

Batch-of-Thought: Cross-Instance Learning for Enhanced LLM Reasoning

Xuan Yang^{1,2*}, Furong Jia¹, Roy Xie¹, Xi Xiong²,

Hengwei Bian², Jian Li², Monica Agrawal¹

¹Duke University ²ByteDance Inc.

{xuan.yang, flora.jia, ruoyu.xie, monica.agrawal}@duke.edu

{xi.xiong, hengwei.bian, limingjun.tsinghua}@bytedance.com

Abstract

Current Large Language Model reasoning systems process queries independently, discarding valuable cross-instance signals such as shared reasoning patterns and consistency constraints. We introduce Batch-of-Thought (BoT), a training-free method that processes related queries jointly to enable cross-instance learning. By performing comparative analysis across queries, BoT identifies high-quality reasoning templates, detects errors through consistency checks, and amortizes computational costs. We instantiate BoT within a multi-agent reflection architecture (BoT-R), where a Reflector performs joint evaluation to unlock mutual information gain unavailable in isolated processing. Experiments across three model families and six benchmarks demonstrate that BoT-R consistently improves accuracy and confidence calibration while reducing inference costs by up to 61%. Our theoretical and experimental analysis reveals *when* and *why* batch-aware reasoning benefits LLM systems.¹

1 Introduction

Large Language Models (LLMs) (Achiam et al., 2023; Madaan et al., 2023; Wei et al., 2022; Shinn et al., 2023; Yao et al., 2022) have achieved strong performance across diverse tasks and are increasingly applied in domains such as medical reasoning, question answering, and scientific problem solving (Singhal et al., 2025; McDuff et al., 2025; Nori et al., 2025; Haas et al., 2025; Wang et al., 2023; Sun et al., 2024). However, producing reliable answers with well-calibrated confidence remains a challenge (Xiong et al., 2024; Ji et al., 2023). LLMs often assign high confidence to incorrect answers, which undermines their practical deployment in high-stakes applications where accuracy

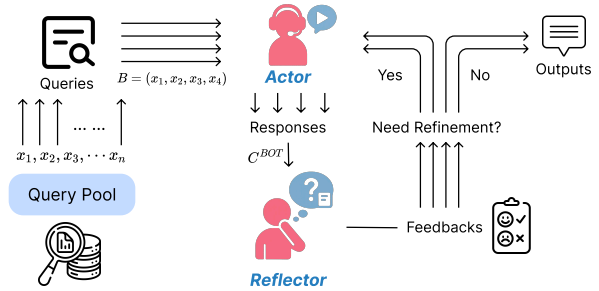


Figure 1: Batch-of-Thought reflection framework. An Actor generates initial responses for a batch of queries. A Reflector then jointly evaluates all responses through comparative analysis, determining whether each should be finalized or refined with feedback.

and reliable uncertainty quantification are essential.

Multi-agent LLM systems (Li et al., 2023; Chan et al., 2024; Guo et al., 2024) extend single-model capabilities through specialized roles and iterative refinement. Despite architectural diversity, existing approaches share a fundamental limitation: they process queries independently. While computationally straightforward, this paradigm discards valuable cross-instance signals. When queries share domain characteristics or structural patterns, isolated processing prevents the system from identifying outliers through comparative assessment, propagating validated knowledge from confident instances to uncertain ones, or detecting errors that emerge only through cross-instance consistency checks.

We introduce **Batch-of-Thought (BoT)**, a training-free framework that processes related queries jointly to enable cross-instance learning and comparative reasoning. Our key insight is that batch-level reasoning unlocks mutual information gain unavailable in isolated processing. By treating queries as a cohort rather than independent instances, BoT enables comparative analysis, reasoning pattern identification, and distributional uncertainty calibration.

* Work done during an internship at ByteDance Inc.

¹Code can be found in <https://github.com/xuanyang19/BoT>.

To illustrate this principle, consider fraud detection: evaluating a single seller in isolation provides limited signal. Examining a cohort simultaneously reveals recurring suspicious patterns, distinguishes legitimate domain practices from genuine anomalies, and enables comparative evidence assessment. This principle mirrors James–Stein estimation (James and Stein, 1961; Stein, 1956; Efron and Morris, 1977): pooling information across similar instances improves individual estimates through shrinkage toward the cohort mean (see Appendix D for theoretical analysis).

We instantiate BoT within a multi-agent reflection architecture (Madaan et al., 2023; Shinn et al., 2023), termed **BoT-R**, though the principle generalizes to other frameworks including Plan-and-Act (Erdogan et al., 2025) and Multi-Agent Debate (Du et al., 2024; Liang et al., 2024). In BoT-R, an Actor generates answer-rationale pairs for a batch of queries, then a Reflector jointly evaluates through comparative analysis—identifying inconsistencies, extracting shared domain knowledge, assessing relative quality, and suggesting refinements. This approach simultaneously improves reasoning quality and computational efficiency by amortizing reflection overhead across the batch. We summarize our main contributions as follows:

1. We propose **Batch-of-Thought (BoT)**, a training-free method that enhances LLM reasoning by processing related queries as cohesive batches, enabling cross-instance learning unavailable in isolated processing.
2. We instantiate BoT in a reflection-based multi-agent system and conduct experiments across six benchmarks and three model families, demonstrating consistent accuracy improvements and **46.9%** average cost reduction.
3. We theoretically and experimentally analyze how task characteristics and batch composition influence BoT’s effectiveness, revealing that interpretive domains benefit substantially from comparative reasoning while symbolic tasks require careful batch design.

2 Methods

Batch-of-Thought (BoT) is a training-free, model-agnostic method that jointly processes batches of queries to improve reasoning quality, confidence

Algorithm 1 BoT-R

Require: Batch $B = \{x_i\}_{i=1}^N$, Actor $\mathcal{A}$, Reflector $\mathcal{R}$, tool set $\mathcal{T}$, max outer rounds T , max tool calls K
Ensure: Final answers $\{a_i\}_{i=1}^N$ with confidences $\{u_i\}_{i=1}^N$
Initialize $c_i \leftarrow \emptyset, \forall i \in [N]$; active set $S \leftarrow [N]$
for $t = 1$ **to** T **do**
 (Parallel) $(a_i, \rho_i) \leftarrow \mathcal{A}(x_i, \mathcal{T}, c_i, K)$ for all $i \in S$
 Build reflective context $\mathcal{C} \leftarrow \langle (x_i, a_i, \rho_i) \rangle_{i=1}^N$
 (Joint) $(u_i, r_i, c_i) \leftarrow \mathcal{R}(\mathcal{C})$ for all $i \in S$
 if $\forall i : r_i = 0$ **then break**
 $S \leftarrow \{i \mid r_i = 1\}$
return $\{a_i\}, \{u_i\}$

calibration, and computational efficiency. We formalize the approach and describe its instantiation within a multi-agent reflection architecture.

2.1 Problem Formulation

Let $\mathcal{X}$ and $\mathcal{Y}$ denote input and output spaces. Queries arrive in batches $B = \{x_i\}_{i=1}^N \subset \mathcal{X}$. We employ a two-agent architecture with iterative refinement:

Actor $\mathcal{A}$. A ReAct agent (Yao et al., 2022) that interleaves reasoning traces with tool execution to generate answer-rationale pairs. At iteration t , given query x_i and optional critique $c_i^{(t-1)}$ from the previous round:

$$(a_i^{(t)}, \rho_i^{(t)}) = \mathcal{A}(x_i, c_i^{(t-1)}; \text{tools}), \quad a_i^{(t)} \in \mathcal{Y}, \quad (1)$$

where $a_i^{(t)}$ is the predicted answer, $\rho_i^{(t)}$ is the reasoning trace, and $c_i^{(0)} = \emptyset$ for the initial iteration.

Reflector $\mathcal{R}$. A reflection agent (Madaan et al., 2023; Shinn et al., 2023) that evaluates the context $\mathcal{C}^{(t)}$ containing all current query-answer-rationale tuples. For each query i , it produces:

$$(r_i^{(t)}, u_i^{(t)}, c_i^{(t)}) = \mathcal{R}(\mathcal{C}^{(t)}), \quad (2)$$

where $r_i^{(t)} \in \{0, 1\}$ indicates whether query i requires refinement, $u_i^{(t)} \in [0, 1]$ is a confidence score, and $c_i^{(t)}$ is an actionable critique. If $r_i^{(t)} = 1$, the query proceeds to iteration $t + 1$ with critique $c_i^{(t)}$; otherwise, $a_i^{(t)}$ is finalized.

Objective. Improve (i) task accuracy, (ii) confidence calibration, and (iii) computational efficiency.

