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 batches, 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) (OpenAI, 2023; Madaan et al., 2023; Wei et al., 2022; Shinn et al., 2023; Yao et al., 2023) 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., 2023b; Sun et al., 2024). However, producing reliable answers with well-calibrated confidence remains a challenge (Xiong et al., 2023; 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., 2023; 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.

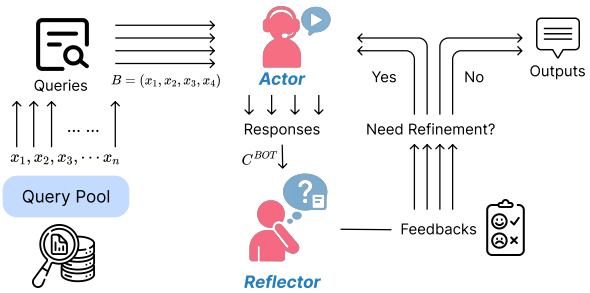

Figure 1: Batch-of-Thought reflection framework. An Actor generates initial responses for a batch of queries. A Reflector then jointly evaluates all responses through comparative analysis, determining whether each should be finalized or refined with feedbacks.

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 and Stein, 1961; Stein, 1956; Efron and Morris, 1977): pooling information across similar instances improves individual estimates through shrinkage toward the cohort distribution (see Appendix E 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., 2023; Liang et al., 2023). In BoT-R, an Actor generates answer-rationale pairs for a batch of queries, then a Reflector performs joint evaluation 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 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.

4. We introduce the **Seller Fraud Detection** benchmark for evaluating agentic reasoning in high-stakes scenarios, which we will publicly release upon publication.

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 $c_i \leftarrow \varnothing$, $u_i \leftarrow 0$ for all $i \in [N]$; active set $S \leftarrow [N]$
for $t = 1$ **to** T **do**
  (Parallel) $(a_i, \rho_i, \mathsf{traj}_i) \leftarrow \mathcal{A}(x_i, \mathcal{T}, c_i, K)$ for all $i \in S$
  Build reflective context $\mathcal{C} \leftarrow \langle (x_i, a_i, \rho_i, \mathsf{traj}_i) \rangle_{i=1}^N$
  (Joint) $(u_i, r_i, c_i) \leftarrow \mathcal{R}(\mathcal{C}, i)$ for all $i \in [N]$
  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., 2023) 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 $c_i^{(0)} = \varnothing$ for the initial iteration.

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

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

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:

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

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**	**0.488**	**0.890**	**0.904**	**0.698**	**0.887**
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**	**0.516**	**0.862**	**0.804**	**0.757**	0.923
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**	**0.657**	**0.874**	**0.860**	0.704	0.919

Table 1: Performance comparison of reasoning methods across various base models and datasets (number of queries listed as n). Scores represent accuracy. The best result in each setting is highlighted in bold.

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

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

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 E 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 five public benchmarks and one newly curated dataset and report results from both API and open-source models. Detailed information can be found in Appendix B.

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, 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., 2023), 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 (Dubey et al., 2024) and Qwen3-Next-80B (Yang et al., 2025). For the *Fraud Detection* dataset, we enable Brave Search and the Brave Summarizer as external tools for retrieval and grounding (Brave Software, Inc., 2025)

3.2 Main Results

As shown in Table 1, we compare task performance after integrating the proposed BoT method into the reflection framework. From this ablation study, BoT-R consistently achieves the highest accuracy across datasets and base API models, demonstrating the effectiveness of our Batch-of-Thought approach.

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)	$3.9	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. Each entry reports the KS statistic between the confidence distributions of correct vs. incorrect answers and the ECE score. Higher KS (↑) is better, while lower ECE (↓) is better.

Beyond accuracy, we examine how incorporating BoT influences efficiency. To enable fair comparison, all token costs are normalized using a unified reference pricing scheme based on the production-grade GPT-4o model. As reported in Table 2, BoT-R significantly reduces reflection overhead, achieving an average token-cost reduction of **46.9%** for the three benchmarks, and up to **61%** savings on the SMS_Spam benchmark.

