) = \tau$.

Proof. We prove the result in a few simple steps.

Claim 1. For fixed sampling, the selection rule in equation 15 is optimal.

Condition on a realized history $h_\sigma = (p, y_{w,1}, \dots, y_{w,\sigma(p)})$. Because the weak-model samples are independent of (Y, Y_s) conditional on P, the conditional correctness probabilities are

$$\Pr(Y_{w,i} = Y \mid H_\sigma = h_\sigma) = q_p(y_{w,i}), \qquad \Pr(Y_s = Y \mid H_\sigma = h_\sigma) = t_p.$$

Hence, for any selection and routing decisions, the conditional expected accuracy is

$$(1 - \pi(h_\sigma)) q_p(y_{w,\phi(h_\sigma)}) + \pi(h_\sigma) t_p.$$

For any fixed routing decision, this is maximized by selecting a weak response with the largest value of $q_p(y_{w,i})$. Therefore

$$\phi^*(h_\sigma) \in \arg\max_{i \le \sigma(p)} q_p(y_{w,i}).$$

With this choice, the conditional expected accuracy becomes

$$\max_{i \le \sigma(p)} q_p(y_{w,i}) + \pi(h_\sigma)\Delta(h_\sigma).$$

Thus, after optimizing selection, routing only needs to decide where to spend the strong-model budget based on the gain $\Delta(h_\sigma)$.

Claim 2. The optimal routing policy is a threshold rule on $\Delta(h_\sigma)$.

For fixed σ, maximizing accuracy is equivalent to

$$\max_{\pi} \mathbb{E}[\pi(H_\sigma)\Delta(H_\sigma)] \qquad \text{subject to} \qquad \mathbb{E}[\pi(H_\sigma)] \le B_s.$$

Therefore the budget should be allocated to histories with the largest values of Δ.

More formally, choose $\tau \ge 0$ and, if necessary, randomize on $\{\Delta = \tau\}$ so that equation 16 holds. For any other feasible routing policy π,

$$(\pi^* - \pi)(\Delta - \tau) \ge 0$$

pointwise. Hence

$$\begin{aligned}\mathbb{E}[(\pi^* - \pi)\Delta] &= \mathbb{E}[(\pi^* - \pi)(\Delta - \tau)] + \tau\big(\mathbb{E}[\pi^*] - \mathbb{E}[\pi]\big) \\ &\ge 0,\end{aligned}$$

where the second term is nonnegative by feasibility and complementary slackness. Thus π^* is optimal, proving part (i).

Claim 3. For fixed prices (β, τ), the optimal number of weak samples satisfies equation 18.

Introduce a price β for each weak-model sample and a price τ for each strong-model query. Consider a prompt p for which we choose n weak samples.

After the samples are observed, Claim 1 shows that the best weak-model accuracy is

$$M_{p,n} = \max_{i \le n} q_p(Y_{w,i}).$$

If we return the weak response, the priced value is $M_{p,n}$. If we query the strong model, the priced value is $t_p - \tau$. Therefore the optimal terminal value is

$$\max\{M_{p,n}, t_p - \tau\}.$$

Before the weak responses are sampled, its expected value is exactly $U_p(n;\tau)$ from equation 17. After subtracting the sampling cost βn, the value of choosing n samples is

$$U_p(n;\tau) - \beta n.$$

Since σ depends only on the prompt, the optimal choice can be made separately for each p, giving

$$\sigma^*(p) \in \arg\max_{1 \le n \le K} \{U_p(n;\tau) - \beta n\}.$$

The sampling rule also has a useful marginal interpretation. Define

$$G_p(n;\tau) := U_p(n+1;\tau) - U_p(n;\tau), \qquad n < K.$$

These marginal gains are nonincreasing in n. To see this, write

$$Z_i := q_p(Y_{w,i}), \qquad R_n := \max\{t_p - \tau, Z_1, \ldots, Z_n\}.$$

Then

$$G_p(n;\tau) = \mathbb{E}[(Z_{n+1} - R_n)_+ \mid P = p].$$

Since $R_{n+1} \geq R_n$ and a fresh weak response has the same distribution at every step, the expected improvement from one additional sample cannot increase with n. Thus equation 18 can equivalently be viewed as increasing the planned number of weak samples while the expected marginal gain exceeds β, and stopping once it falls below β.