2.2 Batch-of-Thought (BoT)

Formalization. Standard per-instance reflection constructs N independent contexts:

$$\mathcal{C}_i^{\text{ind}} = \langle (x_i, a_i, \rho_i) \rangle, \quad (3)$$

Model	Method	FraudDet $n = 1793$	GPQA $n = 448$	Winogrande $n = 1267$	MedQA $n = 1273$	PubMedQA $n = 1000$	SMS Spam $n = 1510$
GPT-4o	ReAct	0.685	0.439	0.872	0.878	0.679	0.796
	Reflection	0.693	0.459	0.879	0.901	0.667	0.854
	BoT-R	0.740(4)	0.488(4)	0.890(8)	0.904(8)	0.698(8)	0.887(8)
Llama-3.3-70B	ReAct	0.635	0.494	0.831	0.783	0.753	0.920
	Reflection	0.679	0.504	0.853	0.797	0.755	0.925
	BoT-R	0.713(4)	0.516(4)	0.862(4)	0.804(4)	0.757(8)	0.923(8)
Qwen3-Next-80B	ReAct	0.633	0.560	0.823	0.814	0.732	0.946
	Reflection	0.639	0.636	0.869	0.846	0.681	0.900
	BoT-R	0.660(4)	0.657(4)	0.874(4)	0.860(8)	0.704(4)	0.919(8)

Table 1: Performance comparison of reasoning methods across various base models and datasets (number of queries listed as n). Scores represent accuracy. For BoT-R, the number in parentheses denotes the batch size $N \in \{4, 8\}$ used to obtain the reported result. The best result in each setting is highlighted in bold.

evaluating each query in isolation. BoT instead constructs a single *shared context*:

$$\mathcal{C}^{\text{BoT}} = \langle (x_1, a_1, \rho_1), \dots, (x_N, a_N, \rho_N) \rangle, \quad (4)$$

and performs joint evaluation:

$$\{(r_i, u_i, c_i)\}_{i=1}^N = \mathcal{R}(\mathcal{C}^{\text{BoT}}). \quad (5)$$

Cross-instance mechanisms. The shared context enables three synergistic mechanisms: **(1) Outlier detection:** $\mathcal{R}$ identifies answers that appear plausible in isolation but are inconsistent with peer patterns, propagating high-quality reasoning templates via critiques $\{c_i\}$. **(2) Distributional calibration:** Confidence scores u_i are calibrated relative to batch statistics $\phi(\mathcal{C}^{\text{BoT}})$ rather than assessed independently, improving uncertainty quantification. **(3) Computational amortization:** Evaluation rubrics are encoded once per batch, reducing input costs, and joint evaluation enables more accurate refinement decisions, reducing unnecessary Actor-Reflector loops.

The complete BoT-R workflow is detailed in Algorithm 1, which alternates between Actor generation and Reflector evaluation steps until convergence or maximum iterations.

Theoretical foundation. Appendix D establishes formal guarantees through information-theoretic and statistical analysis, demonstrating that BoT achieves a **Pareto improvement** over independent processing: simultaneously enhancing accuracy and reducing computational cost.

3 Experiments

3.1 Experimental Setup

We evaluate BoT on six datasets, including five public benchmarks and one newly curated corpus,

using both API-based and open-source large language models. Full experimental details are provided in Appendix A.

Datasets. Our evaluation covers diverse reasoning and decision-making tasks: GPQA (Rein et al., 2023), WinoGrande-debiased (Sakaguchi et al., 2021), PubMedQA (Jin et al., 2019), MedQA (USMLE) (Jin et al., 2021), SMS Spam Detection (Yang et al., 2025b), and a newly curated fraud-seller detection dataset (Appendix I). We additionally use MMLU (Hendrycks et al., 2020) for the subject-level analysis in §4. Together, these benchmarks span scientific reasoning, common-sense inference, biomedical QA, broad academic knowledge, and real-world anomaly detection.

Metrics. We evaluate (i) task accuracy, (ii) token efficiency (input, output, and total token counts), and (iii) confidence calibration using two complementary measures: (a) the Kolmogorov–Smirnov (KS) statistic (Smirnov, 1939) between the confidence distributions of correct vs. incorrect predictions, and (b) Expected Calibration Error (ECE; Guo et al., 2017).

Baselines. To isolate BoT’s contribution, we compare against two training-free reasoning baselines: **ReAct** (Yao et al., 2022), which performs standard single-instance reasoning with optional tool augmentation, and **Reflection** (Shinn et al., 2023; Madaan et al., 2023), a multi-agent framework employing per-instance self-critique and revision. We then augment the same Actor with BoT’s joint, batch-aware reflection to obtain **BoT-R**. All other factors remain constant across methods.

Models. We report results from both API and open-source models: GPT-4o-2024-11-20 (Hurst

Method	SMS_Spam		GPQA		Winogrande	
	Cost	$\Delta\%$	Cost	$\Delta\%$	Cost	$\Delta\%$
<i>Actor</i>						
Reflection	\$3.61	–	\$4.70	–	\$2.57	–
BoT (4)	\$2.41	33.25%	\$3.68	21.71%	\$2.27	11.75%
BoT (8)	\$2.06	42.96%	\$3.42	27.24%	\$1.99	22.54%
<i>Reflector</i>						
Reflection	\$6.39	–	\$2.76	–	\$3.40	–
BoT (4)	\$2.44	61.85%	\$1.49	45.99%	\$1.97	42.10%
BoT (8)	\$1.84	71.12%	\$1.17	57.65%	\$1.52	55.31%
<i>Total</i>						
Reflection	\$10.00	–	\$7.46	–	\$5.97	–
BoT (4)	\$4.85	51.52%	\$5.17	30.68%	\$4.24	29.04%
BoT (8)	\$3.90	60.95%	\$4.59	38.48%	\$3.51	41.21%

Table 2: Total token cost and relative reduction ($\Delta\%$) for each dataset (GPT-4o). $\Delta\%$ is computed for BoT(4) and BoT(8) relative to Reflection cost.

Method	SMS_Spam		GPQA		Winogrande	
	KS $\uparrow$	ECE $\downarrow$	KS $\uparrow$	ECE $\downarrow$	KS $\uparrow$	ECE $\downarrow$
ReAct	0.256	0.176	0.181	0.372	0.273	0.113
Reflection	0.360	0.104	0.265	0.329	0.376	0.035
BoT-R	0.633	0.063	0.368	0.317	0.442	0.013

Table 3: Confidence calibration across datasets (GPT-4o). Each entry reports the KS statistic between the confidence distributions of correct vs. incorrect answers and the ECE score. Higher KS ($\uparrow$) is better, while lower ECE ($\downarrow$) is better.

et al., 2024), Llama-3.3-70B (Grattafiori et al., 2024) and Qwen3-Next-80B (Yang et al., 2025a). For the *Fraud Detection* dataset, we enable Brave Search and the Brave Summarizer as tools for retrieval and grounding (Brave Software, Inc., 2025).

3.2 Main Results

Accuracy. As shown in Table 1, we compare task performance after integrating BoT into the reflection framework. From this ablation study, BoT-R is consistently competitive and typically the strongest overall variant, across datasets and base models, demonstrating the effectiveness of BoT. Gains are most visible on higher-variance, decision-heavy tasks where per-instance reflection can be brittle. For example, under GPT-4o, BoT-R improves FraudDet and GPQA by +4.7 and +2.9 accuracy points over Reflection, respectively, and yields an average improvement of +2.6 points across all six datasets. On near-saturated benchmarks (e.g., SMS Spam with Qwen3), where base performance is already high, the accuracy headroom is limited and BoT-R may not always outperform simpler baselines, suggesting that BoT is most beneficial when cross-instance comparison provides additional cor-

Method	med&bio	hum	social	math	sci
ReAct	0.887	0.805	0.915	0.763	0.797
Reflection	0.886	0.825	0.915	0.865	0.843
BoT-R	0.891	0.837	0.922	0.853	0.832

Table 4: MMLU dataset subject-wise accuracy (GPT-4o). The highest score for each subject is in bold.

rective signal.

Efficiency. We further evaluate efficiency using a unified reference pricing scheme based on production-grade GPT-4o pricing. Table 2 shows that BoT-R substantially reduces overall token cost, achieving **46.9%** average savings across the three representative benchmarks at batch size 8, and up to **61%** reduction on SMS_Spam. Batch formation is not a hidden cost: sequential batching is a single pass over the input stream, and semantic clustering adds only 0.1–0.6% of the token spend (Appendix B.3). BoT-R also dominates majority voting baselines ($s = 3$), which is less accurate at 1.3–6.7 $\times$ the cost (Appendix B.4). These reductions indicate that batch-aware reflection effectively amortizes reflective reasoning across instances while improving task performance.

Calibration. Table 3 shows that BoT-R improves confidence reliability under GPT-4o. It increases KS and reduces ECE across all three datasets (e.g., on SMS_Spam, KS 0.360 $\rightarrow$ 0.633 and ECE 0.104 $\rightarrow$ 0.063). The pattern matches the collective-signal perspective of Lemma D.4: batch statistics carry usable signal when the effective sample size N_{eff} is meaningfully above 1, as expected for moderate error correlation at $N \in \{4, 8\}$, and Proposition D.3 then bounds the calibration gain attainable by conditioning on them.

Overall, the results are consistent with our theoretical analysis: batch-aware reflection yields a favorable accuracy–cost–calibration profile, with robust accuracy gains and calibration improvements on diverse tasks and consistent efficiency. Appendix C complements these aggregate results with two qualitative traces: one illustrates cross-instance contradiction detection, and the other illustrates comparative quality assessment followed by selective re-evaluation. Appendix B provides the full token breakdown, and Appendix G reports alternative calibration metrics, paired significance tests, and run-to-run variance.

Dataset	No-batch	Sequential	Semantic
FraudDet	0.693	0.740	0.768
MedQA	0.901	0.904	0.902
PubMedQA	0.667	0.698	0.697
Winogrande	0.879	0.890	0.897
SMS Spam	0.854	0.887	0.902
GPQA	0.459	0.488	0.486

Table 5: Accuracy across batching strategies (GPT-4o). The highest score for each dataset is in bold.

4 Discussion

RQ1: Which domains benefit most from BoT?

