STOP, CONTINUE, OR ESCALATE: ADAPTIVE CONTROL OF MULTI-AGENT DEBATE

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-agent debate can improve the reasoning of compact language models, but applying the same amount of debate to every query is inefficient. Easy queries are often resolved within one or two rounds, some disagreements benefit from further debate, and others persist or converge to shared errors. When local debate remains unreliable, the query can be escalated to a stronger external model, although access to it may be limited by cost or usage constraints. This raises a practical question of whether, after each round of local debate, the system should stop, continue debating, or escalate the query. We formulate this problem as budget-aware, round-by-round control of multi-agent debate. Our framework learns to resolve the current disagreement from the evolving debate trajectory and chooses among the three actions by estimating the value of the current local answer, further local debate, and escalation, while an online mechanism keeps escalation within a prescribed budget. Across four reasoning benchmarks and four agent configurations (three homogeneous and one heterogeneous), our method achieves a better accuracy-computation trade-off than fixed-depth debate as well as routing, cascading, and adaptive-control baselines. At a 10% escalation rate, it improves accuracy by 2.3 percentage points over the strongest baseline while using 18% fewer local model calls than full-length debate. Further analysis shows that our answer resolver improves answer selection under disagreement, whereas the debate controller avoids unnecessary debate rounds through selective continuation and prioritizes escalation when predicted local outcomes are weak.

1 INTRODUCTION

Multi-agent debate enables compact language models to exchange reasoning, critique alternative solutions, and revise their answers across multiple rounds, thereby improving reasoning quality and answer accuracy (Du et al., 2024; Chen et al., 2024a; Liang et al., 2024). However, its benefits vary significantly across queries (Smit et al., 2024; Wang et al., 2024; Wynn et al., 2025; Choi et al., 2025). Easy queries are typically resolved by initial responses or within a few rounds, rendering further debate redundant. Moderately difficult queries may require a few additional rounds to reach consensus. In contrast, for the hardest queries, continued debate often fails to resolve disagreement or even propagate incorrect reasoning, wasting computation. For such intractable cases, escalating to a more capable external model offers a natural solution. However, because access to this external model may incur monetary cost or be limited by an invocation quota, it should be reserved for queries that local debate cannot effectively resolve.

In this case, after each disagreement round, a critical choice arises: whether to STOP because the local debate is sufficient, CONTINUE because further local debate remains worthwhile, or ESCALATE the query to a stronger external model. The quality of this decision directly affects both answer accuracy and resource efficiency. Stopping debate prematurely may sacrifice answer quality, unnecessary continuation wastes local computation, and escalating queries that remain locally resolvable wastes the limited escalation budget. An effective controller must therefore infer from the evolving debate trajectory the reliability of the current local solution, the expected benefit of further local debate, and the potential gain from external-model escalation.

Prior work has addressed different parts of this decision problem. Adaptive debate methods decide whether to initiate or terminate debate using confidence, agreement, or stability signals (Eo et al.,

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.

2025; Hu et al., 2025; Niu & Zhang, 2026), and typically return an agreed or aggregated answer once debate stops (Kaesberg et al., 2025; Ai et al., 2026). They address how much local debate to perform, but not how further debate should be traded against escalation to a stronger model. Model routing and cascading address the complementary question of when to invoke a stronger model (Chen et al., 2024b; Ong et al., 2025; Dekoninck et al., 2025), but usually decide either before local inference or after a fixed local procedure, without using the intermediate evolution of a debate. Recent work has begun to narrow this gap. CascadeDebate introduces multi-agent consensus stages into a fixed sequence of model tiers (Chang et al., 2026), and Budgeted Act-or-Defer makes act, continue, or defer decisions after each debate round, but targets reliability-guaranteed deferral (Wang et al., 2026b). However, existing approaches still lack a mechanism that jointly allocates local answer resolution, further debate, and limited escalation to a stronger model as the debate unfolds round by round.

To fill this gap, we formulate adaptive control of multi-agent debate as a round-by-round decision task and introduce a framework that jointly resolves the current disagreement and determines the next inference action after each non-converged round. Specifically, an answer resolver selects a local answer from any intermediate debate state by incorporating candidate content, supporting reasoning, and the evolution of agent support. A debate controller then uses signals from the evolving debate to decide whether to return this answer, allocate another round of local deliberation, or escalate the query to a stronger external model, while an online mechanism regulates external-model calls toward a prescribed escalation rate. Together, these components adapt debate depth and external-model use to the evolving debate trajectory, rather than relying on a fixed debate depth or a one-shot escalation decision.

The contributions of our work can be summarized as follows:

- We unify debate termination, continuation, and escalation within a single round-by-round decision problem for multi-agent debate. Unlike approaches that fix the number of rounds or make escalation decisions only before or after debate, our formulation conditions each action on the evolving debate trajectory, under an escalation budget shared across queries.
- We propose an adaptive debate control framework that coordinates local debate and escalation. Specifically, an answer resolver draws on the candidate answers, their supporting reasoning, and how support has evolved, yielding a more reliable local answer at any round and thus making early stopping worthwhile. A debate controller then decides whether to stop, continue, or escalate, spending further rounds only where they are likely to help and reserving the escalation budget for queries that local debate is unlikely to resolve.
- Across four benchmarks and four debate configurations, our method outperforms the strongest baseline, among fixed-depth debate, routing, cascading, and adaptive control, by 2.3 points at a 10% escalation budget, while using 18% fewer local model calls than full-length debate. Results show that the answer resolver improves answer selection under unresolved debate, while the controller reduces unnecessary local computation through selective continuation and maintains the prescribed escalation budget.

2 Problem Setting

Debate process. We consider n compact language models that participate in a local debate process, together with a stronger external model M_e whose use is constrained by a shared call budget. We refer to the compact models as *local agents* to distinguish their computational role from that of the external model. Given an input query x from task $k \in \mathcal{K}$ with ground-truth answer y, where $\mathcal{K}$ denotes the set of tasks, all local agents independently generate an initial response at round $t = 1$. At each subsequent round $t \in \{2, \dots, T\}$, every agent observes the query x together with the answers and reasoning produced by the other agents in round $t-1$, and then generates an updated response. Let $z_i^t = (a_i^t, r_i^t)$ denote the response of agent i at round t, where a_i^t represents its predicted answer and r_i^t denotes the associated reasoning. The set of all agent responses at round t is $S_t = \{z_1^t, \dots, z_n^t\}$. We define the debate as converged at round t when all agents agree on the same answer, i.e., $a_1^t = a_2^t = \cdots = a_n^t$. Following common practice in multi-agent debate, we treat convergence as a terminal state and return the consensus answer directly.

Debate control. At the end of each non-converged round $t \in \{1, \dots, T\}$, a controller determines how local debate should proceed based on the accumulated debate trajectory $H_t = (S_1, S_2, \dots, S_t)$.

For $t < T$, it selects a control action $\xi_t \in \{\text{STOP}, \text{CONTINUE}, \text{ESCALATE}\}$. At the maximum round T, CONTINUE is unavailable, so $\xi_T \in \{\text{STOP}, \text{ESCALATE}\}$. Under STOP, the debate terminates and returns a final local answer $\hat{y} = \psi(H_t)$, where ψ is an answer-selection function (e.g., majority vote or a learned aggregator) that maps the observed debate history to a single answer. Under CONTINUE, the debate proceeds to round $t + 1$. Under ESCALATE, the local debate is terminated, and the query is submitted to the external model, whose response $\hat{y} = M_e(x)$ is adopted as the final answer. Let $\pi(H_t)$ denote a generic control rule that maps the accumulated debate history H_t to an action distribution over the available choices.

Budget constraint. For a workload of N queries, let $c(x; \pi) \in \{0, 1\}$ indicate whether input x is escalated to the external model under policy π, and write $c_j(\pi) = c(x_j; \pi)$ for query j. We constrain the average external-model call rate by a prescribed budget $\rho \in [0, 1]$:

$$\frac{1}{N} \sum_{j=1}^{N} c_j(\pi) \leq \rho. \tag{1}$$

Because this constraint is shared across queries, escalating one query would reduce the budget available to others. Thus, the policy should consider not only whether escalation would benefit the current query, but also whether that benefit is large enough to justify consuming part of a budget that other, possibly harder, queries may need.

Let $\hat{y}(x; \pi)$ denote the final answer returned under policy π. In this work, our goal is to improve expected answer accuracy $\mathbb{E}_{(x,y)}[\mathbf{1}\{\hat{y}(x; \pi) = y\}]$, while satisfying the prescribed escalation budget (1). In addition, among policies with comparable accuracy, we prefer those that invoke fewer local model calls, since unnecessary debate rounds waste local computation.

3 METHOD

3.1 OVERVIEW

In this section, we present a budget-aware adaptive debate control framework that coordinates local debate and external-model escalation. The framework comprises three components. An *answer resolver* maps the evolving debate history to a local answer and an estimate of its confidence. Second, a greedy *debate controller* evaluates the relative value of returning the current answer, continuing local debate, or escalating to the external model using estimates of current and eventual local correctness together with task-level external-model accuracy. Third, a *budget-aware mechanism* adjusts the escalation threshold according to realized external-model usage so that the overall escalation rate aligns with the prescribed budget.

3.2 ANSWER RESOLVER

When agents remain in disagreement yet the controller elects to terminate locally rather than escalate to the external model, the system must commit to a single answer from the available candidates. A straightforward choice is majority voting, but it relies only on the answer distribution at the final round, ignoring the reasoning behind each candidate and how support for it has evolved across earlier rounds. To overcome this, we introduce a learned answer resolver ψ_θ that leverages the debate history H_t to assess candidate answers. Specifically, let $\mathcal{A}_t = \{a_1^t, \ldots, a_n^t\}$ denote the answers produced by the n agents at round t, and let $\mathcal{V}_t$ denote the candidate answer set available at that round. For example, in multiple-choice tasks, $\mathcal{V}_t$ contains all answer options, whereas in open-ended numerical tasks it consists of the distinct answers proposed up to round t, i.e., $\mathcal{V}_t = \bigcup_{t'=1}^{t} \mathcal{A}_{t'}$. Then, the resolver models a probability distribution $\psi_\theta(\cdot \mid H_t)$ over $\mathcal{V}_t$, where $\psi_\theta(v \mid H_t)$ denotes the estimated probability that candidate v is correct. The predicted local answer $\hat{y}_t$ and its confidence γ_t are then obtained as

$$\hat{y}_t = \arg\max_v \psi_\theta(v|H_t), \qquad \gamma_t = \max_v \psi_\theta(v|H_t). \tag{2}$$

Thus, for our method, the generic answer-selection function introduced in Section 2 is $\psi(H_t) = \arg\max_v \psi_\theta(v \mid H_t)$. Details of the answer resolver architecture and input features are provided in Appendix B.3.

3.3 DEBATE CONTROLLER

At each non-converged round t, the debate controller chooses among three actions: STOP the local debate, CONTINUE for another round, or ESCALATE the query to the external model. The decision is based on three estimates of answer correctness: two derived from the current debate history and one task-level estimate for the external model. Specifically, the stopping probability $p_{\text{stop}}(H_t) = P(\hat{y}_t = y \mid H_t)$ estimates the probability that the resolver's current answer $\hat{y}_t$ is correct. The continuation probability $p_{\text{cont}}(H_t)$ estimates the terminal correctness of a fixed local rollout that continues without escalation until the agents converge or the final round T is reached. The external-model probability $p_{\text{ext}}(k)$ is its empirical validation accuracy on task k, and is therefore shared across all queries from the same task. From these estimates, the controller computes the expected benefit of continuing the debate over stopping now, i.e., $g_{\text{cont}}(H_t) = p_{\text{cont}}(H_t) - p_{\text{stop}}(H_t)$, and the expected benefit of escalation over the better of the two local options,

$$g_{\text{esc}}(H_t) = p_{\text{ext}}(k) - \max\{p_{\text{stop}}(H_t),\, p_{\text{cont}}(H_t)\}, \tag{3}$$

and selects

$$\xi_t = \begin{cases} \text{ESCALATE}, & g_{\text{esc}}(H_t) \geq \tau, \\ \text{CONTINUE}, & g_{\text{esc}}(H_t) < \tau,\ g_{\text{cont}}(H_t) > 0,\ t < T, \\ \text{STOP}, & \text{otherwise}, \end{cases} \tag{4}$$

where τ is the escalation threshold described in Section 3.4.

This greedy rule first checks whether the external model's expected accuracy exceeds that of the better local option by at least the threshold, and escalates if so. Otherwise, it chooses CONTINUE when continuing the debate is expected to yield a more accurate final answer than stopping now, and STOP in all remaining cases. A CONTINUE action runs only one additional round, after which the controller recomputes its estimates from the updated debate history and decides again. The policy therefore operates in a receding-horizon manner rather than committing to the full remaining debate. Details of the controller architecture and input features are also provided in Appendix B.3.

3.4 BUDGET-AWARE ESCALATION THRESHOLD

Since the escalation budget is shared across all queries, a positive escalation benefit $g_{\text{esc}}(H_t)$ is not by itself sufficient to justify escalation, because an external-model call spent on the current query is no longer available for later ones. We therefore escalate only when this benefit reaches a threshold τ, which can be interpreted as the price of a single external-model call. A higher threshold reserves escalation for queries with larger expected benefit, whereas a lower threshold permits it more freely. Since $p_{\text{ext}}(k)$ is fixed for all queries from the same task, differences in $g_{\text{esc}}(H_t)$ within that task mainly reflect differences in the estimated quality of the available local alternatives. Because the distribution of escalation scores may vary across workloads, in our framework, we initialize τ on validation data and adjust it online according to the observed deviation from the target call rate. Specifically, we initialize τ on validation data so that escalating every first-round disagreement with $g_{\text{esc}} \geq \tau$ would use a fraction ρ of all queries. During deployment, let $\hat{\rho}_\ell = \frac{1}{\ell}\sum_{j=1}^{\ell} c_j(\pi)$ denote the realized external-model call rate after ℓ queries. Every B queries, τ is updated as

$$\tau \leftarrow \begin{cases} \eta\,\tau, & \hat{\rho}_\ell > (1+\delta)\rho, \\ \tau/\eta, & \hat{\rho}_\ell < (1-\delta)\rho, \\ \tau, & \text{otherwise}, \end{cases} \tag{5}$$

where $\eta > 1$ sets the adjustment rate and δ is a tolerance band. The price therefore rises when usage exceeds the target and falls when it falls short. Algorithm 1 in Appendix B.4 summarizes the complete online inference procedure, including local answer selection, the STOP-CONTINUE-ESCALATE decisions, and online adjustment of the escalation threshold.

3.5 TRAINING PROCEDURE

As stated in Section 3.3, the controller relies on the answer resolver's predictions, so we adopt a two-stage decoupled training procedure. We first train and freeze the resolver and then use its outputs

to train the controller estimators. To prevent the controller from being calibrated to overly optimistic in-sample resolver behavior, the two components are trained on disjoint sets of debate trajectories. Let $\mathcal{D}_{\text{res}}$ and $\mathcal{D}_{\text{ctrl}}$ denote the non-overlapping trajectory sets used for resolver and controller training, respectively.

