

HALT: Evaluator-Guided Clause-Level Pruning and Early Halting for Lower-Cost Multi-Agent Systems

Zongwei Tang^{1*}, Yewen Li^{1*}, Bo Liu^{1,2}, Zezong Zhang¹,
Yiqing Cai¹, Yiwei Guo¹, Houjie Qiu¹, Xingkong Ma^{1†}

¹College of Computer Science and Technology, National University of Defense Technology

²Strategic Evaluation and Consultation Center, Academy of Military Sciences

Email addresses: zongweitang@nudt.edu.cn, yewenlinudt@nudt.edu.cn, kyle.liu@nudt.edu.cn, zhangzezong@nudt.edu.cn, caiyiqing@nudt.edu.cn, guoyiwei20@nudt.edu.cn, qiuhoujie@nudt.edu.cn, Correspondence: maxingkong@nudt.edu.cn.

Abstract

Multi-agent systems improve complex reasoning but create a critical cost and effectiveness dilemma: API costs grow with each additional agent interaction, while long shared contexts can amplify errors and dilute useful evidence, thereby degrading task performance. This paper proposes an evaluator-guided clause-level pruning and early halting method for lower-cost multi-agent systems named HALT, which implements a macro-to-micro intervention. At the macro-level, a difficulty aware router classifies query complexity, bypassing redundant multi-agent systems overhead for simple tasks. At the micro-level, a lightweight hindsight evaluator monitors generation in real time for complex reasoning, driving a dual-axis intervention policy: spatial pruning removes erroneous clauses, while temporal halting stops interaction when the retained clauses indicate sufficient answer confidence. The extensive experiments across three different datasets demonstrate the outperformance of HALT over the baselines, achieving the 97.98%, 90.82%, 87.14% in GSM8K, MMLU, MMLU-Pro in accuracy while drastically mitigating computational overhead, with a maximum reduction of 94.3% in prompt tokens and an unprecedented reduction of 98.9% in completion tokens, which validate the effectiveness of HALT. Additional HotpotQA results show that the evaluator generalizes to relatively open task and consistently reduces inference cost across backbones and MAS topologies. Code is available at [URL](https://github.com/ma-xingkong/HALT).

1 Introduction

With Kahneman theory, while large language models (LLMs) excel at intuitive "System-1" heuristic generation, they often struggle with complex, deliberative "System-2" reasoning (Lin et al., 2023). Multi-agent systems (MAS) have emerged to

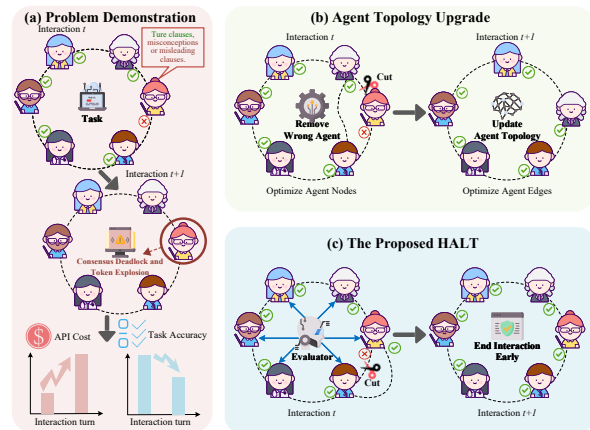


Figure 1: Comparison of MAS intervention paradigms. (a) represents that incorrect clauses made by agents will trigger context degradation, (b) represents the pruning MAS topology, (c) represents using HALT to save API costs and even improve task accuracy.

bridge this gap, attracting widespread research interest for their pivotal role in applications such as software engineering (Lee et al., 2025; Zhang et al., 2025c), mathematical deduction (Zhou et al., 2025; Yuan and Xie, 2025), and scientific discovery (Su et al., 2025). Using collective intelligence through iterative collaboration, MAS effectively marginalize individual cognitive biases to improve performance on these complex reasoning tasks (Qian et al., 2024; Liang et al., 2024a).

However, a counter-intuitive phenomenon consistently emerges in empirical studies (Bai et al., 2026): **increasing the interaction turns among agents does not proportionally improve task performance and can even degrade it, despite the accompanying surge in API costs.** Most strikingly, (Fu et al., 2025) observes that the accuracy of reasoning can even decrease below the baselines of a single agent due to incorrect viewpoints amplified during interaction. As illustrated by the framework shown in Figure 1(a), we identify this phenomenon as the **context degradation paradox**:

*Equal Contribution.

†Corresponding Author.

as interaction rounds increase, API costs grow exponentially while the density of valid information decays relative to noise, exacerbating attention dilution in transformers (Liu et al., 2024b; Xiao et al., 2024). Within the degraded context, localized errors, which we term **toxic anchors**, are semantically flawed reasoning units that propagate and amplify through auto-regressive generation, triggering compounding errors and logical cascading (Peng et al., 2024; Liang et al., 2024a; Bao et al., 2025; Xie et al., 2026).

Therefore, as shown in Figure 1(b), state-of-the-art methods utilize macro-level topology optimization, pruning communication edges or removing agents to optimize the topology (Zhang et al., 2025a,b; Wang et al., 2025c; Li et al., 2025). However, direct topology optimization approaches suffer from a granularity mismatch: treating agent outputs as indivisible units forces a suboptimal trade-off, where pruning nodes causes information loss, while retaining them leads to context degradation. Furthermore, these methods exhibit architectural rigidity: once the communication topology is learned during training, it becomes computationally intractable to dynamically reconfigure agent connections or scale node participation for varying task complexities (Gui and Ji, 2025).

To enable multi-agent collaboration without framework specific redesign, this paper presents HALT, that dynamically calibrates agent participation across different MAS architectures by unifying macroscopic routing with causality-driven microscopic intervention. Specifically, HALT introduces a two-stage intervention pipeline that aggressively reduces API expenditure while preserving task accuracy. Stage 1 employs a difficulty aware router to dispatch trivial queries to lightweight single agent paths, eliminating redundant token overhead at the entry point. Stage 2 targets complex queries routed into MAS loops: a lightweight evaluator performs dual-axis reasoning regulation: (i) clause-level spatial pruning to neutralize harmful cognitive anchors, and (ii) purity-gated temporal halting to detect and terminate spurious consensus formation. Notably, beyond operational cost control via pruning and early stopping, the evaluator operationalizes the autonomous cessation of MAS interactions. In summary, our contributions are as follows:

- **Theoretical Formulation.** We propose HALT, formally cast multi-agent reasoning as a truncatable causal attribution process, de-

composing the global objective into macro-level routing decisions and micro-level intervention policies.

- **Practical Solution.** We design an adaptive difficulty aware router coupled with a lightweight causal evaluator, achieving substantial reduction in aggregate input and output token costs without compromising reasoning accuracy and even enhancing it.
- **Experimental Validation.** Across GSM8K, MMLU, MMLU-Pro, and HotpotQA with three LLMs, HALT achieves state-of-the-art or competitive accuracy on the main reasoning benchmarks (97.98%, 90.82%, and 87.14%) while reducing prompt/completion tokens by up to 94.32%/99.45%; its HotpotQA results further show evaluator transfer without HotpotQA training samples, reducing runtime by 46.39% on average.

2 Related Work

2.1 MAS

MAS mitigate individual cognitive biases through distributed collaboration and adversarial interactions. Different MAS employ different collaborative topological optimization methods. MetaGPT (Hong et al., 2024) uses DAGs to prevent reasoning divergence, Chain-of-Agents (Zhang et al., 2024) partitions long contexts, and hierarchical paradigms enable parallel processing (Ning et al., 2024). Existing efforts frame MAS optimization as joint policy (Wang et al., 2024a; Tang et al., 2026) or reduce costs by role-based aggregation (Wang et al., 2025b). Though adversarial interactions improve factual accuracy (Du et al., 2024), the homogeneous agents risk is also upgrade (Xiong et al., 2023). MAD frameworks address this through tit-for-tat strategies (Liang et al., 2024b), cross examination (Cohen et al., 2023), and stance diversified initialization (Wang et al., 2024c). Self-reflection loops (Shinn et al., 2023) and symbolic solvers (Pan et al., 2023) enhance logical rigor, while game-theoretic formulations ensure consensus (Yi et al., 2025). While effective at macro-level pruning, topology-based methods suffer from granularity mismatch, treating agent outputs as atomic blocks. Therefore, HALT try to utilize the pathway to micro semantic resolution to solve the granularity mismatch problem.

2.2 MAS Optimization

The efficient inference of LLMs has attracted the interest of reserachers due to the prohibitive costs of querying proprietary. The MAS optimization can be classified into static methods, and dynamic methods two categories. Static approaches constrain computation through input compression (Jiang et al., 2023b; Li et al., 2023; Chuang et al., 2024; Jiang et al., 2024) or feature attribution (Packer et al., 2023; Raiyan et al., 2025), but may remove critical reasoning. Topology interventions limit communication overhead yet suffer granularity mismatches (Zhang et al., 2023, 2025a; Wang et al., 2025a). Alternative strategies exploit the heterogeneity of the model by distilling knowledge (Shridhar et al., 2023) and static ensemble voting (Wang et al., 2023), while system optimizations reduce overhead through the reuse and quantification of attention (Li et al., 2024; Gim et al., 2024; Kim et al., 2024). Dynamic methods adapt resource allocation following long-tail distributions. FrugalGPT (Raiyan et al., 2025) pioneered model cascading; subsequent work employs learned difficulty embeddings (Sakota et al., 2024) for improved routing. Dynamic topology orchestration reduces participants by graph autoencoders (Zhang et al., 2025b) and contribution-based pruning (Wang et al., 2025c). Adaptive termination mechanisms halt debates upon consensus detection (Chen et al., 2024) or via learned policies (Jiang et al., 2023a), while early-exit architectures enable token-level skipping (Elhoushi et al., 2024). Static compression and dynamic halting, respectively, disrupt reasoning content and multi-turn dependencies. HALT bridges this gap through a unified framework coupling macro-level difficulty aware routing with micro-level intra-turn content control.

3 Method

To rigorously articulate the challenges addressed by HALT, this paper first describes a mathematical formulation of the problem and the HALT framework, illustrated in Figure 2. The HALT comprises three core components: a difficulty aware router, a hindsight-guided evaluator, and a dual-axis intervention policy. Firstly, HALT employs the router to predict query complexity and bypass redundant MAS overhead for simple tasks. Then, our method utilizes the lightweight evaluator to perform real-time spatial pruning, surgically removing erroneous clauses. Finally, the system terminates

deliberation upon reaching a high-confidence consensus through purity-aware temporal halting, substantially reducing API costs while improving reasoning accuracy. The detailed algorithm of HALT has been described in Appendix A.

3.1 Problem Formulation

This paper assumes that an MAS includes N LLM-based agents $\mathcal{V} = \{V_1, \dots, V_N\}$. Given an input query $x \in \mathcal{X}$ and ground truth answer $y \in \mathcal{Y}$, the MAS engage in T -turn interaction to generate a trajectory of States $\mathcal{S} = (s_1, \dots, s_T)$, where each State s_t encapsulates the dialogue history and intermediate reasoning. The target of MAS is to produce a final generation to accurately answer the query x .

Let c_{single} denote the inference cost of a single agent call and $c_{\text{MAS}} = T \cdot N \cdot c_{\text{single}}$ denote the cost of full MAS interaction. Given a budget constraint B , HALT aims to minimize task error $\mathcal{D}_h$ while reducing computational resources:

$$\min \mathbb{E}_{(x,y) \sim \mathcal{D}_h} [\mathbb{I}\{y^* \neq y\}] \text{ s.t. } c(x) \leq B \quad (1)$$

where y^* is the answer of MAS and $c(x)$ is the realized cost for query x .

3.2 Difficulty Aware Router

The macro policy π_{macro} determines which queries require MAS. In HALT, the difficulty aware router is formulated as $\mathcal{H}_\phi : \mathcal{X} \rightarrow \{0, 1\}$, where 1 indicates that x should be routed to the MAS.

Self-consistency calibration. Training $\mathcal{H}_\phi$ requires difficulty labels without heuristic bias. HALT employs the base model $\mathcal{M}_{\text{base}}$, generating $k = 3$ responses $\{y_1^*, \dots, y_k^*\}$ at temperature $T = 0.7$. A query x is labeled difficult ($y_{\text{hard}} = 1$) if majority voting among the k generations contradicts ground truth y ; otherwise, it is labeled easy ($y_{\text{hard}} = 0$). This yields the routing dataset $\mathcal{D}_{\text{route}} = \{(x_i, y_{\text{hard}}^{(i)})\}_{i=1}^M$.