Claim 4. Suitable prices recover the constrained optimum.

Let

$$A(\sigma, \phi, \pi)$$

denote expected accuracy, and define

$$W(\sigma) := \mathbb{E}[\sigma(P)], \qquad S(\sigma, \pi) := \mathbb{E}[\pi(H_\sigma)].$$

For prices $\beta, \tau \geq 0$, consider the priced objective

$$A(\sigma, \phi, \pi) - \beta W(\sigma) - \tau S(\sigma, \pi).$$

Claims 1–3 characterize a policy that maximizes this objective.

Under the standard randomized-policy formulation with finite K, the set of achievable triples (W, S, A) is convex and compact. Therefore a supporting-hyperplane argument gives prices $\beta, \tau \geq 0$ and boundary randomization, when necessary, satisfying equation 16 and equation 19.

It remains only to verify that these conditions imply constrained optimality. Let $(\sigma^*, \phi^*, \pi^*)$ be the resulting policy, and let (σ, ϕ, π) be any other feasible policy. Since the former maximizes the priced objective,

$$\begin{aligned} A(\sigma, \phi, \pi) &\leq A(\sigma^*, \phi^*, \pi^*) - \beta W(\sigma^*) - \tau S(\sigma^*, \pi^*) \\ &\quad + \beta W(\sigma) + \tau S(\sigma, \pi) \\ &\leq A(\sigma^*, \phi^*, \pi^*) + \beta(B_w - W(\sigma^*)) + \tau(B_s - S(\sigma^*, \pi^*)) \\ &= A(\sigma^*, \phi^*, \pi^*), \end{aligned}$$

where the last equality follows from complementary slackness. Hence the stated sampling, selection, and routing policies are jointly optimal. □