In the first stage, we train the resolver parameters θ over $\mathcal{D}_{\text{res}}$. Since the resolver should be able to produce a local prediction and confidence estimate at any intermediate round of an ongoing debate, we train it on debate histories truncated at varying rounds rather than on complete trajectories alone. For each query $i \in \mathcal{D}_{\text{res}}$ with ground-truth answer y_i^*, we extract the intermediate debate history $H_i^t = (S_i^1, \ldots, S_i^t)$ observed up to each round $t \in \{1, \ldots, T\}$, forming the training set $\mathcal{T}_{\text{res}} = \{(H_i^t, y_i^*) : i \in \mathcal{D}_{\text{res}}, 1 \le t \le T\}$. The resolver is trained by minimizing the negative log-likelihood over the candidate space

$$\mathcal{L}_{\text{res}}(\theta) = - \sum_{(H_i^t, y_i^*) \in \mathcal{T}_{\text{res}}} \log \psi_\theta(y_i^* \mid H_i^t). \tag{6}$$

In the second stage, we freeze the trained resolver $\psi_{\theta^\star}$ and train the controller over $\mathcal{D}_{\text{ctrl}}$. For each trajectory $i \in \mathcal{D}_{\text{ctrl}}$, define its controller-visible rounds as $\mathcal{I}_i = \{t \in \{1, \ldots, T\} : S_i^s \text{ is non-converged for every } s \le t\}$. For a state H_i^t with $t \in \mathcal{I}_i$, let

$$T_i^{\text{roll}}(t) = \min\left(\{s \in \{t, \ldots, T\} : S_i^s \text{ is converged}\} \cup \{T\}\right) \tag{7}$$

be the terminal round reached by the fixed no-escalation local rollout. Let $\tilde{y}_i^{\text{roll}}(t)$ denote the answer that the adopted local protocol actually returns at that terminal round: the consensus answer if convergence occurs, and otherwise the resolver answer at round T. For every $t \in \mathcal{I}_i$, we record two correctness indicators

$$b_{\text{stop}}^{i,t} = \mathbf{1}\{\hat{y}_i^t = y_i^*\}, \qquad b_{\text{cont}}^{i,t} = \mathbf{1}\{\tilde{y}_i^{\text{roll}}(t) = y_i^*\}, \tag{8}$$

where $\hat{y}_i^t = \arg\max_v \psi_{\theta^\star}(v \mid H_i^t)$ is the resolver's answer at round t. The stopping and continuation estimators p_{stop} and p_{cont} are then trained as logistic regressors by minimizing their respective binary cross-entropy losses over these controller-visible states. Finally, because the external model receives the query directly and generates answers, we estimate $p_{\text{ext}}(k)$ as its empirical accuracy on validation queries from task k.

4 Experiments

4.1 Experimental Setup

Datasets and Models. We evaluate our method on four common benchmarks: MMLU (Hendrycks et al., 2021), ECQA (Aggarwal et al., 2021), AQuA (Ling et al., 2017), and GSM8K (Cobbe et al., 2021), covering factual knowledge, commonsense reasoning, algebraic word problems, and open-ended mathematical reasoning. The local agents are instantiated with three compact language models, Gemma 2-9B (Team et al., 2024), Llama 3.1-8B (Grattafiori et al., 2024), and Qwen2.5-7B (Qwen et al., 2025). In our main experiments, we use GPT-oss-20B as the external model, which offers a cost-efficient and reproducible setting for evaluation. To verify that our method does not depend on this particular choice, we additionally evaluate it with a commercial model, DeepSeek-V4.1, accessed through its API. Our method remains the most accurate at every budget in this setting while reducing local compute (Appendix E.1).

Debate configurations and evaluation protocol. By default, we set the number of local agents to $n = 3$ and the number of rounds to $T = 3$. We construct four three-agent debate configurations: three homogeneous configurations consisting entirely of Gemma 2-9B, Llama 3.1-8B, or Qwen2.5-7B, and one heterogeneous configuration containing one agent from each model. We adopt the ReConcile protocol (Chen et al., 2024a) to conduct local multi-agent debate, and use Qwen3-Embedding-4B (Zhang et al., 2025) to encode the query and agents' reasoning. Each debate configuration is evaluated on the same combined workload of test queries from MMLU, ECQA, AQuA, and GSM8K, with the escalation budget ρ shared across the entire workload rather than assigned separately to each task. In the default experiments, the learned components are trained and evaluated on disjoint questions drawn from the same four task distributions. We preserve the question-level partition

across all debate configurations, so a test question from one configuration cannot enter training through another configuration. We evaluate each configuration using eight random seeds (42-49) and report results averaged across configurations and seeds. All model selection, hyperparameter tuning, and escalation-threshold calibration are performed on validation data. Complete data-splitting, architecture, optimization, and budget-calibration details are provided in Appendix B.

Baselines. We compare against five representative strategies. (1) *Fixed-depth debate* runs a fixed number of local debate rounds (one, two, or three), terminating early only if agents converge, and then makes a single answer-or-escalate decision once debate ends. (2) *Pre-debate router* follows a RouteLLM-style design (Ong et al., 2025) and decides whether to use local debate or the external model directly from the input question before executing any debate. (3) *Post-debate cascade* follows a FrugalGPT-style cascade (Chen et al., 2024b). It first completes the local debate, then estimates whether the resulting answer is correct and escalates low-confidence cases. (4) *Heuristic per-round controller* uses the same round-by-round action space as our method, but replaces our debate controller with heuristic decision rules. (5) *Act-or-Defer* adapts the per-round rule of Budgeted Act-or-Defer (Wang et al., 2026b): a k_{NN}-nearest-neighbour lower confidence bound on local correctness is estimated from the debate-prefix state, the query is answered locally once the bound reaches a validation-selected level, and otherwise it is deferred to the external model when the debate ends. For a controlled comparison, all baselines use the same local debate trajectories, confidence-weighted local answer rule, external model, target escalation budget, and online budget-control mechanism. The two learned baselines, the pre-debate router and post-debate cascade, use the same embedding encoder and training-data partitions as our method. More implementation details for these baselines are provided in Appendix D.

4.2 Main Results

Overall accuracy-computation trade-off. Table 1 reports answer accuracy and the average number of local model calls under five escalation budgets. Our method achieves the highest accuracy at every evaluated budget. At $\rho = 0.1$, it reaches 78.7% accuracy with 4.06 local model calls per query. Compared with Fixed-2, Fixed-3, the pre-debate router, the post-debate cascade, the heuristic controller, and Act-or-Defer, our method improves accuracy by 2.3-2.9 percentage points while reducing local calls by 4%-18%. Fixed-1 is the only baseline that uses fewer local model calls, requiring 26% fewer calls than our method, but its accuracy is 5.2 percentage points lower. These results show that our method improves both accuracy and local efficiency over methods with comparable or greater computation, while the only cheaper baseline attains substantially lower accuracy. The same pattern persists across budgets. Relative to Fixed-3, our method reduces local model calls by 9%, 14%, 18%, 26%, and 33% at $\rho = 0$, 0.05, 0.1, 0.2, and 0.3, respectively, while remaining more accurate at every budget. It also consistently uses fewer local calls than the post-debate cascade, the heuristic controller, and Act-or-Defer. Lower-depth strategies can require less local computation at some budgets, and the pre-debate router becomes slightly cheaper at $\rho = 0.3$ by sending more queries directly to the external model, but both alternatives remain less accurate than our method. Results for individual homogeneous and heterogeneous debate configurations are reported in Appendix E.

The lower block of Table 1 compares alternative debate-control policies while using our answer resolver. At $\rho = 0.1$, uniformly debating to the final round achieves 77.9% accuracy, whereas our adaptive controller improves accuracy to 78.7% while reducing local model calls by 18%. This gain does not result from simply terminating debate as early as possible. Skipping debate entirely uses 26% fewer local calls than our method but lowers accuracy to 77.3%. Together, these comparisons show that neither uniformly avoiding debate nor always debating to the final round provides the best trade-off. The Act-or-Defer comparison replaces the complete stopping and deferral policy while using the same answer resolver. Across the five evaluated budgets, our controller is consistently 0.5-1.1 points more accurate than Act-or-Defer (0.84 points on average) and uses 10.1% fewer local calls on average (3.95 versus 4.40), although Act-or-Defer is slightly cheaper at $\rho = 0$ and $\rho = 0.05$.

The online mechanism tracks the escalation budget. The improvements in accuracy and local computation are obtained while keeping external-model usage close to the prescribed escalation rate. As shown in Table 2, the realized rates are .000, .050, .100, .200, and .290 for target rates of 0, .05, .10, .20, and .30, respectively. The online threshold therefore tracks the target closely across the

Table 1: Accuracy (%) under different escalation budgets ρ, averaged over four debate configurations and eight random seeds. Subscripts denote the average number of local model calls per query. The upper block reports the formal baselines. The lower block reports controlled comparisons that use our answer resolver with alternative debate-control policies: no debate, uniform full debate, or the Act-or-Defer stopping and deferral rule. The two Δ rows compare our method with the strongest formal baseline (mean $\pm$ std over eight seeds) and with the best controlled comparison, respectively. The first Δ row is computed from unrounded per-seed accuracies and may therefore differ by 0.1 from the difference of the rounded entries.

Method	$\rho = 0$	$\rho = 0.05$	$\rho = 0.1$	$\rho = 0.2$	$\rho = 0.3$
Fixed-depth debate, 1 round	$69.7_{3.00}$	$71.5_{3.00}$	$73.5_{3.00}$	$76.9_{3.00}$	$80.2_{3.00}$
Fixed-depth debate, 2 rounds	$72.2_{4.23}$	$74.2_{4.23}$	$76.0_{4.23}$	$79.3_{4.23}$	$81.9_{4.23}$
Fixed-depth debate, 3 rounds	$72.7_{4.94}$	$74.7_{4.94}$	$76.5_{4.94}$	$79.8_{4.94}$	$82.1_{4.94}$
Pre-debate router	$72.7_{4.94}$	$74.5_{4.64}$	$76.2_{4.34}$	$78.9_{3.75}$	$81.1_{3.19}$
Post-debate cascade	$72.7_{4.94}$	$74.6_{4.94}$	$76.3_{4.94}$	$79.5_{4.94}$	$81.9_{4.94}$
Heuristic per-round controller	$72.7_{4.80}$	$74.5_{4.66}$	$76.4_{4.54}$	$78.2_{4.41}$	$79.8_{4.29}$
Act-or-Defer	$72.6_{4.58}$	$74.3_{4.57}$	$75.8_{4.61}$	$78.9_{4.78}$	$81.7_{4.94}$
Controlled comparisons with our answer resolver					
No debate + learned escalation	$72.5_{3.00}$	$75.1_{3.00}$	$77.3_{3.00}$	$80.9_{3.00}$	$82.9_{3.00}$
Full debate + learned escalation	$74.3_{4.94}$	$76.2_{4.94}$	$77.9_{4.94}$	$80.6_{4.94}$	$82.5_{4.94}$
Act-or-Defer rule	$74.4_{4.33}$	$76.1_{4.19}$	$77.8_{4.23}$	$80.5_{4.58}$	$82.3_{4.67}$
Ours	$\mathbf{74.9}_{4.48}$	$\mathbf{76.9}_{4.27}$	$\mathbf{78.7}_{4.06}$	$\mathbf{81.6}_{3.65}$	$\mathbf{83.2}_{3.31}$
Δ (Ours $-$ best baseline)	$+2.1_{\pm 0.1}$	$+2.2_{\pm 0.1}$	$+2.3_{\pm 0.2}$	$+1.9_{\pm 0.2}$	$+1.1_{\pm 0.2}$
Δ (Ours $-$ best controlled comparison)	$+0.5$	$+0.7$	$+0.8$	$+0.7$	$+0.3$

Table 2: Target and realized escalation rates of our method, averaged over the four debate configurations and eight random seeds.

Target rate ρ	0	0.05	0.10	0.20	0.30
Realized escalation rate	.000	.050	.100	.200	.290

evaluated budget range, rather than obtaining higher accuracy through additional external-model calls. Appendix G.4 further shows that budget tracking remains stable across different update periods, adjustment factors, tolerance bands, and cumulative caps.

4.3 Inference Cost

Token and processing-time reduction. The main results use local model calls as a hardware-independent measure of agent-side computation. Table 3 further examines whether the reduction in model calls translates into lower token usage and measured processing time after accounting for our answer resolver and controller. At $\rho = 0.1$, our method generates 368 tokens and processes 518 prompt-prefill tokens per query, compared with 452 and 741 for Fixed-3. This corresponds to reductions of 19% in generated tokens and 30% in prefill tokens. The reductions arise because early termination avoids both the outputs generated in later rounds and the increasingly long debate histories included in their prompts. Although the answer resolver in our method incurs extra embedding tokens, these token types have substantially different computational costs. Generated tokens require sequential autoregressive decoding by 7-9B parameter language models, whereas embedding tokens are processed in parallel by a frozen encoder. Under our empirical measurements, the 331 embedding tokens add only approximately 33 ms per query, as shown in Table 9. Our method also uses fewer generation and prefill tokens than the pre-debate router and the heuristic per-round controller. Compared with the router, it reduces generated tokens by 3% and prefill tokens by 15%, while compared with the heuristic controller, the corresponding reductions are 8% and 15%.

On NVIDIA RTX 5090 GPUs, the resulting local processing time is 4.04 s per query, compared with 4.94 s for Fixed-3, an 18% reduction that already includes resolver and controller overhead. Fixed-1 is faster at 2.61 s but is 5.2 percentage points less accurate. Fixed-2 has nearly identical

Table 3: Local processing cost per query at $\rho = 0.1$, measured on NVIDIA RTX 5090 GPUs and averaged over the four debate configurations and eight random seeds.

Method	Acc. (%)	Local calls	Gen. tokens	Prefill tokens	Embed tokens	Local processing time
Fixed-depth debate, 1 round	73.5	3.00	240	193	0	2.61 s
Fixed-depth debate, 2 rounds	76.0	4.23	368	522	0	4.03 s
Fixed-depth debate, 3 rounds	76.5	4.94	452	741	0	4.94 s
Pre-debate router	76.2	4.34	378	609	65	4.13 s
Post-debate cascade	76.3	4.94	452	741	154	4.95 s
Heuristic per-round controller	76.4	4.54	401	610	0	4.37 s
Ours	78.7	4.06	368	518	331	4.04 s

processing time (4.03 s versus 4.04 s) but is 2.7 points less accurate, while the pre-debate router, heuristic controller, and post-debate cascade are both slower and less accurate.

Overhead of the answer resolver and controller. Compared with standard multi-agent debate, our method introduces an embedding model, a resolver head with 0.68M parameters in our implementation, and two logistic regressors with only 16 scalar weights in total. Embedding encoding and the learned heads add approximately 33.4 ms per query (Table 9), accounting for only 0.8% of the measured local processing time, with the learned heads themselves contributing just 0.3 ms. This overhead is substantially smaller than the agent-generation computation avoided through selective continuation and early stopping. Our method additionally requires 7.6 GB of bf16 memory for the embedding encoder, but only a single copy of the encoder is shared across the entire system rather than replicated for each local agent. Appendix F provides the complete token, processing-time, and component-level breakdown.

4.4 Ablation: What Each Component Contributes

Our framework integrates two key design choices: an answer resolver and a debate controller among STOP, CONTINUE, and ESCALATE. Table 4 presents a factorial ablation of these design choices, revealing how their contributions complement one another. Feature-level ablations of the answer resolver and the controller are reported in Appendices G.2 and G.3.

The answer resolver improves local answer quality. When all queries are debated to the final round under the same heuristic escalation rule, replacing confidence-weighted voting with the answer resolver improves accuracy without changing local computation. At $\rho = 0.1$, accuracy increases from 76.5% to 77.3%, while both methods use 4.94 local model calls per query. Without access to the external model, the corresponding improvement is from 72.7% to 74.3%. These results show that the resolver improves the answer returned from unresolved debate states, while leaving the amount of local debate unchanged.

The debate controller reduces local computation through learned per-round decisions. Applying the learned controller only after the final round can improve escalation decisions, but cannot reduce local computation because the full debate has already been completed. In contrast, applying a heuristic controller after each round can terminate debate early. At $\rho = 0.1$, the heuristic per-round controller reduces local calls from 4.94 to 4.54 per query, while accuracy slightly decreases from 76.5% to 76.4%. Combining learned controller with per-round decision-making achieves both benefits. The complete method improves accuracy to 78.7% while further reducing local calls to 4.06 per query, an 18% reduction, because it continues the debate only when further rounds are expected to help.