Finally, we evaluate calibration performance using both the KS statistic and ECE score on three representative datasets under the GPT-4o backbone (Table 3). 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.

These results confirm that comparative, cohort-aware reasoning not only enhances output quality, confidence calibration but also improves cost efficiency. Additional details are provided in the Appendix C.

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.

4 Discussions

RQ1: Which domain benefits most from BoT? Table 4 reveals systematic variation in BoT's effectiveness across MMLU subject domains. BoT-R achieves strongest gains on interpretive tasks—humanities, social sciences, and medicine & biology—where multiple reasoning paths exist and comparative evaluation distinguishes robust from brittle arguments. Conversely, formal domains requiring precise symbolic manipulation (mathematics, physical sciences) show marginal decline, suggesting that batch-level consensus does not work in tasks where correctness depends on exact derivation rather than comparative plausibility. More details can be found in Appendix G.

RQ2: How does batching strategy influence performance? Appendix F shows that sequential batching—grouping queries in their given order achieves competitive performance, as naturally structured datasets often exhibit implicit topical coherence. On the other hand, semantic clustering via embedding-based grouping provides additional gains on heterogeneous datasets.

Larger batches theoretically increase mutual information gain, but two constraints emerge: (1) context window limits force rationale compression, and (2) heterogeneous batches introduce noise. Our experiments show that moderate batch sizes ($N \in \{4, 8\}$) optimize the accuracy-efficiency frontier under current model constraints. Future models with expanded context windows may enable larger batches without sacrificing rationale quality. More discussion in Appendix F.

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 items, and produce better-calibrated confidence while amortizing computation. BoT yields higher accuracy, lower token cost, and improved calibration across settings, including our proposed **Seller Fraud Detection** benchmark.

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 released 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

Anthropic. 2024. The Claude model family: Technical report. Accessed: 2025-10-07.

Brave Software, Inc. 2025. *Brave Web Search API Documentation*. Official API reference.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33*, pages 1877–1901.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2023. Chateval: Towards better llm-based evaluators through multi-agent debate. *arXiv preprint arXiv:2308.07201*.

Catherine Chen, Panupong Pasupat, and Percy Liang. 2024. Reconciling probability and logic in semantic parsing.

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, Singapore. Association for Computational Linguistics.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2023. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

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 *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135, Sydney, Australia. PMLR.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 1321–1330.

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, Jiaqi Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. 2023. MetaGPT: Meta programming for a multi-agent collaborative framework. *Preprint*, arXiv:2308.00352.

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 *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France. PMLR.

W. James and Charles Stein. 1961. Estimation with quadratic loss. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, pages 361–379, Berkeley, CA. 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 W Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. *arXiv preprint arXiv:1909.06146*.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zachary Dodds, Nova DasSarma, Eli Tran-Johnson, and 1 others. 2022. Language models (mostly) know what they know. *Preprint*, arXiv:2207.05221.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *Preprint*, arXiv:2302.09664.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R. Kosiorek, Seungjin Choi, and Yee Whye Teh. 2019. Set transformer: A framework for attention-based permutation-invariant neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3744–3753, Long Beach, CA, USA. 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, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. 2023. Encouraging divergent thinking in large language models through multi-agent debate. *arXiv preprint arXiv:2305.19118*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. In *Transactions on Machine Learning Research*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*.

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*, pages 1–7.

Eric Mitchell, Rafael Rafailov, Archit Sharma, Chelsea Finn, and Christopher D. Manning. 2023. Enhancing self-consistency and performance of pre-trained language models through natural language inference. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11685–11702, Singapore. Association for Computational Linguistics.

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*.

OpenAI. 2023. Gpt-4 technical report. *arXiv:2303.08774*.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2024. ChatDev: Communicative agents for software development. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15174–15186, Bangkok, Thailand. Association for Computational Linguistics.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

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.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. In *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS 2023)*.