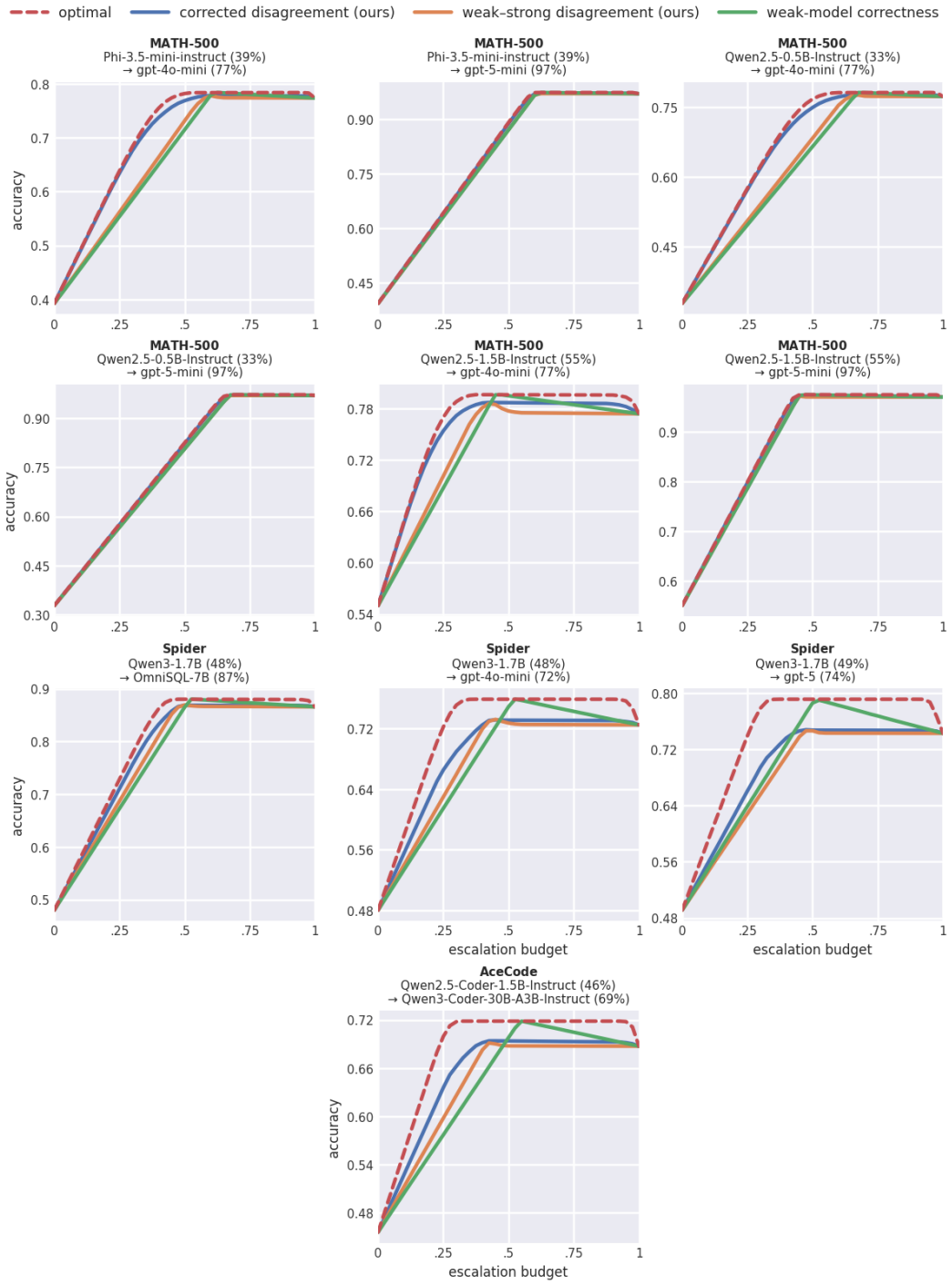


Figure 4: Population-level comparison of routing signals with $N = 1$ weak-model response. Each panel shows routing accuracy over the full budget range $B \in [0, 1]$ for a different dataset and weak–strong model pair. We compare optimal routing, corrected disagreement, raw weak–strong disagreement, and weak-model correctness.

