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

Anonymous ACL submission

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 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 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.

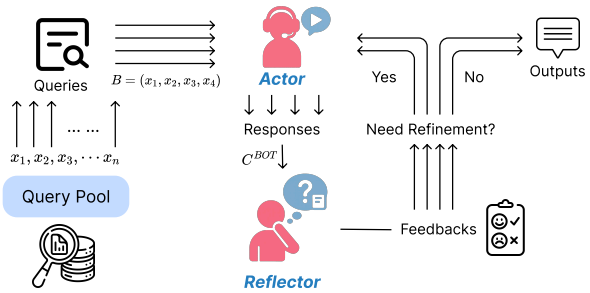

Figure 1: Batch-of-Thought reflection framework. An Actor generates initial responses for a batch of queries. A Reflector then jointly evaluates all responses through comparative analysis, determining whether each should be finalized or refined with feedback.

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.

To illustrate this principle, consider fraud detection: evaluating a single seller in isolation provides limited signal. Examining a cohort simultaneously reveals recurring suspicious patterns, distin-

guishes legitimate domain practices from genuine anomalies, and enables comparative evidence assessment. This principle mirrors James–Stein estimation (James et al., 1961; Stein, 1956; Efron and Morris, 1977): pooling information across similar instances improves individual estimates through shrinkage toward the cohort mean (see Appendix C 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 calibration, and computational efficiency. We formalize the approach and describe its instantiation within a multi-agent reflection architecture.

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 $c_i \leftarrow \varnothing, \forall i \in [N]$; 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 $\{a_i\}, \{u_i\}$

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:

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

where $a_i^{(t)}$ is the predicted answer, $\rho_i^{(t)}$ is the reasoning trace, and $c_i^{(0)} = \varnothing$ 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:

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

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:

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

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

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

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.

and performs joint evaluation:

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

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 C 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.

Metrics. We evaluate (i) task accuracy, (ii) token efficiency (input token count, output token count, and total), and (iii) confidence calibration using two complementary measures: (a) the Kolmogorov–Smirnov (KS) statistic (Smirnov et al., 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 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

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. 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

Method	SMS_Spam		GPQA		Winogrande	
	Cost	Δ%	Cost	Δ%	Cost	Δ%
Actor						
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%
Reflector						
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%
Total						
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 (Δ%) for each dataset (GPT-4o). Δ% is computed for BoT(4) and BoT(8) relative to Reflection cost.

Method	SMS_Spam		GPQA		Winogrande	
	KS↑	ECE↓	KS↑	ECE↓	KS↑	ECE↓
ReAct	0.256	0.176	0.181	0.372	0.273	0.113
Reflect	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.

the three representative benchmarks at batch size 8, and up to **61%** reduction on SMS_Spam. Finally, Table 3 shows that BoT-R improves confidence reliability under GPT-4o. Across all settings, BoT-R yields lower ECE and higher KS scores, indicating that joint reasoning over a batch improves the alignment between confidence scores and answer correctness—enhancing the reliability and interpretability of the model's judgments.

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. Additional details are provided in Appendix B.

4 Discussions

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 especially informative. In contrast, domains dominated by exact symbolic derivation,

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.

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 F) This pattern aligns with the coherence–diversity trade-off in §C.3. Interpretive tasks tend to satisfy moderate similarity and moderate error correlation, yielding a meaningful effective sample size and enabling collective signal amplification.

RQ2: How does batching strategy influence performance? Appendix E.2 indicates that *how* queries are batched matters, but less than expected. Sequential batching—grouping queries in their natural order—already delivers consistent improvements over instance-wise reflection across all datasets. This suggests that many benchmarks exhibit latent topical or structural coherence, allowing BoT-R to extract cross-instance signals even without explicit clustering. Semantic batching further improves performance on heterogeneous datasets by increasing within-batch similarity.

While larger batches theoretically provide richer comparative context, empirical results in Appendix E.1 reveal a non-monotonic relationship between batch size and performance, due to two constraints: (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 aggregation benefits.

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 perform comparative analysis, share knowledge across instances, and produce better-calibrated confidence while amortizing computation. BoT yields higher accuracy, lower token cost, and improved calibration across settings.

Limitations

The efficacy of Batch-of-Thought (BoT) is subject to several constraints. First, performance depends on batch formation: the core comparative reasoning assumes within-batch semantic relatedness. Poorly formed cohorts can cause negative transfer, degrading both calibration and accuracy. Second, BoT inherits the base model's context limits. For long-context queries, concatenating multiple items may approach or exceed the window, leading to truncation or failures. Finally, while we instantiate BoT in an Actor–Reflector architecture and the idea generalizes naturally to other multi-agent designs (e.g., Plan-and-Act, Debate), empirical validation of such integrations remains open. Systematically assessing the portability of batch-aware reasoning across alternative collaborative frameworks is an important direction for future work.

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 Altenschmidt, 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.

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 Xiangliang 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, Charles Stein, and 1 others. 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 Kosiorek, 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. 2021. *Sampling: design and analysis*. Chapman and Hall/CRC.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, 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, Kavita Kulkarni, 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.

Roman F Nalewajski. 2011. Elements of information theory. In *Perspectives in Electronic Structure Theory*, pages 371–395. Springer.

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 and 1 others. 1939. On the estimation of the discrepancy between empirical curves of distribution for two independent samples. *Bull. Math. Univ. Moscou*, 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. Autogen: 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

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), MMLU (Hendrycks et al., 2020), SMS Spam Detection (Yang et al., 2025b), and a newly curated fraud-seller detection dataset (Appendix G). Together, these benchmarks span scientific reasoning, commonsense inference, biomedical QA, broad academic knowledge, and real-world anomaly detection.