Shiyu Si, Keping Bi, Jiafeng Guo, and Xueqi Cheng. 2024. When do LLMs need retrieval augmentation? mitigating LLMs overconfidence helps retrieval augmentation. *Preprint*, arXiv:2402.11457.

Karan Singhal, Tao Tu, Juraj Gottweis, R. Sayres, Ellery Wulczyn, Mohamed Amin, Le Hou, Kevin Clark, Stephen R. Pfohl, Heather Cole-Lewis, Darlene Neal, Q. Rashid, Mike Schaekermann, Amy Wang, Dev Dash, Jonathan H. Chen, Nigam H. Shah, Sami Lachgar, P. Mansfield, and 16 others. 2025. Toward expert-level medical question answering with large language models. *Nature Medicine*, 31:943 – 950.

N. V. Smirnov. 1939. On the estimation of the discrepancy between empirical curves of distribution for two independent samples. *Bulletin of Moscow University*, 2(2).

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems 30*, pages 4077–4087. Curran Associates, Inc.

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*, pages 197–206, Berkeley, CA. 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.

Kevin Tian, Stephanie Lin, Jacob Hilton, and Owain Evans. 2023. Just say what you know: Improving calibration by verbalizing model uncertainty. *Preprint*, arXiv:2310.01846.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2023a. Improving text embeddings with large language models. *arXiv preprint arXiv:2401.00368*.

Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun R Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. 2023b. Scibench: Evaluating college-level scientific problem-solving abilities of large language models. *arXiv preprint arXiv:2307.10635*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023c. Self-consistency improves chain of thought reasoning in language models. *arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*.

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. *Preprint*, arXiv:2308.08155.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2023. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. *arXiv preprint arXiv:2306.13063*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Tiankai Yang, Yi Nian, Shawn Li, Ruiyao Xu, Yuangang Li, Jiaqi Li, Zhuo Xiao, Xiyang Hu, Ryan Rossi, Kaize Ding, and 1 others. 2024. Ad-llm: Benchmarking large language models for anomaly detection. *arXiv preprint arXiv:2412.11142*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. *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 *The Twelfth International Conference on Learning Representations*.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan Salakhutdinov, and Alexander Smola. 2017. Deep sets. In *Advances in Neural Information Processing Systems 30*, pages 3391–3401. Curran Associates, Inc.

A 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.

B Experiment Settings

Datasets. We evaluate BoT on five public benchmarks: GPQA (graduate-level Google-proof multiple-choice questions in biology, physics, and chemistry) (Rein et al., 2024), WinoGrande–debiased (pronoun-resolution problems) (Sakaguchi et al., 2021), PubMedQA (biomedical question answering over PubMed abstracts with yes/no/maybe labels)(Jin et al., 2019), MedQA(USMLE) (multiple-choice questions drawn from medical licensing examinations)(Jin et al., 2021), MMLU (57-task multiple-choice questions spanning broad academic disciplines) (Hendrycks et al., 2020), SMS Spam (an anomaly-detection formulation of the SMS Spam Collection where "spam" is treated as the anomaly class)(Yang et al., 2024), and a newly released fraud-seller detection corpus (Appendix A).

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.

C 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 and Reflector—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.

C.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

Method	SMS_Spam				GQPA				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*, *Reflection*, 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.

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.

C.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 42% to 57% 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.

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., 2023; Kadavath et al., 2022). This miscalibration persists even in state-of-the-art models (OpenAI, 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., 2023c), 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., 2023; Lin et al., 2022) or categorical (Tian et al., 2023) confidence estimates. These methods are ef-

ficient but highly sensitive to prompt formatting (Si et al., 2024) and often produce overconfident predictions (Kadavath et al., 2022). Recent efforts employ chain-of-thought reasoning for confidence elicitation (Xiong et al., 2023) 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., 2023), 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., 2023) 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.