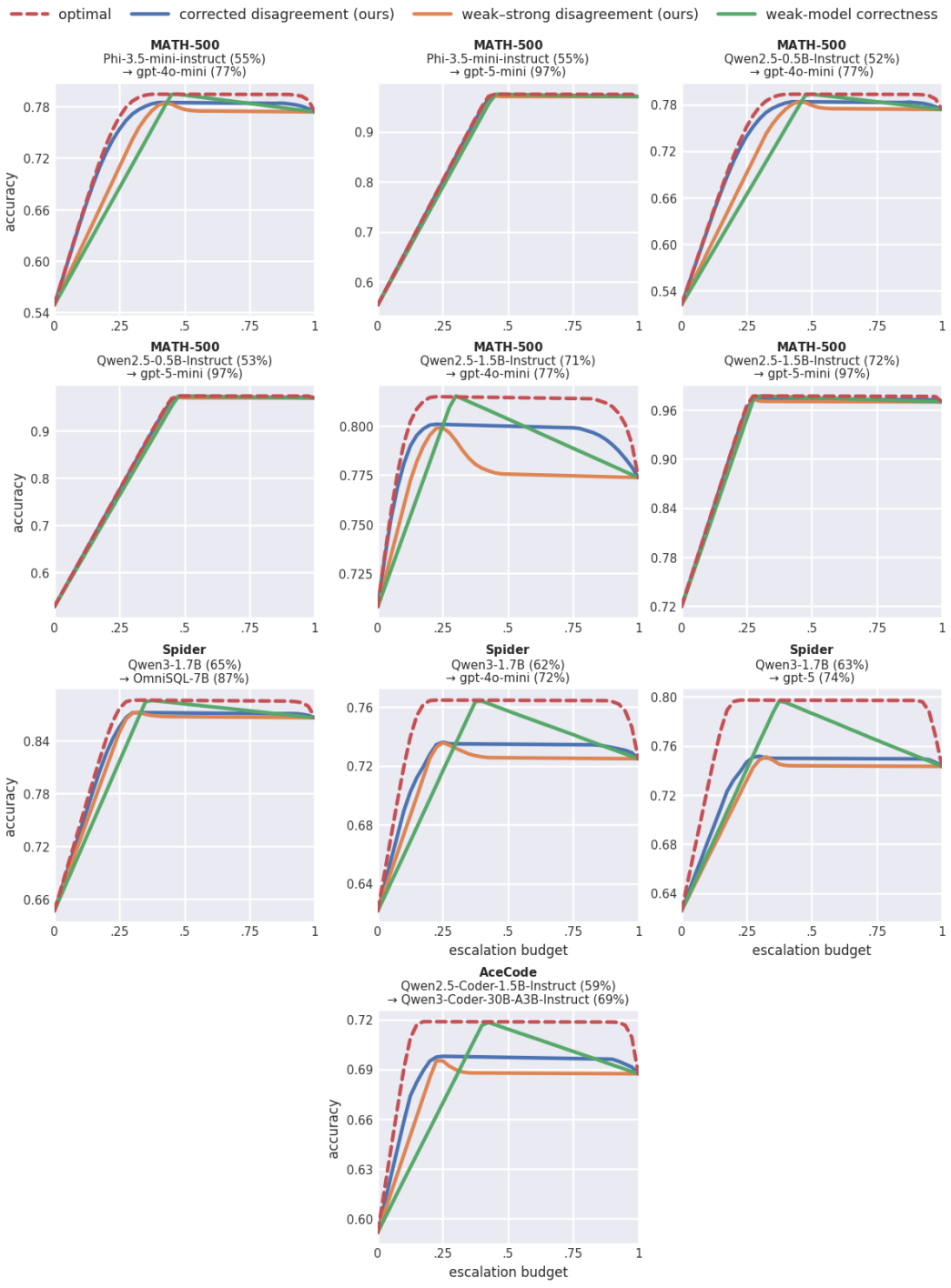


Figure 5: Population-level comparison of routing signals with $N = 4$ weak-model responses per prompt. Panels and routing signals follow Figure 4.

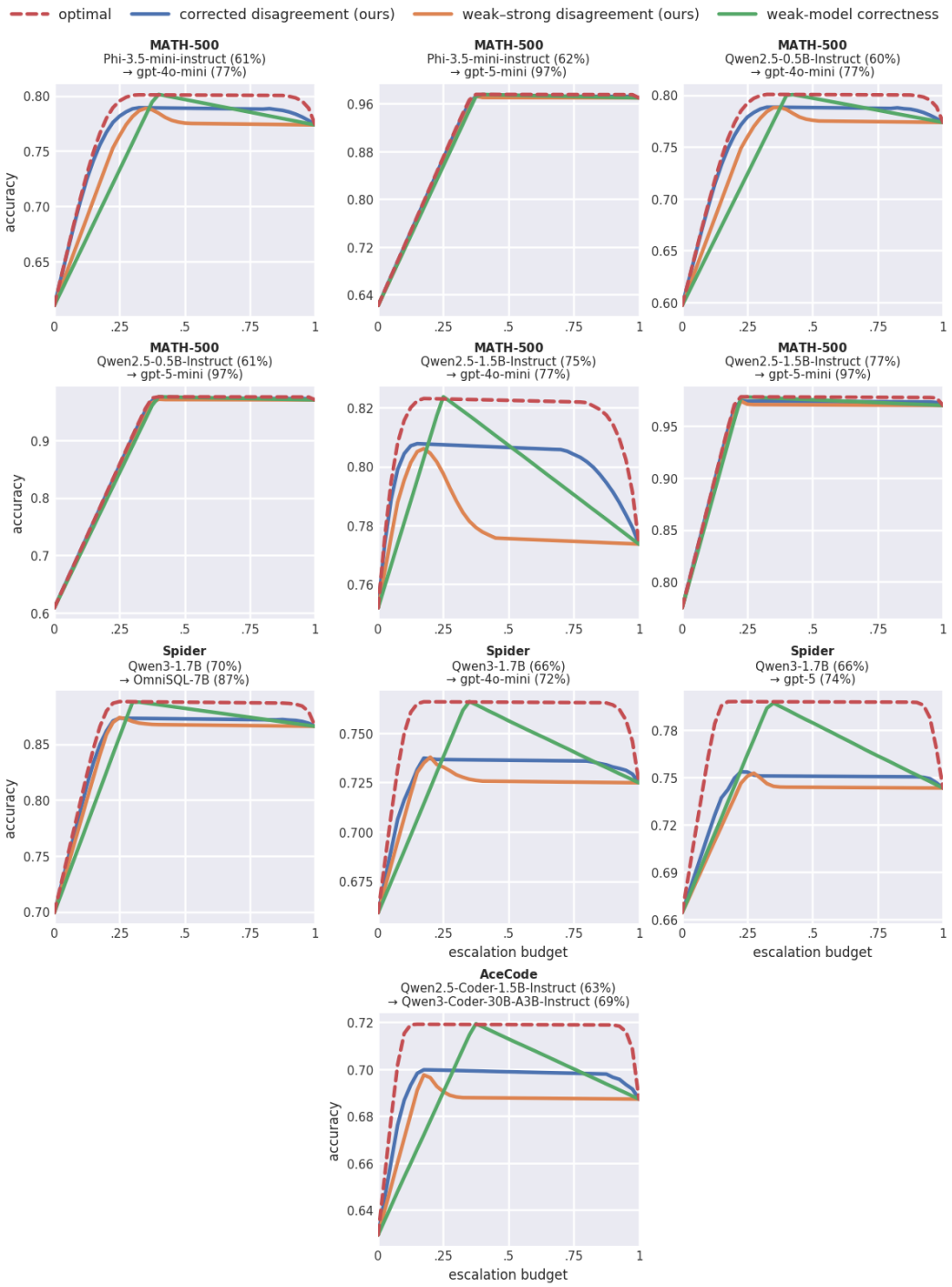
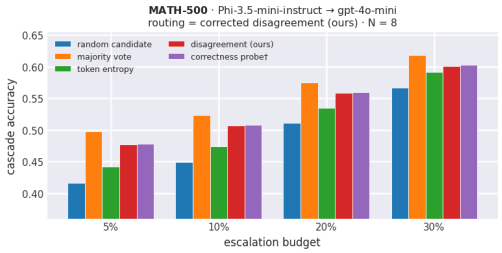