Cost sensitive objective. Unlike traditional routing mechanisms that impose a rigid global budget constraint, the difficulty aware router $\mathcal{H}_\phi$ achieves computational efficiency intrinsically through accurate difficulty classification. The router $\mathcal{H}_\phi$ is applied to minimize the expected cost-sensitive risk over the routing dataset $\mathcal{D}_{\text{route}}$:

$$\min \mathbb{E}_{(x, y_{\text{hard}}) \sim \mathcal{D}_{\text{route}}} [\mathcal{L}_{\text{cs}}(\mathcal{H}_\phi(x), y_{\text{hard}})] \quad (2)$$

where $\mathcal{H}_\phi(x)$ outputs the predicted probability of the query being hard. To prevent the irreversible

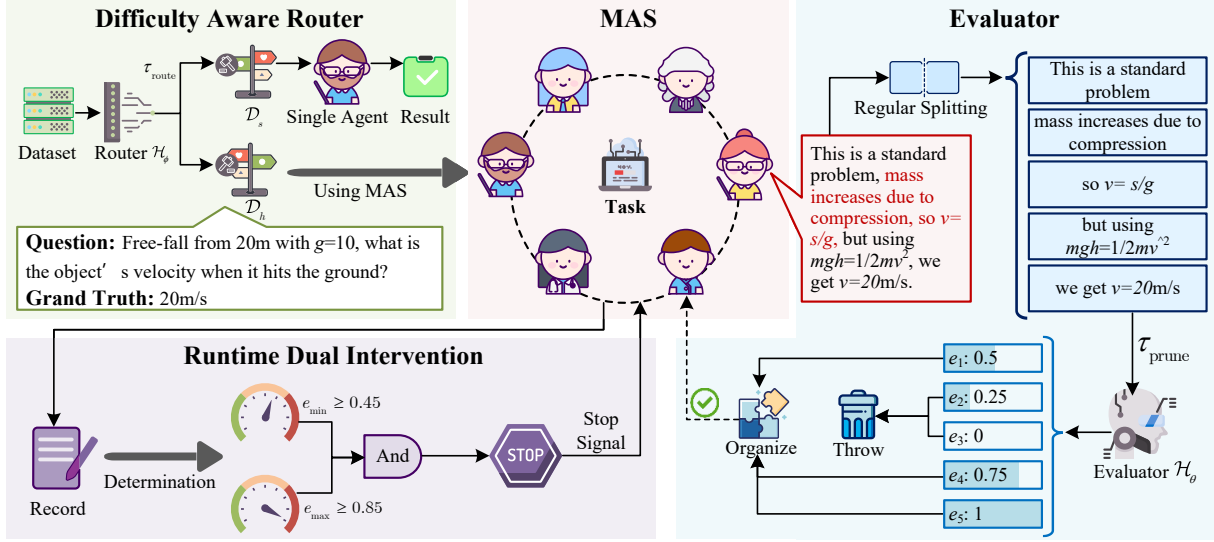


Figure 2: Schematic diagram of the HALT workflow.

accuracy degradation caused by false negatives, HALT introduces an asymmetric weighting scheme into the objective function:

$$\mathcal{L}_{cs}(\mathcal{H}_\phi(x), y_{\text{hard}}) = -\lambda \cdot y_{\text{hard}} \log(\mathcal{H}_\phi(x)) - (1 - y_{\text{hard}}) \log(1 - \mathcal{H}_\phi(x)) \quad (3)$$

By adopting above formulation, the system does not require an explicit computational budget constraint. Instead, cost reduction emerges naturally as a byproduct of accurate classification: queries correctly identified as simple are inherently bypassed to a single agent, implicitly optimizing the global API expenditure while maximizing the safety margin for complex reasoning tasks.

Architecture. The router employs dual-modal feature extraction: the semantic channel encodes x using a pre-trained text encoder, producing $f_{\text{sem}} \in \mathbb{R}^{1024}$; the structural channel extracts normalized query length and lexical diversity to produce $f_{\text{struct}} \in \mathbb{R}^2$. The fused feature $f = [f_{\text{sem}}; f_{\text{struct}}] \in \mathbb{R}^{1026}$ is processed by a gradient boosting classifier outputting probability $\pi_{\text{route}}(x) \in [0, 1]$. By decoupling the probability estimation from the final discrete classification step, we gain the mathematical flexibility to calibrate the decision boundary according to our cost-sensitive objective. During inference, rather than adopting the default neutral threshold, HALT deliberately applies a conservative, risk-averse threshold $\pi_{\text{route}}(x) \geq 0.45$.

3.3 Hindsight-Guided Evaluator

When a hard query enters the MAS, agents engage in multi-turn interaction. To fundamentally eradicate context bloat, we propose a hindsight-guided

evaluator that identifies toxic anchors for targeted pruning while preserving semantic coherence of the remaining context.

Clause level State decomposition. At turn t , we segment the generated content into clauses $\mathcal{E}_t = \{e_1, e_2, \dots, e_{m_t}\}$ using punctuation-based segmentation, where each clause e_j represents a minimal evaluable unit and $m_t = |\mathcal{E}_t|$.

Evaluation scoring. To address the credit assignment problem in long-chain reasoning, specifically, identifying which intermediate clauses are causally responsible for the final outcome, HALT introduces a distillation-based valuation mechanism, which process distinguishes between an offline scorer and a deployable evaluator. During the data collection phase, HALT uses a high-capability LLM $\mathcal{M}_h$ as a scorer. Crucially, the scorer is granted hindsight knowledge, evaluating the MAS transcript with access to both the query x and the ground truth y . The inclusion of y is not to let the model copy the answer, but the scorer identifies which clauses logically align with the verified solution path and which ones constitute toxic anchors or hallucinations that lead to divergence. The high-capability LLM $\mathcal{M}_h$ scores each clause on a 5-point ordinal scale $\mathcal{S} = \{0.0, 0.25, 0.5, 0.75, 1.0\}$ based on three criteria: internal consistency of reasoning, novelty of contribution to the solution, and relevance to the query and ground truth, and detailed prompts are shown in Appendix B.

Objective of distillation. To bridge the gap between offline supervision and real-time deployment, the HALT distills knowledge of $\mathcal{M}_h$ into a

lightweight evaluator $\mathcal{H}_\theta$. It is imperative to note that the evaluator never sees the ground truth during training and inference. The training of evaluator $\mathcal{H}_\theta$ is regarded as a regression task, with the aim of minimizing the discrepancy between the predicted value of the evaluator and the hindsight label of the scorer. Formally, for a problem context x and a set of m_t reasoning clauses $\{e_1, e_2, \dots, e_{m_t}\}$ generated in turn t , the distillation loss is defined as:

$$\mathcal{L}_{\text{eval}} = \frac{1}{m_t} \sum_{j=1}^{m_t} (\mathcal{H}_\theta(e_j | x) - z_j)^2 \quad (4)$$

where $z_j \in [0, 1]$ is the scalar score assigned by the scorer. By optimizing this objective, $\mathcal{H}_\theta$ learns to approximate the latent logic-value mapping without requiring ground truth answers during inference.

Toxic recall criterion. In MAS contexts, failing to identify low-quality clauses is more harmful than misscoring high-quality clauses. Therefore, this paper defines critical recall as a decisive training metric for HALT:

$$\text{CR} = \frac{\sum_j \mathbb{I}\{s_j \leq 0.25, \mathcal{H}_\theta(e_j | x) \leq \tau_{\text{crit}}\}}{\sum_j \mathbb{I}\{s_j \leq 0.25\}} \quad (5)$$

where $\tau_{\text{crit}} = 0.35$ ensures numerical stability, ensuring the evaluator reliably identifies clauses that should be pruned.

3.4 Runtime Dual-Axis Intervention

The micro policy $\pi_{\text{micro}} = (\pi_{\text{prune}}, \pi_{\text{halt}})$ is implemented for the runtime intervention conditioned on evaluator scores.

Spatial intervention π_{prune} : At each round t , the spatial policy selects clauses to retain:

$$a_t^{(j)} = \pi_{\text{prune}}(e_j) = \mathbb{I}\{\mathcal{H}_\theta(e_j | x) \geq \tau_{\text{prune}}\} \quad (6)$$

where $\tau_{\text{prune}} = 0.45$ is the pruning threshold. The state transition under spatial intervention is:

$$s_{t+1} = \text{Concat}(s_t, \{\mathcal{H}_\theta(e_j | x) \geq \tau_{\text{prune}}\}) \quad (7)$$

where $\text{Concat}(\cdot)$ denotes concatenation to the dialogue history. This update enforces context sparsity, retaining only clauses the evaluator deems informative.

Temporal intervention π_{halt} : The temporal policy terminates MAS by checking three conditions simultaneously. First, the agents must produce a parseable answer. Second, at least one clause must

achieve a maximum quality threshold e_{max} of 0.85 or above, indicating a strong reasoning step. Third, all clauses must meet a minimum quality threshold e_{min} of 0.45, ensuring no toxic or erroneous content is output. The system halts deliberation and outputs the answer only when all three conditions are satisfied, preventing both premature termination and redundant deliberation.

4 Experiments

This paper conducts a comprehensive evaluation of HALT to address four research questions: **RQ1:** Does HALT achieve a superior accuracy and cost trade-off compared to baselines? **RQ2:** What are the individual contributions of the router, evaluator, pruning, and halting modules? **RQ3:** Can a 142M-parameter local evaluator effectively supervise billion-scale LLMs? **RQ4:** How well does HALT’s hindsight-guided evaluator transfer to a relatively open task without task-specific evaluator training? **RQ5:** How sensitive is HALT to variations in the purity thresholds ($\tau_{\text{route}}, \tau_{\text{prune}}, e_{\text{max}}, e_{\text{min}}$)?

4.1 Experimental Settings

Datasets and Metrics. We evaluate HALT on three benchmarks of increasing difficulty: GSM8K (Cobbe et al., 2021) for multi-step mathematical reasoning, MMLU (Hendrycks et al., 2021) for knowledge assessment across 57 subjects, and MMLU-Pro (Wang et al., 2024b) for complex logical inference. To evaluate cross-task effectiveness and evaluator generalization, we further introduce HotpotQA (Yang et al., 2018), a relatively open multi-hop question answering benchmark with verifiable answers. We measure task quality by final answer accuracy, runtime by wall-clock time, and cost by prompt tokens (PT) and completion tokens (CT).

Baselines. We compare against three categories of methods: (1) single agent chain of thought reasoning (SCoT); (2) Fully Connected network (FC) with 6 heterogeneous agents, representing upper bound of performance but high cost; and (3) State-of-the-art MAS optimizers: AP (Zhang et al., 2025a), AD (Wang et al., 2025c), and GD (Zhang et al., 2025b). To isolate component contributions, we additionally construct three ablation variants: w/o_r , w/o_p , and w/o_h . All MAS methods use the same agent set, interaction rounds, and prompt configuration, shown in Appendix B for details.

Table 1: The task accuracy on GSM8K, MMLU, and MMLU-pro with different LLMs.

Methods	GPT-4o-mini			DeepSeek-v3			Qwen3-14b		
	GSM8K	MMLU	MMLU-Pro	GSM8K	MMLU	MMLU-Pro	GSM8K	MMLU	MMLU-Pro
SCOT	89.16	73.86	55.71	79.10	86.94	77.14	<u>95.98</u>	78.25	57.14
FC	89.53	<u>83.16</u>	65.71	90.45	87.69	78.57	93.14	81.40	75.71
AP	<u>94.19</u>	79.74	73.91	<u>96.35</u>	90.25	82.86	94.41	<u>83.80</u>	79.71
AD	93.45	78.43	68.10	96.23	91.54	85.71	94.82	83.66	<u>76.81</u>
GD	93.55	83.01	73.91	96.21	<u>91.15</u>	<u>86.96</u>	95.91	82.35	75.36
w/o_r	92.75	82.14	62.86	94.41	88.17	80.00	94.95	82.34	72.60
w/o_p	93.78	81.12	68.57	93.71	86.46	81.43	93.93	83.39	73.61
w/o_h	91.92	80.35	70.00	94.95	87.72	84.29	90.91	83.16	72.86
Ours	96.07	83.40	<u>72.86</u>	97.98	90.82	87.14	96.97	84.21	75.71

Implementation Details. We evaluate across three LLM backbones: GPT-4o-mini (OpenAI, 2024), DeepSeek-v3 (Liu et al., 2024a), and Qwen3-14b (Yang et al., 2025). For evaluator training, test sets are strictly isolated and Gemini-3-Pro (Google DeepMind, 2025) (temperature = 0) serves as hindsight scorer to provide dense reward signals. The router $\mathcal{H}_\phi$ uses BGE-Large embeddings with a HistGradientBoosting classifier (Pedregosa et al., 2011) ($\lambda = 3.0$, $\tau_{route} = 0.5$). The evaluator $\mathcal{H}_\theta$ is based on DeBERTa-v3-Small (He et al., 2021) with task-specific τ_{prune} , e_{max} , and e_{min} . To maximise reproducibility, all experiments are conducted with temperature = 0, yielding deterministic outputs; reported results are from a single run. All experiments run on six NVIDIA A100-80GB GPUs.

4.2 Performance Analysis

Accuracy analysis: We first evaluate HALT’s impact on task accuracy to address RQ1. The results across three benchmarks and three LLM backbones are demonstrated/shown in Table 1. In GSM8K, HALT achieves the highest accuracy across all models: 96.07% with GPT-4o-mini, 97.98% with DeepSeek-v3, and 96.97% with Qwen3-14b, outperforming the strongest FC baseline by 6.54%, 7.53%, and 3.83%, respectively. In MMLU, HALT attains top performance with GPT-4o-mini at 83.40% and Qwen3-14b at 84.21%, while remaining competitive with DeepSeek-v3 at 90.82%, marginally below AD at 91.54%. For MMLU-Pro, HALT achieves the best result with DeepSeek-v3 at 87.14%, though it is slightly surpassed by AP and GD on GPT-4o-mini and trails AP in Qwen3-14b. These results demonstrate that HALT excels particularly in structured mathematical reasoning while maintaining strong performance on broad knowl-

edge benchmarks. Moreover, HALT provides an example how clause-level pruning achieves a 46.4% compression ratio without compromising reasoning correctness in the Appendix C.

Cost analysis: Table 2 reveals the token efficiency gains of HALT. HALT consistently delivers substantial reductions in both prompt tokens and completion tokens across all configurations. Notably, on GSM8K with Qwen3-14b, HALT reduces prompt tokens by 92.87% from 3,212 to 229 and completion tokens by 99.45% from 18,817 to 103. Similarly, on MMLU-Pro with DeepSeek-v3, HALT achieves an 83.52% reduction in prompt tokens and a 96.43% reduction in completion tokens. These gains scale robustly to larger and more complex tasks, confirming the effectiveness of HALT’s selective reasoning mechanism in significantly lowering inference overhead. Detailed cost breakdowns are provided in Appendix D. Appendix E also provides a detailed demonstration of the effectiveness of HALT in MAS with varying numbers and structures of agents.

4.3 Comparison Analysis

Table 1 and Table 2 reveal a consistent pattern: existing MAS optimizers improve efficiency at the expense of either accuracy or stability, whereas HALT achieves both objectives simultaneously. In GSM8K with GPT-4o-mini, GD reduces PT by 46.61% but attains only 93.55% accuracy. HALT surpasses it on both dimensions, cutting PT by 83.13% while reaching 96.07% accuracy, a 2.52 absolute percentage point improvement with nearly half the prompt overhead. The gap widens further with DeepSeek-v3: HALT reduces PT and CT by 88.96% and 98.88%, respectively, compared with 40.04% and 41.42% for AP, while also outperforming AP in accuracy by 1.63 points. In the

Table 2: The average task tokens on GSM8K, MMLU, and MMLU-pro with different LLMs.

