Speculative Decoding for Multi-Sample Inference

Yiwei Li[1], **Jiayi Shi**[1], **Shaoxiong Feng**[2†], **Peiwen Yuan**[1], **Xinglin Wang**[1], **Yueqi Zhang**[1], **Ji Zhang**[1], **Chuyi Tan**[1], **Boyuan Pan**[2], **Yao Hu**[2], **Kan Li**[1†]
[1] School of Computer Science, Beijing Institute of Technology
[2] Xiaohongshu Inc
{liyiwei,shijiayi,peiwenyuan,wangxinglin}@bit.edu.cn
{shaoxiongfeng2023}@gmail.com {panboyuan,xiahou}@xiaohongshu.com
{zhangyq,jizhang,tanchuyi,likan}@bit.edu.cn

Abstract

We propose a novel speculative decoding method tailored for multi-sample reasoning scenarios, such as self-consistency and Best-of-N sampling. Our method exploits the intrinsic consensus of parallel generation paths to synthesize high-quality draft tokens without requiring auxiliary models or external databases. By dynamically analyzing structural patterns across parallel reasoning paths through a probabilistic aggregation mechanism, it identifies consensus token sequences that align with the decoding distribution. Evaluations on mathematical reasoning and code generation benchmarks demonstrate a substantial improvement in draft acceptance rates over baselines, while reducing the latency in draft token construction. This work establishes a paradigm shift for efficient multi-sample inference, enabling seamless integration of speculative decoding with sampling-based reasoning techniques.

1 Introduction

Nowadays, large language models (LLMs) increasingly rely on multi-sample aggregation strategies, such as majority voting (self-consistency) (Wei et al., 2022) and Best-of-N sampling (Cobbe et al., 2021a), to enhance prediction accuracy in complex reasoning tasks. By generating and aggregating multiple candidate outputs, these methods mitigate individual sampling errors and improve task performance, particularly in mathematical reasoning (Lu et al., 2023). However, this benefit comes at a significant computational cost: generating multiple samples inherently prolongs inference latency, posing a critical challenge for real-world applications where efficiency is paramount.

To address latency issues, speculative decoding (Leviathan et al., 2023; Chen et al., 2023a; Miao et al., 2023) has emerged as a promising acceleration method. By predicting draft tokens in advance and verifying them against the original LLM, this approach reduces the number of full autoregressive steps. However, existing methods depend on external draft models (Li et al., 2024d) or retrieval-augmented databases (He et al., 2024) to generate drafts, introducing three critical limitations: First, draft tokens sourced externally struggle to fully align with the original LLM's output distribution, which lowers token acceptance rates and diminishes the potential speedup. Second, querying these external modules or databases inevitably incurs additional latency. Third, relying on auxiliary resources incurs extra costs—training a draft model or maintaining a database require additional computational and storage overhead.

A critical observation motivates our work: multi-sample inference yields a rich reservoir of high-quality draft tokens. When generating multiple parallel reasoning paths (e.g., for majority voting), the LLM produces outputs that, while varied in expression, are all drawn from the same underlying distribution. Frequently, these outputs share common substeps—such as intermediate equations in math problems—while differing in ordering or phrasing. Importantly, because all paths stem from the same model, their substeps naturally mirror its output tendencies, unlike externally sourced drafts. This inherent alignment makes them ideal candidates for speculative decoding, provided we can effectively extract and aggregate consensus patterns.

In this paper, we introduce a novel speculative decoding method explicitly designed for multi-sample reasoning scenarios. Our approach begins by leveraging the overlapping token subsequences generated across parallel reasoning paths to build a dynamic draft token pool. At each inference step, the most recent tokens from any path are used as queries to retrieve matching prefixes from other paths via suffix-based matching. The tokens immediately following these matches are aggregated as candidate drafts. These candidates are then orga-

†Corresponding authors.

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 12523–12533
November 4-9, 2025

nized into a weighted directed acyclic graph (DAG), where the edge weights reflect transition probabilities derived from the LLM's distribution. A confidence-weighted search is performed on this DAG to extract the highest-likelihood token sequence—prioritizing paths with strong agreement. This consensus-driven process yields draft tokens that closely align with the model's output distribution, ultimately accelerating inference by reducing the number of full autoregressive steps without the need for external modules or datastores.