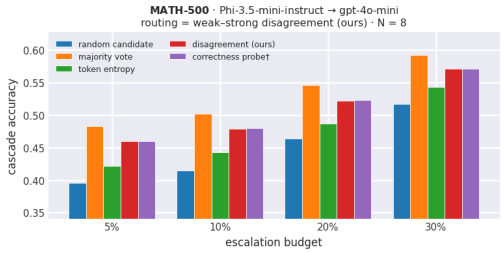
Figure 6: Population-level comparison of routing signals with $N = 8$ weak-model responses per prompt. Panels and routing signals follow Figure 4.

Routing: corrected disagreement (ours)



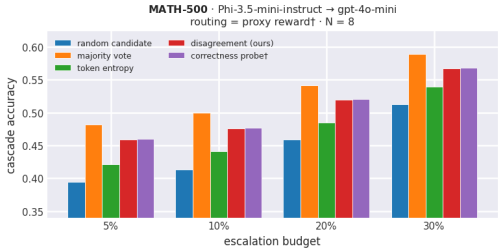
Selection	$b = 0\%$	$b = 5\%$	$b = 10\%$	$b = 20\%$	$b = 30\%$
Random candidate	0.377	0.416	0.449	0.511	0.567
Majority vote	**0.467**	**0.498**	**0.524**	**0.575**	**0.619**
Token entropy	0.404	0.443	0.474	0.535	0.592
Disagreement (ours)	0.449	0.478	0.507	0.558	0.601
Correctness probe†	0.450	0.478	0.508	0.560	0.602

Routing: weak–strong disagreement (ours)



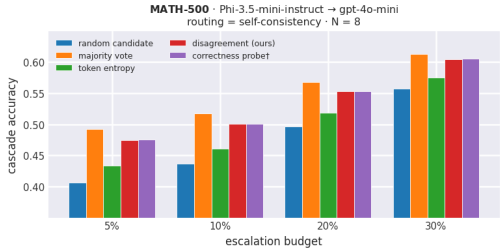
Selection	$b = 0\%$	$b = 5\%$	$b = 10\%$	$b = 20\%$	$b = 30\%$
Random candidate	0.377	0.395	0.415	0.464	0.517
Majority vote	**0.467**	**0.483**	**0.502**	**0.546**	**0.593**
Token entropy	0.404	0.422	0.443	0.487	0.543
Disagreement (ours)	0.449	0.460	0.479	0.522	0.571
Correctness probe†	0.450	0.460	0.480	0.523	0.571