Method	GSM8K		MMLU		MMLU-pro	
	PT	CT	PT	CT	PT	CT
GPT-4o-mini						
FC	3924	22017	5511	31059	5838	33715
AP	3790 $\downarrow$ 3.41%	17053 $\downarrow$ 22.55%	2751 $\downarrow$ 50.08%	10447 $\downarrow$ 66.36%	6552 $\uparrow$ 12.23%	24108 $\downarrow$ 28.49%
AD	4128 $\uparrow$ 5.20%	18064 $\downarrow$ 17.95%	3633 $\downarrow$ 34.08%	11099 $\downarrow$ 64.26%	4730 $\downarrow$ 18.98%	15607 $\downarrow$ 53.71%
GD	2095 $\downarrow$ 46.61%	9051 $\downarrow$ 58.89%	2265 $\downarrow$ 58.90%	7626 $\downarrow$ 75.45%	6802 $\uparrow$ 16.51%	28178 $\downarrow$ 16.42%
w/o_r	3282 $\downarrow$ 16.36%	18352 $\downarrow$ 16.65%	1830 $\downarrow$ 66.79%	10186 $\downarrow$ 67.20%	2624 $\downarrow$ 55.05%	15055 $\downarrow$ 55.35%
w/o_p	720 $\downarrow$ 81.65%	1215 $\downarrow$ 94.48%	3511 $\downarrow$ 36.29%	13409 $\downarrow$ 56.83%	4443 $\downarrow$ 23.90%	17866 $\downarrow$ 47.01%
w/o_h	986 $\downarrow$ 74.87%	3547 $\downarrow$ 83.89%	1230 $\downarrow$ 77.68%	6566 $\downarrow$ 78.86%	1628 $\downarrow$ 72.11%	8703 $\downarrow$ 74.19%
Ours	662 $\downarrow$ 83.13%	1181 $\downarrow$ 94.64%	879 $\downarrow$ 84.05%	3140 $\downarrow$ 89.89%	1488 $\downarrow$ 74.51%	7353 $\downarrow$ 78.19%
DeepSeek-v3						
FC	4401	24771	7409	40291	7450	44912
AP	2639 $\downarrow$ 40.04%	14510 $\downarrow$ 41.42%	3908 $\downarrow$ 47.25%	13197 $\downarrow$ 67.25%	9627 $\uparrow$ 29.22%	32746 $\downarrow$ 27.09%
AD	2947 $\downarrow$ 33.04%	13996 $\downarrow$ 43.50%	4413 $\downarrow$ 40.44%	13008 $\downarrow$ 67.71%	6759 $\downarrow$ 9.28%	21050 $\downarrow$ 53.13%
GD	1562 $\downarrow$ 64.51%	8231 $\downarrow$ 66.77%	3088 $\downarrow$ 58.32%	9625 $\downarrow$ 76.11%	9438 $\uparrow$ 26.68%	39083 $\downarrow$ 12.98%
w/o_r	932 $\downarrow$ 78.82%	1794 $\downarrow$ 92.76%	538 $\downarrow$ 92.74%	2846 $\downarrow$ 92.94%	4312 $\downarrow$ 42.12%	24742 $\downarrow$ 44.91%
w/o_p	1328 $\downarrow$ 69.83%	4469 $\downarrow$ 81.96%	5667 $\downarrow$ 23.50%	23661 $\downarrow$ 41.27%	4832 $\downarrow$ 35.14%	17957 $\downarrow$ 60.02%
w/o_h	558 $\downarrow$ 87.32%	1921 $\downarrow$ 92.24%	423 $\downarrow$ 94.29%	2137 $\downarrow$ 94.70%	2485 $\downarrow$ 66.64%	10825 $\downarrow$ 75.90%
Ours	486 $\downarrow$ 88.96%	277 $\downarrow$ 98.88%	421 $\downarrow$ 94.32%	1780 $\downarrow$ 95.58%	1228 $\downarrow$ 83.52%	1603 $\downarrow$ 96.43%
Qwen3-14b						
FC	3212	18817	8656	45840	9094	50811
AP	2569 $\downarrow$ 20.02%	15702 $\downarrow$ 16.55%	8265 $\downarrow$ 4.52%	25655 $\downarrow$ 44.03%	12410 $\uparrow$ 36.46%	43053 $\downarrow$ 15.27%
AD	2271 $\downarrow$ 29.30%	12658 $\downarrow$ 32.73%	5511 $\downarrow$ 36.33%	15179 $\downarrow$ 66.89%	9428 $\uparrow$ 3.67%	36409 $\downarrow$ 28.34%
GD	3029 $\downarrow$ 5.7%	19796 $\uparrow$ 5.20%	8479 $\downarrow$ 2.04%	31734 $\downarrow$ 30.77%	11048 $\uparrow$ 21.49%	42521 $\downarrow$ 16.32%
w/o_r	268 $\downarrow$ 91.66%	15448 $\downarrow$ 17.90%	4997 $\downarrow$ 42.27%	27069 $\downarrow$ 40.95%	5035 $\downarrow$ 44.63%	26858 $\downarrow$ 47.14%
w/o_p	2083 $\downarrow$ 35.15%	6895 $\downarrow$ 63.36%	6587 $\downarrow$ 23.90%	18776 $\downarrow$ 59.04%	6904 $\downarrow$ 24.08%	21109 $\downarrow$ 58.46%
w/o_h	856 $\downarrow$ 73.35%	2545 $\downarrow$ 86.47%	3780 $\downarrow$ 56.33%	19171 $\downarrow$ 58.18%	3765 $\downarrow$ 58.60%	18461 $\downarrow$ 63.67%
Ours	229 $\downarrow$ 92.87%	103 $\downarrow$ 99.45%	2502 $\downarrow$ 71.10%	822 $\downarrow$ 98.21%	3848 $\downarrow$ 57.69%	18222 $\downarrow$ 64.14%

more challenging MMLU-Pro benchmark, the limitations of single-dimension optimization become pronounced. With GPT-4o-mini, GD actually increases PT consumption by 16.51% relative to the FC baseline, and AD achieves merely an 18.98% PT reduction while sacrificing accuracy to 68.10%. In contrast, HALT reduces both PT and CT by 74.51% and 78.19% while maintaining a competitive accuracy of 72.86%. A similar pattern holds for DeepSeek-v3 on MMLU-Pro: HALT attains the highest accuracy of 87.14% and the lowest token consumption, whereas GD increases PT by 26.68% and reduces CT by only 12.98%. These results confirm that topology pruning and dynamic elimination alone are insufficient for long-context tasks; HALT’s clause-level content filtering provides the necessary granularity for effective cost control.

4.4 Ablation Analysis

The ablation experiments address RQ2 and RQ3 by isolating the contributions of the router, pruning, and halting modules. Table 2 shows that each component governs a distinct cost channel.

Effect of removing the router (w/o_r). Without adaptive dispatch, the system defaults to MAS processing for all queries, inflating PT On GSM8K with DeepSeek-v3, w/o_r consumes 932 prompt tokens, nearly twice the 486 tokens of the full HALT, and suffers a 10-point accuracy drop on MMLU-Pro with GPT-4o-mini, from 72.86% to 62.86%.

Effect of removing pruning (w/o_p). Retaining agent decoupling but eliminating content pruning yields low PT, 720 tokens versus 662 for HALT on GSM8K with GPT-4o-mini, because relevance data

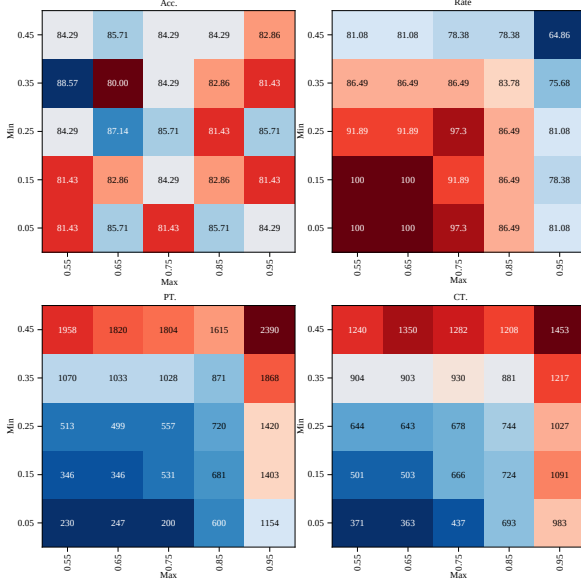


Figure 3: Sensitivity of temporal halting thresholds with DeepSeek-v3 in MMLU-pro.

is no longer appended to the prompt. However, unfiltered agent outputs accumulate across interaction rounds, causing severe CT inflation: on GSM8K with DeepSeek-v3, CT reaches 4,469, which is $16.1 \times$ the 277 tokens of the full system.

Effect of removing halting (w/o_h). Eliminating early termination primarily affects CT through unchecked over-generation. On GSM8K with Qwen3-14b, CT rises from 103 to 2,545 tokens, a $24.7 \times$ increase. Iterative context accumulation also increases PT, from 229 to 856 on the same configuration, degrading accuracy from 96.97% to 90.91%.

Collectively, the router controls whether to activate MAS, the pruning module filters what content enters the context, and the halting module determines when to terminate. Removing any one component disrupts this complementary design, leading to suboptimal trade-offs.

4.5 Parameter Sensitivity Analysis

The experiments in this section answer RQ4 by examining whether HALT is controlled by interpretable validation-calibrated thresholds rather than brittle test-time constants. The thresholds affect different parts of the controller: τ_{route} determines whether a query enters MAS, τ_{prune} controls how much generated content is retained, and (e_{min}, e_{max}) determine when the retained context is sufficiently clean and confident for early termination. Figures 3 and 4 focus on the micro-control

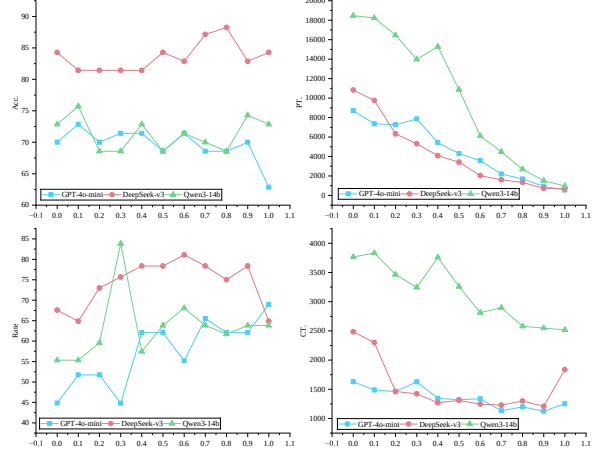


Figure 4: Impact of pruning thresholds on LLMs Performance in MMLU-pro.

thresholds, while additional backbone and dataset level sweeps are reported in Appendix E.

Temporal halting thresholds. Figure 3 shows that the temporal thresholds expose a controllable accuracy and cost frontier on MMLU-Pro with DeepSeek-v3. Across the tested grid, accuracy remains between 80.00% and 88.57%, with most settings falling in the 81-87% range; this indicates that HALT is not dependent on a single isolated threshold pair. The best observed accuracy is 88.57% at $e_{min} = 0.35$ and $e_{max} = 0.55$, while the operating point $e_{min} = 0.25$, $e_{max} = 0.65$ still reaches 87.14%. The thresholds mainly change how aggressively HALT terminates deliberation: at $e_{max} = 0.55$, reducing e_{min} from 0.45 to 0.05 increases the early-stopping ratio from 81.08% to 100%, and reduces max prompt and completion tokens from 1958/1240 to 230/371, while accuracy decreases moderately from 84.29% to 81.43%. Thus, lower bounds define latency-oriented operating points, whereas stricter bounds preserve longer deliberation for accuracy-oriented settings.

Spatial pruning threshold. Figure 4 evaluates τ_{prune} by varying the retained-clause boundary from 0.0 to 1.0 across three backbones. Increasing τ_{prune} generally removes more low-scored clauses before they enter the shared context, which reduces prompt/completion tokens and often increases the early-stopping ratio. The accuracy response is model-dependent: DeepSeek-v3 remains comparatively stable, staying in the 81-88% range across the sweep and remaining competitive around $\tau_{prune} = 0.7$; GPT-4o-mini and Qwen3-14b show wider variation, roughly spanning 45-73% and 55-76%, respectively. This pattern supports the benefit

Table 3: HotpotQA results across MAS topologies.

Method	DeepSeek-v3				GPT-4o-mini				Qwen3-14b			
	Acc. $\uparrow$	PT $\downarrow$	CT $\downarrow$	Time (s) $\downarrow$	Acc. $\uparrow$	PT $\downarrow$	CT $\downarrow$	Time (s) $\downarrow$	Acc. $\uparrow$	PT $\downarrow$	CT $\downarrow$	Time (s) $\downarrow$
FC	0.5450	7943	387	74.35	0.5495	7954	396	56.44	0.5495	13679	1125	50.63
FC+HALT	0.5766	3879	228	32.65	0.5405	3719	203	25.15	0.5586	3708	898	30.10
Net	0.5541	7688	384	75.37	0.5495	7737	407	42.40	0.5631	13369	1153	53.69
Net+HALT	0.5315	2630	172	23.27	0.5450	3692	206	23.47	0.5541	3787	969	31.89
Linear	0.5495	6582	411	77.41	0.5495	6539	389	39.11	0.5631	9743	1443	64.51
Linear+HALT	0.5541	3633	249	36.27	0.5405	3533	205	22.03	0.5676	3911	992	43.73
Tree	0.5586	6682	386	75.91	0.5541	6672	377	49.28	0.5721	10807	1595	63.49
Tree+HALT	0.5676	3547	227	32.00	0.5450	3526	203	23.44	0.5676	3710	1009	46.81
Star	0.5541	6635	378	74.36	0.5676	6649	371	42.66	0.5631	10830	1620	60.84
Star+HALT	0.5631	3475	234	32.96	0.5586	3516	197	25.61	0.5495	3660	1148	43.70

of clause-level filtering over a broad middle range, while also showing why overly aggressive pruning should be avoided for weaker or less calibrated backbones.