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. However, these methods rely on predefined examples or retrieval systems rather than jointly reasoning over a batch of target queries. Recent work on **self-consistency with rationalization** (Mitchell et al., 2023) 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 and comparable accuracy to self-consistency 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 Theoretical Analysis

This section establishes formal foundations for batch-aware reasoning in LLM-based systems. We prove that joint processing of related queries provides information-theoretic advantages over independent processing, characterize conditions under

which these benefits manifest, and derive efficiency guarantees for batch-level computation.

E.1 Preliminaries and Problem Formulation

Definition E.1 (Batch reasoning problem)**.** *Let $\mathcal{D}$ be a distribution over query-answer pairs $\mathcal{X} \times \mathcal{Y}$. A batch $B = \{(x_i, y_i^*)\}_{i=1}^N$ consists of N instances drawn from $\mathcal{D}$. An Actor agent $\mathcal{A}$ produces initial predictions $\hat{y}_i^{(0)} = \mathcal{A}(x_i)$ with reasoning traces ρ_i. The batch context is*

$$\mathcal{C}^{\mathrm{BoT}} = \{(x_j, \hat{y}_j^{(0)}, \rho_j)\}_{j=1}^N. \tag{6}$$

A Reflector agent $\mathcal{R}$ performs joint analysis over $\mathcal{C}^{\mathrm{BoT}}$ to produce for each instance $i \in [N]$:

$$(\mathbf{r}_i, u_i, c_i) = \mathcal{R}(\mathcal{C}^{\mathrm{BoT}}, i), \tag{7}$$

where $\mathbf{r}_i \in \{0, 1\}$ indicates re-evaluation necessity, $u_i \in [0, 1]$ quantifies confidence in correctness, and c_i provides actionable critique.

Assumption E.2 (Batch coherence)**.** *The batch exhibits structural coherence with the following properties:*

(a) ***Exchangeability:*** *The joint distribution of $\{(x_i, y_i^*)\}_{i=1}^N$ is invariant under permutations.*

(b) ***Similarity structure:*** *There exists a similarity function $\mathrm{sim} : \mathcal{X} \times \mathcal{X} \to [0, 1]$ such that the average pairwise similarity satisfies $\mathbb{E}[\mathrm{sim}(x_i, x_j)] \geq \kappa$ for some coherence parameter $\kappa \in (0, 1]$.*

(c) ***Error correlation:*** *Define error indicators $e_i = \mathbf{1}[\hat{y}_i^{(0)} \neq y_i^*]$. Under coherence, errors exhibit positive correlation: $\mathrm{Cor}(e_i, e_j) = \rho_e(\kappa) > 0$ for $i \neq j$, where $\rho_e(\cdot)$ is non-decreasing in coherence strength.*

E.2 Information-Theoretic Foundation for Calibration Improvement

We first establish that batch-level processing provides strictly more information for confidence estimation than independent processing.

Theorem E.3 (Batch processing improves proper scoring rules)**.** *Let $\mathcal{G}_0 = \sigma(\hat{y}_i^{(0)}, \rho_i)$ represent the σ-algebra generated by instance i alone, and $\mathcal{G}_1 = \sigma(\hat{y}_i^{(0)}, \rho_i, \phi)$ where $\phi = \phi(\mathcal{C}^{\mathrm{BoT}})$ denotes batch-level statistics. For any strictly proper scoring rule $S : [0, 1] \times \{0, 1\} \to \mathbb{R}$ (e.g., Brier score, log-loss), the batch-aware confidence predictor*

$$u_i^{\mathrm{BoT}} = \mathbb{P}(\hat{y}_i^{(0)} = y_i^* \mid \mathcal{G}_1) \tag{8}$$

satisfies

$$\mathbb{E}[S(u^{\mathrm{BoT}}, z)] \leq \mathbb{E}[S(u^{\mathrm{ind}}, z)], \tag{9}$$

where $u^{\mathrm{ind}} = \mathbb{P}(\hat{y}_i^{(0)} = y_i^* \mid \mathcal{G}_0)$ and $z = \mathbf{1}[\hat{y}_i^{(0)} = y_i^*]$. The inequality is strict when ϕ is informative: $I(z; \phi \mid \mathcal{G}_0) > 0$.