Routing: proxy reward†



Selection	$b = 0\%$	$b = 5\%$	$b = 10\%$	$b = 20\%$	$b = 30\%$
Random candidate	0.377	0.394	0.413	0.459	0.513
Majority vote	**0.467**	**0.482**	**0.500**	**0.542**	**0.589**
Token entropy	0.404	0.421	0.441	0.485	0.539
Disagreement (ours)	0.449	0.459	0.476	0.520	0.567
Correctness probe†	0.450	0.460	0.477	0.521	0.568

Routing: self-consistency



Selection	$b = 0\%$	$b = 5\%$	$b = 10\%$	$b = 20\%$	$b = 30\%$
Random candidate	0.377	0.407	0.437	0.496	0.557
Majority vote	**0.467**	**0.492**	**0.518**	**0.567**	**0.613**
Token entropy	0.404	0.433	0.461	0.519	0.575
Disagreement (ours)	0.449	0.475	0.501	0.553	0.604
Correctness probe†	0.450	0.475	0.501	0.553	0.605

Figure 7: Effect of weak-response selection on MATH-500 for **Phi-3.5-mini-instruct → GPT-4o-mini**, using $N = 8$ weak responses per prompt and 500 test prompts. Each panel fixes the routing score and reports cascade accuracy for different selection policies and routing budgets. At $b = 0$, accuracy depends only on the selection policy. Bold values indicate the highest accuracy in each column. † denotes methods that use ground-truth supervision during training.

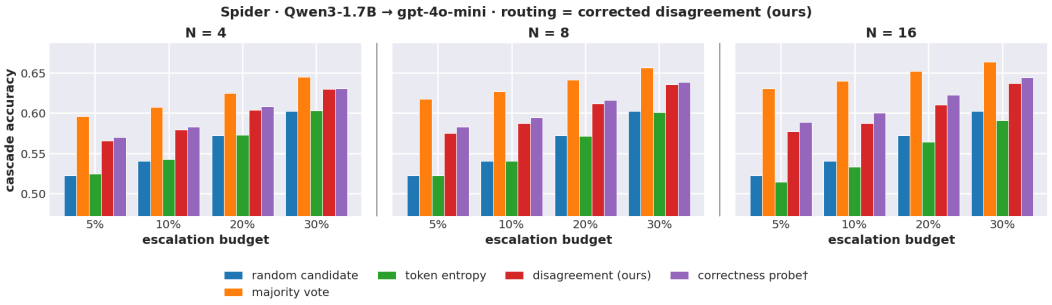


Figure 8: Effect of weak-response selection on SPIDER. Routing uses corrected disagreement throughout. We compare selection policies across routing budgets and different numbers of weak-model samples N.

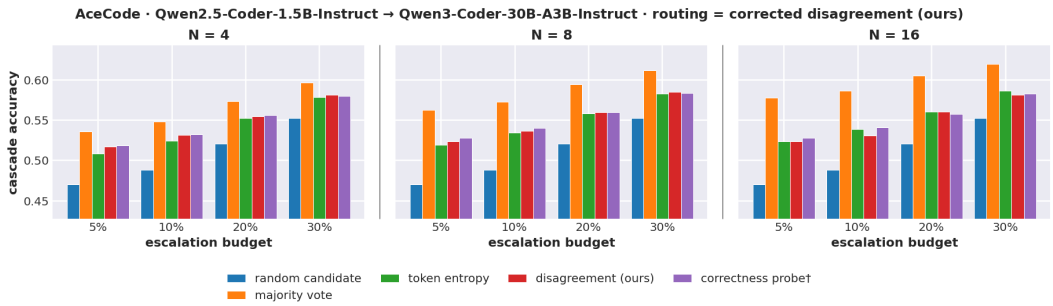


Figure 9: Effect of weak-response selection on AceCode. Routing uses corrected disagreement throughout. We compare selection policies across routing budgets and different numbers of weak-model samples N.