Table 4 shows that BoT-R yields the largest gains on interpretive and judgment-driven domains, including humanities, social sciences, and medicine & biology. These tasks admit multiple plausible reasoning paths and partial cues, making comparative evaluation across instances especially informative. In contrast, domains dominated by exact symbolic derivation, such as mathematics and parts of the physical sciences, exhibit marginal or slightly negative changes. This suggests that batch-level consensus is less effective when correctness depends on exact derivation rather than comparative plausibility (More details in Appendix H). This pattern aligns with the coherence–diversity trade-off formalized in Appendix D.3: interpretive tasks tend to satisfy moderate similarity and moderate error correlation, yielding a meaningful effective sample size and enabling collective signal amplification. By contrast, symbolic domains often sit at the wrong end of this trade-off: small derivation errors are highly correlated within a batch, limiting the benefit of aggregation and occasionally amplifying shared mistakes.

RQ2: How does batching strategy influence performance? We compare three strategies (Table 5): **no-batch** (per-instance reflection), **sequential** batching (queries in natural arrival order; our default), and **semantic** batching (K-means over E5-Mistral-7B embeddings (Wang et al., 2024)). Two findings stand out. First, *how* queries are batched matters less than *that* they are batched: sequential grouping already improves over no-batch on all six datasets (+3.87% average relative gain). Many benchmarks carry latent topical coherence that BoT-R exploits without explicit clustering. Second, semantic coherence adds the most on heterogeneous datasets: FraudDet improves from 0.740 to 0.768 and SMS_Spam from 0.887 to 0.902 over sequential batching (+4.83% average relative gain over no-

batch). For datasets drawn from a narrow latent distribution (MedQA, PubMedQA), the two strategies are nearly indistinguishable: naive batching already satisfies the coherence conditions (Appendix F.2).

RQ3: How does batch size interact with task structure?

While larger batches theoretically provide richer comparative context, a sweep over $N \in \{1, \dots, 16\}$ (Table 12, Appendix F.1) reveals a non-monotonic relationship between batch size and performance. Moderate batch sizes ($N \in \{4, 8\}$) are a strong default, though the best setting is domain-dependent. Larger batches are practically constrained by (i) context window saturation, which forces rationale compression and degrades fine-grained reasoning, and (ii) increased heterogeneity, which dilutes informative cross-instance comparisons. As a result, performance often peaks at intermediate batch sizes despite stronger theoretical aggregation benefits.

Implications and outlook. Overall, these findings suggest that BoT is most effective when batches exhibit sufficient, but not excessive, coherence, and when tasks benefit from comparative judgment rather than exact symbolic derivation. Importantly, the robustness of sequential batching indicates that BoT-R remains practical in streaming or latency-sensitive settings, where semantic clustering may be infeasible. Future work may explore adaptive cohorting strategies that dynamically balance coherence, batch size, and latency, as well as extensions to domains requiring symbolic guarantees.

5 Conclusion

We introduced **Batch-of-Thought (BoT)**, a training-free, model-agnostic approach that processes related queries as a batch so an agent can compare across instances, share knowledge, and produce better-calibrated confidence while amortizing computation. BoT yields higher accuracy, lower token cost, and improved calibration across settings, including on our newly curated Seller Fraud Detection benchmark, and characterizes *when* batching helps and *when* it degrades gracefully.

Limitations

The efficacy of Batch-of-Thought (BoT) is subject to several constraints. Its gains depend on batch composition, and poorly matched queries

can cause negative transfer. Joint prompting also increases context length, limiting applicability to long-input tasks. While we instantiate BoT in an Actor–Reflector architecture, the idea generalizes naturally to other multi-agent designs (e.g., Plan-and-Act, Debate), empirical validation of such integrations remains open. Finally, our experiments batch items within a single public benchmark and evaluation session, so they do not create cross-user privacy risks. In multi-user deployments, however, a shared prompt could expose one user’s content to another; batches should therefore remain within the trusted task boundary.

Ethics Statement

In writing this paper, we used an AI assistant to correct grammatical errors. During the coding process, we utilized AI tools for code completion. All datasets and models used in our experiments are publicly accessible. Our newly introduced Seller Fraud Detection benchmark contains only publicly available information and does not include any private or sensitive data. The Seller Fraud Detection benchmark was developed with human expert annotation. All annotators were compensated fairly for their time and expertise at rates exceeding standard professional compensation in their region. Annotators were provided with clear guidelines and had the option to decline participation at any time.

References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Anthropic. 2024. *The Claude 3 model family: Opus, Sonnet, Haiku*. Technical report, Anthropic. Accessed: 2025-10-07.
- Brave Software, Inc. 2025. Brave web search API documentation. <https://api-dashboard.search.brave.com/app/documentation/web-search/get-started>. Accessed: 2025-10-07.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2024. Chateval: Towards better llm-based evaluators through multi-agent debate. In *International conference on learning representations*, volume 2024, pages 9079–9093.
- Justin 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)*, pages 7066–7085.
- Zhoujun Cheng, Jungo Kasai, and Tao Yu. 2023. Batch prompting: Efficient inference with large language model apis. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 792–810.
- Thomas M. Cover and Joy A. Thomas. 2006. *Elements of Information Theory*, 2/e. John Wiley and Sons.
- 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*.
- Bradley Efron and Carl Morris. 1977. Stein’s paradox in statistics. *Scientific American*, 236(5):119–127.
- Lutfi Eren Erdogan, Nicholas Lee, Sehoon Kim, Suhong Moon, Hiroki Furuta, Gopala Anumanchipalli, Kurt Keutzer, and Amir Gholami. 2025. Plan-and-act: Improving planning of agents for long-horizon tasks. *arXiv preprint arXiv:2503.09572*.
- Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR.
- Tilmann Gneiting and Adrian E Raftery. 2007. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR.
- Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V Chawla, Olaf Wiest, and Xi-angliang Zhang. 2024. Large language model based multi-agents: A survey of progress and challenges. *arXiv preprint arXiv:2402.01680*.
- Lukas Haas, Gal Yona, Giovanni D’Antonio, Sasha Goldshtein, and Dipanjan Das. 2025. Simpleqa verified: A reliable factuality benchmark to measure parametric knowledge. *arXiv preprint arXiv:2509.07968*.

- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.
- Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Steven Yau, Zijuan Lin, Liyang Zhou, and 1 others. 2024. Metagpt: Meta programming for a multi-agent collaborative framework. In *International Conference on Learning Representations*, volume 2024, pages 23247–23275.
- Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.
- Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr.
- William James and Charles Stein. 1961. Estimation with quadratic loss. In *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 361–379. University of California Press.
- Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38.
- Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421.
- Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 2567–2577.
- Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, and 1 others. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.
- Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*.
- Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosior, Seungjin Choi, and Yee Whye Teh. 2019. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International conference on machine learning*, pages 3744–3753. PMLR.
- Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. Camel: Communicative agents for “mind” exploration of large language model society. *Advances in neural information processing systems*, 36:51991–52008.
- Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. 2024. Encouraging divergent thinking in large language models through multi-agent debate. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pages 17889–17904.
- Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. *arXiv preprint arXiv:2205.14334*.
- Sharon L Lohr. 2023. *Sampling: design and analysis*, volume 1. Duxbury press Pacific Grove, California.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, and 1 others. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594.
- Daniel McDuff, Mike Schaekermann, Tao Tu, Anil Palepu, Amy Wang, Jake Garrison, Karan Singhal, Yash Sharma, Shekoofeh Azizi, Kavitaulkarni, and 1 others. 2025. Towards accurate differential diagnosis with large language models. *Nature*, 642(8067):451–457.
- Eric Mitchell, Joseph Noh, Siyan Li, Will Armstrong, Ananth Agarwal, Patrick Liu, Chelsea Finn, and Christopher D Manning. 2022. Enhancing self-consistency and performance of pre-trained language models through natural language inference. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1754–1768.
- Allan H Murphy. 1973. A new vector partition of the probability score. *Journal of Applied Meteorology and Climatology*, 12(4):595–600.
- Harsha Nori, Mayank Daswani, Christopher Kelly, Scott Lundberg, Marco Tulio Ribeiro, Marc Wilson, Xiaoxuan Liu, Viknesh Sounderajah, Jonathan Carlson, Matthew P Lungren, and 1 others. 2025. Sequential diagnosis with language models. *arXiv preprint arXiv:2506.22405*.
- Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, and 1 others. 2024. Chatdev: Communicative agents for software development. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 15174–15186.