4.5 Why Adaptive Debate Depth Matters

The preceding results show that our method improves the accuracy-computation trade-off, but they do not reveal why a fixed debate depth is insufficient. We therefore examine how local answer quality changes across debate rounds for queries whose agents disagree after the initial responses. Figure 1

Table 4: Factorial ablation across escalation budgets $\rho \in \{0, 0.05, 0.10, 0.20, 0.30\}$. Accuracy (%) is reported with average local model calls per query as subscripts.

Answer Selection	Decision Policy	$\rho = 0$	$\rho = 0.05$	$\rho = 0.10$	$\rho = 0.20$	$\rho = 0.30$
End of full debate						
Conf.-weighted	Heuristic rules	$72.7_{4.94}$	$74.7_{4.94}$	$76.5_{4.94}$	$79.8_{4.94}$	$82.1_{4.94}$
Conf.-weighted	Learned controller	$72.7_{4.94}$	$75.0_{4.94}$	$76.8_{4.94}$	$80.1_{4.94}$	$82.2_{4.94}$
Resolver	Heuristic rules	$74.3_{4.94}$	$75.9_{4.94}$	$77.3_{4.94}$	$80.0_{4.94}$	$82.1_{4.94}$
Resolver	Learned controller	$74.3_{4.94}$	$76.2_{4.94}$	$77.9_{4.94}$	$80.6_{4.94}$	$82.5_{4.94}$
Adaptive per-round control						
Conf.-weighted	Heuristic rules	$72.7_{4.80}$	$74.5_{4.66}$	$76.4_{4.54}$	$78.2_{4.41}$	$79.8_{4.29}$
Resolver	Heuristic rules	$74.2_{4.80}$	$75.7_{4.66}$	$77.2_{4.54}$	$78.6_{4.41}$	$79.9_{4.29}$
Conf.-weighted	Learned controller	$72.7_{4.93}$	$74.8_{4.81}$	$76.5_{4.62}$	$79.7_{4.24}$	$82.0_{3.77}$
Resolver	**Learned controller (Ours)**	$\mathbf{74.9}_{4.48}$	$\mathbf{76.9}_{4.27}$	$\mathbf{78.7}_{4.06}$	$\mathbf{81.6}_{3.65}$	$\mathbf{83.2}_{3.31}$

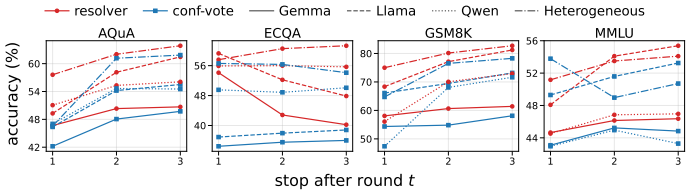


Figure 1: Local answer accuracy on initially disagreeing queries when debate is terminated after round t. Colors distinguish the resolver from confidence-weighted voting, and line styles denote debate configurations.

reports local answer accuracy when initially disagreeing debates are stopped after different rounds. Additional rounds improve accuracy on AQuA, GSM8K, and MMLU, with gains of up to 16.8 points. On ECQA, however, local answer accuracy decreases by 13.9 points with Gemma and 11.4 points with Llama, remains nearly unchanged with Qwen, and improves by only 3.7 points in the heterogeneous configuration. These results show that the benefit of further debate differs substantially across tasks and debate configurations. Our debate controller responds to these differences by allocating additional rounds selectively. As shown in Table 16, among first-round disagreements, the controller continues 79.4% on GSM8K, where additional local debate is generally beneficial, and 53.4% of these continued debates converge before another controller decision is required. In contrast, it stops 59.9% of first-round ECQA disagreements, where further debate is frequently unhelpful. The observed action patterns therefore align with the task-dependent value of further debate and illustrate why a fixed stopping rule is inadequate. Appendix H provides further analysis of local answer quality, wrong convergence, escalation, and controller decisions across debate rounds.

5 Conclusion

We study adaptive control of multi-agent debate when both local computation and access to a stronger external model are allocated selectively. Our framework combines a learned answer resolver with a greedy debate controller that decides whether to stop, continue, or escalate. Across four reasoning benchmarks and four debate configurations, the method shows favorable accuracy-computation trade-offs relative to a range of fixed and adaptive baselines while following the prescribed escalation budget. Our ablations reveal complementary roles for the two components: the resolver improves local answer selection under disagreement, while the controller reduces unnecessary local computation through learned per-round decisions. Further analysis shows that additional debate can substantially improve local answers in some tasks and configurations but degrade them in others. The controller responds to this heterogeneity by allocating further rounds selectively and reserving external assistance for states with weak predicted local outcomes. Together, these results highlight the importance of jointly resolving intermediate disagreements, adapting debate depth, and allocating limited escalation in multi-agent reasoning. Limitations are discussed in Appendix J.

AI USE STATEMENT

In this work, we used generative AI tools to summarize and analyse existing literature, explore research topics and potential gaps, and source and search for information. We also used generative AI tools to provide feedback on the research methodology or experiments. We did not use them for other tasks requiring disclosure. We checked AI-generated summaries, suggested sources, and methodological feedback against the original literature and independently assessed their relevance and accuracy. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Pranjal Aggarwal, Aman Madaan, Yiming Yang, and Mausam. Let's sample step by step: Adaptive-consistency for efficient reasoning and coding with LLMs. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12375–12396, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.761. URL https://aclanthology.org/2023.emnlp-main.761/.

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 Mausam. Automix: Automatically mixing language models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 131000–131034. Curran Associates, Inc., 2024. doi: 10.52202/079017-4164. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/ecda225cb187b40ea8edc1f46b03ffda-Paper-Conference.pdf.

Shourya Aggarwal, Divyanshu Mandowara, Vishwajeet Agrawal, Dinesh Khandelwal, Parag Singla, and Dinesh Garg. Explanations for CommonsenseQA: New Dataset and Models. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 3050–3065, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.238. URL https://aclanthology.org/2021.acl-long.238/.

Rui Ai, Yuqi Pan, David Simchi-Levi, Milind Tambe, and Haifeng Xu. Beyond majority voting: LLM aggregation by leveraging higher-order information. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=ZVyd4r9Xl5.

Santiago Balseiro, Haihao Lu, and Vahab Mirrokni. Dual mirror descent for online allocation problems. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 613–628. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/balseiro20a.html.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. Chateval: Towards better llm-based evaluators through multi-agent debate. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 9079–9093, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/25cc3adf8c85f7c70989cb8a97a691a7-Paper-Conference.pdf.

Raeyoung Chang, Dongwook Kwon, Jisoo Lee, and Nikhil Verma. CascadeDebate: Multi-agent deliberation for cost-aware LLM cascades. In Yunyao Li, Georg Rehm, and Mei Tu (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pp. 1329–1340, San Diego, California, USA, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-394-4. doi: 10.18653/v1/2026.acl-industry.93. URL https://aclanthology.org/2026.acl-industry.93/.

Justin Chen, Swarnadeep Saha, and Mohit Bansal. ReConcile: Round-table conference improves reasoning via consensus among diverse LLMs. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7066–7085, Bangkok, Thailand, August 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.381. URL https://aclanthology.org/2024.acl-long.381/.

Lingjiao Chen, Matei Zaharia, and James Zou. FrugalGPT: How to use large language models while reducing cost and improving performance. *Transactions on Machine Learning Research*, 2024b. ISSN 2835-8856. URL https://openreview.net/forum?id=cSimKw5p6R. Featured Certification.

Hyeong Kyu Choi, Jerry Zhu, and Sharon Li. Debate or vote: Which yields better decisions in multi-agent large language models? In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 101732–101764. Curran Associates, Inc., 2025. doi: 10.52202/085713-3405. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/934252acd87f254d5d4672fbde283bd2-Paper-Conference.pdf.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Jasper Dekoninck, Maximilian Baader, and Martin Vechev. A unified approach to routing and cascading for LLMs. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 12987–13010. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/dekoninck25a.html.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Rühle, Laks Lakshmanan, and Ahmed H Awadallah. Hybrid llm: Cost-efficient and quality-aware query routing. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 41348–41366, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/b47d93c99fa22ac0b377578af0a1f63a-Paper-Conference.pdf.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Sugyeong Eo, Hyeonseok Moon, Evelyn Hayoon Zi, Chanjun Park, and Heuiseok Lim. Debate only when necessary: Adaptive multiagent collaboration for efficient llm reasoning. 2025. URL https://arxiv.org/abs/2504.05047.

Wei Fan, JinYi Yoon, and Bo Ji. iMAD: Intelligent multi-agent debate for efficient and accurate LLM inference. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(35):29403–29411, 2026. doi: 10.1609/aaai.v40i35.40181. URL https://ojs.aaai.org/index.php/AAAI/article/view/40181.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Tianyu Hu, Zhen Tan, Song Wang, Huaizhi Qu, and Tianlong Chen. Multi-agent debate for llm judges with adaptive stability detection. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz,

M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 46504–46540. Curran Associates, Inc., 2025. doi: 10.52202/085713-1548. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/42475c537936b2394b5015e871765056-Paper-Conference.pdf.

Lars Benedikt Kaesberg, Jonas Becker, Jan Philip Wahle, Terry Ruas, and Bela Gipp. Voting or consensus? decision-making in multi-agent debate. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 11640–11671, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.606. URL https://aclanthology.org/2025.findings-acl.606/.

Zachary Kenton, Noah Y. Siegel, János Kramár, Jonah Brown-Cohen, Samuel Albanie, Jannis Bulian, Rishabh Agarwal, David Lindner, Yunhao Tang, Noah Goodman, and Rohin Shah. On scalable oversight with weak llms judging strong llms. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 75229–75276. Curran Associates, Inc., 2024. doi: 10.52202/079017-2395. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/899511e37a8e01e1bd6f6f1d377cc250-Paper-Conference.pdf.

Akbir Khan, John Hughes, Dan Valentine, Laura Ruis, Kshitij Sachan, Ansh Radhakrishnan, Edward Grefenstette, Samuel R. Bowman, Tim Rocktäschel, and Ethan Perez. Debating with more persuasive LLMs leads to more truthful answers. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 23662–23733. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/khan24a.html.

Yiwei Li, Peiwen Yuan, Shaoxiong Feng, Boyuan Pan, Xinglin Wang, Bin Sun, Heda Wang, and Kan Li. Escape sky-high cost: Early-stopping self-consistency for multi-step reasoning. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 14751–14768, 2024a. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/3fe2a777282299ecb4f9e7ebb531f0ab-Paper-Conference.pdf.

Yunxuan Li, Yibing Du, Jiageng Zhang, Le Hou, Peter Grabowski, Yeqing Li, and Eugene Ie. Improving multi-agent debate with sparse communication topology. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 7281–7294, Miami, Florida, USA, November 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.427. URL https://aclanthology.org/2024.findings-emnlp.427/.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. Encouraging divergent thinking in large language models through multi-agent debate. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 17889–17904, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.992. URL https://aclanthology.org/2024.emnlp-main.992/.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In Regina Barzilay and Min-Yen Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 158–167, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1015. URL https://aclanthology.org/P17-1015/.

Andrea Morandi. Sequential consensus for multi-agent llm debates: A wald-sprt compute governor with calibration-based failure detection. 2026. URL https://arxiv.org/abs/2605.19193.

Fuqiang Niu and Bowen Zhang. Armor-mad: Adaptive routing for heterogeneous multi-agent debate in large language model reasoning. 2026. URL https://arxiv.org/abs/2606.13197.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E Gonzalez, M Kadous, and Ion Stoica. Routellm: Learning to route llms from preference data. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 34433–34448, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/5503a7c69d48a2f86fc00b3dc09de686-Paper-Conference.pdf.

Pranoy Panda, Raghav Magazine, Chaitanya Devaguptapu, Sho Takemori, and Vishal Sharma. Adaptive LLM routing under budget constraints. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 23934–23949, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.1301. URL https://aclanthology.org/2025.findings-emnlp.1301/.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Andries Smit, Nathan Grinsztajn, Paul Duckworth, Thomas D. Barrett, and Arnu Pretorius. Should we be going mad? a look at multi-agent debate strategies for llms. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Yiliu Sun, Zicheng Zhao, Sheng Wan, and Chen Gong. CortexDebate: Debating sparsely and equally for multi-agent debate. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 9503–9523, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.495. URL https://aclanthology.org/2025.findings-acl.495/.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Akshay Verma, Swapnil Gupta, Deepak Gupta, Prateek Sircar, and Siddharth Pillai. SELENE: Selective and evidence-weighted LLM debating for efficient and reliable reasoning. In Yevgen Matusevych, Gülşen Eryiğit, and Nikolaos Aletras (eds.), *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 5: Industry Track)*, pp. 95–104, Rabat, Morocco, March 2026. Association for Computational Linguistics. ISBN 979-8-89176-384-5. doi: 10.18653/v1/2026.eacl-industry.7. URL https://aclanthology.org/2026.eacl-industry.7/.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Y Zou. Mixture-of-agents enhances large language model capabilities. 2025:33944–33963, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/5434be94e82c54327bb9dcaf7fca52b6-Paper-Conference.pdf.

Mengdie Flora Wang, Haochen Xie, Guanghui Wang, Aijing Gao, Guang Yang, Ziyuan Li, Qucy Wei Qiu, Fangwei Han, Hengzhi Qiu, Yajing Huang, et al. From debate to decision: Conformal social choice for safe multi-agent deliberation. *arXiv preprint arXiv:2604.07667*, 2026a.

Mengdie Flora Wang, Haochen Xie, Guanghui Wang, Devin Zhang, and Jae Oh Woo. Budgeted act-or-defer multi-agent LLM deliberation with local reliability bounds. 2026b. URL https://openreview.net/forum?id=5g04fnFmLz.

Qineng Wang, Zihao Wang, Ying Su, Hanghang Tong, and Yangqiu Song. Rethinking the bounds of LLM reasoning: Are multi-agent discussions the key? In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for*

Computational Linguistics (Volume 1: Long Papers), pp. 6106–6131, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.331. URL https://aclanthology.org/2024.acl-long.331/.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

Florian Valentin Wunderlich, Lars Benedikt Kaesberg, Jan Philip Wahle, Terry Ruas, and Bela Gipp. Multi-agent reasoning improves compute efficiency: Pareto-optimal test-time scaling. In Santosh T.Y.S.S., Juan Diego Rodriguez, and Ona de Gibert (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pp. 1–14, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-393-7. doi: 10.18653/v1/2026.acl-srw.1. URL https://aclanthology.org/2026.acl-srw.1/.

Andrea Wynn, Harsh Satija, and Gillian Hadfield. Talk isn't always cheap: Understanding failure modes in multi-agent debate, 2025. URL https://arxiv.org/abs/2509.05396.

Zhongling Xu, Shunan Zheng, and Wei Wang. Seqroute: Global budget-aware sequential llm routing via offline reinforcement learning. 2026. URL https://arxiv.org/abs/2605.25424.

Murong Yue, Jie Zhao, Min Zhang, Liang Du, and Ziyu Yao. Large language model cascades with mixture of thought representations for cost-efficient reasoning. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 21691–21728, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/5de11e930c1bbfda5d4fc9d2b0924032-Paper-Conference.pdf.

Yanwei Yue, Guibin Zhang, Boyang Liu, Guancheng Wan, Kun Wang, Dawei Cheng, and Yiyan Qi. MasRouter: Learning to route LLMs for multi-agent systems. 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. 15549–15572, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.757. URL https://aclanthology.org/2025.acl-long.757/.