Proof. By definition of conditional expectation and the tower property,

$$u_i^{\mathrm{BoT}} = \mathbb{E}[z \mid \mathcal{G}_1] = \mathbb{E}[\mathbb{E}[z \mid \mathcal{G}_0] \mid \mathcal{G}_1] = \mathbb{E}[u^{\mathrm{ind}} \mid \mathcal{G}_1]. \tag{10}$$

Since S is strictly proper, the predictor u^{BoT} is optimal among all $\mathcal{G}_1$-measurable predictors. By the law of total expectation,

$$\mathbb{E}[S(u^{\mathrm{BoT}}, z)] = \mathbb{E}[\mathbb{E}[S(u^{\mathrm{BoT}}, z) \mid \mathcal{G}_1]] \tag{11}$$

$$\leq \mathbb{E}[\mathbb{E}[S(u^{\mathrm{ind}}, z) \mid \mathcal{G}_1]] = \mathbb{E}[S(u^{\mathrm{ind}}, z)], \tag{12}$$

where the inequality follows from optimality of u^{BoT} with respect to $\mathcal{G}_1$. Strict inequality holds when $u^{\mathrm{BoT}} \neq u^{\mathrm{ind}}$ with positive probability, which occurs precisely when $I(z; \phi \mid \mathcal{G}_0) > 0$. □

Remark E.4 (Connection to Expected Calibration Error)**.** *While Expected Calibration Error (ECE) is not a proper scoring rule, empirical calibration typically improves when confidence predictors condition on additional informative statistics. Theorem E.3 provides theoretical justification for observed ECE reductions under batch processing: by extracting cross-instance statistics ϕ through comparative analysis, the Reflector produces better-calibrated confidence estimates than those based solely on single-instance features.*

E.3 Collective Signal Amplification for Error Detection

We now quantify how batch-level aggregation amplifies signals for error detection, explaining improved separation in confidence distributions between correct and incorrect predictions.

Proposition E.5 (Effective sample size under correlation)**.** *Let $z_i = \mathbf{1}[\hat{y}_i^{(0)} = y_i^*]$ denote correctness indicators with $\mathbb{E}[z_i] = p$ and equicorrelation structure $\mathrm{Cor}(z_i, z_j) = \rho_c \in [0, 1)$ for all $i \neq j$. Define the effective sample size*

$$N_{\mathrm{eff}} = \frac{N}{1 + (N - 1)\rho_c}. \tag{13}$$

Then the batch-average correctness $M_N = \frac{1}{N}\sum_{i=1}^{N} z_i$ *has variance*

$$\mathrm{Var}(M_N) = \frac{p(1-p)}{N_{\text{eff}}}. \tag{14}$$

Furthermore:

(i) If $\rho_c = O(1/N)$*, then* $N_{\text{eff}} = \Theta(N)$ *and* M_N *concentrates at rate* $O(1/\sqrt{N})$*.*

(ii) If $\rho_c = \rho_0 > 0$ *is constant, then* $N_{\text{eff}} \to 1/\rho_0$ *as* $N \to \infty$*, and concentration gains saturate.*

Proof. For exchangeable binary random variables with equicorrelation ρ_c,

$$\mathrm{Var}(M_N) = \frac{1}{N^2}\sum_{i=1}^{N}\mathrm{Var}(z_i) + \frac{1}{N^2}\sum_{i\neq j}\mathrm{Cov}(z_i, z_j) \tag{15}$$

$$= \frac{1}{N^2}\cdot N\cdot p(1-p) + \frac{1}{N^2}\cdot N(N-1)\cdot \rho_c p(1-p) \tag{16}$$

$$= \frac{p(1-p)}{N}\left[1 + (N-1)\rho_c\right] = \frac{p(1-p)}{N_{\text{eff}}}. \tag{17}$$