- David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2023. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*.
- Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.
- Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652.
- Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Mohamed Amin, Le Hou, Kevin Clark, Stephen R Pfohl, Heather Cole-Lewis, and 1 others. 2025. Toward expert-level medical question answering with large language models. *Nature medicine*, 31(3):943–950.
- Nikolai V Smirnov. 1939. On the estimation of the discrepancy between empirical curves of distribution for two independent samples. *Bull. Math. Univ. Moscow*, 2(2):3–14.
- Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30.
- Charles Stein. 1956. Inadmissibility of the usual estimator for the mean of a multivariate normal distribution. In *Proceedings of the third Berkeley symposium on mathematical statistics and probability, volume 1: Contributions to the theory of statistics*, volume 3, pages 197–207. University of California Press.
- Liangtai Sun, Yang Han, Zihan Zhao, Da Ma, Zhennan Shen, Baocai Chen, Lu Chen, and Kai Yu. 2024. Scieval: A multi-level large language model evaluation benchmark for scientific research. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19053–19061.
- Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Improving text embeddings with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11897–11916.
- Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun R Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. 2023. Scibench: Evaluating college-level scientific problem-solving abilities of large language models. *arXiv preprint arXiv:2307.10635*.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.
- Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. 2023. Auto-gen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155*, 3(4).
- Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2024. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. In *International Conference on Learning Representations*, volume 2024, pages 23650–23678.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.
- Tiankai Yang, Yi Nian, Li Li, Ruiyao Xu, Yuangang Li, Jiaqi Li, Zhuo Xiao, Xiyang Hu, Ryan A Rossi, Kaize Ding, and 1 others. 2025b. Ad-llm: Benchmarking large language models for anomaly detection. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 1524–1547.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.
- Michihiro Yasunaga, Xinyun Chen, Yujia Li, Panupong Pasupat, Jure Leskovec, Percy Liang, Ed H Chi, and Denny Zhou. 2024. Large language models as analogical reasoners. In *International Conference on Learning Representations*, volume 2024, pages 17019–17045.
- Junchi Yu, Ran He, and Rex Ying. 2024. Thought propagation: An analogical approach to complex reasoning with large language models. In *International Conference on Learning Representations*, volume 2024, pages 10989–11015.
- Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. 2017. Deep sets. *Advances in neural information processing systems*, 30.

A Experiment Settings

Confidence Calibration Metrics. We use two complementary measures: (a) the Kolmogorov–Smirnov (KS) statistic (Smirnov, 1939), which measures the maximum difference between the cumulative distributions of confidence scores for correct versus incorrect predictions—higher KS values indicate better separation and thus more reliable confidence estimates; and (b) Expected Calibration Error (ECE (Guo et al., 2017)), which quantifies the average gap between predicted confidence and actual accuracy across binned predictions—lower ECE indicates that confidence scores accurately reflect true correctness probabilities. Together, these metrics assess both discriminative power (KS) and absolute calibration quality (ECE).

Protocol. All methods are training-free. Prompts, temperature (0.0), tool access, and stopping rules (maximum 5 reflection iterations) are held constant across conditions; seeds are fixed for comparability. For batched settings, we vary batch size $N \in \{4, 8\}$ and use a sequential batching strategy in the main experiments.

B Token Efficiency Results

We provide a comprehensive breakdown of token usage and costs across different experimental configurations. To ensure fair comparison, all costs are normalized using the production-grade GPT-4o pricing scheme (input: \$2.50 per 1M tokens; output: \$10.00 per 1M tokens as of the experimental period).

Table 6 presents detailed token counts for each pipeline stage—Actor, Reflector, and combined total—across three representative datasets. For each method, we report input tokens, output tokens, and total cost in USD. The Actor stage includes all reasoning and answer generation, while the Reflector stage encompasses evaluation, critique generation, and refinement decisions.

B.1 Efficiency Gains from Batch Processing

BoT achieves substantial efficiency improvements through two complementary mechanisms. First, **instruction amortization**: the Reflector’s evaluation rubric and reasoning guidelines are encoded once per batch rather than repeated for each query, saving $(N - 1) \times T_{\text{inst}}$ tokens where N is batch size and T_{inst} is instruction length. Second, **reduced**

iteration overhead: joint evaluation enables more accurate refinement decisions, reducing unnecessary Actor-Reflector loops.

As reported in Table 2, BoT-R achieves an average total cost reduction of **46.9%** across the three benchmarks when using batch size 8. Savings are most pronounced on the SMS_Spam dataset (**61%** reduction), where the homogeneous task structure enables highly effective batch-level evaluation. Even with batch size 4, BoT-R consistently reduces costs by 29-52% while maintaining or improving accuracy.

B.2 Stage-Level Analysis

The efficiency gains distribute differently across pipeline stages:

Actor Stage: BoT introduces minimal overhead at the Actor level, as answer generation remains largely independent. The modest savings arise from reduced refinement iterations due to more accurate Reflector feedback.

Reflector Stage: BoT delivers dramatic savings (42-71% reduction) by replacing N independent reflection calls with a single joint evaluation. Larger batch sizes amplify these gains: moving from $N = 4$ to $N = 8$ increases Reflector savings from 46% to 58% on GPQA.

Total Cost: The combined effect yields 29-61% total cost reduction depending on dataset characteristics and batch size. These results demonstrate that BoT achieves a Pareto improvement: simultaneously enhancing both task performance (Table 1) and computational efficiency, making it particularly valuable for production deployments where cost and accuracy are both critical.

B.3 Cost of Batch Formation

A natural question is whether the cost of assembling batches offsets the token savings reported above. It does not.

Sequential batching is effectively free. All results in Tables 1, 2 and 6 use sequential batching, which groups queries in their natural dataset order (Appendix F.2) in a single pass over the input stream, so there is no formation cost to report for those configurations. BoT-R does not depend on optimized cohorts: sequential batching alone improves over per-instance Reflection on all six datasets (Table 5), because many workloads carry latent topical coherence in their natural order—MedQA and PubMedQA queries, for instance, are drawn from a narrow biomedical distribution. The 29–61% token

Method	SMS_Spam				GPQA				Winogrande			
	In	Out	Cost	$\Delta\%$	In	Out	Cost	$\Delta\%$	In	Out	Cost	$\Delta\%$
<i>Actor</i>												
Reflection	105,936	334,429	\$3.61		196,763	421,168	\$4.70		115,788	228,050	\$2.57	
BoT (4)	75,838	221,962	\$2.41	33.25%	176,381	324,153	\$3.68	21.71%	103,757	200,847	\$2.27	11.75%
BoT (8)	65,268	189,556	\$2.06	42.96%	146,350	305,647	\$3.42	27.24%	91,716	176,136	\$1.99	22.54%
<i>Reflector</i>												
Reflection	429,383	531,556	\$6.39		602,293	125,173	\$2.76		378,801	245,424	\$3.40	
BoT (4)	216,070	189,716	\$2.44	61.85%	370,783	56,221	\$1.49	45.99%	257,260	132,619	\$1.97	42.10%
BoT (8)	186,413	137,899	\$1.84	71.12%	317,082	37,507	\$1.17	57.65%	229,118	94,713	\$1.52	55.31%
<i>Total</i>												
Reflection	535,319	865,985	\$10.00		799,056	546,341	\$7.46		494,589	473,474	\$5.97	
BoT (4)	291,908	411,678	\$4.85	51.52%	547,164	380,374	\$5.17	30.68%	361,017	333,466	\$4.24	29.04%
BoT (8)	251,681	327,455	\$3.90	60.95%	463,432	343,154	\$4.59	38.48%	320,834	270,849	\$3.51	41.21%

Table 6: Input and output token usage and cost across different methods, decomposed into *Actor*, *Reflector*, and *Total* stages (GPT-4o). 1M input tokens cost \$2.5 and 1M output tokens cost \$10. $\Delta\%$ is the cost reduction, computed for BoT(4) and BoT(8) relative to Reflection for the same dataset and stage.

Dataset	n	Embed	k -means
GPQA	448	\$0.0275	\$0.00007
Winogrande	1267	\$0.0053	\$0.00046
SMS Spam	1510	\$0.0103	\$0.00066
<i>Formation total (% of BoT(4) token cost)</i>			
GPQA		\$0.028	(0.53%)
Winogrande		\$0.006	(0.14%)
SMS Spam		\$0.011	(0.23%)

Table 7: One-off cost of forming semantically coherent batches. Percentages are taken relative to the corresponding BoT(4) total token cost in Table 2.

savings in Table 2 therefore come with essentially free batch formation.

Semantic batching costs 0.1–0.6% of the LLM token cost. Semantic batching, which yields the further gains in Table 5, adds two compute steps: an E5-Mistral-7B embedding pass and MiniBatchK-Means clustering with balanced assembly. Table 7 reports the measured one-off cost per corpus. We price the embedding pass on an RTX 4090 at \$0.69/hr, conservatively assuming 20% FLOPs utilization and padded token counts, and price clustering on an AWS m7i.2xlarge CPU instance at \$0.4032/hr using measured wall-clock time (median of three runs).

Two observations follow. First, the routing and clustering step itself costs under \$0.001 on every dataset—seconds of CPU time—so the total is dominated by the one-off embedding pass, which is computed once per corpus and cached across batch sizes and repeated runs. Second, the savings that batching enables exceed the formation cost by

Method	Winogrande		GPQA	
	Acc	Cost	Acc	Cost
ReAct	87.20	\$1.92	43.90	\$2.28
+ majority vote ($s=3$)	86.98	\$5.71	41.39	\$6.75
Reflection	87.90	\$5.97	45.90	\$7.46
+ majority vote ($s=3$)	88.86	\$23.52	46.65	\$34.21
BoT-R	89.00	\$3.51	48.80	\$5.17

Table 8: Majority voting compared with BoT-R (GPT-4o). Accuracy in percent, cost in USD over the full dataset. Majority-vote rows aggregate $s = 3$ samples at temperature 0.7; all other rows use temperature 0. BoT-R uses the batch size reported in Table 1.

roughly two orders of magnitude ($83\text{--}557\times$ across these three datasets): on SMS_Spam, batch-aware reflection saves \$6.10 relative to per-instance Reflection at $N=8$, against a \$0.011 formation cost.