Yuting Zeng, Weizhe Huang, Lei Jiang, Tongxuan Liu, XiTai Jin, Chen Tianying Tiana, Jing Li, and Xiaohua Xu. S^2-MAD: Breaking the token barrier to enhance multi-agent debate efficiency. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 9393–9408, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.475. URL https://aclanthology.org/2025.naacl-long.475/.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models, 2025. URL https://arxiv.org/abs/2506.05176.

A Related Work

Multi-agent debate and collaborative reasoning. Multi-agent debate has been explored as a test-time reasoning strategy in which multiple agents exchange solutions, critique one another, and revise their answers over several rounds (Du et al., 2024; Chen et al., 2024a; Liang et al., 2024). Related test-time aggregation methods combine multiple reasoning paths or model outputs through self-consistency, voting, or staged synthesis (Wang et al., 2023; 2025). Debate has also been studied for scalable oversight and LLM-based evaluation, where a judge selects between competing arguments or multiple agents deliberate to evaluate model outputs (Khan et al., 2024; Kenton et al., 2024; Chan et al., 2024). The benefits of additional interaction, however, are not uniform across queries or debate configurations. Systematic evaluations show that debate can be sensitive to prompts, agreement criteria, agent composition, and the number of rounds, and does not consistently outperform simpler single-agent or ensemble baselines (Smit et al., 2024; Wang et al., 2024). Longer interaction can also propagate incorrect reasoning or induce agents to abandon initially correct answers (Wynn et al., 2025). These observations motivate allocating debate computation selectively rather than applying a fixed interaction protocol to every query.

Efficient and adaptive debate. A growing body of work reduces the computational cost of debate by changing its communication or execution structure. Sparse communication topologies, selective message exchange, and compact debate protocols reduce redundant interactions while retaining much of the benefit of collaboration (Li et al., 2024b; Zeng et al., 2025; Sun et al., 2025). Complementary studies characterize the accuracy-compute frontier of multi-agent reasoning and other test-time scaling strategies (Wunderlich et al., 2026). Several methods instead adapt whether or when deliberation is performed. DOWN and iMAD predict from an initial response whether a query is likely to benefit from debate (Eo et al., 2025; Fan et al., 2026), while SELENE combines selective debate initiation with evidence-weighted answer aggregation (Verma et al., 2026). ARMOR-MAD combines agreement-based debate routing, early agreement stopping, and outlier-aware aggregation in a heterogeneous model pool (Niu & Zhang, 2026). These approaches primarily avoid initiating unnecessary debate or stop once a prescribed agreement condition is reached.

Stopping, aggregation, and deferral in debate. Adaptive termination and final decision-making have also received increasing attention. In LLM-judge debates, (Hu et al., 2025) model collective judge dynamics with a time-varying Beta-Binomial mixture and stop when successive distributions become sufficiently stable, while Sequential Consensus applies a sequential probability ratio test to judge-derived consensus scores (Morandi, 2026). For single-model sampling, Adaptive-Consistency and early-stopping self-consistency stop drawing samples once the answers agree sufficiently (Aggarwal et al., 2023; Li et al., 2024a). Other work studies how voting, consensus, and aggregation protocols affect the final decision (Kaesberg et al., 2025; Ai et al., 2026). Conformal Social Choice aggregates agents' verbalized probability distributions and calibrates them with split conformal prediction, acting autonomously on singleton prediction sets and escalating larger sets to human review (Wang et al., 2026a). Most directly related to our per-round control setting, Budgeted Actor-Defer also acts, continues, or defers after each debate round, but it provides a conditional bound on the probability of acting wrongly and defers to human review, using a fixed aggregation of agent beliefs (Wang et al., 2026b). We instead target higher accuracy under a prescribed escalation rate: a trajectory-aware resolver selects the local answer, and a greedy controller compares these local alternatives with a stronger but fallible external model.

Cost-aware LLM routing and cascading. LLM routing selects among models with different capabilities and costs. Input-level routers predict query difficulty or model preference before generation and assign each query to a suitable model (Ding et al., 2024; Ong et al., 2025). Cascaded systems instead execute a lower-cost model first and invoke more capable models when the initial output appears unreliable (Chen et al., 2024b; Aggarwal et al., 2024); answer consistency across samples from the weaker model has also been used as a signal for this decision (Yue et al., 2024). Recent work unifies routing and cascading through quality-cost optimization (Dekoninck et al., 2025), learns routing policies from bandit feedback under budget constraints (Panda et al., 2025), and extends routing to model and role selection in multi-agent systems (Yue et al., 2025). The broader idea of adapting a resource price according to observed consumption is related to budget pacing and dual-price methods in online allocation (Balseiro et al., 2020).

Deliberation and budgeted routing. Recent work increasingly connects deliberation with model routing and sequential resource allocation. CascadeDebate inserts selective multi-agent deliberation at the escalation boundaries of a multi-tier model cascade and adapts its confidence thresholds online (Chang et al., 2026). SeqRoute formulates multi-turn routing under a global budget as a finite-horizon Markov decision process whose state includes the remaining budget (Xu et al., 2026). Together, these studies show that deliberation, routing, and budget allocation can be coupled within a broader inference system.

Our work focuses on a different control granularity. Rather than deciding only before debate whether to deliberate, selecting a cascade tier, or routing a conversation turn, our controller acts after every non-converged round of an ongoing local debate. At each such state, it compares three alternatives: returning the answer selected by a learned answer resolver, allocating another round of local debate, or escalating to a stronger external model. These alternatives are reconsidered as the debate evolves, with escalation regulated by an escalation budget shared across the workload.

B Experimental Details

B.1 Debate and External-Model Escalation Protocols

To ensure a fair comparison across methods, we pre-collect debate trajectories and external-model answers once and share them across all baselines, rather than having each method regenerate its own. Debate trajectories are collected once per debate configuration using the ReConcile protocol (Appendix C). Each trajectory records every agent's answer, self-reported confidence, and reasoning at each round. External-model escalation is simulated using a single direct response from GPT-oss-20B to the same query. All methods therefore share the same stored local trajectories and external-model responses. Each escalation counts as one unit of the escalation budget.

B.2 Data Splits and Evaluation Protocol

Queries are partitioned into training, validation, and test sets in a 70/15/15 ratio using a hash of the question text and random seed. Since all four debate configurations are evaluated on the same queries, the partition assignment is shared across configurations: a query assigned to the test set is held out from training under every configuration. This prevents leakage when trajectories from multiple configurations are pooled for training. We repeat the partitioning with eight seeds (42-49). For each seed, the resolver is trained on disagreeing trajectories from the training partitions of all 16 task-configuration combinations, while the controller is fitted on their combined validation partitions. All reported results are evaluated exclusively on the corresponding test partitions. For each debate configuration and seed, the four task-specific test sets are combined into a single evaluation workload containing $N = 5277$ queries, of which 27–54% exhibit first-round disagreement depending on the debate configuration. Experiments are performed using NVIDIA RTX 5090 GPUs.

B.3 Implementations of Our Answer Resolver, Controller, and Escalation Threshold

Embeddings. All texts are embedded once with Qwen3-Embedding-4B served by vLLM. A multiple-choice candidate is embedded as the question text followed by the full option text as it appears in the dataset, for example "A)21" in AQuA, "A. 0" in MMLU, and the option phrase alone, such as "bloody mess", in ECQA. The candidate embedding therefore encodes the option's content in the context of the question, not an option label. A GSM8K candidate is embedded as `Q:<question> A:<answer>`, and the candidate set of a GSM8K query consists of the distinct answers proposed up to the current round, capped at five in order of first appearance. GSM8K queries whose gold answer is never proposed are excluded from resolver training and from the answer-selection analyses; in the end-to-end results they are answered by the confidence-weighted vote. Each agent's reasoning is embedded as generated. The question is not embedded separately and enters the resolver only through the candidate text.

Answer resolver. For each candidate answer v, the resolver uses three inputs. First, the embedding of the candidate text (the question followed by the option or answer; see above) is linearly projected

to 128 dimensions. Second, the mean reasoning embedding across all agent-round pairs (i, t') where $t' \leq t$ and candidate $a_i^{t'} = v$ is projected to 128 dimensions. Third, the vector of cumulative per-round support ratios $(u_1(v), \ldots, u_T(v))$, with unobserved future rounds set to zero, is projected to 32 dimensions. Each projection is passed through a ReLU activation function followed by Layer Normalization. This concatenated feature vector is processed by a multi-layer perceptron with a single 64-unit hidden layer, ReLU activation, and a dropout probability of 0.5 to produce an unnormalized logit for candidate v. A softmax transformation over all valid candidates then yields the predicted probability distribution $\psi_\theta(v \mid H_t)$. For multiple-choice queries all options are scored, including options that no agent proposed; such options receive a zero supporter embedding and zero vote shares. Optimization is performed using Adam with a learning rate of 10^{-4}, weight decay of 5×10^{-4}, batch size of 64, and gradient clipping at a maximum norm of 1.0. Training minimizes a cross-entropy loss with label smoothing $\varepsilon = 0.1$. We apply dynamic learning rate decay by a factor of 0.5 upon validation loss plateaus (patience of 6) and early stopping (patience of 20) based on validation accuracy over non-converged debates. To enable a unified architecture across varying debate lengths, every full-length debate generates multiple training samples, one for each intermediate round $t \in \{1, \ldots, T\}$. Unobserved future rounds in these intermediate states are simply zero-masked, allowing a single set of network weights to evaluate $\psi_\theta(v \mid H_t)$ seamlessly across all debate stages.

Controller. At each non-converged debate round t, the controller estimates $p_{\text{stop}}(H_t)$ and $p_{\text{cont}}(H_t)$ using a feature vector from the debate trajectory H_t, comprising the current round index t, the majority support ratio $\max_v u_t(v)$, a one-hot task domain indicator, and the resolver confidence $\max_v \psi_\theta(v \mid H_t)$. The stopping probability $p_{\text{stop}}(H_t)$ and continuation probability $p_{\text{cont}}(H_t)$ are estimated using independent ℓ_2-regularized logistic regression models (inverse regularization strength $C_{\text{LR}} = 1.0$) trained on all non-converged validation trajectories across debate configurations. The external-model correctness estimate $p_{\text{ext}}(k)$ is given by the empirical accuracy of the external model on validation queries from task k.

Escalation threshold. The escalation threshold τ is initialized to the empirical quantile of the task-prior-based escalation score $g_{\text{esc}}(H_t)$ over first-round validation trajectories that yields an expected escalation rate of ρ. During online inference, τ is updated every $B = 50$ queries using a multiplicative factor $\eta = 1.1$. τ is increased when the realized external-model call rate exceeds $\rho(1 + 0.05)$ and decreased when it falls below $\rho(1 - 0.05)$. We additionally cap cumulative external-model calls at $1.05\,\rho N$; once the cap is reached, escalation is disabled.

B.4 Online Inference Algorithm

Algorithm 1 summarizes how the trained resolver and controller are combined at inference time. Every non-converged state is evaluated for external-model escalation, further local debate, or local termination. The online threshold paces calls across the stream, while a cap of $1.05\,\rho N$ bounds transient overspending. Once the cap is reached, escalation is disabled, and the threshold is updated after every block of B processed queries.

Algorithm 1 Adaptive debate control with budget-aware external-model escalation

Require: Query stream $\{(x_j, k_j)\}_{j=1}^{N}$; local agents $\{M_i\}_{i=1}^{n}$; external model M_e; trained resolver ψ_θ; estimators p_{stop}, p_{cont}, and p_{ext}; maximum round $T = 3$; target rate ρ; calibrated threshold τ; update period B; factor $\eta > 1$; tolerance δ
Ensure: Final predictions $\{\hat{y}_j\}_{j=1}^{N}$

$K_{\text{esc}} \leftarrow 1.05\,\rho N$ ▷ cap on cumulative external-model calls
$C_{\text{esc}} \leftarrow 0$ ▷ number of external-model calls used
for $j = 1, \ldots, N$ **do**
  $S_j^1 \leftarrow \text{INITIALRESPONSES}(x_j, \{M_i\}_{i=1}^{n})$; $H_j^1 \leftarrow (S_j^1)$
  for $t = 1, \ldots, T$ **do**
    if S_j^t is converged **then**
      $\hat{y}_j \leftarrow$ consensus answer in S_j^t; **break**
    end if
    $\hat{y}_j^t \leftarrow \arg\max_v \psi_\theta(v \mid H_j^t)$; $\gamma_j^t \leftarrow \max_v \psi_\theta(v \mid H_j^t)$
    $q_{\text{stop}} \leftarrow p_{\text{stop}}(H_j^t)$; $q_{\text{cont}} \leftarrow p_{\text{cont}}(H_j^t)$
    $g_{\text{cont}} \leftarrow q_{\text{cont}} - q_{\text{stop}}$
    $g_{\text{esc}} \leftarrow p_{\text{ext}}(k_j) - \max\{q_{\text{stop}}, q_{\text{cont}}\}$
    if $C_{\text{esc}} < K_{\text{esc}}$ **and** $g_{\text{esc}} \geq \tau$ **then**
      $\hat{y}_j \leftarrow M_e(x_j)$; $C_{\text{esc}} \leftarrow C_{\text{esc}} + 1$; **break** ▷ ESCALATE
    else if $t < T$ **and** $g_{\text{cont}} > 0$ **then**
      $S_j^{t+1} \leftarrow \text{DEBATEROUND}(x_j, H_j^t, \{M_i\}_{i=1}^{n})$
      $H_j^{t+1} \leftarrow (H_j^t, S_j^{t+1})$ ▷ CONTINUE
    else
      $\hat{y}_j \leftarrow \hat{y}_j^t$; **break** ▷ STOP
    end if
  end for
  if $j \bmod B = 0$ **then**
    $\hat{\rho}_j \leftarrow C_{\text{esc}}/j$
    if $\hat{\rho}_j > (1+\delta)\rho$ **then**
      $\tau \leftarrow \eta\tau$
    else if $\hat{\rho}_j < (1-\delta)\rho$ **then**
      $\tau \leftarrow \tau/\eta$
    end if
  end if
end for

C Debate Prompt Templates

The debate trajectories used in our experiments are collected following the ReConcile framework[1]. The templates below reproduce the prompt structures used for debate collection. Since the answer resolver and the controller operate on the resulting trajectories, they are independent of the specific prompt wording. In the prompt templates, bracketed notes such as `[multiple-choice only]` are explanatory annotations added for readability in this appendix and are not literal prompt text sent to the model.

Initial Response. Each agent receives the input question and, for multiple-choice tasks, the labelled answer options. It is then asked to provide step-by-step reasoning, report a confidence level, and return a JSON object.

```
Q: {question}
[multiple-choice only] The options are: {options}. Please
select an option as your answer.
Please answer the question with step-by-step reasoning.
Also, evaluate your confidence level (between 0.0 and 1.0)
to indicate the possibility of your answer being right.
Output your answer in json format, with the format
as follows: {"reasoning": "", "answer": "",
"confidence_level": ""}
```

[1] https://github.com/dinobby/ReConcile

```
Please strictly output in JSON format.
{dataset-specific answer instruction}
Do not output irrelevant content.
```

The dataset-specific instructions are: *"Only place A,B,C,D representing your choice in the "answer" field"* (MMLU); *"Only place A,B,C,D,E representing your choice in the "answer" field"* (AQuA); *"Only place 1,2,3,4,5 representing your choice in the "answer" field"* (ECQA); *"Only place a single numeric value in the "answer" field"* (GSM8K). The `confidence_level` reported here is also used directly by the confidence-weighted voting baseline, and no separate prompt is needed for that method.