The sensitivity analysis shows that HALT’s parameters behave as interpretable deployment controls. τ_{route} should be selected to avoid sending genuinely hard queries to a single agent, τ_{prune} should preserve the task’s necessary evidence while removing redundant or harmful clauses, and (e_{min}, e_{max}) should match the desired latency and accuracy operating point. In deployment, these thresholds are selected on held-out validation data and then fixed before test inference. The advantage is therefore not threshold-free operation, but a stable and explainable calibration procedure that yields useful cost reductions across a wide range of reasonable operating points.

4.6 Cross-Task Analysis

To answer RQ5, we evaluate HALT on HotpotQA, a relatively open multi-hop QA benchmark with verifiable answers. The evaluator is not trained on any HotpotQA samples, so this experiment tests whether its query and clause scoring generalizes to an unseen task format. As shown in Table 3, we report answer accuracy, prompt tokens, completion tokens, and wall-clock runtime under the same backbone and MAS topology.

Efficiency and Runtime. Table 3 shows that the transferred evaluator provides stable efficiency gains on HotpotQA across all tested backbones and MAS topologies. Across the 15 matched Base/+HALT pairs, HALT reduces prompt tokens in every setting by 44.80%–72.89% (55.89% on average), completion tokens by 15.96%–55.21% (39.12% on average), and wall-clock runtime by 26.27%–69.13% (46.39% on average). Accuracy

is topology and backbone dependent rather than uniformly improved: the largest gain is observed for DeepSeek-v3 under FC, from 0.5450 to 0.5766, while the largest drop is observed for DeepSeek-v3 under Net, from 0.5541 to 0.5315.

In response to RQ5, these results show that HALT transfers to relatively open yet answer-verifiable QA as a cost-control mechanism: even without HotpotQA training samples, its evaluator consistently reduces inference cost.

5 Conclusion

This paper presents HALT comprising three core mechanisms: a difficulty aware router for adaptive query dispatch, a lightweight evaluator for clause-level quality assessment and pruning, and a confidence-based halting module for early termination. Extensive experiments in GSM8K, MMLU, and MMLU-Pro demonstrate that HALT achieves the highest accuracy among all evaluated methods in GSM8K and MMLU-Pro, reaching 97.98% and 87.14%, respectively, and remains competitive in MMLU at 90.82%, while cutting prompt tokens by up to 94.32% and completion tokens by up to 99.45%. The HotpotQA cross-task study further shows that the evaluator, without specific training samples, can still guide cost-effective control on relatively open task across multiple backbones and MAS topologies. These results confirm that adaptive, quality-guided control of multi-agent interactions is able to significantly reduce agent costs without compromising task performance. Future work will explore online threshold adaptation, multimodal reasoning, and integration with complementary efficiency methods such as task memory management.

Limitations

Although HALT significantly reduced costs without compromising the performance of the MAS mission, indeed, it actually improved it. We recognize several limitations that open up avenues for future research:

Generalization of the lightweight evaluator:

The foundation of HALT hinges on a local evaluator with 142 million parameters, responsible for spatial pruning and temporal halting. Although empirical evidence indicates its exceptional performance on complex reasoning benchmarks (e.g., GSM8K, MMLU-pro), its capacity is inherently limited by its parameter size and training distribution. In highly specialized domains (e.g., advanced medical diagnostics) or scenarios involving ultra-long context reasoning that surpasses its receptive field, the local evaluator’s ability to accurately allocate credit or identify logical fallacies may diminish, potentially resulting in premature halting or incorrect spatial pruning.

Dependency on heuristic thresholds: As elaborated in the parameter sensitivity analysis, HALT’s performance is closely linked to the selection of purity thresholds and halting bounds (τ_{route} , τ_{prune} , e_{max} , e_{min}). Although the system maintains stable performance within certain parameter ranges, deploying HALT on a completely novel dataset currently necessitates a calibration phase on a validation set to pinpoint the optimal best spot. Developing an adaptive, parameter-free mechanism for dynamically determining these thresholds during inference remains an unresolved challenge.

Applicability to subjective and open-ended tasks: The design of our temporal halting and spatial pruning mechanisms implicitly presupposes the presence of a convergent, objective ground truth (e.g., a mathematical solution or a specific factual answer). For highly subjective, creative, or divergent brainstorming tasks (e.g., creative writing, open-ended policy debate), where achieving agent consensus is not necessarily the objective and diverse "hallucinations" may be interpreted as creativity, HALT’s aggressive pruning and early termination could inadvertently suppress the generative diversity of the MAS.

Ethical Statement

Data and model sources. We evaluate HALT in GSM8K, MMLU, and MMLU-Pro which are publicly available benchmarks released under MIT li-

censes. And HotpotQA is distributed under a CC BY-SA 4.0 License. We use these datasets only for evaluation and do not collect any new human-subject data or personally identifiable information. We access GPT-4o-mini, DeepSeek-V3, Qwen3-14B, and Gemini-3-Pro through their official public APIs under the respective terms of service. The router uses BGE-Large-en-v1.5, and the evaluator is based on DeBERTa-v3-Small; both are released under MIT. We report the accuracy of the final answer together with the counts of prompts and completion tokens to measure both effectiveness and cost.

Use of model-generated supervision. HALT trains its lightweight evaluator using hindsight supervision from an LLM conditioned on ground-truth answers. This supervision is used only to learn clause-level quality estimates for pruning and early halting, rather than to generate new benchmark labels or alter test-set answers. Test sets are strictly isolated from training data for both the router and evaluator.

Potential risks and intended use. HALT is designed to reduce the computational cost of multi-agent reasoning by avoiding unnecessary agent interactions and pruning low-quality clauses. While this can improve efficiency, the framework may still inherit factual errors, biases, or unsafe reasoning patterns from the underlying LLMs. Our experiments are limited to academic reasoning benchmarks, and the method should not be deployed in high-stakes settings without additional domain-specific validation and safety checks.

Environmental and computational impact. The motivation of HALT is to reduce token consumption in multi-agent systems. We report prompt and completion token counts alongside accuracy to make the cost-quality trade-off explicit. Although training the router and evaluator incurs additional offline computation, these components are lightweight compared with repeated multi-agent LLM calls and are intended to amortize cost during inference.

AI Disclosure: We used ChatGPT and Deepseek for language polishing and grammatical correction. All scientific content was authored and verified by the authors.

References

Longju Bai, Zhemin Huang, Xingyao Wang, Jiao Sun, Rada Mihalcea, Erik Brynjolfsson, Alex Pentland,

- and Jiaxin Pei. 2026. [How do AI agents spend your money? analyzing and predicting token consumption in agentic coding tasks](#). *CoRR*, abs/2604.22750.
- Guangsheng Bao, Hongbo Zhang, Cunxiang Wang, Linyi Yang, and Yue Zhang. 2025. [How likely do llms with cot mimic human reasoning?](#) In *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025*, pages 7831–7850. Association for Computational Linguistics.
- Justin Chih-Yao Chen, Swarnadeep Saha, and Mohit Bansal. 2024. [Reconcile: Round-table conference improves reasoning via consensus among diverse llms](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024*, pages 7066–7085. Association for Computational Linguistics.
- Yu-Neng Chuang, Tianwei Xing, Chia-Yuan Chang, Zirui Liu, Xun Chen, and Xia Ben Hu. 2024. [Learning to compress prompt in natural language formats](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024*, pages 7756–7767. Association for Computational Linguistics.
- 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. 2021. [Training verifiers to solve math word problems](#). *CoRR*, abs/2110.14168.
- Roi Cohen, May Hamri, Mor Geva, and Amir Globerson. 2023. [LM vs LM: detecting factual errors via cross examination](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023*, pages 12621–12640. Association for Computational Linguistics.
- Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2024. [Improving factuality and reasoning in language models through multiagent debate](#). In *Forty-first International Conference on Machine Learning, ICML 2024*, Proceedings of Machine Learning Research, pages 11733–11763. PMLR / OpenReview.net.
- Mostafa Elhoushi, Akshat Shrivastava, Diana Liskovich, Basil Hosmer, Bram Wasti, Liangzhen Lai, Anas Mahmoud, Bilge Acun, Saurabh Agarwal, Ahmed Roman, Ahmed A Aly, Beidi Chen, and Carole-Jean Wu. 2024. [Layerskip: Enabling early exit inference and self-speculative decoding](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024*, pages 12622–12642. Association for Computational Linguistics.
- Zhizhang Fu, Guangsheng Bao, Hongbo Zhang, Chenkai Hu, and Yue Zhang. 2025. [Correlation or causation: Analyzing the causal structures of LLM and LRM reasoning process](#). *CoRR*, abs/2509.17380.
- In Gim, Guojun Chen, Seung-Seob Lee, Nikhil Sarda, Anurag Khandelwal, and Lin Zhong. 2024. [Prompt cache: Modular attention reuse for low-latency inference](#). In *Proceedings of the Seventh Annual Conference on Machine Learning and Systems, MLSys 2024*. mlsys.org.
- Google DeepMind. 2025. Gemini 3. <https://deepmind.google/models/gemini/>. Accessed: 2026.
- Shurui Gui and Shuiwang Ji. 2025. [Mitigating spurious correlations in llms via causality-aware post-training](#). *CoRR*, abs/2506.09433.
- Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. [Measuring massive multitask language understanding](#). In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.
- Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiwu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. 2024. [Metagpt: Meta programming for A multi-agent collaborative framework](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024*. OpenReview.net.
- Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023a. [Llm-blender: Ensembling large language models with pairwise ranking and generative fusion](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023*, pages 14165–14178. Association for Computational Linguistics.
- Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023b. [Llmllingua: Compressing prompts for accelerated inference of large language models](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023*, pages 13358–13376. Association for Computational Linguistics.
- Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2024. [Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024*, pages 1658–1677. Association for Computational Linguistics.
- Sehoon Kim, Coleman Hooper, Amir Gholami, Zhen Dong, Xiuyu Li, Sheng Shen, Michael W. Mahoney,

- and Kurt Keutzer. 2024. [Squeezellm: Dense-and-sparse quantization](#). In *Forty-first International Conference on Machine Learning, ICML 2024*, Proceedings of Machine Learning Research, pages 23901–23923. PMLR / OpenReview.net.
- Cheryl Lee, Chunqiu Steven Xia, Longji Yang, Jentse Huang, Zhouruixin Zhu, Lingming Zhang, and Michael R. Lyu. 2025. [Unidebugger: Hierarchical multi-agent framework for unified software debugging](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pages 18237–18266. Association for Computational Linguistics.
- Boyi Li, Zhonghan Zhao, Der-Horng Lee, and Gaoang Wang. 2025. [Adaptive graph pruning for multi-agent communication](#). In *ECAI 2025 - 28th European Conference on Artificial Intelligence, 25-30 October 2025, Bologna, Italy - Including 14th Conference on Prestigious Applications of Intelligent Systems (PAIS 2025)*, Frontiers in Artificial Intelligence and Applications, pages 4305–4312. IOS Press.
- Junyou Li, Qin Zhang, Yangbin Yu, Qiang Fu, and Dengheng Ye. 2024. [More agents is all you need](#). *Transactions on Machine Learning Research*, 2024.
- Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. 2023. [Compressing context to enhance inference efficiency of large language models](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023*, pages 6342–6353. Association for Computational Linguistics.
- Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. 2024a. [Encouraging divergent thinking in large language models through multi-agent debate](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024*, pages 17889–17904. Association for Computational Linguistics.
- Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. 2024b. [Encouraging divergent thinking in large language models through multi-agent debate](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024*, pages 17889–17904. Association for Computational Linguistics.
- Bill Yuchen Lin, Yicheng Fu, Karina Yang, Faeze Brahman, Shiyu Huang, Chandra Bhagavatula, Prithviraj Ammanabrolu, Yejin Choi, and Xiang Ren. 2023. Swiftsage: A generative agent with fast and slow thinking for complex interactive tasks. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023*.
- Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024a. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.
- Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024b. [Lost in the middle: How language models use long contexts](#). *Transactions of the Association for Computational Linguistics*, 12:157–173.
- Xuefei Ning, Zinan Lin, Zixuan Zhou, Zifu Wang, Huazhong Yang, and Yu Wang. 2024. [Skeleton-of-thought: Prompting llms for efficient parallel generation](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024*. OpenReview.net.
- GPT OpenAI. 2024. Gpt-4o mini: advancing cost-efficient intelligence. <https://openai.com>. Accessed: 2026.
- Charles Packer, Vivian Fang, Shishir G. Patil, Kevin Lin, Sarah Wooders, and Joseph E. Gonzalez. 2023. [Memgpt: Towards llms as operating systems](#). *CoRR*, abs/2310.08560.
- Liangming Pan, Alon Albalak, Xinyi Wang, and William Yang Wang. 2023. [Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, Findings of ACL, pages 3806–3824. Association for Computational Linguistics.
- Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. 2011. [Scikit-learn: Machine learning in python](#). *The Journal of Machine Learning Research*, 12:2825–2830.
- Jingyu Peng, Maolin Wang, Xiangyu Zhao, Kai Zhang, Wanyu Wang, Pengyue Jia, Qidong Liu, Ruocheng Guo, and Qi Liu. 2024. [Stepwise reasoning error disruption attack of llms](#). *CoRR*, abs/2412.11934.
- Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2024. [Chatdev: Communicative agents for software development](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024*, pages 15174–15186. Association for Computational Linguistics.
- Syed Rifat Raiyan, Md Farhan Ishmam, Abdullah Al Imran, and Mohammad Ali Moni. 2025. [Frugalprompt: Reducing contextual overhead in large language models via token attribution](#). *CoRR*, abs/2510.16439.
- Marija Sakota, Maxime Peyrard, and Robert West. 2024. [Fly-swat or cannon? cost-effective language model choice via meta-modeling](#). In *Proceedings of the*