B.4 Comparison with Majority Voting

Self-consistency (Wang et al., 2022) is the standard sampling-based route to higher accuracy and better calibration, and is a natural point of comparison. Reliable gains from self-consistency typically require many forward passes—often 10–40 samples depending on the task (Appendix E.1)—which multiplies inference cost, whereas BoT-R obtains its gains within a single batch pass while *reducing* cost. This difference in cost profile is why our baseline ladder (ReAct $\rightarrow$ Reflection $\rightarrow$ BoT-R) varies only the batch context.

For completeness we evaluate majority voting with $s = 3$, the smallest sample count that supports voting; larger s would widen the cost gap further. Table 8 reports accuracy and total cost under GPT-

40.

Three observations follow. First, voting over three ReAct samples *degrades* accuracy on both datasets (−0.2 on Winogrande, −2.5 on GPQA) despite roughly tripling cost: with few samples, majority voting amplifies sampling noise, consistent with reports that self-consistency requires substantially larger sample counts to stabilize. Second, voting over three full Reflection pipelines does improve Reflection, yet still trails BoT-R on both datasets (−0.14 and −2.15 points) while costing 6.7× and 6.6× more. Third, BoT-R therefore occupies the accuracy–cost frontier in this comparison: the highest accuracy of any method evaluated, at a fraction of the cost of either aggregation baseline.

C Case Studies

The two runs below show the cross-instance mechanisms claimed in §2.2 operating in practice.

Case 1: cross-instance comparison exposes a contradiction (Winogrande). Two items in one batch shared the premise “*The children were not vaccinated. Betty was fine with this, but Mary was annoyed,*” and asked opposite questions: who *knew* vaccines do not cause autism, and who *believed* they do. Processed independently, each question is a self-contained interpretation task, and the Actor answered “Mary” to both—each answer individually plausible. Presented together, the pair carries a constraint that neither instance carries alone: the two opposing beliefs cannot both belong to Mary. Evaluating the batch jointly, the Reflector kept “Mary” for the first item but flagged the second as inconsistent with the characters’ stated reactions—Mary’s annoyance at non-vaccination implies she regards vaccines as safe, which makes Betty the coherent holder of the opposing belief—and requested re-evaluation, upon which the Actor corrected the answer (Table 9). The error is invisible from either instance alone and detectable only in the shared batch context: cross-instance consistency checking in its purest form.

Case 2: comparative quality assessment singles out the weakest answer. A four-item batch contained three answers supported by mechanism-level justifications and one—a diene-reactivity ranking—asserted without support (option B, ordering 4-2-1-3). Seen in isolation an unsupported answer is not obviously wrong; seen beside three well-justified peers it stands out. The joint Reflector accepted

Question	Initial	After BoT-R
Who <i>knew</i> vaccines do not cause autism?	Mary ✓	Mary ✓
Who <i>believed</i> vaccines do cause autism?	Mary ×	Betty ✓

Table 9: Case 1. Answered in isolation, the Actor assigns both opposing beliefs to the same character; joint evaluation surfaces the contradiction and corrects the second item.

the three and flagged only the fourth (Table 10). Its critique—that the answer “is incomplete, and the correct sequence of diene reactivity is not clearly explained”—requested a justification grounded in the reaction mechanism and molecular properties. On re-evaluation the Actor produced the correct option A (3-1-2-4) together with the missing rationale: cyclopentadiene is most reactive because of its favorable locked geometry, and the hindered (Z,Z)-diene least reactive. Two mechanisms are visible at once. The batch supplies a *relative quality bar*, making the unsupported answer salient, and refinement is *selective*: only one of four items is re-run, which is precisely the source of the reduced iteration overhead quantified in Appendix B.1.

Batch item	Reflector decision
Organic named-reaction	Accept
Genetic inheritance	Accept
Stem-cell culture	Accept
Diene reactivity	Re-evaluate

Table 10: Case 2. Joint evaluation of a four-item batch flags only the answer that lacks mechanism-level support, leaving the other three untouched.

Together these examples instantiate the mechanisms claimed in §2.2—consistency checking across instances and comparative outlier detection—in real runs.

D Theoretical Analysis

We establish three guarantees for batch-aware reflection. The first two are information-theoretic and follow from a single observation — that the batch context refines the per-instance σ -algebra — and are therefore *attainability* statements: they bound what a Bayes-optimal Reflector can achieve. The third is an exact token-counting identity that holds independently of Reflector quality. The substantive batch-specific content is the effective-sample-size analysis (§D.3), which makes the coherence–diversity trade-off quantitative.

D.1 Setup

Definition D.1 (Batch reasoning problem). *Let $\mathcal{D}$ be a joint distribution on $\mathcal{X} \times \mathcal{Y}$ and $B = \{(x_i, y_i^*)\}_{i=1}^N$ a batch drawn from $\mathcal{D}^{\otimes N}$. The Actor emits predictions $\hat{y}_i = \mathcal{A}(x_i)$ with rationales ρ_i , and the Reflector consumes $\mathcal{C}^{\text{BoT}} = \{(x_j, \hat{y}_j, \rho_j)\}_{j=1}^N$ to produce, for each $i \in [N]$, a confidence $u_i \in [0, 1]$, a re-evaluation flag $r_i \in \{0, 1\}$, and a critique c_i . Let $z_i = \mathbf{1}[\hat{y}_i = y_i^*]$ and $e_i = 1 - z_i$.*

Fix i and let $\mathcal{G}_0 = \sigma(\hat{y}_i, \rho_i)$. For any $\mathcal{C}^{\text{BoT}}$ -measurable summary ϕ , write $\mathcal{G}_1 = \sigma(\mathcal{G}_0, \phi) \supseteq \mathcal{G}_0$. The Bayes-optimal confidences are $u_i^{\text{ind}} := \mathbb{P}(z_i = 1 \mid \mathcal{G}_0)$ and $u_i^{\text{BoT}} := \mathbb{P}(z_i = 1 \mid \mathcal{G}_1)$.

Assumption D.2 (Coherence). *The pairs $\{(x_i, y_i^*)\}_{i=1}^N$ are exchangeable, and the error indicators are equicorrelated: $\text{Cor}(e_i, e_j) = \rho_e \in [0, 1)$ for $i \neq j$. Higher semantic similarity within a batch is associated with larger ρ_e , since related queries induce correlated reasoning errors.*

D.2 Calibration and discrimination gain

Proposition D.3 (Bayes-optimal batch reflection dominates). *For any strictly proper scoring rule S in loss form (e.g., Brier loss, log-loss), $\mathbb{E}[S(u_i^{\text{BoT}}, z_i)] \leq \mathbb{E}[S(u_i^{\text{ind}}, z_i)]$, $I(z_i; u_i^{\text{BoT}}) \geq I(z_i; u_i^{\text{ind}})$, with both inequalities strict iff $I(z_i; \phi \mid \mathcal{G}_0) > 0$.*

Proof. Both claims follow from $\mathcal{G}_0 \subseteq \mathcal{G}_1$. Strict propriety (Gneiting and Raftery, 2007) implies that $q \mapsto \mathbb{E}[S(q, z_i) \mid \mathcal{G}_1](\omega)$ is uniquely minimized at $q = \mathbb{P}(z_i = 1 \mid \mathcal{G}_1)(\omega) = u_i^{\text{BoT}}(\omega)$. Since u_i^{ind} is $\mathcal{G}_0$ -measurable and hence $\mathcal{G}_1$ -measurable, plugging it in yields $\mathbb{E}[S(u_i^{\text{ind}}, z_i) \mid \mathcal{G}_1] \geq \mathbb{E}[S(u_i^{\text{BoT}}, z_i) \mid \mathcal{G}_1]$ a.s.; taking expectations gives the first inequality, with equality iff $u_i^{\text{ind}} = u_i^{\text{BoT}}$ a.s., i.e. $z_i \perp\!\!\!\perp \phi \mid \mathcal{G}_0$. For the second, since z_i is binary the Bayes posterior is sufficient: $I(z_i; u_i^{\text{ind}}) = I(z_i; \mathcal{G}_0)$ and $I(z_i; u_i^{\text{BoT}}) = I(z_i; \mathcal{G}_1)$, and the chain rule (Cover and Thomas, 2006) gives $I(z_i; \mathcal{G}_1) - I(z_i; \mathcal{G}_0) = I(z_i; \phi \mid \mathcal{G}_0) \geq 0$. $\square$

Scope. Proposition D.3 requires no batch structure beyond the σ -algebra refinement $\mathcal{G}_0 \subseteq \mathcal{G}_1$; it bounds what is *attainable* by any $\mathcal{G}_1$ -measurable predictor. The Reflector LLM produces an approximation $\tilde{u}_i^{\text{BoT}}$ to the Bayes posterior, so the realized gain is upper-bounded by the attainable gain. Whether this gain is non-trivial depends on

(a) whether the batch context carries residual information about correctness, i.e. $I(z_i; \phi \mid \mathcal{G}_0) > 0$, and (b) how closely the Reflector approximates the Bayes posterior. The first is an empirical question about batch composition that §D.3 characterizes; the second is validated by the calibration improvements in Table 3.

From proper losses to ECE. The Brier loss admits the decomposition $\text{BS} = \text{REL} - \text{RES} + \text{UNC}$ (Murphy, 1973), so Proposition D.3 implies a strictly lower $(\text{REL} - \text{RES})$ whenever the batch is informative — realizable through better calibration, higher resolution, or both. ECE estimates REL alone and is not a proper scoring rule, so the proposition does not directly imply ECE reduction; the empirical ECE reduction in Table 3, alongside the improved KS separation, is nevertheless consistent with the decomposition gain.