Debate Rounds. In each subsequent round, the agent's prior context is retained and extended with a debate summary compiled from all agents' outputs in the previous round. The appended instructions are:

```
Carefully review the following solutions from other agents
as additional information, and provide your own answer and
step-by-step reasoning to the question.
Clearly state which viewpoint you agree or disagree with and
why.
There are {count} agents that think the answer is {answer}.
One agent solution: I think the answer is {answer} because
{reasoning}. My confidence level is {confidence_level}. [...
repeated for each supporting agent within each answer group
...]
Output your answer in json format, with the format
as follows: {"reasoning": "", "answer": "",
"confidence_level": ""}
Please strictly output in JSON format.
```

The debate summary groups agents by their current answer. For each answer group, it first reports how many agents currently support that answer and then lists one entry per supporting agent in the form *"I think the answer is X because ... My confidence level is Y"*. Thus, in the data collection used for our experiments, each agent in later rounds receives the other agents' reasoning content from the previous round, together with their reported answers and confidence levels.

For GSM8K, the same overall structure is used, except that the prompt begins with `Problem:` rather than `Question:`, and the final answers are disputed numeric values rather than option-labelled choices.

D Baseline Implementations

We implement all baselines under the same escalation-budget protocol as our method. Unless otherwise stated, their local answer is the confidence-weighted voting at the state where local debate terminates. A query is escalated when its baseline-specific score $s(H_t) \geq \tau$ and fewer than $1.05\,\rho N$ external-model calls have been issued. The threshold τ is initialized from validation scores and updated online using the same procedure as for our method. Thus, every method operates under the same budget control, while its routing score determines which queries receive the available calls. The two learned baselines use the same frozen Qwen3-Embedding-4B encoder, question-level data splits, and training-task mixture as our method; their classifier heads are fitted on the training split, while validation AUC is used only for model selection. The fixed-depth and heuristic per-round baselines rank eligible states using a hand-designed score based on the agents' self-reported confidence κ_i^t at round t. For a three-way disagreement, $s(H_t) = 3 - \max_i \kappa_i^t$. For a two-to-one split, $s(H_t) = 2 - \min\{1, \max(0, \bar{\kappa}_{\mathrm{maj}}^t - \kappa_{\mathrm{min}}^t)\}$, where $\bar{\kappa}_{\mathrm{maj}}^t$ is the mean confidence of the two majority agents and κ_{min}^t is the dissenting agent's confidence. For convergence, $s(H_t) = 1 - \bar{\kappa}^t$. Hence, three-way disagreements, two-to-one splits, and convergence occupy disjoint score ranges, with lower-confidence states ranked earlier within each range.

Fixed-Depth Baselines. Fixed-depth strategies run local debate for a fixed number of rounds (1, 2, or 3), terminating earlier if the debate converges. Once debate stops, either upon convergence or after reaching the predefined depth, the resulting state is scored using the heuristic escalation score above. The query is escalated if this score meets or exceeds the threshold τ, in which case the external-model answer is returned. Otherwise, if escalation is rejected or the budget has been exhausted, the confidence-weighted voting of the final local state is returned. Running only one round makes the escalation decision based on the agents' initial independent responses.

Pre-debate router. This baseline follows a RouteLLM-style design (Ong et al., 2025) and routes each query before invoking any local agent. For every debate configuration and random seed, we collect the L2-normalized question embeddings from the training partitions of all four tasks and assign the binary target $z_{\text{route}} = \mathbf{1}\{\text{external model correct}\}\,\mathbf{1}\{\text{complete local debate incorrect}\}$. We fit both an ℓ_2-regularized logistic regressor and an MLP with one 256-unit hidden layer (ℓ_2 penalty 10^{-3} and early stopping), selecting the head with the higher validation AUC. Its predicted probability $P(z_{\text{route}} = 1 \mid x)$ is the escalation score. If this score meets or exceeds the threshold τ, the query is sent directly to the external model and incurs no local model calls; otherwise, local debate proceeds until convergence or the maximum round and returns the confidence-weighted voting of the final round.

Post-debate cascade. This baseline follows a FrugalGPT-style cascade (Chen et al., 2024b). It first runs the complete local debate, terminating upon convergence or at the maximum round, and takes the confidence-weighted vote of the final round as the candidate answer. A verifier then predicts whether this candidate is correct from three separately L2-normalized embeddings: the question, the candidate answer, and the mean terminal-round reasoning embedding of the agents supporting it. We fit one verifier per debate configuration and seed on the training partitions of all four tasks, excluding first-round consensus queries, since the cascade returns these immediately without ever invoking the verifier. As above, we fit logistic regression and a one-hidden-layer 256-unit MLP and retain the head with the higher validation AUC. The escalation score is $s(H_T) = 1 - P(\text{candidate correct} \mid H_T)$. If the score meets or exceeds the escalation threshold τ, the query is sent to the external model; otherwise, the candidate local answer is returned.

Heuristic per-round controller. This baseline uses the same per-round action space as our method but replaces our learned controller with the following rules. If the debate converges, the shared answer is returned immediately. For a two-to-one split with confidence margin $\bar{\kappa}_{\text{maj}} - \kappa_{\min} \geq \mu$, or whenever the maximum debate round is reached, the controller requests external-model escalation using the heuristic score defined above. If escalation is granted, the external-model answer is returned; otherwise, the controller returns the confidence-weighted voting. For a two-to-one split with margin below μ, or for a three-way disagreement before the maximum round, the controller also requests escalation, but continues to the next debate round if the request is rejected. The margin threshold $\mu \in \{0.05, 0.10, 0.20\}$ is selected independently for each seed by maximizing validation accuracy at zero escalation budget.

Act-or-Defer. This baseline adapts Budgeted Act-or-Defer (Wang et al., 2026b). Each debate prefix is mapped to a standardized low-dimensional state: the round index, the maximum probability and top-two margin of the linear opinion pool of the agents' answers and self-reported confidences, the majority share, the task indicator, and, when our resolver supplies the answer, its confidence. For a state, the $k_{\text{NN}} = 50$ nearest validation states give an empirical correctness rate for the local answer, from which we subtract the Hoeffding slack $\sqrt{\log(1/\delta_{\text{H}})/(2k_{\text{NN}})}$ with $\delta_{\text{H}} = 0.05$ to obtain a lower confidence bound (LCB). At each non-converged round, the query is answered locally if the LCB reaches $1 - \alpha$, and debate continues otherwise. When the debate ends at the final round or on consensus, the query is deferred to the external model if its LCB is below $1 - \alpha$ and the paced threshold admits it. The original method defers to human review and bounds the probability of acting wrongly; here deferral goes to the external model under the same budget control as every other method. We select $\alpha \in \{0.05, 0.1, 0.2, \ldots, 0.8\}$ per seed and budget by validation accuracy. The local answer is the confidence-weighted vote, or our resolver in the controlled comparison.

Table 5: Accuracy (%) / external-model calls per query / local calls per query, averaged over four debate configurations and eight random seeds. The final row reports the paired accuracy difference between our method and the strongest baseline at each budget (mean ± std. over seeds). The three italic rows are controlled comparisons that answer with our resolver and are excluded from this difference.

Method	$\rho = 0$	$\rho = 0.02$	$\rho = 0.05$
Fixed-depth debate, 1 round	69.7 / .000 / 3.00	70.4 / .020 / 3.00	71.5 / .050 / 3.00
Fixed-depth debate, 2 rounds	72.2 / .000 / 4.23	73.0 / .020 / 4.23	74.2 / .050 / 4.23
Fixed-depth debate, 3 rounds	72.7 / .000 / 4.94	73.5 / .020 / 4.94	74.7 / .050 / 4.94
Pre-debate router	72.7 / .000 / 4.94	73.5 / .020 / 4.82	74.5 / .050 / 4.64
Post-debate cascade	72.7 / .000 / 4.94	73.5 / .020 / 4.94	74.6 / .050 / 4.94
Heuristic per-round	72.7 / .000 / 4.80	73.5 / .020 / 4.75	74.5 / .050 / 4.66
Act-or-Defer	72.6 / .000 / 4.58	73.3 / .020 / 4.66	74.3 / .050 / 4.57
No debate + learned escalation	72.5 / .000 / 3.00	73.6 / .020 / 3.00	75.1 / .050 / 3.00
Full debate + learned escalation	74.3 / .000 / 4.94	75.2 / .020 / 4.94	76.2 / .049 / 4.94
Act-or-Defer rule + our resolver	74.4 / .000 / 4.33	75.1 / .020 / 4.31	76.1 / .050 / 4.19
Ours	**74.9** / .000 / 4.48	**75.8** / .020 / 4.40	**76.9** / .050 / 4.27
Δ vs. best baseline	$+2.1 \pm 0.1$	$+2.3 \pm 0.2$	$+2.2 \pm 0.1$

Method	$\rho = 0.1$	$\rho = 0.2$	$\rho = 0.3$
Fixed-depth debate, 1 round	73.5 / .100 / 3.00	76.9 / .200 / 3.00	80.2 / .290 / 3.00
Fixed-depth debate, 2 rounds	76.0 / .100 / 4.23	79.3 / .200 / 4.23	81.9 / .290 / 4.23
Fixed-depth debate, 3 rounds	76.5 / .100 / 4.94	79.8 / .200 / 4.94	82.1 / .290 / 4.94
Pre-debate router	76.2 / .100 / 4.34	78.9 / .200 / 3.75	81.1 / .300 / 3.19
Post-debate cascade	76.3 / .100 / 4.94	79.5 / .200 / 4.94	81.9 / .290 / 4.94
Heuristic per-round	76.4 / .100 / 4.54	78.2 / .160 / 4.41	79.8 / .210 / 4.29
Act-or-Defer	75.8 / .101 / 4.61	78.9 / .198 / 4.78	81.7 / .295 / 4.94
No debate + learned escalation	77.3 / .100 / 3.00	80.9 / .201 / 3.00	82.9 / .291 / 3.00
Full debate + learned escalation	77.9 / .100 / 4.94	80.6 / .200 / 4.94	82.5 / .293 / 4.94
Act-or-Defer rule + our resolver	77.8 / .099 / 4.23	80.5 / .197 / 4.58	82.3 / .270 / 4.67
Ours	**78.7** / .100 / 4.06	**81.6** / .200 / 3.65	**83.2** / .290 / 3.31
Δ vs. best baseline	$+2.3 \pm 0.2$	$+1.9 \pm 0.2$	$+1.1 \pm 0.2$

E Complete Main Results

Table 5 extends Table 1 to a wider range of escalation budgets and additionally reports the realized escalation rate.

Performance across budgets. Our method achieves the highest accuracy at every reported budget, improving over the strongest baseline by 2.0 percentage points on average. Averaged across these budgets, it also uses 18.5% fewer local model calls than the three-round fixed-depth debate. The fixed-depth, pre-debate router, post-debate cascade, and Act-or-Defer baselines, as well as our method, all closely track the target escalation rate across the reported budgets. The heuristic per-round controller is the exception, reaching escalation rates of only 0.16 and 0.21 at $\rho = 0.2$ and 0.3, respectively. This is because its heuristic rules allow external-model escalation only from a restricted set of debate states. The Act-or-Defer rule with our resolver also underspends at $\rho = 0.3$ (0.27), because it defers only when a debate ends.

Performance across tasks and debate configurations. Table 7 breaks down the results at $\rho = 0.1$ across all 16 combinations of task and debate configuration. Our method achieves the highest accuracy on 12 combinations and ties for the best on one additional case. On the remaining three, it is narrowly outperformed by a different baseline in each case: the heuristic per-round controller on GSM8K/Qwen, the pre-debate router on MMLU/Qwen, and the post-debate cascade on MMLU/Heterogeneous. The largest gains in our method occur with the Llama debate configuration. Relative to the three-round fixed-depth debate baseline (Fixed-3), our method improves accuracy by 6.2 points on AQuA and 9.8 points on ECQA, while reducing local model calls from 7.13 to 5.88 and from 6.21 to 3.78, respectively. The gains are smaller for the Qwen debate configuration, where the heuristic per-round

Table 6: Accuracy (%) for each debate configuration at $\rho = 0.1$, with the average number of local model calls per query shown as a subscript. The first three rows use homogeneous groups of three agents instantiated from the same model, and the heterogeneous configuration contains one Gemma, one Llama, and one Qwen agent. Results are averaged over eight random seeds. Fixed-1 and Fixed-3 denote one- and three-round fixed-depth debate; Router and Cascade denote the pre-debate router and the post-debate cascade.

Debate pool	Disagree	Fixed-1	Fixed-3	Router	Cascade	**Ours**
Gemma 2-9B $\times 3$	32%	$71.7_{3.00}$	$73.0_{4.36}$	$72.8_{3.81}$	$73.1_{4.36}$	$\mathbf{74.7}_{3.61}$
Llama 3.1-8B $\times 3$	54%	$68.1_{3.00}$	$71.5_{5.73}$	$71.0_{5.03}$	$71.4_{5.73}$	$\mathbf{76.3}_{4.64}$
Qwen2.5-7B $\times 3$	27%	$79.2_{3.00}$	$81.6_{4.19}$	$81.3_{3.69}$	$81.0_{4.19}$	$\mathbf{82.4}_{3.55}$
Heterogeneous	52%	$74.9_{3.00}$	$79.8_{5.49}$	$79.8_{4.81}$	$79.8_{5.49}$	$\mathbf{81.6}_{4.46}$
Mean		$73.5_{3.00}$	$76.5_{4.94}$	$76.2_{4.34}$	$76.3_{4.94}$	$\mathbf{78.7}_{4.06}$

Table 7: Accuracy (%) and average local model calls per query (subscripts) across sixteen task and debate configuration pairs evaluated at escalation budget $\rho = 0.10$ (averaged over eight random seeds). FIXED-1, FIXED-2, and FIXED-3 denote one-, two-, and three-round fixed-depth debate baselines, respectively, and Router, Cascade, and Heuristic denote the pre-debate router, the post-debate cascade, and the heuristic per-round controller. The rightmost column reports the escalation rate of our method.