- 17th ACM International Conference on Web Search and Data Mining, WSDM 2024, pages 606–615. ACM.
- Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. [Reflexion: language agents with verbal reinforcement learning](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023*.
- Kumar Shridhar, Alessandro Stolfo, and Mrinmaya Sachan. 2023. [Distilling reasoning capabilities into smaller language models](#). In *Findings of the Association for Computational Linguistics: ACL 2023, Findings of ACL*, pages 7059–7073. Association for Computational Linguistics.
- Haoyang Su, Renqi Chen, Shixiang Tang, Zhenfei Yin, Xinzhe Zheng, Jinzhe Li, Biqing Qi, Qi Wu, Hui Li, Wanli Ouyang, Philip Torr, Bowen Zhou, and Nanqing Dong. 2025. [Many heads are better than one: Improved scientific idea generation by A llm-based multi-agent system](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 28201–28240. Association for Computational Linguistics.
- Hao Tang, Feng Zhang, Kejia Wan, Songyi Lu, Xi Zhang, Yiping Yao, and Xinhai Xu. 2026. Veritas: Structuring and verifying llm knowledge for logically consistent behavior tree generation in llm-based agents. *Information Processing & Management*, page 104830.
- Dongzi Wang, Fangwei Zhong, Minglong Li, Muning Wen, Yuanxi Peng, Teng Li, and Adam Yang. 2024a. [Romat: Role-based multi-agent transformer for generalizable heterogeneous cooperation](#). *Neural Networks*, 174:106129.
- Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Zou. 2025a. [Mixture-of-agents enhances large language model capabilities](#). In *The Thirteenth International Conference on Learning Representations, ICLR 2025*. OpenReview.net.
- Xiao Wang, Jia Wang, Yijie Wang, Pengtao Dang, Sha Cao, and Chi Zhang. 2025b. [MARS: toward more efficient multi-agent collaboration for LLM reasoning](#). *CoRR*, abs/2509.20502.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. [Self-consistency improves chain of thought reasoning in language models](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023*. OpenReview.net.
- Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyang Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhui Chen. 2024b. [Mmlu-pro: A more robust and challenging multi-task language understanding benchmark](#). In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.
- Zhenhailong Wang, Shaoguang Mao, Wenshan Wu, Tao Ge, Furu Wei, and Heng Ji. 2024c. [Unleashing the emergent cognitive synergy in large language models: A task-solving agent through multi-persona self-collaboration](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024*, pages 257–279. Association for Computational Linguistics.
- Zhexuan Wang, Yutong Wang, Xuebo Liu, Liang Ding, Miao Zhang, Jie Liu, and Min Zhang. 2025c. [Agentdropout: Dynamic agent elimination for token-efficient and high-performance llm-based multi-agent collaboration](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025*, pages 24013–24035. Association for Computational Linguistics.
- Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. [Efficient streaming language models with attention sinks](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024*. OpenReview.net.
- Yizhe Xie, Congcong Zhu, Xinyue Zhang, Tianqing Zhu, Dayong Ye, Minfeng Qi, Huajie Chen, and Wanlei Zhou. 2026. [From spark to fire: Modeling and mitigating error cascades in llm-based multi-agent collaboration](#). *CoRR*, abs/2603.04474.
- Kai Xiong, Xiao Ding, Yixin Cao, Ting Liu, and Bing Qin. 2023. [Examining inter-consistency of large language models collaboration: An in-depth analysis via debate](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023, Findings of ACL*, pages 7572–7590. Association for Computational Linguistics.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. [HotpotQA: A dataset for diverse, explainable multi-hop question answering](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.
- Xie Yi, Zhanke Zhou, Chentao Cao, Qiyu Niu, Tongliang Liu, and Bo Han. 2025. [From debate](#)

to equilibrium: Belief-driven multi-agent LLM reasoning via bayesian nash equilibrium. In *Forty-second International Conference on Machine Learning, ICML 2025, Proceedings of Machine Learning Research*. PMLR / OpenReview.net.

Yurun Yuan and Tengyang Xie. 2025. Reinforce LLM reasoning through multi-agent reflection. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025, Proceedings of Machine Learning Research*. PMLR / OpenReview.net.

Guibin Zhang, Yanwei Yue, Zhixun Li, Sukwon Yun, Guancheng Wan, Kun Wang, Dawei Cheng, Jeffrey Xu Yu, and Tianlong Chen. 2025a. Cut the crap: An economical communication pipeline for llm-based multi-agent systems. In *The Thirteenth International Conference on Learning Representations, ICLR 2025*. OpenReview.net.

Guibin Zhang, Yanwei Yue, Xiangguo Sun, Guancheng Wan, Miao Yu, Junfeng Fang, Kun Wang, Tianlong Chen, and Dawei Cheng. 2025b. G-designer: Architecting multi-agent communication topologies via graph neural networks. In *Forty-second International Conference on Machine Learning, ICML 2025, Proceedings of Machine Learning Research*. PMLR / OpenReview.net.

Jieyu Zhang, Ranjay Krishna, Ahmed Hassan Awadallah, and Chi Wang. 2023. Ecoassistant: Using LLM assistant more affordably and accurately. *CoRR*, abs/2310.03046.

Shaokun Zhang, Ming Yin, Jieyu Zhang, Jiale Liu, Zhiguang Han, Jingyang Zhang, Beibin Li, Chi Wang, Huazheng Wang, Yiran Chen, and Qingyun Wu. 2025c. Which agent causes task failures and when? on automated failure attribution of LLM multi-agent systems. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025, Proceedings of Machine Learning Research*. PMLR / OpenReview.net.

Yusen Zhang, Ruoxi Sun, Yanfei Chen, Tomas Pfister, Rui Zhang, and Sercan Ö. Arik. 2024. Chain of agents: Large language models collaborating on long-context tasks. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024*.

Heng Zhou, Hejia Geng, Xiangyuan Xue, Li Kang, Yiran Qin, Zhiyong Wang, Zhenfei Yin, and Lei Bai. 2025. Reso: A reward-driven self-organizing llm-based multi-agent system for reasoning tasks. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pages 15979–15998. Association for Computational Linguistics.

A Algorithm of HALT

This section presents the algorithmic flow of HALT, as shown in Algorithm 1.

Algorithm 1 HALT

Input: Query x , Difficulty router $\mathcal{H}_\phi$, Evaluator $\mathcal{H}_\theta$, Agents $\mathcal{V}$, Max rounds T

Parameters: $\tau_{route} = 0.45$, $\tau_{prune} = 0.45$, $e_{max} = 0.85$, $e_{min} = 0.45$

Output: Final Answer y^*

```

1:  $\pi_{route}(x) \leftarrow \mathcal{H}_\phi(x)$ 
2: if  $\pi_{route}(x) < \tau_{route}$  then
3:   return  $\mathcal{M}_{base}(x)$ 
4: end if
5:  $s_0 \leftarrow x$ 
6: for  $t = 1$  to  $T$  do
7:    $\mathcal{O}_t \leftarrow \text{Generate\_Responses}(\mathcal{V}, s_{t-1})$ 
8:    $\mathcal{E}_t \leftarrow \text{SPMP\_Pipeline}(\mathcal{O}_t)$ 
9:    $\mathcal{E}_{retain} \leftarrow \emptyset$ 
10:  for each  $e_j \in \mathcal{E}_t$  do
11:     $z_j \leftarrow \mathcal{H}_\theta(e_j | x)$ 
12:    if  $z_j \geq \tau_{prune}$  then
13:       $\mathcal{E}_{retain} \leftarrow \mathcal{E}_{retain} \cup \{e_j\}$ 
14:    end if
15:  end for
16:   $s_t \leftarrow \text{Concat}(s_{t-1}, \mathcal{E}_{retain})$ 
17:   $\hat{y}_t \leftarrow \text{Extract\_Answer}(s_t)$ 
18:   $C_{conf} \leftarrow \max(z_j) \geq e_{max}$ 
19:   $C_{qual} \leftarrow \min(z_j) \geq e_{min}$ 
20:  if  $\hat{y}_t$  &  $C_{conf}$  &  $C_{qual}$  then
21:    break
22:  end if
23: end for
24: return  $\text{Extract\_Answer}(s_t)$ 

```

B Prompts

The prompts utilized in this paper for scoring the LLM during training and evaluation are depicted as shown in Figure 5(a), offering a comprehensive overview of the task requirements, scoring rationale, and illustrative examples. In Figure 5(b), the identity role configurations employed by MAS, along with the prompts associated with each role, are presented.

Figure 6 presents two alternative prompt templates for constructing evaluator supervision under different segmentation granularities. The sentence-level template in Figure 6(a) treats each complete sentence as a single scoring unit, while the

(a) LLM scoring prompt for training evaluator		
You are an impartial, rigorous logic arbiter with "oracle/hindsight" knowledge. Your task is to analyze a multi-turn AI interaction transcript at an EXTREMELY FINE-GRAINED (split by commas/periods) level. [Evaluation Task] Knowing the ultimate Ground Truth, evaluate the logical value of EACH segmented clause. You MUST assign strictly one of these five discrete scores (0.0, 0.25, 0.5, 0.75, 1.0): - 1.0 (Crucial): Vital correct formula, decisive breakthrough, or fatal error correction. - 0.75 (Helpful): A correct intermediate reasoning step or parameter extraction that advances the solution. - 0.5 (Neutral): Transitional phrases (e.g., "Wait,", "Let's see,"), conversational fillers, or harmless restatements. - 0.25 (Distracting): Irrelevant rambling or unnecessary complex hypotheses that clutter the context. - 0.0 (Toxic/Fatal): Hallucinated physics, math calculation errors, or entirely wrong directions that contradict the Ground Truth. [Few-Shot Examples for Calibration] Problem: A 10kg block is on a frictionless surface. A 50N force is applied. What is the acceleration? (Ground Truth: 5 m/s²) Transcript: [T1_S0]: Let's break down this problem, -> Score: 0.5 (Rationale: Neutral transitional phrase) [T1_S1]: we should use Newton's second law, -> Score: 0.75 (Rationale: Correct principle identified) [T1_S2]: $F = ma$. -> Score: 1.0 (Rationale: Crucial formula stated) [T1_S3]: However, -> Score: 0.5 (Rationale: Neutral conversational filler) [T1_S4]: since there is friction, -> Score: 0.0 (Rationale: Toxic hallucination, contradicts "frictionless") [T1_S5]: we need the coefficient first. -> Score: 0.0 (Rationale: Toxic continuation of the wrong assumption) [T2_S0]: Wait, -> Score: 0.5 (Rationale: Neutral transition) [T2_S1]: the problem explicitly states the surface is frictionless! -> Score: 1.0 (Rationale: Crucial correction of a fatal error)		
(b) Role prompts for MAS		
Role: Explorer  Role prompt: You are the Explorer. Based on the problem and current discussion, propose the initial step-by-step reasoning. Break down the core principles and formulas. Do not jump to conclusions hastily.	Role: Critic  Role prompt: You are the Critic. Your job is to strictly scrutinize the Explorer's reasoning. Find logical loopholes, hallucinated physics formulas, or missing conditions. Correct any errors aggressively.	Role: Lateral Thinker  Role prompt: You are the Lateral Thinker. Provide an alternative perspective or a different method to solve the problem to cross-verify the previous steps. (e.g., if they used kinematics, try energy conservation; if algebra, try geometry).
Role: Verifier  Role prompt: You are the Verifier. Focus strictly on mathematical execution, dimensional analysis, and unit consistency. Recalculate everything proposed by the previous agents step-by-step to ensure absolute numerical precision.	Role: Synthesizer  Role prompt: You are the Synthesizer. Review all previous arguments, conflicts, and calculations. Resolve any remaining disagreements and push the derivation forward toward a unified, flawless, and undeniable truth.	Role: Decider  Role prompt: [Task] Provide an answer based on the question. [Interaction History] [Instruction] Please output ONLY the correct answer.

Figure 5: The prompts used in HALT.

semantic-proposition template in Figure 6(b) decomposes a transcript into minimal, independently verifiable claims. Both templates retain the same hindsight setting, access to the verified answer during offline labeling, a five-point ordinal scoring scale, and the few-shot calibration structure as the default scorer. These prompts provide guidelines at various levels of detail for training optional evaluator criteria.

C Pruning Visualization Example

To illustrate the effect of our clause-level pruning mechanism, we present a concrete example from the MMLU-pro dataset. Table 4 shows a side-by-side comparison of the raw agent output and the pruned version after applying our evaluator with threshold $\tau_{route} = 0.5$, $\tau_{prune} = 0.6$, $e_{max} = 0.85$, $s_{min} = 0.45$.