D.3 Effective sample size

The dominance in Proposition D.3 is non-trivial only when batch-level statistics carry usable signal. Under Assumption D.2, the precision of the simplest batch statistic — the empirical mean — admits a closed form.

Lemma D.4 (Effective sample size (Lohr, 2023)). *Let $p = \mathbb{P}(z_i = 1)$. Under Assumption D.2,*

$$\text{Var}(\bar{z}_N) = \frac{p(1-p)}{N_{\text{eff}}}, \quad N_{\text{eff}} := \frac{N}{1 + (N-1)\rho_e}. \quad (6)$$

N_{eff} is strictly increasing in N for $\rho_e < 1$, with $N_{\text{eff}} \rightarrow 1/\rho_e$ as $N \rightarrow \infty$ when $\rho_e > 0$, and $N_{\text{eff}} = N$ when $\rho_e = 0$.

This is the standard design-effect formula from survey sampling; we record it because it makes the coherence–diversity trade-off quantitative. Within-batch similarity makes ϕ more informative about each z_i but inflates ρ_e , which suppresses N_{eff} . The two effects pull in opposite directions, predicting that BoT is most effective at *moderate* coherence — neither uncorrelated batches (where ϕ carries no signal, $I(z_i; \phi \mid \mathcal{G}_0) \rightarrow 0$) nor highly correlated batches (where $N_{\text{eff}} \rightarrow 1$). This matches the subject-wise pattern of §4: interpretive domains gain the most; mathematics and parts of physical sciences, where mistakes cluster more tightly, gain little.

We do not estimate ρ_e directly. Our experiments measure discrimination via the Kolmogorov–Smirnov distance between the conditional distributions of u given $z_i = 1$ and $z_i = 0$; KS and mutual

information are different functionals of the joint law and are not monotonically related in general, but both vanish iff $u \perp\!\!\!\perp z_i$. The KS improvements in Table 3 are thus directionally consistent with Proposition D.3.

D.4 Input-cost identity

Proposition D.5 (Input-cost identity). *Let T_{inst} and T_{ctx} denote the token counts of the shared reflection instructions and per-instance context. Per-instance reflection re-emits instructions N times for input cost $C_{\text{in}}^{\text{ind}} = N(T_{\text{inst}} + T_{\text{ctx}})$; batch reflection emits them once for $C_{\text{in}}^{\text{BoT}} = T_{\text{inst}} + N T_{\text{ctx}}$. The savings are exactly $(N - 1) T_{\text{inst}}$ tokens, with relative reduction*

$$1 - \frac{C_{\text{in}}^{\text{BoT}}}{C_{\text{in}}^{\text{ind}}} = \frac{(N - 1) T_{\text{inst}}}{N(T_{\text{inst}} + T_{\text{ctx}})} \xrightarrow{N \rightarrow \infty} \frac{T_{\text{inst}}}{T_{\text{inst}} + T_{\text{ctx}}}. \quad (7)$$

The identity is exact and independent of Reflector quality.

The output side is not amenable to closed-form analysis because critique length is determined by the Reflector. Empirically (Table 6), joint Reflector output exhibits sublinear growth: critiques cross-reference shared reasoning structures (“items 2 and 5 share the same flawed assumption”) and reuse insights across instances. Combined with the input-side identity, this drives the 46.9% average total-cost reduction reported in §3.

D.5 When BoT helps, and when it degrades gracefully

The preceding results identify three regimes in which BoT converges to per-instance reflection without regressing. *No coherence* ($\rho_e \rightarrow 0$, so $I(z_i; \phi \mid \mathcal{G}_0) \rightarrow 0$): ϕ carries no information; the calibration and discrimination gains vanish, but the input-cost saving persists. *High correlation* ($\rho_e \rightarrow 1$): $N_{\text{eff}} \rightarrow 1$ and the collective-signal contribution saturates. *Small N* : the input-cost advantage $(N - 1) T_{\text{inst}}$ becomes negligible. All three degradations are continuous in (ρ_e, N) , so BoT converges to the per-instance baseline rather than regressing catastrophically.

E Related Work

E.1 Confidence Calibration in Large Language Models

Reliable uncertainty quantification remains critical for deploying LLMs in high-stakes applica-

tions. Modern LLMs frequently exhibit poor calibration, assigning high confidence to incorrect predictions (Guo et al., 2017; Xiong et al., 2024; Kadavath et al., 2022). This miscalibration persists even in state-of-the-art models (Achiam et al., 2023; Anthropic, 2024), undermining trust in automated decision-making systems.

Existing calibration approaches fall into three categories. **Post-hoc calibration methods** apply temperature scaling (Guo et al., 2017) or Platt scaling to model outputs, but require held-out calibration sets and fail to capture semantic uncertainty (Kuhn et al., 2023). **Sampling-based methods** estimate uncertainty through self-consistency (Wang et al., 2022), semantic entropy (Kuhn et al., 2023), or ensemble disagreement (Chen et al., 2024). While effective, these approaches incur substantial computational overhead—self-consistency requires 10-40 samples per query—and process each instance independently, missing opportunities for cross-instance calibration; Appendix B.4 quantifies this accuracy-cost trade-off empirically. **Verbalized confidence approaches** directly prompt models for numerical (Xiong et al., 2024; Lin et al., 2022) estimates. These methods are efficient but highly sensitive to prompt formatting and often produce overconfident predictions (Kadavath et al., 2022). Recent efforts employ chain-of-thought reasoning for confidence elicitation (Xiong et al., 2024) or fine-tune models on calibration data (Lin et al., 2022), yet these remain per-instance techniques that cannot leverage distributional signals.

Our work introduces **comparative calibration through batch processing**: confidence scores are grounded in cross-instance statistics rather than isolated assessments. This approach combines the efficiency of verbalized confidence (no additional sampling) with the distributional awareness of ensemble methods, achieving superior calibration without multiplicative computational costs.

E.2 Multi-Agent Reasoning Systems

Recent work has explored sophisticated communication protocols (Wu et al., 2023), dynamic role allocation (Hong et al., 2024), and multi-agent collaboration on complex tasks (Li et al., 2023; Qian et al., 2024). However, a fundamental limitation persists: **existing multi-agent systems process queries independently**. Even when multiple agents collaborate on a single query, the framework treats each query in isolation, discarding cross-instance signals.

AutoGen (Wu et al., 2023) and MetaGPT (Hong et al., 2024) enable multi-agent workflows but apply them instance-by-instance. CAMEL (Li et al., 2023) studies role-playing conversations yet maintains per-query boundaries.

The closest work to ours is **batch prompting** (Cheng et al., 2023), which groups multiple queries into a single API call for efficiency. However, batch prompting lacks reflective evaluation mechanisms and does not perform comparative analysis—it simply concatenates queries without leveraging cross-instance reasoning. Our work fundamentally differs by introducing **batch-aware reflection**: the Reflector explicitly performs comparative evaluation, consistency checking, and knowledge propagation across the batch.

E.3 Cross-Instance Learning

In deep learning, batch normalization (Ioffe and Szegedy, 2015) leverages mini-batch statistics during training, while recent work explores cross-example attention (Lee et al., 2019) and set-based reasoning (Zaheer et al., 2017). Meta-learning approaches (Finn et al., 2017; Snell et al., 2017) learn from task distributions rather than individual instances, demonstrating benefits of comparative learning.

For LLM inference, **in-context learning** (Brown et al., 2020) uses examples to guide reasoning, and **analogical prompting** (Yasunaga et al., 2024) retrieves similar cases to aid problem-solving. Building on this, recent analogical question-answering frameworks systematically leverage past examples to guide complex reasoning steps (Yu et al., 2024). Furthermore, in the realm of model training, recent reinforcement learning methods increasingly leverage cohort-level feedback signals and distribution-aware rewards, such as Group Relative Policy Optimization (GRPO), to stabilize training and align collective reasoning patterns (Shao et al., 2024). However, these methods rely on predefined examples, retrieval systems, or training-time optimization rather than jointly reasoning over a batch of target queries at inference. Recent work on **self-consistency with rationalization** (Mitchell et al., 2022) aggregates multiple reasoning paths for a single query but does not transfer knowledge across distinct queries.

BoT differs by enabling **mutual information gain across queries at inference time**: each query in the batch provides signal for evaluating others through comparative reflection. This creates a feed-

back loop where batch-level patterns inform individual assessments, analogous to how James-Stein estimation improves individual predictions through the group mean, but applied dynamically to LLM reasoning rather than static parameter estimation.

E.4 Positioning of Our Work

Our contributions address gaps in existing literature along three dimensions:

(1) Efficiency-calibration trade-off: We achieve better calibration than verbalized confidence while reducing costs by 46.9% (vs. per-instance reflection) rather than increasing costs 10-40× (self-consistency overhead).

(2) Cross-instance reasoning: We introduce the first multi-agent framework that explicitly performs comparative evaluation across queries, going beyond batch prompting’s simple concatenation to enable consistency checking, knowledge propagation, and distributional calibration.

(3) Training-free generality: Unlike calibration methods requiring fine-tuning (Lin et al., 2022) or specialized architectures, BoT is model-agnostic and integrates with existing multi-agent frameworks (Reflection, Plan-and-Act, Debate) without additional training.

F Batching Strategy Analysis

We investigate how batch composition and size influence BoT’s performance through systematic experiments across six benchmarks using GPT-4o.