Task / Config.	Fixed-1	Fixed-2	Fixed-3	Router	Cascade	Heuristic	**Ours**	Escalation
AQuA / Gemma	$66.4_{3.00}$	$70.1_{4.62}$	$70.1_{5.39}$	$71.1_{4.11}$	$69.3_{5.39}$	$71.0_{4.67}$	$\mathbf{72.3}_{4.26}$	0.21
ECQA / Gemma	$51.0_{3.00}$	$50.7_{4.04}$	$50.5_{4.45}$	$50.3_{3.98}$	$53.4_{4.45}$	$50.6_{4.23}$	$\mathbf{54.1}_{3.23}$	0.03
GSM8K / Gemma	$91.2_{3.00}$	$92.3_{3.46}$	$92.3_{3.72}$	$90.2_{3.70}$	$90.8_{3.72}$	$92.4_{3.52}$	$\mathbf{92.4}_{3.54}$	0.04
MMLU / Gemma	$78.1_{3.00}$	$78.2_{3.57}$	$78.3_{3.75}$	$78.4_{3.47}$	$78.4_{3.75}$	$78.2_{3.62}$	$\mathbf{79.5}_{3.32}$	0.09
AQuA / Llama	$61.5_{3.00}$	$68.0_{5.34}$	$69.6_{7.13}$	$69.3_{5.69}$	$66.8_{7.13}$	$69.8_{5.98}$	$\mathbf{75.8}_{5.88}$	0.24
ECQA / Llama	$47.9_{3.00}$	$47.7_{4.99}$	$48.0_{6.21}$	$47.6_{5.44}$	$52.8_{6.21}$	$48.5_{5.72}$	$\mathbf{57.8}_{3.78}$	0.01
GSM8K / Llama	$89.3_{3.00}$	$90.8_{4.00}$	$92.2_{4.63}$	$90.2_{4.61}$	$91.1_{4.63}$	$91.6_{4.42}$	$\mathbf{94.1}_{4.42}$	0.03
MMLU / Llama	$74.2_{3.00}$	$75.2_{4.11}$	$75.8_{4.80}$	$76.1_{4.38}$	$75.5_{4.80}$	$75.7_{4.50}$	$\mathbf{77.4}_{4.21}$	0.08
AQuA / Qwen	$74.4_{3.00}$	$77.3_{4.19}$	$77.0_{4.73}$	$77.3_{3.74}$	$77.0_{4.73}$	$76.5_{4.40}$	$\mathbf{78.1}_{3.90}$	0.16
ECQA / Qwen	$82.7_{3.00}$	$83.0_{3.42}$	$83.2_{3.66}$	$82.7_{3.66}$	$82.7_{3.66}$	$83.2_{3.59}$	$\mathbf{83.5}_{3.07}$	0.01
GSM8K / Qwen	$84.4_{3.00}$	$90.0_{4.10}$	$90.2_{4.59}$	$86.5_{4.39}$	$87.4_{4.59}$	$\mathbf{91.1}_{4.03}$	$90.7_{4.00}$	0.11
MMLU / Qwen	$77.3_{3.00}$	$78.3_{3.49}$	$78.6_{3.76}$	$\mathbf{80.4}_{3.15}$	$78.9_{3.76}$	$78.0_{3.66}$	$79.4_{3.22}$	0.10
AQuA / Hetero	$62.3_{3.00}$	$72.5_{5.27}$	$73.1_{6.51}$	$75.6_{4.97}$	$73.5_{6.51}$	$73.0_{5.89}$	$\mathbf{76.0}_{5.28}$	0.19
ECQA / Hetero	$78.9_{3.00}$	$78.9_{4.06}$	$78.9_{4.90}$	$77.8_{4.90}$	$78.8_{4.90}$	$78.9_{4.49}$	$\mathbf{79.3}_{3.25}$	0.01
GSM8K / Hetero	$83.0_{3.00}$	$89.3_{4.68}$	$90.4_{5.45}$	$87.8_{5.32}$	$88.3_{5.45}$	$89.9_{5.03}$	$\mathbf{93.0}_{4.96}$	0.06
MMLU / Hetero	$78.5_{3.00}$	$78.1_{4.13}$	$79.1_{4.96}$	$79.5_{4.19}$	$\mathbf{80.4}_{4.96}$	$78.5_{4.58}$	$80.3_{4.17}$	0.10

controller performs better on GSM8K and the pre-debate router performs better on MMLU. These results indicate that our adaptive control method is beneficial across the large majority of task and configuration combinations, with only narrow losses on the remainder.

Task-dependent allocation of the shared escalation budget. Table 7 also shows our controller dynamically adjusts escalation rates rather than distributing external-model calls uniformly, as shown in the rightmost column. Across the four debate configurations, our method makes 0.16-0.24 external-model calls per query on AQuA, compared with only 0.01-0.03 on ECQA. This allocation reflects two separate factors: how often agents disagree in the first place, and how the controller acts once a disagreement occurs.

The first factor is disagreement frequency. As shown in Table 16, AQuA produces 3704 first-round controller decisions per seed across the four debate configurations, compared with only 1661 for MMLU, meaning substantially more AQuA queries enter the control process to begin with. Conditional on reaching this first-round decision, however, the controller escalates AQuA and MMLU at nearly identical rates, 23.6% and 23.4% respectively. The larger escalation budget allocation to AQuA is therefore explained primarily by how often AQuA queries disagree in the first round, rather than by a stronger per-disagreement preference for escalating AQuA specifically.

The second factor is the combination of task-level external-model accuracy and state-dependent local outcome estimates. On final-round AQuA disagreements, the external model outperforms the resolver by approximately 35–43 percentage points on average, whereas the corresponding aggregate gap on ECQA ranges from -7 to $+11$ points (Table 15). Consistent with this smaller task-level gap, the controller stops 59.9% of first-round ECQA disagreements and escalates only 3.9%. GSM8K exhibits a different pattern: additional local debate is often beneficial (Figure 1), and the controller therefore continues 79.4% of first-round disagreements while escalating 14.5% (Table 16). Taken together, these patterns show that the controller does not apply a fixed task-level escalation quota. The observed allocation instead emerges from disagreement frequency, the task-level external-model prior, and the state-dependent estimates of current and terminal local correctness. The controller does not predict query-specific external–local complementarity.

Importantly, a low escalation rate does not imply a small performance gain. On ECQA with the Llama debate configuration, our method improves over the three-round fixed-depth debate baseline by 9.8 percentage points while escalating only 1% of queries and reducing average local calls from 6.21 to 3.78. Since external-model escalation directly affects only a very small fraction of queries, most of this improvement comes from better local answer resolution and adaptive stopping. On ECQA, the resolver relies largely on the semantics of the answer options (Appendix G.2).

E.1 Evaluation with a Commercial External Model

Our main experiments use GPT-oss-20B as the external model. To examine whether the effectiveness of our framework depends on this particular external-model backend, we use the commercial API model DeepSeek-V4.1. The evaluation uses exactly the same data and seeds as in the main experiments. Over the full question set, DeepSeek-V4.1 answers 91.2% of AQuA, 83.8% of ECQA, 96.8% of GSM8K, and 93.3% of MMLU questions correctly, compared with 89.7%, 70.6%, 94.2%, and 88.5% for GPT-oss-20B. It is stronger on every task, and by a particularly large margin on ECQA. Table 8 shows that the advantage of adaptive control persists when GPT-oss-20B is replaced by the stronger commercial external model. At $\rho = 0$, no query is escalated, so all methods reproduce their corresponding results in Table 5 exactly, confirming that the two evaluations differ only in the external-model answers. From $\rho = 0.05$ onward, every method benefits from the stronger external model, and our method remains the most accurate at every budget. Across all reported budgets, it also reduces the average number of local model calls from 4.94 for the three-round fixed-depth debate baseline to 4.05, corresponding to an average reduction of approximately 18%. These results show that the performance gains of our framework persist with a different and stronger external model.

Table 8: Accuracy (%) with DeepSeek-V4.1 as the external model, with average local model calls per query as subscripts. The debate trajectories, data splits, resolver, controller, and seeds are identical to those in Table 5, and only the external-model answers differ. Results are averaged over eight seeds. The last row reports the paired per-seed accuracy difference between our method and the strongest baseline at each budget.

Method	$\rho = 0$	0.02	0.05	0.1	0.2	0.3
Fixed-depth debate, 1 round	$69.7_{3.00}$	$70.6_{3.00}$	$71.8_{3.00}$	$74.0_{3.00}$	$78.1_{3.00}$	$81.8_{3.00}$
Fixed-depth debate, 2 rounds	$72.2_{4.23}$	$73.1_{4.23}$	$74.5_{4.23}$	$76.7_{4.23}$	$80.6_{4.23}$	$83.6_{4.23}$
Fixed-depth debate, 3 rounds	$72.7_{4.94}$	$73.6_{4.94}$	$75.0_{4.94}$	$77.1_{4.94}$	$81.1_{4.94}$	$83.9_{4.94}$
Pre-debate router	$72.7_{4.94}$	$73.5_{4.82}$	$74.7_{4.64}$	$76.6_{4.34}$	$79.7_{3.75}$	$82.4_{3.19}$
Post-debate cascade	$72.7_{4.94}$	$73.6_{4.94}$	$74.9_{4.94}$	$77.0_{4.94}$	$80.9_{4.94}$	$83.8_{4.94}$
Heuristic per-round controller	$72.7_{4.80}$	$73.6_{4.75}$	$74.8_{4.66}$	$77.0_{4.54}$	$79.1_{4.41}$	$81.0_{4.29}$
Ours	$\mathbf{74.9}_{4.48}$	$\mathbf{75.8}_{4.40}$	$\mathbf{77.1}_{4.28}$	$\mathbf{79.1}_{4.08}$	$\mathbf{82.5}_{3.69}$	$\mathbf{84.7}_{3.37}$
Δ vs. best baseline	$+2.1 \pm 0.2$	$+2.1 \pm 0.2$	$+2.1 \pm 0.2$	$+2.0 \pm 0.2$	$+1.5 \pm 0.2$	$+0.9 \pm 0.3$

F Additional Experiments on Inference Cost

This section provides a component-level breakdown of local inference cost and evaluates the accuracy-cost trade-off across various escalation budgets.

F.1 Component-Level Cost

Table 9 decomposes local processing time by component on NVIDIA RTX 5090 GPUs, isolating the cost of the added answer resolver and debate controller from the cost of agent-side computation. In processing time, the added components are small relative to autoregressive agent inference: resolver encoding and the learned heads add 33.4 ms per query in total, accounting for about 0.8% of the measured local processing. This comparison does not imply a similarly small memory footprint, nor does it include orchestration or communication overhead. By enabling dynamic early stopping and selective external-model escalation, these components avoid agent-generation work that is more expensive on the evaluated hardware.

Table 9: Local processing time per query at $\rho = 0.1$, decomposed by component on NVIDIA RTX 5090 GPUs and averaged across the four debate configurations. The values quantify aggregate component processing under this setup rather than end-to-end latency in a production system.

Method	Agent decoding	Agent prefill	Resolver embedding	Learned head	Total
Fixed-depth debate, 1 round	2595 ms	19.3 ms	0.0 ms	0.0 ms	2.61 s
Fixed-depth debate, 2 rounds	3975 ms	52.2 ms	0.0 ms	0.0 ms	4.03 s
Fixed-depth debate, 3 rounds	4862 ms	74.1 ms	0.0 ms	0.0 ms	4.94 s
Pre-debate router	4061 ms	60.9 ms	6.5 ms	0.3 ms	4.13 s
Post-debate cascade	4862 ms	74.1 ms	15.4 ms	0.3 ms	4.95 s
Heuristic per-round controller	4311 ms	61.0 ms	0.0 ms	0.0 ms	4.37 s
Ours	3957 ms	51.8 ms	33.1 ms	0.3 ms	4.04 s

F.2 Accuracy-Cost Trade-off

Figure 2 compares accuracy against measured local processing time under three fixed escalation budgets. Across all budgets, our method provides a favorable accuracy-processing trade-off on the evaluated hardware. At $\rho = 0.1$, only the one- and two-round fixed-depth debate baselines require less measured local processing time than our method. The one-round baseline reduces processing time by 1.43 s but sacrifices 5.2 percentage points of accuracy, while the two-round baseline has nearly identical time (4.03 s vs. 4.04 s) but remains 2.7 points less accurate. All other baselines are both slower and less accurate under the same measurement setup. Specifically, the pre-debate router requires 4.13 s for 76.2% accuracy, the heuristic per-round controller 4.37 s for 76.4%, the three-round fixed-depth baseline 4.94 s for 76.5%, and the post-debate cascade 4.95 s for 76.3%. At $\rho = 0.3$ the pre-debate router also becomes cheaper, since escalated queries skip the local debate, but at every budget each cheaper method is at least two points less accurate.

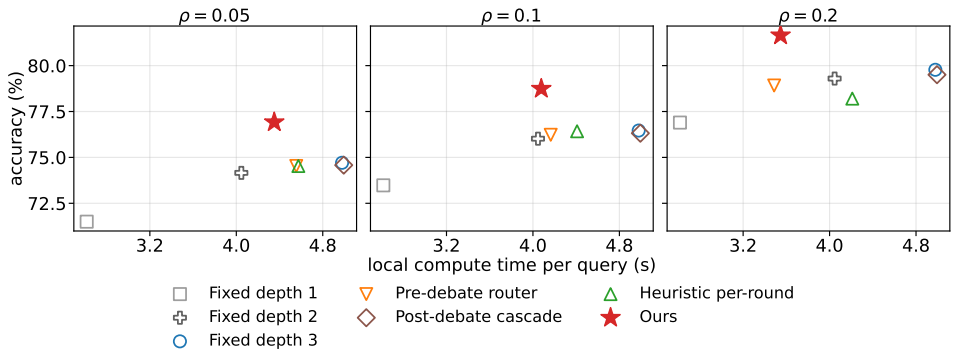


Figure 2: Accuracy against measured local processing time per query at three escalation budgets on NVIDIA RTX 5090 GPUs, averaged over the four debate configurations.

G Additional Ablations and Component Analysis

This section extends the factorial ablation in Section 4.4, which examines the contributions of the resolver and controller components, and evaluates the sensitivity of the budget-aware escalation threshold.

G.1 Uniform versus Selective Debate with the Resolver

Table 10 compares variants that all answer with our resolver and escalate with our debate controller, and differ only in how much local debate they run. The uniform variants run a fixed depth (stopping early on consensus) and take a single stop-or-escalate decision at its end; one round means no debate. Our controller is shown without CONTINUE and with a continuation margin m_{cont}: it continues only if $p_{\text{cont}}(H_t) - p_{\text{stop}}(H_t) > m_{\text{cont}}$, and $m_{\text{cont}} = 0$ is the deployed setting. At every budget, the margin sweep contains a setting that is more accurate than uniform debate to two or three rounds while using fewer local calls; for example, at $\rho = 0.1$, a margin of 0.05 reaches 78.1% with 3.29 calls, above full debate with 4.94 calls (77.9%). No debate uses the fewest calls (3.00); our controller improves on it by spending a few additional calls, e.g., 0.29 more calls for 0.8 points at $\rho = 0.1$ with $m_{\text{cont}} = 0.05$. Margins of 0.01 and 0.02 reduce local calls by 4% and 9% relative to the deployed setting without a significant change in accuracy. We keep $m_{\text{cont}} = 0$, which was fixed in advance rather than tuned on test data; the margin offers a simple knob for trading accuracy against local computation.

Table 10: Accuracy (%) with average local model calls per query as subscripts for variants that share our resolver and learned escalation, averaged over the four debate configurations and eight random seeds. The deployed setting is our controller with margin $m_{\text{cont}} = 0$.