(a) Sentence-level segmentation prompt
You are an impartial, rigorous logic arbiter with "oracle/hindsight" knowledge. Your task is to analyze a multi-turn AI interaction transcript at the SENTENCE level. [Evaluation Task] Knowing the ultimate Ground Truth, evaluate the logical value of EACH sentence level segment. You MUST assign strictly one of these five discrete scores (0.0, 0.25, 0.5, 0.75, 1.0): - 1.0 (Crucial): Vital correct formula, decisive breakthrough, final correct inference, or fatal error correction. - 0.75 (Helpful): A correct intermediate reasoning step or parameter extraction that advances the solution. - 0.5 (Neutral): Transitional phrases, harmless restatements, or procedural descriptions with little logical contribution. - 0.25 (Distracting): Irrelevant rambling or unnecessary hypotheses that clutter the context. - 0.0 (Toxic/Fatal): Hallucinated facts, math errors, or wrong directions that contradict the Ground Truth. [Few-Shot Examples for Calibration] Problem: A 10kg block is on a frictionless surface. A 50N force is applied. What is the acceleration? (Ground Truth: 5 m/s²) Transcript: [T1_S0]: Let's break down this problem, we should use Newton's second law, $F = ma$. -> Score: 1.0 (Rationale: The sentence contains the decisive correct formula, despite a neutral opening phrase.) [T1_S1]: However, since there is friction, we need the coefficient first. -> Score: 0.0 (Rationale: The sentence introduces a false friction assumption that contradicts the problem.) [T2_S0]: Wait, the problem explicitly states the surface is frictionless! -> Score: 1.0 (Rationale: The sentence corrects a fatal error.)
(b) Semantic-proposition segmentation prompt
You are an impartial, rigorous logic arbiter with "oracle/hindsight" knowledge. Your task is to analyze a multi-turn AI interaction transcript at the SEMANTIC-PROPOSITION level. [Evaluation Task] Knowing the ultimate Ground Truth, evaluate the logical value of EACH segmented clause. You MUST assign strictly one of these five discrete scores (0.0, 0.25, 0.5, 0.75, 1.0): - 1.0 (Crucial): Vital correct formula, decisive breakthrough, or fatal error correction. - 0.75 (Helpful): A correct intermediate reasoning step or parameter extraction that advances the solution. - 0.5 (Neutral): Transitional phrases, harmless restatements, or procedural descriptions with little logical contribution. - 0.25 (Distracting): Irrelevant rambling or unnecessary complex hypotheses that clutter the context. - 0.0 (Toxic/Fatal): Hallucinated physics, math calculation errors, or entirely wrong directions that contradict the Ground Truth. [Few-Shot Examples for Calibration] Problem: A 10kg block is on a frictionless surface. A 50N force is applied. What is the acceleration? (Ground Truth: 5 m/s²) Transcript: [T1_S0]: Let's break down this problem. -> Score: 0.5 (Rationale: Neutral procedural statement.) [T1_S1]: We should use Newton's second law. -> Score: 0.75 (Rationale: Correct principle identified.) [T1_S2]: $F = ma$. -> Score: 1.0 (Rationale: Crucial formula for solving the problem.) [T1_S3]: The surface has friction. -> Score: 0.0 (Rationale: False assumption contradicting the problem statement.) [T1_S4]: We need the coefficient of friction first. -> Score: 0.0 (Rationale: Wrong direction caused by the false friction assumption.) [T2_S0]: The problem explicitly states the surface is frictionless. -> Score: 1.0 (Rationale: Crucial correction of the fatal assumption.)

Figure 6: Alternative segmentation prompts for offline evaluator-supervision construction. (a) uses sentence-level segmentation, whereas (b) uses semantic-proposition segmentation.

C.1 Analysis of Pruning Effects

Table 5 quantifies the compression achieved through spatial pruning on this example.

C.2 Clause-Level Score Distribution

Tabel 5 summarizes the evaluator’s score distribution across clauses in the raw output. Clauses with scores below $\tau_{\text{prune}} = 0.6$ (shaded region) are filtered out during pruning.

- **Low-score clauses** (< 0.6): Section headers ('Step 1', 'Step 2'), meta-commentary ('Alright, let's break this down'), transitional

phrases ('Thus', 'So')

- **High-score clauses** (≥ 0.6): Mathematical definitions, formal proofs, and concluding statements

This example demonstrates that our clause-level pruning effectively removes conversational filler content while preserving the logical structure and mathematical rigor of the reasoning process.

D Actual Cost

Offline supervision cost: Controller labels are constructed once before deployment, so we separately

Table 4: Comparison of raw agent output (left) and pruned output (right) after applying clause-level filtering with $\tau_{route} = 0.5$, $\tau_{prune} = 0.6$, $e_{max} = 0.85$, $s_{min} = 0.45$. Pruned content includes meta-commentary ('Alright, let's break this down'), section headers ('Step X'), and redundant explanations.

Raw Output (Before Pruning)	Pruned Output (After Pruning)
<i>Question: Find the characteristic of the ring $2\mathbb{Z}$.</i> Alright, let's break this down step by step. Step 1: Understanding the question The question says: Find the characteristic of the ring $2\mathbb{Z}$. Here $2\mathbb{Z}$ means the set of even integers: $2\mathbb{Z} = \{\dots, -4, -2, 0, 2, 4, \dots\}.$ It is a ring under the usual addition and multiplication of integers (but without multiplicative identity 1). Step 2: Definition of characteristic For a ring R (with unity 1_R), the characteristic is the smallest positive integer n such that $n \cdot 1_R = 0_R$ if such n exists; otherwise, the characteristic is 0. If the ring does not have a multiplicative identity (like $2\mathbb{Z}$), we define the characteristic as the smallest positive integer n such that $n \cdot a = 0$ for all $a \in R$. If no such n exists, the characteristic is 0. Step 3: Apply to $2\mathbb{Z}$ Take any element $a \in 2\mathbb{Z}$, say $a = 2m$ for some $m \in \mathbb{Z}$. We ask: Is there a positive integer n such that $n \cdot a = 0$ for all $a \in 2\mathbb{Z}$? That means: $n \cdot (2m) = 0 \quad \text{for all integers } m.$ But $n \cdot (2m) = 2mn$. For this to be 0 for all m, we would need $2n \cdot m = 0$ for all m, which is impossible unless $2n = 0 \Rightarrow n = 0$. So no positive n works. Thus, the characteristic is 0. Step 4: Check possible confusion Sometimes students confuse $2\mathbb{Z}$ with $\mathbb{Z}_2$ or $\mathbb{Z}/2\mathbb{Z}$, but $2\mathbb{Z}$ is infinite and has no torsion in its additive group except 0 itself. The additive group $(2\mathbb{Z}, +)$ is isomorphic to $\mathbb{Z}$, and $\mathbb{Z}$ has characteristic 0. Step 5: Conclusion The characteristic of $2\mathbb{Z}$ is 0. A	Find the characteristic of the ring $2\mathbb{Z}$. Here $2\mathbb{Z}$ means the set of even integers: $2\mathbb{Z} = \{\dots, -4, -2, 0, 2, 4, \dots\}$. It is a ring under the usual addition and multiplication of integers (but without multiplicative identity 1). For a ring R (with unity 1_R), the characteristic is the smallest positive integer n such that $n \cdot 1_R = 0_R$ if such n exists; the characteristic is 0. If the ring does not have a multiplicative identity (like $2\mathbb{Z}$), we define the characteristic as the smallest positive integer n such that $n \cdot a = 0$ for all $a \in R$. the characteristic is 0. Take any element $a \in 2\mathbb{Z}$, say $a = 2m$ for some $m \in \mathbb{Z}$. We ask: Is there a positive integer n such that $n \cdot a = 0$ for all $a \in 2\mathbb{Z}$? $n \cdot (2m) = 0$ for all integers m. But $n \cdot (2m) = 2mn$. For this to be 0 for all m, we would need $2n \cdot m = 0$ for all m, which is impossible unless $2n = 0 \Rightarrow n = 0$. So no positive n works. the characteristic is 0. but $2\mathbb{Z}$ is infinite and has no torsion in its additive group except 0 itself. The additive group $(2\mathbb{Z}, +)$ is isomorphic to $\mathbb{Z}$, and $\mathbb{Z}$ has characteristic 0. The characteristic of $2\mathbb{Z}$ is 0. A

Table 5: Compression statistics for the pruning example in Table 4.

Metric	Raw	Pruned
Character count	1,847	987
Token count	412	221
Compression ratio	–	46.4%
Clauses removed	–	12
Clauses retained	23	11

report their token cost instead of mixing this cost with inference-time savings. Router label cost is measured across GSM8K, MMLU, and MMLU-

Pro, while evaluator supervision cost is measured from scored MAS trajectory records. Table 6 shows that router label construction is lightweight, averaging 155 prompt tokens and 104 completion tokens per labeled example. Evaluator supervision is more expensive because it is built from MAS trajectories, averaging 25,301 prompt tokens and 3,959 completion tokens per example. This accounting makes the data-construction overhead explicit while preserving a clean distinction between offline calibration cost and online inference cost.

To complement the theoretical token reduction analyzed in Section 4.2, this section bridges the gap between empirical token counts and real-world

Table 6: Offline token cost for constructing controller supervision with GPT-4o-mini.

Supervision	Average PT	Average CT
Router	155	104
Evaluator	25301	3959

financial implications. The inclusion of diverse pricing structures (Official and two third-party tiers, as shown in Table 7, measured in USD) is crucial to evaluating the robustness of the system, as the LLM API costs are subject to significant market volatility.

Disproportionate financial impact of CT: A critical observation from Table 7 is that the pricing for CT is consistently and strictly higher than PT across all models—frequently by a multiplier of 2 times to 4 times. Consequently, multi-agent frameworks that fail to constrain agent generation are heavily penalized economically. HALT’s temporal halting mechanism directly targets this vulnerability. As detailed in Table 8, under Pricing 1 for GPT-4o-mini on the GSM8K dataset, the Fully-Connected (FC) baseline incurs a CT cost of \$17.42, whereas HALT aggressively reduces this to a mere \$0.93. By intercepting over-generation early, HALT neutralizes the most expensive dimension of API billing.

Extreme cost mitigation in volatile markets: Table 9 demonstrates that HALT acts as a robust economic buffer against severe API price hikes. Consider the DeepSeek-v3 model under the highly inflated "Third-party price 3" tier on the GSM8K dataset. The baseline FC network spirals out of control, accumulating a prohibitive CT cost of \$98.02 and a PT cost of \$11.61 (Total: \$109.63 per benchmark run). In stark contrast, HALT intercepts this runaway reasoning, maintaining a CT cost of only \$1.10 and a PT cost of \$1.28 (Total: \$2.38). This represents a remarkable 97.8% reduction in actual financial expenditure, proving that unbounded agent interactions are financially inviable in inflated markets without HALT’s intervention.

Comparative economic superiority over MAS optimizers: Even when benchmarked against competitive multi-agent optimizers, HALT maintains a definitive economic edge. As detailed in Table 10, examining the Qwen3-14b model under Pricing 3 on the MMLU dataset reveals that GD incurs a total API cost of \$38.60 (\$2.42 PT + \$36.18 CT). HALT successfully executes the complex reason-

ing task for a fraction of the price, totaling only \$1.65 (\$0.71 PT + \$0.94 CT). This immense disparity highlights that spatial-only or topology-only optimizations (like GD and AP) still fall victim to output token bloat. HALT’s integrated three-tier approach is therefore uniquely positioned to enable the deployment of sustainable, commercial-grade MAS at scale.

Table 7: Different prices for per 1M tokens LLMs.

Model	GPT-4o-mini		DeepSeek-v3		Qwen3-14b	
	Input	Output	Input	Output	Input	Output
Official price	0.15	0.6	0.28	0.42	0.35	1.4
Third-party price 1	0.15	0.6	0.26	0.38	0.06	0.24
Third-party price 2	0.09	0.36	2	3	1	4

E Relevant Supplementary Experiments

E.1 Runtime and Controller Overhead

Token reduction does not automatically imply end-to-end speedup, because the local router and evaluator introduce additional computation. We therefore measure wall-clock latency, API time, local controller time, API calls, and token usage on MMLU-Pro with GPT-4o-mini under the Star topology. The study is intended to characterize runtime overhead.

End-to-end runtime: Table 11 shows that HALT’s token savings translate into practical latency reduction in the measured setting. Compared with full MAS execution, HALT reduces mean wall-clock time from 166.31s to 34.20s per query, while reducing API calls from 11.00 to 2.15. The wall-clock reduction of 79.44% closely tracks the 80.06% API-time reduction, indicating that most of the speedup comes from avoiding remote LLM calls rather than from local post-processing artifacts.

Local controller overhead: The router and evaluator add 1.02s of local controller time per query, which accounts for only 2.98% of HALT’s wall-clock runtime and 0.61% of the FC wall-clock runtime. This overhead is small relative to the latency saved by reducing API calls, prompt tokens, and completion tokens.

E.2 Cross-Dataset Router Adaptation

We further evaluate whether routing requires one separate parameterization for every dataset. We construct a class-balanced router dataset whenever possible: GSM8K, MMLU and MMLU-Pro. Since the positive class denotes queries that require

Table 8: Detailed API costs for different methods with GPT-4o-mini.

Prices	Datasets	GSM8K		MMLU		MMLU-pro	
	Method	PT	CT	PT	CT	PT	CT
Official price	FC	0.78	17.42	0.24	5.31	0.06	1.42
	AP	0.75	13.50	0.12	1.79	0.07	1.01
	AD	0.82	14.30	0.16	1.90	0.05	0.66
	GD	0.41	7.16	0.10	1.30	0.07	1.18
	HALT	0.13	0.93	0.04	0.54	0.02	0.31
Third-party 1	FC	0.78	17.42	0.24	5.31	0.06	1.42
	AP	0.75	13.50	0.12	1.79	0.07	1.01
	AD	0.82	14.30	0.16	1.90	0.05	0.66
	GD	0.41	7.16	0.10	1.30	0.07	1.18
	HALT	0.13	0.93	0.04	0.54	0.02	0.31
Third-party 2	FC	0.47	10.45	0.14	3.19	0.04	0.85
	AP	0.45	8.10	0.07	1.07	0.04	0.61
	AD	0.49	8.58	0.09	1.14	0.03	0.39
	GD	0.25	4.30	0.06	0.78	0.04	0.71
	HALT	0.08	0.56	0.02	0.32	0.01	0.19

Table 9: Detailed API costs for different methods with DeepSeek-v3.

Prices	Datasets	GSM8K		MMLU		MMLU-pro	
	Method	PT	CT	PT	CT	PT	CT
Official price	FC	1.63	13.72	0.59	4.82	0.15	1.32
	AP	0.97	8.04	0.31	1.58	0.19	0.96
	AD	1.09	7.75	0.35	1.56	0.13	0.62
	GD	0.58	4.56	0.25	1.15	0.18	1.15
	HALT	0.18	0.15	0.03	0.21	0.02	0.05
Third-party 1	FC	1.51	12.42	0.55	4.36	0.14	1.19
	AP	0.91	7.27	0.29	1.43	0.18	0.87
	AD	1.01	7.02	0.33	1.41	0.12	0.56
	GD	0.54	4.13	0.23	1.04	0.17	1.04
	HALT	0.17	0.14	0.03	0.19	0.02	0.04
Third-party 2	FC	11.61	98.02	4.22	34.45	1.04	9.43
	AP	6.96	57.42	2.23	11.28	1.35	6.88
	AD	7.77	55.38	2.52	11.12	0.95	4.42
	GD	4.12	32.57	1.76	8.23	1.32	8.21
	HALT	1.28	1.10	0.24	1.52	0.17	0.34

Table 10: Detailed API costs for different methods with Qwen3-14b.