The asymptotic regimes follow directly from the definition of N_{eff}. □

Corollary E.6 (Confidence separation for error detection)**.** *When* N_{eff} *is large, batch-level consensus* M_N *provides a reliable collective signal. Instances whose predictions deviate from consensus receive adjusted confidence scores. For fixed instance-level accuracy* $p > 1/2$*, the Kolmogorov-Smirnov (KS) statistic measuring separation between confidence distributions of correct and incorrect predictions increases with* N_{eff}*.*

Remark E.7 (Optimal batch composition)**.** *Proposition E.5 reveals a fundamental trade-off: high coherence* κ *enables effective cross-instance learning but may increase error correlation* ρ_e*, reducing* N_{eff}*. Optimal batches satisfy:*

- ***Sufficient similarity:*** $\kappa > \kappa_{\min}$ *to enable pattern extraction and knowledge transfer.*
- ***Sufficient diversity:*** $\rho_e < 0.5$ *to maintain* $N_{\text{eff}} > 0.67N$*, ensuring reliable collective signals.*

For batch sizes $N \in \{4, 8\}$ *used in practice, this yields* $N_{\text{eff}} \in [2.7, 5.3]$ *when* $\rho_e \approx 0.3$*, providing meaningful collective signal while preserving diversity.*

E.4 Computational Efficiency Through Amortization

Proposition E.8 (Sublinear cost scaling)**.** *Let* T_{inst}*,* T_{ctx}*, and* T_{out} *denote token counts for reflection instructions, per-instance context, and per-instance output, respectively. Independent reflection incurs total cost*

$$C_{\text{ind}} = N\cdot(T_{\text{inst}} + T_{\text{ctx}} + T_{\text{out}}). \tag{18}$$

Batch-aware reflection with shared comparative analysis costs

$$C_{\text{BoT}} = T_{\text{inst}} + N\cdot T_{\text{ctx}} + S(N), \tag{19}$$

where $S(N)$ *is the joint Reflector output length. When critiques reference shared reasoning structures and cross-instance insights are reused,* $S(N)$ *exhibits sublinear growth:* $S(N) = O(N^{\beta})$ *with* $\beta < 1$*.*

E.5 Characterization of Favorable Conditions

We synthesize the preceding results to characterize when batch-aware reasoning provides advantages.

Theorem E.9 (Conditions for BoT effectiveness)**.** *Batch-aware reasoning via BoT provides improvements over independent processing in calibration (lower ECE), error detection (higher KS statistic), and efficiency when the following conditions hold:*

(i) ***Coherence:*** *Batch exhibits sufficient similarity structure with* $\kappa > \kappa_{\min}$*, enabling pattern extraction and knowledge transfer.*

(ii) ***Moderate correlation:*** *Error correlation satisfies* $\rho_e \in (0, 0.5)$*, ensuring* $N_{\text{eff}} > 0.5N$ *for collective signal reliability while preserving diversity.*

(iii) ***Informative batch statistics:*** *Cross-instance features* $\phi(\mathcal{C}^{\text{BoT}})$ *satisfy* $I(z_i; \phi \mid \mathcal{G}_0) > 0$*, providing additional information beyond single-instance features.*

(iv) ***Adequate batch size:*** $N \geq N_{\min}$ *for reliable collective signal extraction. For typical correlation* $\rho_e \approx 0.3$*, batch sizes* $N \in \{4, 8\}$ *yield* $N_{\text{eff}} \in [2.7, 5.3]$*.*

Under these conditions, the following guarantees hold:

- ***Calibration:*** *By Theorem E.3, batch-aware confidence* u^{BoT} *achieves lower expected loss for proper scoring rules.*

- ***Error detection:*** *By Corollary E.6, confidence distributions exhibit increased separation with* N_{eff}.