Variant	$\rho = 0$	$\rho = 0.05$	$\rho = 0.1$	$\rho = 0.2$	$\rho = 0.3$
Uniform, 1 round (no debate)	$72.5_{3.00}$	$75.1_{3.00}$	$77.3_{3.00}$	$80.9_{3.00}$	$82.9_{3.00}$
Uniform, 2 rounds	$73.9_{4.23}$	$76.1_{4.23}$	$77.7_{4.23}$	$80.5_{4.23}$	$82.6_{4.23}$
Uniform, 3 rounds (full)	$74.3_{4.94}$	$76.2_{4.94}$	$77.9_{4.94}$	$80.6_{4.94}$	$82.5_{4.94}$
Ours without CONTINUE	$72.5_{3.00}$	$74.9_{3.00}$	$77.2_{3.00}$	$80.7_{3.00}$	$82.9_{3.00}$
Ours, $m_{\text{cont}} = 0.2$	$72.5_{3.01}$	$74.9_{3.00}$	$77.2_{3.00}$	$80.7_{3.00}$	$82.9_{3.00}$
Ours, $m_{\text{cont}} = 0.1$	$73.4_{3.22}$	$75.6_{3.13}$	$77.5_{3.06}$	$80.8_{3.02}$	$82.9_{3.00}$
Ours, $m_{\text{cont}} = 0.05$	$74.2_{3.61}$	$76.3_{3.44}$	$78.1_{3.29}$	$81.1_{3.11}$	$83.0_{3.02}$
Ours, $m_{\text{cont}} = 0.02$	$74.8_{4.10}$	$76.9_{3.91}$	$78.7_{3.71}$	$81.6_{3.36}$	$83.2_{3.12}$
Ours, $m_{\text{cont}} = 0.01$	$74.9_{4.29}$	$76.9_{4.09}$	$78.8_{3.89}$	$81.6_{3.49}$	$83.2_{3.20}$
Ours, $m_{\text{cont}} = 0$ **(deployed)**	$\mathbf{74.9}_{4.48}$	$\mathbf{76.9}_{4.27}$	$\mathbf{78.7}_{4.06}$	$\mathbf{81.6}_{3.65}$	$\mathbf{83.2}_{3.31}$

G.2 Analysis of Resolver Inputs

Table 11 evaluates the contribution of the resolver's three input features by retraining the model with each one removed. All three inputs improve answer-selection accuracy, but their contributions vary across tasks. Removing the candidate text costs 13.2 points on ECQA but at most 0.6 points elsewhere. ECQA options are natural-language phrases, so the embedding of the question followed by an option directly reflects how plausible that option is, whereas AQuA and MMLU options are mostly numbers or expressions and GSM8K candidates are bare numbers, whose embeddings carry little evidence of correctness. A resolver that reads only the candidate text confirms this: it is below confidence-weighted voting on AQuA, GSM8K, and MMLU and averages 36.8% over the four tasks, but reaches 50.6% on ECQA. Restricted to the options proposed by the agents, it reaches 53.9% on ECQA, close to the full resolver under the same restriction (55.5%), but only 35.8% and 39.1% on AQuA and MMLU, against 46.9% and 45.4% for the restricted full resolver. On ECQA the resolver thus draws mainly on option semantics, while on the other tasks it relies on the debate. Conversely, a resolver without candidate text still reaches 50.9%, retaining about 70% of the full resolver's gain over voting, so most of the gain does not rely on answer semantics. Removing vote shares costs 22.3 points on GSM8K and 4.7 on AQuA but essentially nothing on ECQA, since a numeric answer conveys little beyond its digits, leaving the resolver to rely on who supported it and when. Supporter reasoning provides consistent gains on AQuA, GSM8K, and MMLU, while contributing little on ECQA. These results show that the resolver is not purely a structural aggregator:

it combines task-dependent semantic evidence from candidate content and supporting reasoning with structural signals from the evolution of agent support. The strong dependence on candidate text for ECQA also anticipates the larger loss observed when the resolver is transferred to an unseen task (Appendix I).

Because the resolver has access to the full set of multiple-choice options, it is not restricted to answers proposed by the agents. In practice, however, it selects an unproposed option only rarely, in 0.5%, 6.3%, and 2.1% of final-round disagreements on AQuA, ECQA, and MMLU, respectively. To test whether this additional flexibility is responsible for its gains, we restrict the resolver to agent-proposed options. The resulting accuracy changes are small at 0.0, -0.9, and $+0.5$ percentage points, suggesting that the resolver's advantage over voting is generally driven by how it evaluates the available proposals rather than by its ability to introduce a new option.

Table 11: Ablation analysis of answer resolver input features on unresolved post-debate queries. We report answer-selection accuracy (%) on test queries exhibiting agent disagreement after the final debate round, averaged across four agent setups and eight random seeds. GSM8K queries whose gold answer is never proposed by any agent are excluded.

Resolver inputs	AQuA	ECQA	GSM8K	MMLU	Mean
Confidence-weighted voting (no resolver)	40.4	39.1	55.7	37.6	43.2
All three (ours)	**46.9**	**56.5**	**69.3**	**44.9**	**54.4**
without candidate text	46.7	43.3	68.7	44.8	50.9
without supporter reasoning	42.9	56.1	65.9	42.8	51.9
without per-round vote shares	42.2	56.1	47.0	42.3	46.9
candidate text only	28.3	50.6	37.6	30.8	36.8

G.3 Analysis of Controller Inputs and Action Choices

Table 12 evaluates the state inputs and the action space of the controller at $\rho = 0.1$. Removing the task indicator, the resolver confidence, or the majority share lowers accuracy by 0.41, 0.39, and 0.27 points, respectively ($p \leq 0.013$, paired over seeds). The resolver confidence also has the largest effect on local computation, since removing it raises local calls from 4.06 to 4.25, which suggests that it helps the controller recognize when stopping is safe. The round index can be removed without loss of accuracy. The effects of single inputs are modest because the controller does not produce answers itself and can change correctness only on queries where its alternatives lead to different outcomes.

The action space has a larger effect. Without CONTINUE, every non-converged query must stop or escalate after the initial responses, and accuracy drops by 1.5 points while local calls fall to 3.00. The full controller thus converts about one additional local call per query into 1.5 points of accuracy. Without early stopping, accuracy drops by 0.5 points and local calls rise from 4.06 to 4.52. Selective continuation and early stopping therefore both improve accuracy, and early stopping additionally reduces local computation.

Table 12: Ablation analysis of the debate controller at $\rho = 0.1$. We report accuracy (%), average local model calls per query, and escalation rate, averaged over the four debate configurations and eight random seeds.

Variant	Accuracy (%)	Local calls	Escalation rate
Deployed controller	**78.74**	4.06	0.100
Remove each state input			
without round index	78.73	3.91	0.100
without majority share	78.47	4.11	0.100
without task indicator	78.33	4.02	0.100
without resolver confidence	78.35	4.25	0.101
Change the action space			
without CONTINUE action	77.20	3.00	0.101
without early stop	78.20	4.52	0.100

G.4 Sensitivity of the Budget-Aware Escalation Threshold

The escalation-threshold update introduced in Section 3.4 has three pacing constants: the update period B, the multiplicative factor η, and the tolerance band δ; the cumulative cap is a fourth safeguard. Table 13 varies each around its default ($B = 50$, $\eta = 1.1$, band $\pm 5\%$, cap $1.05\,\rho N$). Across the periodically updated variants, accuracy differs from the default by at most 0.4 points at any target rate, and the realized rate stays within 2% of the target in every setting. Removing the cap or raising it to $1.10\,\rho N$ leaves the results almost unchanged, because the threshold update already keeps usage near the target. A strict cap of ρN keeps the realized rate below the target and lowers accuracy by at most 0.12 points. In contrast, fixing the threshold overspends the budget, by 2%-5% with the cap and by 14%-41% without it, and the slightly higher accuracy of these two variants comes from the additional external-model calls.

Table 13: Sensitivity of the online escalation-threshold update. Each cell reports accuracy (%) and escalation rate, with the relative deviation from the target rate ρ in parentheses. The default setting is $B = 50$, $\eta = 1.1$, band $\pm 5\%$, and cap $1.05\,\rho N$. Results are averaged over four debate configurations and eight random seeds. The last two rows disable the threshold update.

Setting	$\rho = 0.05$	$\rho = 0.1$	$\rho = 0.2$
Period $B = 20$	76.8 — .051 (+1.3%)	78.5 — .100 (0.0%)	81.4 — .200 (−0.2%)
Period $B = 50$ (default)	76.9 — .050 (+1.0%)	78.7 — .100 (+0.2%)	81.6 — .200 (0.0%)
Period $B = 100$	77.0 — .050 (−0.9%)	78.8 — .100 (−0.2%)	81.7 — .201 (+0.5%)
Period $B = 200$	76.9 — .049 (−2.0%)	78.8 — .098 (−1.8%)	81.7 — .200 (+0.1%)
Factor $\eta = 1.05$	77.0 — .051 (+1.1%)	78.8 — .101 (+0.8%)	81.7 — .201 (+0.5%)
Factor $\eta = 1.2$	76.8 — .050 (+0.4%)	78.6 — .100 (−0.2%)	81.5 — .201 (+0.4%)
Factor $\eta = 1.5$	76.7 — .050 (−0.1%)	78.3 — .099 (−0.7%)	81.2 — .199 (−0.4%)
Band $\pm 2\%$	76.9 — .050 (+0.7%)	78.7 — .100 (+0.3%)	81.5 — .200 (−0.2%)
Band $\pm 10\%$	76.9 — .050 (−1.0%)	78.7 — .099 (−1.2%)	81.7 — .199 (−0.7%)
No cap	76.9 — .051 (+1.4%)	78.7 — .100 (+0.2%)	81.6 — .200 (0.0%)
Cap 1.00 (strict)	76.9 — .049 (−1.3%)	78.7 — .099 (−1.3%)	81.5 — .197 (−1.5%)
Cap 1.10	76.9 — .051 (+1.4%)	78.7 — .100 (+0.2%)	81.6 — .200 (0.0%)
No update, cap only	77.0 — .051 (+2.2%)	78.8 — .105 (+4.8%)	81.8 — .207 (+3.5%)
No update, no cap	77.8 — .070 (+40.7%)	80.0 — .136 (+36.5%)	82.4 — .228 (+13.9%)

H Additional Analysis of Adaptive Debate Control

This section provides additional evidence for how the controller chooses among STOP, CONTINUE, and ESCALATE. We first examine when another local debate round improves or degrades the current answer, then compare local resolution with external-model escalation on persistent disagreements, and finally analyze how the controller allocates its actions across rounds.

H.1 When Is Further Local Debate Useful?

Table 14 reports the detailed results underlying Figure 1. Here, $t = 1$ denotes the agents' initial independent responses, while $t = 2$ and $t = 3$ denote the two subsequent revision rounds. In addition to answer accuracy, the table reports how often initially disagreeing agents reach convergence and how often the final converged answer is incorrect. The results show that further debate is beneficial on most tasks, but its effect is not uniform across tasks and debate configurations. For every AQuA, GSM8K, and MMLU configuration, the accuracy at $t = 3$ exceeds its value at $t = 1$, with gains of up to 12.2 points for AQuA with Llama, 12.9 points for GSM8K with Llama, and 16.8 points for GSM8K with Qwen. ECQA exhibits a markedly different pattern: accuracy decreases by 13.9 points with Gemma and 11.4 points with Llama, remains nearly unchanged with Qwen, and improves by 3.7 points under the heterogeneous configuration. These results show that additional debate can substantially improve local answer quality, but can also degrade it in some settings. A fixed debate depth therefore cannot capture this variation, since applying the same number of rounds to every query may preserve useful reasoning in some cases while allowing harmful revisions in others.

Figure 1 also shows that, averaged over the homogeneous configurations, the resolver's advantage over confidence-weighted voting shrinks from 6.4 points after the initial responses to 4.0 points after the final round as agents approach consensus, most markedly on ECQA. Early disagreements expose multiple candidate answers and their supporting reasoning, providing the resolver with useful evidence that may disappear once the agents converge. This explains why the resolver and round-by-round control are complementary: the resolver provides a reliable intermediate answer, while the controller determines when that answer should be used.

The wrong-convergence statistics provide further evidence for this heterogeneity. Averaged across the four tasks, the final wrong-convergence rate is approximately 38% for Gemma, 27% for Qwen, and 17% for Llama, compared with 12% for the heterogeneous configuration, which is at most 17% on every task. This suggests that homogeneous agents are more susceptible to converging on shared errors, whereas heterogeneous agents are less frequently driven toward the same incorrect answer. Together, these observations indicate that disagreement alone does not determine whether debate should continue, since the terminal outcome of further local debate depends on both the task and the evolving debate state.

Table 14: Accuracy (%) on test queries with disagreement after round 1, averaged over eight random seeds. For each round t, we report the accuracy obtained by returning either the resolver output or the confidence-weighted vote, together with the cumulative fraction of queries that have converged by that round. The final column reports the fraction that have converged to an incorrect answer by $t = 3$. All comparisons discussed in the text are paired across seeds on the same queries.

		Resolver			Conf.-weighted			Converged		Wrong convergence
Configuration	N	t=1	t=2	t=3	t=1	t=2	t=3	t=2	t=3	t=3
AQuA / Gemma	809	46.7	50.3	50.7	42.2	48.1	49.7	52%	73%	35%
AQuA / Llama	1170	49.3	58.1	61.5	46.4	54.2	55.5	23%	38%	8%
AQuA / Qwen	593	51.1	55.3	56.1	47.0	54.7	54.5	54%	75%	29%
AQuA / Hetero	1132	57.6	62.0	63.9	46.6	61.2	61.8	45%	62%	17%
ECQA / Gemma	396	54.1	42.8	40.2	34.4	35.5	36.0	61%	78%	50%
ECQA / Llama	758	59.3	52.2	47.9	36.9	38.0	38.8	39%	59%	33%
ECQA / Qwen	160	55.9	56.0	55.7	49.5	48.9	50.1	43%	68%	31%
ECQA / Hetero	404	57.6	60.5	61.3	56.6	56.3	54.1	21%	34%	10%
GSM8K / Gemma	174	58.1	60.6	61.4	54.4	54.8	58.1	44%	61%	21%
GSM8K / Llama	376	68.3	77.1	81.2	66.1	69.5	73.2	37%	54%	4%
GSM8K / Qwen	415	56.1	70.1	72.9	47.4	68.1	71.6	56%	74%	14%
GSM8K / Hetero	631	75.0	80.2	82.7	64.7	76.5	78.3	54%	72%	7%
MMLU / Gemma	285	44.7	46.1	46.4	43.1	45.2	44.8	68%	84%	44%
MMLU / Llama	559	48.1	54.1	55.4	49.3	51.6	53.2	38%	57%	22%
MMLU / Qwen	249	44.5	46.8	47.0	43.0	45.0	43.3	46%	65%	34%
MMLU / Hetero	568	51.2	53.5	54.1	53.8	49.0	50.7	26%	39%	15%

H.2 Local Resolution versus External-Model Escalation

Table 15 isolates answer selection from sequential debate control by evaluating majority voting, confidence-weighted voting, the learned resolver, and the external model on the same test queries that remain non-converged after the final local round. This comparison characterizes the alternatives available when persistent disagreement must be resolved either locally or through external-model escalation. The results show that our learned resolver substantially improves local answer selection. Averaged across the sixteen combinations of task and debate configuration, its accuracy reaches 54.4%, compared with 43.2% for confidence-weighted voting, and it outperforms the voting rule in every combination by at least 4.4 percentage points. The higher-order aggregators of Ai et al. (2026) do not improve on voting: over the twelve multiple-choice combinations they reach 38.5% (OW) and 38.7% (ISP), against 39.0% for confidence-weighted voting and 49.4% for the resolver. Debate makes the agents' final answers strongly correlated, which limits what agent-level weighting can extract.

In addition, the aggregate accuracy gap between external-model escalation and local resolution varies across tasks. Relative to the resolver, direct external-model escalation improves accuracy by 34.9–43.3 points on AQuA and 31.0–42.6 points on MMLU. The gap is smaller on GSM8K, at 12.5–27.4

points, and ranges from -7.1 to $+10.5$ points on ECQA. In the Qwen and heterogeneous ECQA configurations, the resolver is more accurate than the external model. Although these results are restricted to final-round disagreements and do not establish query-specific external-model advantage, they motivate using different task-level external-model priors rather than a single prior shared across all tasks. This is consistent with our controller allocating relatively few external-model calls to ECQA.

Table 15: Accuracy (%) on final-round disagreements evaluated on the same queries for each seed. Values are means ± standard deviations over eight seeds. GSM8K queries whose gold answer is never proposed by any agent are excluded. OW and ISP are the aggregators of Ai et al. (2026), fitted on the training split. They require a fixed label space and do not apply to GSM8K.