Confidence Calibration Metrics. We use two complementary measures: (a) the Kolmogorov–Smirnov (KS) statistic (Smirnov et al., 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 5 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

Method	SMS_Spam				GPQA				Winogrande			
	In	Out	Cost	Δ%	In	Out	Cost	Δ%	In	Out	Cost	Δ%
Actor												
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%
Reflector												
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%
Total												
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 5: 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.

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 30-50% 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.

C 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 (§C.3), which makes the coherence–diversity trade-off quantitative.

C.1 Setup

Definition C.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 $\mathcal{C}^{\mathrm{BoT}}$-measurable summary ϕ, write $\mathcal{G}_1 = \sigma(\mathcal{G}_0, \phi) \supseteq \mathcal{G}_0$. The Bayes-optimal confidences are $u_i^{\mathrm{ind}} := \mathbb{P}(z_i = 1 \mid \mathcal{G}_0)$ and $u_i^{\mathrm{BoT}} := \mathbb{P}(z_i = 1 \mid \mathcal{G}_1)$.

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

C.2 Calibration and discrimination gain

Proposition C.3 (Bayes-optimal batch reflection dominates)**.** *For any strictly proper scoring rule* S *in loss form (e.g., Brier loss, log-loss),* $\mathbb{E}\big[S(u_i^{\mathrm{BoT}}, z_i)\big] \leq \mathbb{E}\big[S(u_i^{\mathrm{ind}}, z_i)\big]$, $I(z_i; u_i^{\mathrm{BoT}}) \geq I(z_i; u_i^{\mathrm{ind}})$, *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 $q = \mathbb{P}(z_i = 1 \mid \mathcal{G}_1)(\omega) = u_i^{\mathrm{BoT}}(\omega)$. Since u_i^{ind} is $\mathcal{G}_0$-measurable and hence $\mathcal{G}_1$-measurable, plugging it in yields $\mathbb{E}[S(u_i^{\mathrm{ind}}, z_i) \mid \mathcal{G}_1] \geq \mathbb{E}[S(u_i^{\mathrm{BoT}}, z_i) \mid \mathcal{G}_1]$ a.s.; taking expectations gives the first inequality, with equality iff $u_i^{\mathrm{ind}} = u_i^{\mathrm{BoT}}$ 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: $I(z_i; u_i^{\mathrm{ind}}) = I(z_i; \mathcal{G}_0)$ and $I(z_i; u_i^{\mathrm{BoT}}) = I(z_i; \mathcal{G}_1)$, and the chain rule (Nalewajski, 2011, Thm. 2.5.2) gives $I(z_i; \mathcal{G}_1) - I(z_i; \mathcal{G}_0) = I(z_i; \phi \mid \mathcal{G}_0) \geq 0$. □

Scope. Proposition C.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 $\tilde{u}_i^{\mathrm{BoT}}$ 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 §C.3 characterizes; the second is validated by the calibration improvements in Table 3.

From proper losses to ECE. The Brier loss admits the decomposition $\mathrm{BS} = \mathrm{REL} - \mathrm{RES} + \mathrm{UNC}$ (Murphy, 1973), so Proposition C.3 implies a strictly lower $(\mathrm{REL} - \mathrm{RES})$ 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.

C.3 Effective sample size

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

Lemma C.4 (Effective sample size (Lohr, 2021, Ch. 5))**.** *Let* $p = \mathbb{P}(z_i = 1)$. *Under Assumption C.2,*

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

N_{eff} *is strictly increasing in* N *for* $\rho_e < 1$, *with* $N_{\mathrm{eff}} \to 1/\rho_e$ *as* $N \to \infty$ *when* $\rho_e > 0$, *and* $N_{\mathrm{eff}} = N$ *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, $I(z_i; \phi \mid \mathcal{G}_0) \to 0$) nor highly correlated batches (where $N_{\mathrm{eff}} \to 1$). 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 C.3.

C.4 Input-cost identity

Proposition C.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*

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

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 5), 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.

C.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* ($\rho_e \to 0$, so $I(z_i;\phi \mid \mathcal{G}_0) \to 0$): ϕ carries no information; the calibration and discrimination gains vanish, but the input-cost saving persists. *High correlation* ($\rho_e \to 1$): $N_{\text{eff}} \to 1$ 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.

D Related Work

D.1 Confidence Calibration in Large Language Models

Reliable uncertainty quantification remains critical for deploying LLMs in high-stakes applications. 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. **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.

D.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 prompt-**

ing (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.

D.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 feedback 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.

D.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.

E Batching Strategy Analysis

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

E.1 Batch Size Effects

Table 6 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 7.

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 C.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 C) 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.

E.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 8.

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 C.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.

F 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 (+1.4% 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

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 6: 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 7: 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.

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% respectively), BoT-R exhibits marginal decline relative to Reflection (-1.2% and -1.2%). 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.

Method	FraudDet	MedQA	PubMedQA	Winogrande	SMS Spam	GPQA
No-batch	0.693	0.901	0.667	0.879	0.854	0.459
Sequential	0.740 (4)	**0.904** (8)	**0.698** (8)	0.890 (8)	0.887 (8)	**0.488** (4)
Semantic	**0.768** (4)	0.902 (8)	0.697 (4)	**0.897** (8)	**0.902** (4)	0.486 (8)

Table 8: Accuracy comparison across batching strategies (GPT-4o). Numbers in parentheses indicate the batch size yielding the best result.

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

G 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.

H 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}]”