- ***Efficiency:*** *By Proposition E.8, sublinear output scaling yields* $C_{\text{BoT}}/C_{\text{ind}} < 1$ *for* $N \geq 2$.

Remark E.10 (Failure modes and graceful degradation)**.** *BoT degrades toward independent processing when:*

- ***No coherence*** *(*$\kappa \approx 0$*): Instances lack shared structure; cross-instance statistics* ϕ *are uninformative.*

- ***High correlation*** *(*$\rho_e \to 1$*): All instances make identical errors;* $N_{\text{eff}} \to 1$*, eliminating collective signal benefits.*

- ***Insufficient size*** *(*$N < N_{\min}$*): Collective signals are unreliable due to high sampling variance.*

Importantly, performance degrades gracefully as N_{eff} *decreases continuously with* ρ_e*, rather than exhibiting catastrophic failure.*

E.6 Summary

Our theoretical analysis establishes rigorous foundations for batch-aware reasoning:

- **Information gain (Theorem E.3):** Batch statistics ϕ provide additional information, improving calibration through optimal conditioning on $\mathcal{G}_1 \supset \mathcal{G}_0$.

- **Effective sample size (Proposition E.5):** Quantifies collective signal strength via N_{eff}, explaining KS statistic improvements under moderate correlation.

- **Computational efficiency (Proposition E.8):** Sublinear output scaling yields provable cost reductions for $N \geq 2$.

- **Effectiveness conditions (Theorem E.9):** Characterizes when BoT succeeds, providing actionable guidance for batch construction and domain selection.

These results not only explain empirical findings but also provide principled guidelines for applying batch-aware reasoning to new domains and tasks.

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 6 presents accuracy across batch sizes $N \in \{1, 4, 8\}$, where $N = 1$ corresponds to standard per-instance reflection. The results reveal non-monotonic relationships between batch size and performance, with optimal configurations varying by task characteristics.

Most datasets exhibit peak performance at moderate batch sizes, suggesting that batch size interacts with task structure in complex ways. Our theoretical analysis (Appendix E) predicts that larger batches increase mutual information gain by providing richer comparative context. However, three 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 $N = 8$, 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

To isolate the effect of query relatedness, we compare three batching strategies on Winogrande and PubMedQA:

Sequential batching: Group queries in their original dataset order (default strategy in main experiments).

Semantic batching: Cluster queries by embedding similarity using K-means on E5-Mistral-7B embeddings (Wang et al., 2023a), then form batches from cluster members sorted by centroid proximity.

Batch	FraudDet	GPQA	Winogrande	MedQA	PubMedQA	SMS Spam
1	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

Dataset	Strategy	Batch=4	Batch=8
Winogrande	Sequential	0.888	0.890
	Semantic	0.883	**0.897**
PubMedQA	Sequential	0.683	**0.698**
	Semantic	0.697	0.696

Table 7: Comparison of sequential vs. semantic (K-means clustering) batching strategies. Semantic batching groups queries by E5-Mistral-7B embedding similarity before forming fixed-size batches.

Table 7 presents results. On Winogrande, semantic batching at $N = 8$ achieves 0.897 accuracy (+0.7% over sequential), demonstrating that grouping structurally similar pronoun resolution problems enhances comparative reasoning. The gain is smaller at $N = 4$ and PubMedQA dataset, suggesting that semantic clustering provides diminishing returns when batches are already moderately coherent due to dataset structure.

Implications for batch composition. These results suggest that batching strategy should adapt to dataset characteristics:
(1) For heterogeneous datasets with diverse query types (MMLU, FraudDet), semantic clustering can improve performance by ensuring within-batch coherence, particularly at larger batch sizes where random grouping risks mixing incompatible problems.
(2) For naturally structured datasets (PubMedQA, domain-specific benchmarks), sequential batching suffices, as the original ordering already reflects meaningful clustering.

G 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 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.

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 8: MMLU per-subject accuracy summary grouped by category.

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}]"