Configuration	Majority	Conf. vote	OW	ISP	Resolver	External
AQuA / Gemma	40.1±4.1	40.2±3.9	37.6	39.6	44.6±2.8	87.9±1.0
AQuA / Llama	40.3±1.6	41.7±1.2	39.6	38.5	51.6±1.4	86.5±1.4
AQuA / Qwen	35.3±2.0	36.1±2.2	34.8	35.1	41.6±3.4	82.4±1.8
AQuA / Hetero	44.5±2.0	43.6±2.0	45.4	44.4	49.6±1.8	86.1±1.5
ECQA / Gemma	36.4±3.9	36.9±4.2	36.4	36.6	55.6±5.5	60.2±3.8
ECQA / Llama	31.3±2.5	32.2±2.6	32.8	32.0	53.9±2.5	64.4±1.6
ECQA / Qwen	40.7±5.4	41.1±5.6	40.8	40.8	59.7±6.9	52.6±5.2
ECQA / Hetero	46.4±3.3	46.2±3.2	45.7	46.1	56.6±2.2	55.4±2.4
GSM8K / Gemma	49.4±4.0	54.6±2.7	–	–	63.8±7.1	86.9±2.8
GSM8K / Llama	56.4±3.1	54.3±2.7	–	–	75.6±3.3	88.1±2.5
GSM8K / Qwen	55.2±6.3	57.5±5.1	–	–	62.9±5.3	90.3±4.8
GSM8K / Hetero	57.9±4.6	56.4±3.9	–	–	74.7±4.6	89.9±1.3
MMLU / Gemma	28.6±3.9	29.0±4.5	27.8	27.6	38.0±6.7	80.6±4.2
MMLU / Llama	41.2±2.7	42.7±3.1	41.8	43.3	47.2±2.3	84.0±2.1
MMLU / Qwen	34.9±3.4	35.0±3.7	35.5	36.0	45.4±3.4	81.2±4.6
MMLU / Hetero	43.6±0.7	43.7±1.0	43.5	44.2	49.1±1.9	80.1±1.4
Mean (all)	42.6	43.2	–	–	54.4	78.5
Mean (multiple choice)	38.6	39.0	38.5	38.7	49.4	75.1

H.3 How Does the Controller Act across Rounds?

Table 16 reports the controller's decisions over the full debate trajectory. Percentages are conditional on the queries carried into each round. *Converged* denotes queries that reach agreement after the preceding CONTINUE decision and, under the adopted agreement-based termination protocol, exit before another controller decision. At the first decision, the controller frequently stops on ECQA, continues on GSM8K, and escalates more often on AQuA and MMLU.

The later rounds show how CONTINUE converts additional local computation into resolved debates. Among queries carried into round 2, 26.3–53.4% have already converged before another controller decision is required. A further 18.5–34.7% of the queries reaching round 3 converge after the additional debate step. This effect is particularly obvious on GSM8K, where 79.4% of round-1 disagreements are continued and 53.4% of the resulting round-2 states have already converged. These trajectories illustrate why continuation can be valuable even when the initial responses disagree.

Persistent disagreement, however, does not automatically trigger use of the external model. At the final round, where CONTINUE is no longer available, the controller returns the resolver's local answer for 59.0–79.1% of the remaining states and escalates only 2.4–14.7%. External-model escalation is most frequent at the initial decision for every task and remains selective in later rounds. The controller therefore uses additional debate to resolve many queries locally and reserves external-model access for states whose predicted local outcomes are sufficiently weak relative to the task-level external-model prior.

These trajectories are consistent with the preceding analyses of local debate and answer selection. GSM8K is dominated by CONTINUE, consistent with the substantial gains from additional local reasoning. ECQA is dominated by early STOP, where further debate provides limited or even negative gains and the external model offers relatively little advantage over the resolver. AQuA and MMLU receive more frequent escalation, consistent with the larger observed gap between external-model

and local resolver accuracy. Overall, the controller adapts not only whether to use the external model, but also how much local reasoning to allocate before making that decision.

Table 16: Controller outcomes on initially disagreeing queries at $\rho = 0.1$, reported by task and debate round and aggregated across the four debate configurations. The third column gives the number of queries carried into each round per seed. Converged denotes queries that reach agreement following the preceding CONTINUE decision and terminate according to the adopted local debate protocol before another controller decision.

Task	Round	Decisions	STOP	CONTINUE	ESCALATE	Converged
AQuA	1	3704	8.6%	67.8%	23.6%	–
	2	2513	8.6%	42.4%	8.0%	41.0%
	3	1065	62.9%	–	10.8%	26.3%
ECQA	1	1718	59.9%	36.2%	3.9%	–
	2	622	33.0%	38.7%	2.0%	26.3%
	3	241	79.1%	–	2.4%	18.5%
GSM8K	1	1595	6.1%	79.4%	14.5%	–
	2	1267	5.5%	39.0%	2.1%	53.4%
	3	494	62.1%	–	3.2%	34.7%
MMLU	1	1661	9.7%	66.9%	23.4%	–
	2	1112	19.0%	32.7%	10.3%	38.0%
	3	363	59.0%	–	14.7%	26.3%

I TRANSFER EXPERIMENTS

We here evaluate whether the learned resolver and controller transfer along two distinct axes: to a previously unseen debate configuration and to a previously unseen task.

I.1 TRANSFER PROTOCOL

Unseen debate configuration. To evaluate transfer to an unseen debate configuration, we hold out one of the four configurations (Gemma, Llama, Qwen, or heterogeneous) and train both the answer resolver and controller on the remaining three configurations. Because all configurations are evaluated on the same underlying questions, we preserve the same question-level data split across them. Model fitting uses only training questions from the remaining three configurations, while the held-out configuration is evaluated exclusively on the disjoint test partition. This ensures that no evaluation question is observed during training through another debate configuration.

Unseen task. To evaluate transfer to an unseen task, we hold out one of the four tasks (MMLU, ECQA, AQuA, or GSM8K) and train both the resolver and controller on the remaining three tasks. All debate configurations of the held-out task are excluded from training and validation. The resolver is trained using only questions from the remaining three tasks, and the controller is fitted on their validation trajectories. Because the held-out task is never observed during fitting, its task-indicator feature is inactive, and p_{ext} is set to the mean external-model accuracy over the remaining three tasks. No questions from the held-out task are used for zero-shot training, validation, or model selection.

Evaluation variants. All transfer experiments use $\rho = 0.1$ and report averages over eight random seeds. In the *zero-shot* setting, both the answer resolver and controller are trained only on the remaining three tasks or debate configurations, with no labeled data from the held-out setting. To separate the contributions of the two components, we also evaluate a zero-shot controller together with a resolver trained on the held-out setting. In addition, we consider few-shot controller calibration using n_{cal} labeled validation queries from the held-out task while keeping either the zero-shot or in-domain resolver fixed. These calibration queries are distributed evenly across the four debate configurations and are also used to estimate p_{ext}. All resolver variants use the same training procedure, and model selection for zero-shot transfer relies only on validation data from the remaining training settings.

I.2 Transfer to an Unseen Debate Configuration

To test whether our method generalizes to an unseen debate configuration, we hold out one of the four configurations at a time, as listed in the first column of Table 17, and evaluate on its test queries. We here compare four settings. *Fixed-3* is the three-round fixed-depth debate baseline, which involves no learned component and serves as a reference. In *Both transferred*, the answer resolver and the controller are both trained only on the other three configurations, so neither has seen any debate from the held-out configuration. In *Controller transferred*, the controller is trained in the same way, while the resolver is trained on all four configurations, including the held-out one. *No transfer* trains both components on all four configurations, which is the setting used in our main experiments and serves as an upper reference. Comparing these settings shows how much accuracy is lost when each component must work on a configuration it has never seen.

Table 17: Transfer to an unseen debate configuration at $\rho = 0.1$. Each row holds out one configuration during training and evaluates on it. Accuracy (%) is reported with average local model calls per query as subscripts, averaged over the four tasks and eight seeds. The column settings are described in the text. Fixed-3 calibrates its escalation threshold on the source configurations, so its values differ slightly from Table 6.

Held-out configuration	Fixed-3	Both transferred	Controller transferred	No transfer
Gemma	$72.9_{4.36}$	$74.4_{3.70}$	$74.6_{3.70}$	$74.7_{3.61}$
Llama	$71.6_{5.73}$	$74.9_{4.88}$	$75.9_{4.73}$	$76.3_{4.64}$
Qwen	$81.6_{4.19}$	$82.1_{3.56}$	$82.3_{3.54}$	$82.4_{3.55}$
Hetero	$79.8_{5.49}$	$81.4_{4.45}$	$81.6_{4.35}$	$81.6_{4.46}$
Mean	$76.5_{4.94}$	$78.2_{4.15}$	$78.6_{4.08}$	$78.7_{4.06}$

The results show that our method transfers to an unseen debate configuration with only a small loss in accuracy. With both components transferred, the system reaches 78.2% accuracy, only 0.5 points below the no-transfer setting, while still outperforming Fixed-3 by 1.7 points and using approximately 16% fewer local model calls. The largest gap to the no-transfer setting occurs on Llama, at 1.4 points. Comparing the middle two columns shows that this loss comes mainly from the resolver. When only the controller is transferred, accuracy rises from 78.2% to 78.6%, within 0.1 points of the no-transfer setting. The controller therefore transfers to a new debate configuration almost without loss, whereas the resolver benefits more from training data of the unseen debate configuration.

I.3 Transfer to an Unseen Task

Table 18: Transfer to an unseen task at $\rho = 0.1$. Each column holds out one task during training and evaluates on it, and the settings are described in the text. Accuracy (%) is reported with average local model calls per query as subscripts, averaged over the four debate configurations and eight seeds.

Setting	AQuA	ECQA	GSM8K	MMLU	Mean
Baselines					
Fixed-3	$70.4_{5.94}$	$65.6_{4.81}$	$92.3_{4.60}$	$79.1_{4.32}$	$76.8_{4.92}$
Heuristic controller	$69.1_{5.00}$	$65.5_{4.48}$	$91.4_{4.21}$	$79.0_{4.04}$	$76.2_{4.43}$
Resolver transferred					
Controller transferred	$67.1_{3.26}$	$66.1_{3.55}$	$91.8_{3.14}$	$77.8_{3.03}$	$75.7_{3.25}$
Controller calibrated, $n_{\text{cal}} = 100$	$70.6_{5.46}$	$66.4_{4.21}$	$92.8_{3.99}$	$78.7_{3.51}$	$77.1_{4.29}$
Controller calibrated, $n_{\text{cal}} = 300$	$70.8_{5.37}$	$66.4_{4.02}$	$93.0_{4.05}$	$78.8_{3.70}$	$77.3_{4.29}$
Resolver trained with target task					
Controller transferred	$68.9_{3.30}$	$69.8_{3.33}$	$92.6_{3.10}$	$78.7_{3.04}$	$77.5_{3.19}$
Controller calibrated, $n_{\text{cal}} = 300$	$71.3_{5.03}$	$69.7_{3.33}$	$93.2_{3.93}$	$79.4_{3.70}$	$78.4_{4.00}$
No transfer	$71.6_{5.27}$	$70.1_{3.12}$	$93.5_{4.04}$	$79.3_{3.72}$	$78.6_{4.04}$

In Table 18, we evaluate whether our method generalizes to an unseen task. Each column of Table 18 holds out one task, and the remaining three tasks are used for training. Each held-out task is evaluated

as its own query stream with budget $\rho = 0.1$, so its numbers are not directly comparable to the task-level results in Table 7, where the four tasks share one budget. The settings in Table 18 differ in whether each learned component has been trained with data from the target task. A *transferred* resolver or controller is trained only on the other three tasks. A *calibrated* controller is transferred in the same way and then refitted on n_{cal} labeled queries from the target task, which are also used to estimate p_{ext} for that task. A resolver *trained with the target task* uses the target task's training data, and *No transfer* trains both components with the target task, as in our main experiments. Fixed-3 and the heuristic controller involve no learned component and serve as references.

The results show that transfer across tasks is substantially more challenging than transfer across debate configurations. With both components transferred, the system reaches 75.7% accuracy, 2.9 points below No transfer and 1.1 points below Fixed-3, although it uses approximately 34% fewer local model calls than Fixed-3. Replacing the transferred resolver with one trained on the target task improves accuracy by 1.8 points to 77.5%, while calibrating the controller on 300 target queries improves accuracy by 1.6 points to 77.3%. Combining both adaptations reaches 78.4%, within 0.2 points of No transfer. Thus, both components contribute to the cross-task gap, and neither adaptation alone recovers the full performance.

The loss attributable to the controller is concentrated on AQuA. With the answer resolver trained on the target task, a transferred controller reaches 68.9% on AQuA, compared with 71.6% under No transfer, while using only 3.30 local calls per query instead of 5.27. This is consistent with a mismatch in the transferred task-level external-model prior. For a held-out task, p_{ext} is set to the mean external-model accuracy of the three source tasks, whereas the aggregate gap between the external model and resolver accuracy on AQuA is much larger than on the other tasks (Table 15). The resulting prior makes escalation less favorable on AQuA and contributes to premature stopping. Calibrating the controller with 300 labeled AQuA queries substantially reduces this mismatch, increasing accuracy to 71.3%.

Isolating resolver transfer. To separate resolver behavior from controller calibration, Table 19 evaluates answer selection alone on the same final-round disagreements. Holding out the evaluated debate configuration reduces resolver accuracy only from 54.4% to 53.2%, retaining approximately 89% of its gain over confidence-weighted voting. Holding out the entire task produces a larger drop to 48.3%, but still retains approximately 46% of the in-domain gain. The largest task-holdout losses occur on ECQA, where the input ablation in Appendix G.2 shows the strongest dependence on candidate-answer text. These results argue against dependence on a particular evaluated agent configuration as the main explanation, while showing that part of the resolver's benefit is task-specific rather than universally transferable.

Table 19: Resolver answer-selection accuracy (%) on final-round disagreements under different training settings, averaged over all sixteen task–configuration combinations and eight seeds. All settings use the same evaluation queries.

Answer-selection rule / resolver training setting	Accuracy
Confidence-weighted voting (no resolver)	43.2
Target task and configuration included	54.4
Evaluated task held out	48.3
Evaluated debate configuration held out	53.2

I.4 Practical Implications

These results highlight a clear difference between transfer to an unseen debate configuration and transfer to an unseen task. Across the configurations considered here, a resolver and controller trained on the other debate configurations can be applied to a new one with only a small loss in accuracy. However, the resolver combines transferable debate signals with task-dependent answer semantics, and zero-shot transfer to a new task is therefore more challenging. Our main results should accordingly be interpreted under the stated workload-adaptive setting, in which representative labeled trajectories are available for resolver training. When no such data are available, the zero-shot results quantify the resulting loss; when target-task data can be collected, retraining the resolver provides the

largest individual recovery, while the controller can be further adapted through few-shot calibration using a few hundred labeled queries from that task.

J Discussion and Limitations

While our results show that jointly controlling local debate and external-model escalation improves both accuracy and efficiency, several aspects of our approach and evaluation remain limited and point to directions for future work.

First, although the answer resolver adds little processing time relative to autoregressive generation, it relies on a 4B-parameter embedding model that occupies an additional 7.6 GB of memory in bf16. Smaller or quantized encoders and more memory-efficient representations are therefore important directions for reducing this overhead. Second, our debate controller operates only on non-converged debate states. Following the agreement-based termination convention of the debate protocol, unanimous consensus is returned immediately. This provides a clear stopping rule, but agreement is not a correctness guarantee: agents can converge through shared biases or error propagation, as reflected by the wrong-convergence statistics in Appendix H. Post-consensus verification or selective escalation would extend the decision scope beyond that considered here. Third, the external-model estimate is task-level rather than query-specific. Within each task, the escalation score is therefore driven primarily by predicted local outcomes and cannot directly model whether the external and local models have complementary errors on a particular query. The controller may spend calls on queries that are difficult for both systems or miss cases in which the external model could correct a high-confidence local error. Learning query-conditional external-model advantage estimates would address these limitations. Finally, our formulation assumes discrete answer candidates with unambiguous ground-truth targets. This assumption is well suited to multiple-choice tasks and reasoning tasks with short, well-defined answers, but does not directly extend to open-ended generation, where outputs may be semantically equivalent, partially correct, or subjectively judged. Extending the framework to such settings would require semantic grouping of candidate answers together with graded or preference-based evaluation rather than exact matching and binary correctness.