We evaluate our method on four benchmarks using three widely adopted LLMs under multi-sample inference settings. Our approach achieves much higher token acceptance rate at identical draft lengths, demonstrating superior alignment with the original model's distribution. Crucially, the draft construction process incurs lower latency than baselines, as it eliminates costly database queries or auxiliary model inferences.

2 Background

Multi-Sample Inference Aggregation strategies aim to enhance reasoning reliability by generating and aggregating multiple candidate outputs. Let $Y = \{y_1, y_2, \ldots, y_N\}$ denote N parallel samples drawn from a language model p_θ. Two dominant paradigms include:

- Self-Consistency: Selects the final answer via majority voting over Y:

$$\hat{y} = \arg\max_{a \in \mathcal{A}} \sum_{i=1}^{N} \mathbb{I}(a = y_i), \tag{1}$$

  where $\mathcal{A}$ is the answer space and $\mathbb{I}(\cdot)$ is the indicator function.

- Best-of-N: Scores each sample y_i with a reward function $G(y_i|x)$ and selects the highest-ranked candidate:

$$\hat{y} = \arg\max_{y_i \in Y} G(y_i|x), \tag{2}$$

  where G may quantify sequence likelihood, alignment with domain-specific heuristics, or external verification signals.

While aggregation techniques enhance accuracy, generating multiple samples increases latency, thereby motivating the development of acceleration methods.

Speculative Decoding Speculative decoding reduces inference latency by predicting K-step draft tokens $\{\tilde{y}^1, \ldots, \tilde{y}^K\}$ in advance from draft model q and verifying them via the original model p_θ. Traditional approaches rely on an auxiliary draft model or a retrieval datastore D to propose drafts. For each step t:

- Draft Proposal: Generate $\tilde{y}^t \sim q(\cdot|y^{1:t-1})$ or retrieve $\tilde{y}^t$ from D.
- Parallel Verification: Compute $p_\theta(\cdot|y^{1:t-1})$ for all $\tilde{y}^{1:t}$.
- Draft verification: Accept the longest prefix $\tau \leq K$ where $p_\theta(\tilde{y}^t|y^{1:t-1}) \geqq (\tilde{y}^t|y^{1:t-1})$.

3 Method

Our method transforms the inherent consensus of multi-sample reasoning paths into high-quality draft tokens through three key components: (1) dynamic construction of a draft pool via cross-path suffix searching, (2) fuzzy suffix matching to handle lexical variations and (3) extraction of consensus subsequences using a DAG structure built over aligned reasoning paths. The entire process operates during parallel decoding, requiring no external models or datastores. Please refer to Appendix A for the algorithm.

Dynamic Draft Pool Construction During parallel generation of N reasoning paths $\{y_1, y_2, \ldots, y_N\}$, we iteratively build a draft pool by identifying overlapping token subsequences across paths. For any partial sequence $y_i^{1:t}$ on the path i, we use its k-token suffix $y_i^{t-k+1:t}$ as a query to search for matching prefixes in other paths $\{y_j\}_{j \neq i}$. Matched prefixes' subsequent tokens are aggregated as draft candidates. Formally, the candidate set $\mathcal{C}_t$ at step t is:

$$\mathcal{C}_t = \bigcup_{j \neq i} \left\{ y_j^{t':} \mid \exists t' \leq t, y_j^{1:t'} \text{ ends with } y_i^{t-k+1:t} \right\}. \tag{3}$$

Candidates inherently align with p_θ, as all paths are derived from the same model.

Fuzzy Suffix Matching To handle minor lexical variations (e.g., "x²" vs. "x^2" in mathematical steps), we extend exact suffix matching with edit distance tolerance ϵ. For query suffix s, we retrieve all s' where:

$$\delta(s, s') \leq \epsilon, \tag{4}$$

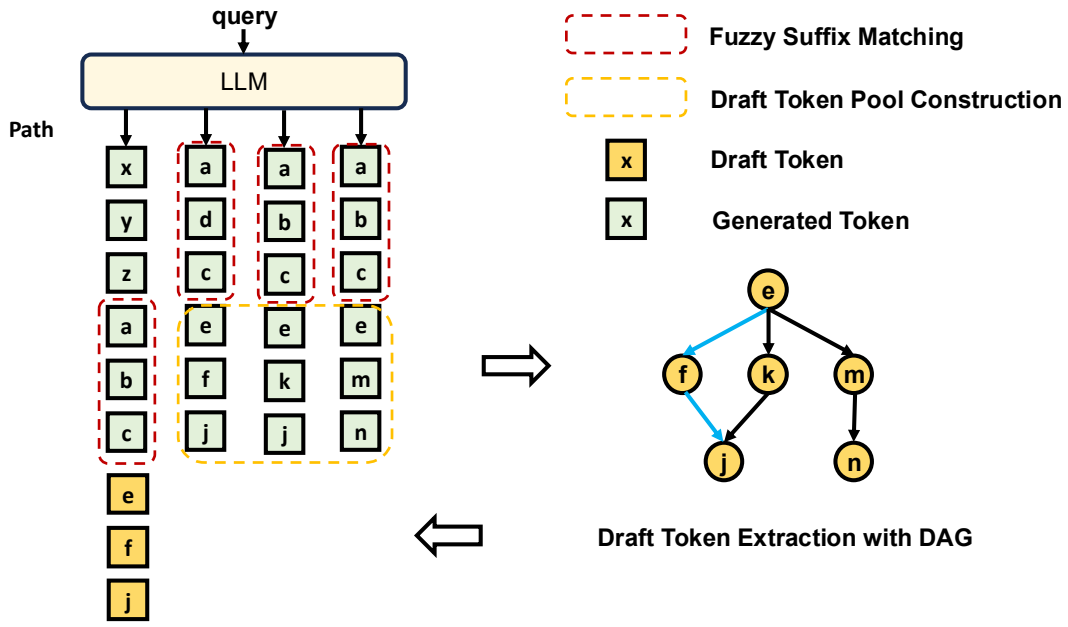

Figure 1: The framework of proposed method.

ensuring semantically equivalent tokens contribute to the draft pool.

Consensus-Driven Draft Extraction with DAG We organize candidates into a **directed acyclic graph (DAG)** $\mathcal{G} = (V, E)$, where nodes (V) represent unique tokens, and identical tokens that appear within the same layer (i.e., at the same time step) across different candidates are merged to form the graph. Edges (E) encode transitions weighted by:

$$w(u, v) = \alpha \cdot \sum_i p_\theta(v_i|u_i) + (1-\alpha) \cdot \text{Freq}(u \to v), \tag{5}$$

where $p_\theta(v_i|u_i)$ represents the model's probability of generating v from u in the i-th instance. We weight through generation probabilities to prevent the selection of low-probability tokens, thereby improving the accept rate. We extract the optimal draft sequence by greedily selecting the token with the highest weight ($\sum_{u:(u,v)\in E} w(u, v)$) at each layer. The search terminates upon reaching either the maximum draft token length L or a leaf node.

There are three advantages of DAG over tree structure: **Compact Structure**: Merges shared tokens (e.g., common math operators) into single nodes. **Multi-Path Support**: Captures divergent reasoning branches (e.g., "x+1" vs. "x-1") via multiple edges. **Probabilistic Ranking**: Edge weights quantify consensus between paths.

Draft acceptance We follow He et al. (2024), where draft tokens are validated by comparing them with tokens sampled from the model's conditional distribution at each position. Accepted draft tokens are those that match the sampled tokens, and any mismatch leads to rejection of subsequent drafts. This method ensures that the generated sequences align with standard autoregressive generation without any loss of accuracy.

4 Experiment

4.1 Experimental Setup

Datasets and Models We evaluate our method on two widely-used mathematical reasoning benchmarks: GSM8K (Cobbe et al., 2021b) and MATH (Hendrycks et al., 2021) and two code generation benchmarks: MBPP (Austin et al., 2021) and HumanEval (Chen et al., 2021). We test on Qwen2.5-7B-Instruct, Qwen2.5-14B-Instruct (Yang et al., 2024) and Llama-3-8B-Instruct (Dubey et al., 2024) models. For simplicity, edit distance tolerance ϵ and α is set to 1. The effect of the hyperparameters is detailed in Appendix C.