F.1 Batch Size Effects

Table 11 presents accuracy across batch sizes $N \in \{1, 4, 8\}$, where $N = 1$ corresponds to standard per-instance reflection. To systematically stress-test scaling behavior and robustness, we further evaluate performance across a wider spectrum of batch sizes ($N \in \{1, 2, 4, 6, 8, 12, 16\}$) on three representative datasets. These extended results are shown in Table 12.

The results reveal that while BoT-R remains stable across a broad range of batch sizes, optimal scaling behavior is highly domain-dependent and often non-monotonic. Datasets with relatively homogeneous task structures, such as SMS Spam, benefit from larger batches (reaching peak accuracy at $N = 16$). This aligns with our theoretical framework (Lemma D.4): larger batches increase the “effective sample size,” which enhances comparative reasoning and the reliability of collective signals for error detection.

Conversely, tasks requiring deep logical derivation or exhibiting high intra-domain variance, such as GPQA, peak at smaller batch sizes ($N \in \{2, 4\}$) and gradually degrade as N increases. Our theoretical analysis (Appendix D) predicts that while larger batches theoretically increase mutual information gain, two practical factors constrain this relationship in practice:

(1) Context window saturation. As batch size approaches model context limits, the system must compress individual rationales to fit all items. Near capacity, models produce overly concise responses that sacrifice reasoning depth for brevity, diminishing the comparative analysis benefits. For GPQA—which requires detailed scientific reasoning—this compression effect becomes apparent at $N \geq 8$, where individual responses average 30% shorter than at $N = 4$.

(2) Batch heterogeneity. When queries within a batch are too dissimilar, cross-instance signals become noisy rather than informative. Sequential batching—our default strategy—groups adjacent queries without explicit similarity filtering. For datasets with high within-domain variance (e.g., GPQA spanning biology, physics, and chemistry), larger batches increase the likelihood of mixing incompatible problem types, diluting useful comparative signals.

F.2 Semantic vs. Sequential Batching

We further study how batch composition influences BoT-R by comparing three batching strategies across six datasets. **No-batch** applies reflection independently to each query without any cross-instance context. **Sequential batching** groups queries in their original dataset order—without explicitly enforcing semantic similarity—and is used as the default strategy in our main experiments. **Semantic batching** clusters queries by embedding similarity using K-means over E5-Mistral-7B embeddings (Wang et al., 2024), and forms fixed-size batches from cluster members sorted by proximity to the cluster centroid to maximize within-batch coherence. Results are summarized in Table 5.

Robust gains from simple batching. A key observation is that BoT-R delivers substantial improvements even under simple sequential batching. Compared to the no-batch baseline, sequential grouping improves performance on *all six datasets*, yielding an average relative gain of **+3.87%**. This demonstrates that BoT-R is not overly sensitive to

imperfect batch coherence and can reliably extract useful cross-instance signals even when batches are formed without explicit semantic optimization.

Additional benefits from semantic coherence.

Semantic batching provides further gains on several datasets, particularly those where cross-instance comparison and distributional cues are informative. On FraudDet, accuracy improves from 0.740 to 0.768, and on SMS Spam from 0.887 to 0.902 when moving from sequential to semantic batching. Winogrande shows a similar trend. Averaged across all six datasets, semantic batching achieves a relative improvement of **+4.83%** over the no-batch baseline, exceeding that of sequential batching. These results align with our theoretical analysis: increasing within-batch coherence strengthens the informativeness of batch-level statistics, improving collective error detection and refinement decisions.

When batching strategy matters less. For datasets characterized by shared domain knowledge or homogeneous reasoning styles, such as MedQA and PubMedQA, the difference between sequential and semantic batching is minimal. This suggests that when queries already originate from a narrow latent distribution, even naive batching satisfies the coherence conditions required for effective batch-aware reasoning, consistent with the graceful-degradation properties in Appendix D.5.

Practical considerations. While semantic clustering is beneficial in offline or high-throughput evaluation settings, it introduces practical constraints in streaming scenarios (e.g., online fraud detection), where queries arrive sequentially and delaying processing to form semantically coherent batches may increase latency. The strong performance of sequential batching indicates that BoT-R remains effective under such constraints. Designing adaptive cohorting strategies that balance coherence, latency, and throughput is an important direction for future work.

G Additional Calibration Metrics and Significance Analysis

G.1 Bias-Robust Calibration Metrics

The ECE reported in Table 3 uses equal-width bins, whose value can depend on the choice of bin edges. To check that the calibration improvement is not an artifact of the binning scheme, we recompute

Batch	FraudDet	GPQA	Winogrande	MedQA	PubMedQA	SMS Spam
1 (Reflection)	0.693	0.459	0.879	0.901	0.667	0.854
4	0.740	0.488	0.888	0.895	0.683	0.881
8	0.732	0.471	0.890	0.904	0.698	0.887

Table 11: Batch size influence on accuracy (GPT-4o) across the full benchmark suite. Batch size $N = 1$ is functionally equivalent to per-instance Reflection.

Dataset	$N = 1$	$N = 2$	$N = 4$	$N = 6$	$N = 8$	$N = 12$	$N = 16$
Winogrande	0.879	0.886	0.888	0.893	0.890	0.893	0.887
SMS Spam	0.854	0.887	0.881	0.909	0.887	0.912	0.921
GPQA	0.459	0.488	0.488	0.468	0.471	0.472	0.456

Table 12: Extended batch size stress test on three representative datasets (GPT-4o). Accuracy is reported across a wider range of batch sizes to evaluate scaling behavior, stability, and context saturation boundaries.

Method	EF-ECE ↓	Brier ↓	NLL ↓	KS ↑
<i>GPQA</i>				
ReAct	51.61%	0.5023	1.7006	0.181
Reflection	44.81%	0.4412	1.5211	0.265
BoT-R	40.66%	0.4015	1.4761	0.368
<i>Winogrande</i>				
ReAct	6.58%	0.1107	0.3720	0.273
Reflection	3.16%	0.0942	0.3278	0.376
BoT-R	2.50%	0.0901	0.3135	0.442

Table 13: Bias-robust calibration metrics (GPT-4o). EF-ECE is equal-frequency ECE with 10 bins; Brier and binary NLL are strictly proper scoring rules; KS is reproduced from Table 3. Best value per column and dataset in bold.

Dataset	Comparison	Exact p
GPQA	ReAct → Reflection	0.011
	ReAct → BoT-R	0.008
	Reflection → BoT-R	0.048
Winogrande	ReAct → Reflection	0.007
	ReAct → BoT-R	0.001
	Reflection → BoT-R	0.324

Table 14: Exact McNemar tests on paired per-item outcomes from the deterministic (temperature 0) runs, GPT-4o.

calibration with equal-frequency (equal-mass) bins and add two strictly proper scoring rules—the class of losses for which Proposition D.3 states its guarantee.

Metric definitions. Equal-frequency ECE sorts predictions by confidence into equal-sized bins and averages the |confidence—accuracy| gap, removing the sensitivity to bin placement that affects equal-width ECE (Guo et al., 2017). The Brier score is the mean squared probability error and binary NLL

Dataset	ReAct	Reflection	BoT-R
GPQA	42.53 ± 1.31	45.58 ± 0.44	47.44 ± 1.18
Winogrande	86.77 ± 0.12	88.23 ± 0.22	89.27 ± 0.15

Table 15: Run-to-run variance at temperature 0.7 over three seeds (GPT-4o); accuracy in percent, mean ± SD. Reported separately from Table 14, since stochastic runs cannot enter a paired test.

the negative log-likelihood of correctness; both are strictly proper (Gneiting and Raftery, 2007), and NLL penalizes confident errors especially heavily. The KS statistic measures the maximum separation between the confidence distributions of correct and incorrect answers (Smirnov, 1939); it captures separation only, not direction, which is why we pair it with the proper scores rather than reporting it alone.

Table 13 reports all four metrics under GPT-4o on GPQA and Winogrande. BoT-R attains the best equal-frequency ECE, Brier score and NLL on both datasets, alongside the largest KS separation. The ordering of the three methods is therefore stable across binning schemes and across both proper and separation-based measures, and its direction is the one predicted by Proposition D.3.

G.2 Significance Testing and Run-to-Run Variance

We assess the accuracy differences in Table 1 in two complementary ways: exact McNemar tests on the paired per-item outcomes of the deterministic runs (Table 14), and accuracy variance over three seeds at temperature 0.7 (Table 15). The two are reported separately because stochastic runs cannot enter a paired per-item test.

BoT-R attains the highest mean accuracy on both

datasets. On GPQA its gain over Reflection is significant ($p = 0.048$) and larger than the run-to-run standard deviation of either method. On Winogrande the gain is reproducible across seeds—the ± 1 SD bands of Reflection and BoT-R do not overlap—but does not reach per-item significance ($p = 0.324$), and is best described as consistent but modest. BoT-R nonetheless reaches that accuracy at 41% lower cost on Winogrande and 31% lower on GPQA (Table 2), so it dominates Reflection on the accuracy–cost plane under even the most conservative reading.

A note on MedQA, where Reflection and BoT-R differ by +0.3 points, or roughly four items out of 1273. A difference this small cannot reach significance in principle: even if every discordant pair favored BoT-R, the exact two-sided McNemar p -value could not fall below $2 \cdot 2^{-4} = 0.125$. Near-ceiling accuracy leaves little headroom on this benchmark, and the result is best read as comparable accuracy at lower cost, with the input-side saving guaranteed by the exact identity of Proposition D.5.