Prices	Datasets	GSM8K		MMLU		MMLU-pro	
	Method	PT	CT	PT	CT	PT	CT
Official price	FC	1.48	34.75	0.86	18.29	0.22	4.98
	AP	1.19	29.00	0.82	10.24	0.30	4.22
	AD	1.05	23.37	0.55	6.06	0.23	3.57
	GD	1.40	36.56	0.85	12.66	0.27	4.17
	HALT	0.11	0.19	0.25	0.33	0.09	1.79
Third-party 1	FC	0.25	5.96	0.15	3.14	0.04	0.85
	AP	0.20	4.97	0.14	1.75	0.05	0.72
	AD	0.18	4.01	0.09	1.04	0.04	0.61
	GD	0.24	6.27	0.14	2.17	0.05	0.71
	HALT	0.02	0.03	0.04	0.06	0.02	0.31
Third-party 2	FC	4.24	99.28	2.47	52.26	0.64	14.23
	AP	3.39	82.84	2.36	29.25	0.87	12.05
	AD	3.00	66.78	1.57	17.30	0.66	10.19
	GD	4.00	104.44	2.42	36.18	0.77	11.91
	HALT	0.30	0.54	0.71	0.94	0.27	5.10

MAS, we select operating points by validation F1. Figure 7 shows that dataset-specific routers are

strongest on their own source datasets, as expected, while the fused router reaches the best average F1 of 0.6776. This result supports a practical deployment claim: mixed validation data can provide a shared router with nontrivial cross-dataset transfer, although dataset-specific calibration remains useful when sufficient validation data are available.

E.3 The Effectiveness of HALT in Different MAS

E.3.1 Different Numbers of MAS

A central concern for any MAS optimization method is whether its effectiveness is contingent on a specific system scale. To verify that HALT generalizes across different multi-agent configurations, we evaluate HALT under varying agent counts. If HALT were only effective at a particular scale, its practical utility would be severely limited; conversely, consistent effectiveness across scales would demonstrate that the method captures scale-invariant principles of multi-agent coordination.

Table 12 presents the full results. We analyze the behavior of each HALT module under varying system scales.

The router operates consistently regardless of system scale. A well-designed routing module should allocate queries based on inherent problem difficulty rather than the available agent pool. Table 12 confirms this: the MAS proportion remains strictly invariant across all agent counts for every backbone and dataset pair. For example, GPT-4o-mini routes 79.80% of GSM8K queries to MAS regardless of whether two agents or five agents. This stability is by design, the router evaluates query complexity before observing the agent pool, and it ensures that the routing policy generalizes without retraining when the system is reconfigured. The cross-backbone variation further validates the router’s discriminative ability: on GSM8K, DeepSeek-v3 routes only 43.35% to MAS (correctly identifying that a strong backbone needs less collaboration), while Qwen3-14b routes 87.19% (appropriately relying on multi-agent deliberation to compensate for weaker single-agent reasoning).

The halting module adapts its trigger rate to agent density, correctly identifying redundant computation. As the number of agents increases, the halting proportion generally rises, indicating that the evaluator detects earlier convergence when more agents participate. This is the expected behav-

Table 11: End-to-end Wall-clock runtime and controller overhead on MMLU-Pro with GPT-4o-mini under the Star topology.

Method	Wall-clock runtime	API time	Local controller time	API calls	PT	CT
FC	166.31	166.30	0.00	11.00	15,773	5,466
HALT	34.20	33.17	1.02	2.15	987	858
Reduction	79.44%	80.06%	—	80.45%	93.74%	84.30%

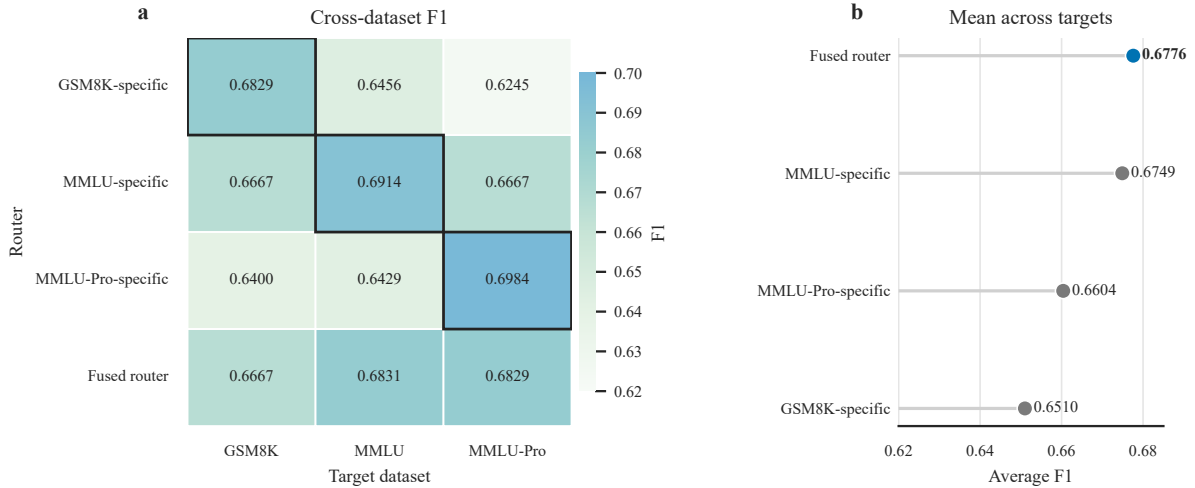


Figure 7: Routers trained on individual datasets and a fused router on GSM8K, MMLU, and MMLU-Pro. (a) Heatmap of F1 scores on each target dataset; black boxes mark the in-domain evaluations of dataset-specific routers. (b) Average F1 across the three target datasets.

ior: with a larger agent pool, individual proposals are more likely to converge quickly through majority agreement, and the evaluator correctly identifies that further deliberation is unnecessary. For instance, on DeepSeek-v3 in MMLU, the halting proportion increases from 43.90% (two agents) to 73.17% (five agents); on Qwen3-14b in GSM8K, it rises from 71.75% to 96.61%. Critically, this adaptive behavior occurs without any scale-specific re-tuning of the evaluator’s thresholds, demonstrating that the learned confidence representation generalizes across system sizes. The halting module thus provides increasing cost savings precisely when they are most needed, in larger systems where the raw token cost of full execution grows combinatorially.

HALT maintains competitive accuracy across all scales, with the accuracy and cost trade-off consistently favoring early halting. The key question for scalability is not whether accuracy varies with agent count, some variation is inevitable, but whether HALT’s optimization modules continue to deliver meaningful cost reductions without degrading quality. Table 12 shows that HALT achieves

competitive accuracy across all $|\mathcal{A}|$ values while the halting module prevents the explosive cost growth that would otherwise occur.

Backbone capability mediates the interaction between scale and accuracy. While HALT is effective across all backbones, the relationship between agent count and accuracy is modulated by backbone capability. Strong backbones (DeepSeek-v3) exhibit the most stable accuracy across scales: on MMLU, accuracy varies only 1.91 points (88.04%–89.95%), and on GSM8K it even peaks at 96.55%. This indicates that high-quality per-agent reasoning enables genuine complementarity in larger ensembles. For weaker backbones (GPT-4o-mini, Qwen3-14b), accuracy can decline with more agents, most dramatically Qwen3-14b on GSM8K (92.12%→83.25% from two agents to three agents). This is not a failure of HALT’s optimization modules, but rather reflects a fundamental limit of multi-agent coordination: when individual agents produce unreliable reasoning chains, adding more agents amplifies noise rather than reducing it. Importantly, HALT’s halting module partially mitigates this effect by terminating deliberation early

Table 12: HALT performance across varying agent counts.

Datasets	GSM8K					MMLU					MMLU-pro				
Number	Acc.	PM.	PH.	PT	CT	Acc.	PM.	PH.	PT	CT	Acc.	PM.	PH.	PT	CT
GPT-4o-mini															
2	94.58	79.80	78.40	280	495	81.82	33.49	45.71	407	1075	72.86	41.43	44.83	474	391
3	93.60	79.80	79.00	568	635	81.82	33.49	44.29	828	569	68.57	41.43	48.28	1021	691
4	94.09	79.80	78.40	1283	994	81.82	33.49	47.14	1406	781	70.00	41.43	44.83	1885	1028
5	93.10	79.80	83.33	1236	2377	81.34	33.49	58.57	2089	952	67.14	41.43	65.52	1965	1005
DeepSeek-v3															
2	95.57	43.35	72.73	236	290	88.04	39.23	43.90	413	366	83.86	52.86	48.65	568	496
3	95.07	43.35	75.00	434	401	89.47	39.23	43.90	914	688	85.71	52.86	48.65	1278	2267
4	96.55	43.35	81.82	167	492	88.04	39.23	64.63	1253	874	82.86	52.86	59.46	1693	1076
5	94.09	43.35	92.05	710	518	89.95	39.23	73.17	1390	908	81.43	52.86	75.68	2315	1255
Qwen3-14b															
2	92.12	87.19	71.75	419	486	88.52	57.42	25.83	770	789	74.29	67.14	30.00	897	884
3	83.25	87.19	64.97	739	629	86.60	57.42	25.83	1975	1668	71.43	67.14	31.91	2375	1896
4	83.74	87.19	73.45	1522	892	85.32	57.42	46.38	3375	2190	71.43	67.14	51.06	3645	2350
5	84.73	87.19	96.61	1146	822	81.65	57.42	66.96	3604	2309	67.14	67.14	61.70	4865	2595

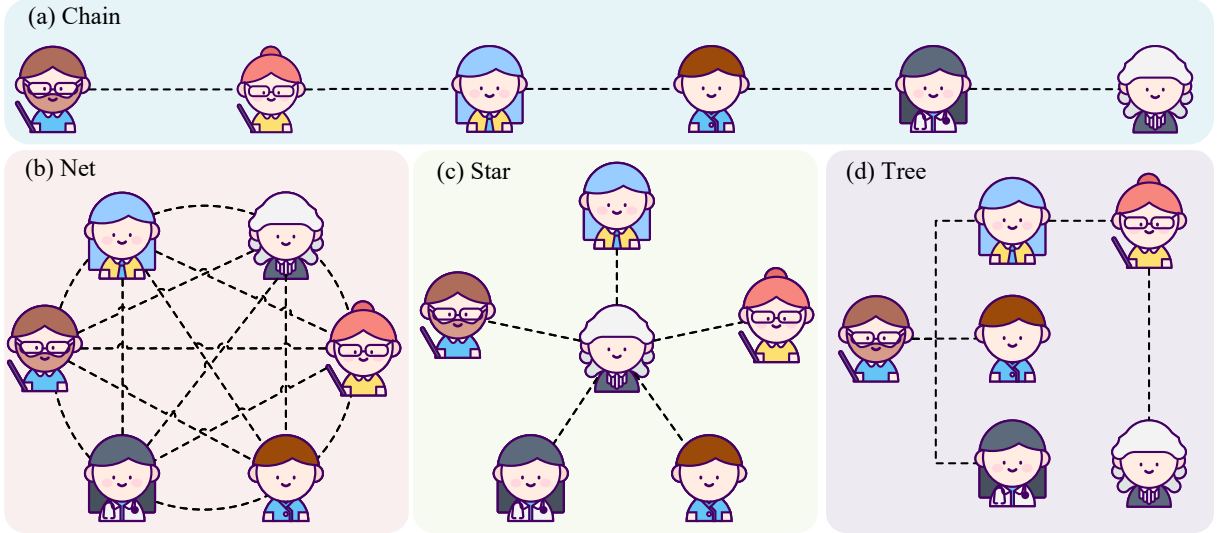


Figure 8: Several representative types of MAS topologies.

before error amplification can compound, as evidenced by the high halting proportions observed for weaker backbones at larger agent counts (e.g., 96.61% for Qwen3-14b in GSM8K at five agents).

The scalability analysis demonstrates that HALT operate effectively across multi-agent systems of varying size, without requiring any scale-specific reconfiguration. The router maintains stable query allocation, the halting module adapts its trigger rate to agent density, and the cost savings grow precisely when they matter most. Although backbone capability mediates the accuracy and scale relationship, HALT consistently delivers its core value proposition: adaptive cost reduction without accuracy degradation, regardless of the scale of the

system.

E.3.2 Different Topologies of MAS

A practical MAS optimization method must not be tied to a specific communication topology. Real-world deployments may adopt fully connected, hierarchical, or hub-spoke architectures depending on latency constraints, privacy requirements, or organizational structure. This section evaluates whether HALT remain effective when applied to several classic MAS topologies as shown in Figure 8: **Chain**, **Net**, **Star**, and **Tree**. If HALT were only effective on a particular topology, its applicability would be narrow; consistent effectiveness across topologies would confirm that the method captures topology-invariant principles of multi-agent

cost optimization. And Table 13 presents the full results under the same variables ($\tau_{route} = 0.5$, $\tau_{prune} = 0.6$, $e_{max} = 0.85$, $e_{min} = 0.45$). We emphasize that these unified configurations do not represent the optimal performance for each individual topology, fine-tuning topology specific parameters could potentially yield better results. Rather, this controlled setup isolates the contribution of HALT itself from confounding variables, providing a conservative estimate of its generalizability. The complete results are presented in Table 13.

Token efficiency gains. HALT delivers dramatic reductions in token consumption across all topologies, models, and datasets. On average, HALT reduces prompt tokens by 92.4% (median: 93.7%, ranging from 51.7% to 97.9%) and completion tokens by 79.7% (median: 80.8%, ranging from 46.0% to 88.7%). Notably, the most substantial PT reduction is observed for DeepSeek-v3 with Star+HALT on GSM8K, which cuts PT by 97.9% (from 8,566 to merely 183 tokens) while maintaining strong accuracy (95.48%, within 0.5% of the Star baseline). CT is consistently high across configurations, typically exceeding 77%. These results confirm that HALT’s early-stopping mechanism effectively prevents redundant inter-agent communication, regardless of the underlying collaboration structure.