H MMLU Detailed Results

Table 4 reveals a systematic pattern in how BoT’s effectiveness varies across subject domains within the MMLU benchmark. We identify two distinct task categories with markedly different responses to batch-level reasoning.

Subjective and interpretive domains. BoT-R achieves its strongest gains on humanities (+1.2 points over Reflection), social sciences (+0.7), and medicine & biology (+0.5). These domains share three key characteristics: (1) questions often admit multiple defensible reasoning paths, (2) answer quality depends on contextual interpretation rather than strict logical derivation, and (3) comparative evaluation helps identify robust reasoning patterns across similar cases. For instance, in social science questions about policy implications or historical interpretation, batch-level reflection enables the model to distinguish well-grounded arguments from superficially plausible but contextually inconsistent reasoning.

Formal and symbolic domains. In contrast, mathematics and physical sciences show qualitatively different behavior. While Reflection substantially improves over ReAct in these domains (+10.2 and +4.6 points respectively), BoT-R ex-

hibits marginal decline relative to Reflection (−1.2 and −1.1). This pattern suggests that batch-level consensus can occasionally mislead reasoning in domains where correctness is determined by precise symbolic manipulation rather than comparative plausibility. In mathematical problem-solving, an incorrect but superficially consistent approach across multiple batch items may receive spurious validation through cross-instance agreement, whereas per-instance reflection focuses more directly on logical rigor.

Implications for batch composition. These findings indicate that BoT’s effectiveness depends critically on task structure. Domains requiring interpretive judgment and context-dependent reasoning benefit from distributional signals and comparative calibration. Conversely, domains demanding exact symbolic computation may require alternative batch strategies—such as explicitly instructing the Reflector to prioritize logical correctness over cross-instance consensus, or segregating formal reasoning tasks into separate batches. Future work should investigate adaptive reflection strategies that modulate the weight given to batch-level signals based on detected task characteristics.

Subject	ReAct	Reflection	BoT-R
Biology			
anatomy	0.907	0.918	0.910
clinical_knowledge	0.913	0.898	0.913
college_biology	0.941	0.931	0.965
college_medicine	0.852	0.855	0.866
high_school_biology	0.959	0.961	0.964
human_aging	0.821	0.820	0.833
human_sexuality	0.917	0.923	0.923
medical_genetics	0.970	0.970	0.970
nutrition	0.902	0.911	0.902
professional_medicine	0.953	0.963	0.959
virology	0.573	0.564	0.582
General			
global_facts	0.667	0.667	0.697
high_school_european_history	0.896	0.878	0.890
high_school_geography	0.935	0.944	0.944
high_school_us_history	0.942	0.946	0.946
high_school_world_history	0.951	0.941	0.945
miscellaneous	0.961	0.960	0.964
prehistory	0.957	0.943	0.963
Humanities			
management	0.882	0.892	0.902
marketing	0.936	0.927	0.936
moral_disputes	0.871	0.868	0.881
moral_scenarios	0.707	0.767	0.767
philosophy	0.890	0.884	0.900
public_relations	0.739	0.752	0.743
Law			
business_ethics	0.831	0.838	0.838
high_school_government_and_politics	0.982	0.984	0.979
international_law	0.898	0.900	0.908
jurisprudence	0.907	0.916	0.916
professional_law	0.758	0.761	0.763
us_foreign_policy	0.939	0.939	0.960
Math			
abstract_algebra	0.561	0.727	0.697
college_mathematics	0.470	0.697	0.636
econometrics	0.716	0.779	0.743
elementary_mathematics	0.767	0.936	0.936
formal_logic	0.682	0.728	0.752
high_school_macroeconomics	0.906	0.915	0.913
high_school_mathematics	0.478	0.814	0.758
high_school_microeconomics	0.956	0.966	0.966
high_school_statistics	0.797	0.860	0.874
logical_fallacies	0.881	0.883	0.883
professional_accounting	0.761	0.886	0.886
Science			
astronomy	0.932	0.927	0.934
college_chemistry	0.514	0.616	0.626
college_computer_science	0.767	0.869	0.859
college_physics	0.635	0.842	0.802
computer_security	0.851	0.848	0.848
conceptual_physics	0.892	0.902	0.917
electrical_engineering	0.821	0.819	0.847
high_school_chemistry	0.775	0.871	0.861
high_school_computer_science	0.934	0.960	0.970
high_school_physics	0.737	0.867	0.827
machine_learning	0.766	0.802	0.829
security_studies	0.809	0.820	0.824
Social Science			
high_school_psychology	0.953	0.945	0.956
professional_psychology	0.898	0.895	0.904
sociology	0.924	0.925	0.932
world_religions	0.904	0.918	0.906

Table 16: MMLU per-subject accuracy summary grouped by category (GPT-4o).

I Dataset: Fraud Detection

We introduce a seller-level dataset for fraud-seller detection tailored to evaluating LLM-based agent frameworks. Each instance corresponds to a single online seller and is annotated by domain experts as fraudulent (1) or non-fraudulent (0). The release contains 1,793 labeled sellers: 1,055 positives (58.8%) and 738 negatives (41.2%).

For each seller, we provide both a seller profile and one representative product profile from the seller. Seller profiles include `shop_name`, `company_name`, `email_domain`, and `product_categories`.

Product profiles include `product_name`, `product_description`, `detailed_subcategory`, `detailed_category`, and `minimum_list_price_in_USD` and `maximum_list_price_in_USD`. These fields enable models to reason over heterogeneous attributes rather than relying on free text alone.

The target label (`is_fraudulent_shop`) $\in \{0, 1\}$ was assigned by domain experts following internal guidelines that emphasize deceptive practices and policy-violating behavior. While positively labeled cases reflect a consensus judgment of fraud, borderline cases may retain residual ambiguity typical of human annotation.

The corpus captures only the information present in the provided schema. External signals such as reputation scores, user reviews, temporal activity traces, or platform enforcement outcomes are not included. As a result, models evaluated on this dataset reason over supplied profile attributes rather than broader ecosystem signals.

J Prompts

This is the system prompt for the Fraud Detection Dataset:

You are a risk analyst expert working for an e-commerce company. Your job is to protect the platform and its customers by identifying fraudulent sellers. A fraudulent seller might engage in fraudulent activities, sell counterfeit goods, misrepresent products, or provide poor customer service. Your task is to conduct a holistic assessment based on the seller's profile and the sample product of the seller.

You are provided with the seller's shop name, company name (some sellers may not have) and email domain, enclosed in triple backticks:

- shop name: `“SHOP_NAME”`
- company name: `“COMPANY_NAME”`
- email domain: `“EMAIL_DOMAIN”`

You are also given the categories of products sold by the seller, enclosed in triple backticks:

- product categories: `“PRODUCT_CATEGORY”`

You are also given the sample product of the seller, enclosed in triple backticks:

- product_name: `“PRODUCT_NAME”`
- product_description: `“PRODUCT_DESCRIPTION”`
- detailed_subcategory: `“DETAILED_SUBCATEGORY”`
- detailed_category: `“DETAILED_CATEGORY”`
- min_list_price_usd: `“MIN_LIST_PRICE_USD”`
- max_list_price_usd: `“MAX_LIST_PRICE_USD”`

Note: you can use provided tools many times until you think the collected information is sufficient to answer the questions, but do avoid unnecessary tool calls.

Based on the information provided and collected by tools, answer the following questions:

1. ****Shop/Company Name Verification:**** Based on the shop name and company name, does this appear to be a reliable/established seller?

- If names seem generic, suspicious, or unfamiliar, search for the company/shop name to verify legitimacy
 - Note: Only use search results if they are clearly relevant to the specific shop or company name
 - 2. **Email Domain Assessment:** Based on the email domain, does this suggest a professional business?
 - If using unfamiliar business domains, consider searching to check if it belongs to an established company
 - Note: Only use search results if they are clearly relevant to the email domain
 - 3. **Product Information Check:** Based on the sample product name, description and the product categories, do you think it is reasonable for the seller to sell the products in the shop?
 - 4. **Product Price Verification:** Does the product pricing seem reasonable for the category?
 - If pricing appears suspiciously low or high, search for typical market prices of similar products
 - 5. Based on all the information, do you think this seller is a fraudulent seller?
- Assign a confidence score: rate your confidence in the assessment.
- Return your response in a single JSON object with the following keys:
- 'is_fraudulent_shop': (boolean) 'true' if the shop exhibits indicators of fraudulent operations, otherwise 'false'.
 - 'confidence_score': (float) A score from 0.0 to 1.0 indicating your confidence in the assessment.
 - 'summary_reasoning': (string) Comprehensive explanation of your fraud assessment, including all factors that led to your conclusion.

This is the system prompt for the reflector:

You are a reflection agent to help refine the answers. Here are N questions, each with the previous model's answer.

For each, critique the model answer for accuracy, completeness, and reasoning, comparing across all answers and their reasoning paths in the batch to identify areas for improvement and give a peer confidence score to quantify how possible the answer is correct.

Make sure you understand each question-answer pair and give detailed explanations to them, Carefully decide if a reevaluation is needed for each case.

For each, provide: (1) whether to trigger reevaluation (true/false) and improve answer, (2) summary assessment, (3) peer confidence score for the current answer(0.0-1.0), (4) suggestions for improvement(empty if reevaluation is false).

Output a JSON list, one entry per question, strictly in format:

"response:{trigger_reevaluation: bool, summary_comment: str, confidence_score: float(0.0-1.0), suggestions: str}]"