Accuracy preservation and model specific behavior. The impact of HALT on accuracy exhibits considerable variance across models, suggesting that the backbone capability of LLM to perform confident self-assessment under limited context is critical to HALT’s success:

DeepSeek-v3 demonstrates the most favorable accuracy–efficiency trade-off. Averaged across all topologies and datasets, DeepSeek-v3 with HALT achieves a near-zero accuracy change (+0.08%) compared to its HALT-free counterparts, with 6 out of 12 topology, dataset combinations yielding accuracy improvements. Particularly striking is DeepSeek-v3 with Star + HALT on MMLU, which matches the full Star baseline at 95.48% while reducing PT by 94.5% and CT by 81.8%. The strongest gains are observed for DeepSeek-v3 with Net + HALT on GSM8K (+3.02%) and with Chain + HALT on MMLU-pro (+4.29%), suggesting that HALT’s context truncation not only preserves but occasionally enhances reasoning by filtering distracting inter-agent exchanges.

GPT-4o-mini shows moderate sensitivity to HALT, with an average accuracy drop of -1.73%

across configurations. Interestingly, HALT improves accuracy on GSM8K for three out of four topologies (e.g., +1.01% Chain + HALT) and yields a substantial +2.70% gain for Net+HALT on MMLU. However, performance degradation is more pronounced on MMLU-pro, where all HALT variants underperform (up to -7.14% for Star+HALT). This pattern suggests that GPT-4o-mini’s confidence calibration is less reliable for complex, domain-diverse tasks compared to mathematical reasoning.

Qwen3-14b exhibits the highest vulnerability to aggressive context truncation, with an average accuracy decline of -3.28% and no positive cases across all 12 configurations. The degradation is most severe for Star+HALT on GSM8K (-6.03%) and MMLU (-6.16%), despite PT and CT reductions of 95.3%/78.5% and 92.1%/76.4%, respectively. We hypothesize that Qwen3-14b’s relatively smaller parameter scale (14B) limits its capacity to maintain reasoning coherence under severely compressed multi-agent contexts, indicating that HALT’s effectiveness is model-capacity dependent.

Topology specific analysis. The relative effect of HALT also varies with the MAS topologies:

Net+HALT, despite showing substantial token reductions (e.g., 95.2% PT reduction for GPT-4o-mini on GSM8K), suffers from accuracy instability, particularly on MMLU-pro. The fully connected topology’s high baseline cost makes it less suitable for HALT optimization when model capacity is limited.

Chain+HALT demonstrates strong token efficiency (e.g., 95.0% PT reduction for GPT-4o-mini on MMLU, the lowest absolute PT among all HALT configurations at 636 tokens) but incurs notable accuracy drops on knowledge-intensive tasks.

Tree+HALT provides the best accuracy–efficiency balance for GPT-4o-mini (88.76% average accuracy with 93.4% PT reduction and 81.9% CT reduction), while maintaining competitive performance for DeepSeek-v3. Star+HALT consistently achieves the lowest token consumption across all models (e.g., average PT reduction of 94.5% for GPT-4o-mini, 94.5% for DeepSeek-v3, and 93.0% for Qwen3-14b), attributable to the centralized hub-and-spoke architecture naturally limiting communication paths. This makes Star an ideal topology for cost-sensitive deployments when paired with a robust backbone.

These results establish several key findings:

(1) HALT achieves consistent and substantial

Table 13: The average accuracy and task tokens on GSM8K, MMLU, and MMLU-pro with different LLMs and MAS topologies.

Datasets	GSM8K			MMLU			MMLU-pro		
Topologies	Acc.	PT	CT	Acc.	PT	CT	Acc.	PT	CT
GPT-4o-mini									
Net	93.47	20775	4000	85.59	30997	5982	77.14	15818	4907
Net+HALT	93.97	1003	769	88.29	1981	1066	74.29	1062	919
Chain	92.96	2299	4014	86.43	12754	4987	80.00	13324	1783
Chain+HALT	93.97	1111	790	80.90	636	665	74.29	995	963
Tree	95.48	11540	3934	87.44	14212	4770	87.14	13588	5261
Tree+HALT	94.97	610	754	85.59	956	799	85.71	1046	959
Star	94.47	11338	3853	86.93	14496	4889	75.71	13528	5066
Star+HALT	94.97	677	819	86.43	736	817	68.57	736	825
DeepSeek-v3									
Net	90.45	17029	3318	89.95	13927	4618	85.71	16926	5185
Net+HALT	93.47	439	388	91.46	869	851	88.57	901	908
Chain	93.47	8127	3140	91.46	13817	5526	80.00	14865	5802
Chain+HALT	95.48	227	354	90.45	932	941	84.29	1250	1174
Tree	95.48	9441	3114	93.47	14668	5007	87.14	13440	5186
Tree+HALT	95.98	389	455	87.44	879	884	84.29	1028	1038
Star	95.98	8566	2826	95.48	13467	4736	88.57	12532	4968
Star+HALT	95.48	183	391	95.48	744	862	85.71	1114	1132
Qwen3-14b									
Net	93.53	17529	3279	87.08	45459	9134	71.43	22454	7317
Net+HALT	87.79	833	632	84.21	3106	1866	71.43	2189	1934
Chain	95.98	8264	3101	87.44	19996	8166	75.71	20783	8330
Chain+HALT	95.48	508	620	84.92	1764	1893	71.43	2141	2251
Tree	92.46	11508	3950	89.45	21177	7471	78.57	19244	7541
Tree+HALT	89.24	608	747	87.11	1837	1827	74.29	1935	2050
Star	96.48	9755	3218	90.45	20329	7281	75.71	18581	7409
Star+HALT	90.45	455	691	84.29	1601	1719	74.29	1585	1857

token reductions (averaging 92.4% for PT and 79.7% for CT) across all MAS topologies; (2) the accuracy–efficiency trade-off is highly model-dependent, with DeepSeek-v3 exhibiting near-lossless compression (+0.08%), GPT-4o-mini showing task-selective benefits, and Qwen3-14b suffering from noticeable degradation (−3.28%); and (3) Star and Tree topologies provide the most favorable substrate for HALT optimization. These findings underscore the importance of jointly selecting the MAS topology, backbone model, and

halting strategy to optimize the cost-performance Pareto frontier in practical deployments.

E.4 Halting Thresholds

This appendix complements the DeepSeek-v3 sensitivity analysis in Section 4.5 by testing the same temporal halting thresholds on Qwen3-14b and GPT-4o-mini. The goal is not to identify one universal threshold pair, but to verify whether (e_{min}, e_{max}) consistently acts as a calibration interface across backbone capacities. Figures 10 and 9 report accuracy, early-stopping rate, prompt tokens,

and completion tokens on MMLU-Pro.

GPT-4o-mini. Figure 9 shows that GPT-4o-mini is more sensitive to aggressive temporal halting than DeepSeek-v3. Permissive halting thresholds reduce PT and CT substantially by stopping deliberation early, but they can also terminate before weaker agents have accumulated enough corrective evidence. Stricter thresholds delay the halt decision and recover accuracy, with the best observed setting reaching 72.86% on MMLU-Pro. This result clarifies the role of temporal thresholds: for smaller or less reliable backbones, HALT should use a more conservative validation-selected operating point, trading some additional tokens for a safer stopping decision.

Qwen3-14b. Figure 10 presents a less brittle trade-off surface for Qwen3-14b. Moderate halting bounds preserve a broad usable region in which accuracy remains competitive while PT and CT stay well below full deliberation. Very permissive thresholds still favor latency, and overly strict thresholds reduce the savings by allowing more rounds to continue. The important observation is that a middle range of validation-selected thresholds provides consistent benefit: HALT removes redundant deliberation without requiring the exact same threshold pair used by DeepSeek-v3 or GPT-4o-mini.

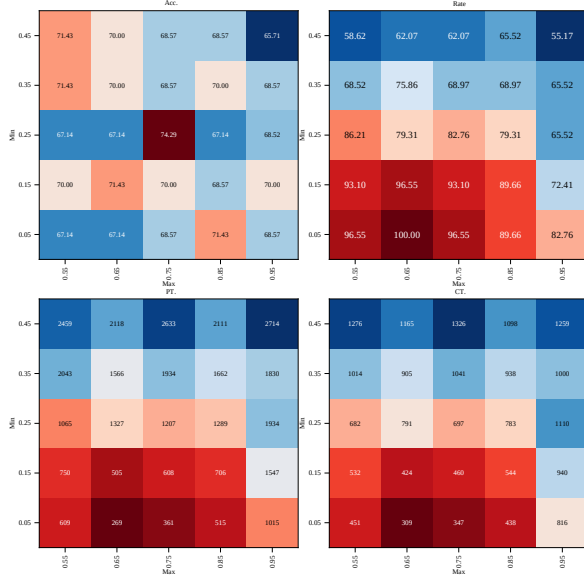


Figure 9: Sensitivity of temporal halting thresholds with GPT-4o-mini in MMLU-pro.

E.5 Pruning Thresholds

This section extends the spatial pruning analysis to GSM8K and MMLU, two tasks with dif-

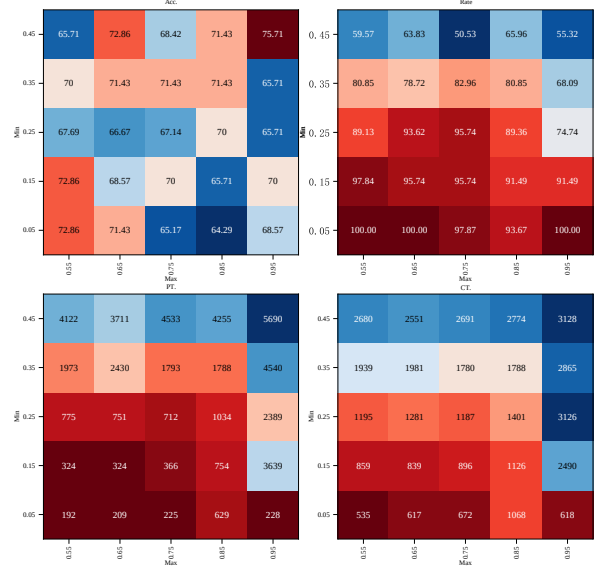


Figure 10: Sensitivity of temporal halting thresholds with Qwen-14b in MMLU-pro.

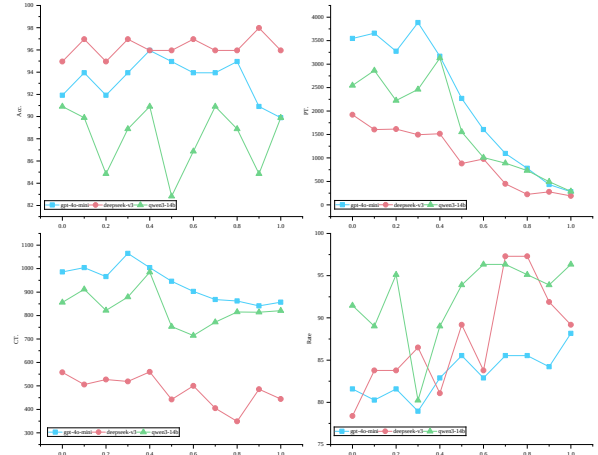


Figure 11: Impact of pruning thresholds on LLMs Performance in GSM8K.

ferent evidence structures. GSM8K requires preserving ordered mathematical derivations, whereas MMLU often contains overlapping factual rationales and multiple-choice elimination cues. Figures 11 and 12 therefore test whether τ_{prune} provides robust context compression rather than a dataset-specific pruning shortcut.

GSM8K. Figure 11 shows that mathematical reasoning benefits most from conservative-to-moderate pruning. In this range, HALT removes conversational fillers, duplicated statements, and low-value detours while retaining the equations and intermediate quantities needed for later reasoning. As τ_{prune} increases, token usage decreases, demonstrating that the evaluator can compress MAS context at the clause level. However, very high prun-

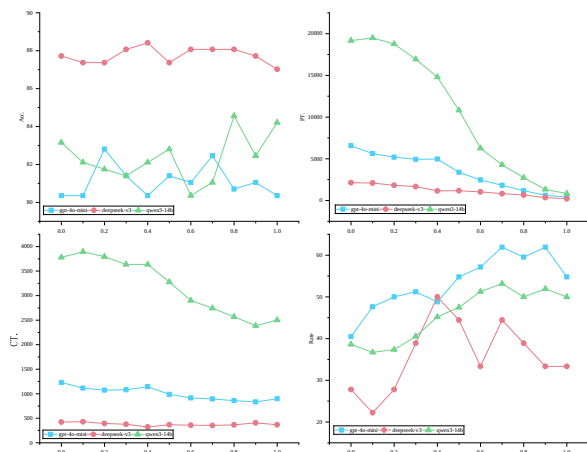


Figure 12: Impact of pruning thresholds on LLMs Performance in MMLU.

ing thresholds become risky because small mathematical clauses may carry essential information; removing them can break the derivation chain. This behavior supports a practical calibration rule: GSM8K-like tasks should use pruning to reduce redundancy, but should avoid thresholds that treat short intermediate equations as expendable context.

MMLU. Figure 12 shows a wider usable pruning region on knowledge-intensive multiple-choice questions. Compared with GSM8K, MMLU reasoning often contains redundant factual explanations and overlapping option-elimination arguments, so moderate pruning can remove repeated rationales without immediately discarding the decisive evidence. This makes τ_{prune} an effective density control for factual collaboration: increasing it improves token efficiency, while validation prevents the threshold from crossing into a regime where useful evidence is lost. The result is consistent with the main MMLU-Pro sweep: pruning is beneficial over a broad calibrated range, but the preferred operating point depends on task structure and backbone reliability.

Task-adaptive context compression. Taken together, the GSM8K, MMLU, and MMLU-Pro sensitivity curves show that spatial pruning should be interpreted as task-adaptive context compression. The same threshold does not have identical effects across datasets, because a short mathematical clause and a repeated factual rationale carry different risks when removed. Nevertheless, after validation calibration, a broad middle range of τ_{prune} consistently reduces context size and supports earlier halting without collapsing task performance. This is the intended advantage of HALT’s

micro-level intervention: it provides a controllable mechanism for reducing MAS communication cost, while making the accuracy and efficiency trade-off visible before test deployment.