Baselines We compare our method with REST (He et al., 2024) and EAGLE-2 (Li et al., 2024d). REST retrieves draft tokens from a pre-built datastore, where we built it with NuminalMath-CoT (Li et al., 2024a) and BigCode dataset (Kocetkov et al., 2023). EAGLE-2 leverages both the token sequence and the hidden state sequence to sequentially predict subsequent draft tokens. When comparing performance, we adopt accept length and

Table 1: Accepted length under different draft lengths (L) for REST, EAGLE-2, and our method on GSM8K and MBPP. Please refer to Appendix D for MATH and HumanEval dataset.

Model	Dataset	Qwen2.5-7B-Instruct				Qwen2.5-14B-Instruct				LLaMA3-8B-Instruct			
		L=6	4	3	2	L=6	4	3	2	L=6	4	3	2
REST	GSM8K	0.56	0.49	0.41	0.25	0.28	0.24	0.19	0.20	0.52	0.46	0.41	0.26
EAGLE-2	GSM8K	0.97	0.87	0.70	0.65	0.88	0.78	0.65	0.63	1.71	1.41	1.32	**1.12**
Ours	GSM8K	**1.53**	**1.46**	**1.35**	**1.12**	**1.45**	**1.42**	**1.31**	**1.10**	**2.17**	**1.79**	**1.53**	**1.12**
REST	MBPP	0.19	0.19	0.18	0.18	0.26	0.23	0.17	0.17	0.31	0.26	0.21	0.20
EAGLE-2	MBPP	0.78	0.78	0.75	0.73	0.96	0.86	0.70	0.69	1.63	1.33	1.32	**1.06**
Ours	MBPP	**0.95**	**0.94**	**0.93**	**0.82**	**2.95**	**2.15**	**1.76**	**1.26**	**1.93**	**1.56**	**1.33**	1.02

latency of draft token construction as evaluation metrics, since baseline methods do not optimize inference time for batch size greater than one, making them incompatible with multi-sample inference. However, these two metrics ensure a fair comparison, as they are orthogonal to parallel speculative decoding. Please refer to Appendix B for details.

4.2 Main Results

Draft Tokens Acceptation Table 1 presents the accept length of our method and the baseline approaches under different draft lengths, where our method demonstrates a clear advantage. Datastore-based method exhibits lower reception lengths, primarily because they rely on the distribution of the database, which does not align with the model's distribution. Draft model based method achieves better alignment with the original LLM due to dedicated training, but their smaller parameter size limits their ability to fully match the original model. In comparison, draft tokens from our method are consistently sourced from the same LLM, ensuring a perfectly matched distribution, which results in a higher acceptance rate.

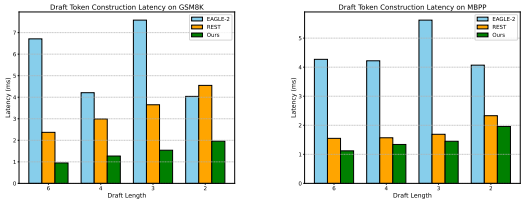

Figure 2: Latency (ms) of draft token construction by Accept Length on Qwen2.5-7B-Instruct for GSM8K and MBPP. Our method generally requires less time to construct draft tokens.

Draft Tokens Construction Latency Figure 2 reports the time consumption of different methods for constructing draft tokens. Among them, EAGLE-2 incurs the highest computational cost due to its reliance on GPU-based inference during construction. Our method is more efficient than REST, as our candidate set is significantly smaller than that of large-scale databases.

4.3 Analysis

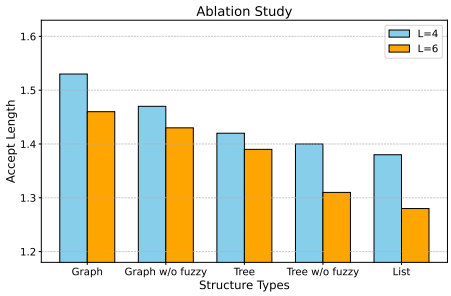

Figure 3: Results of ablation study on Qwen2.5-7B-Instruct for GSM8K. DAG data structure with fuzzy matching achieved the best performance.

Ablation Study Figure 3 presents the results of the ablation study, demonstrating the effectiveness of different components of our method. It can be observed that, in modeling the data structure of the draft pool, DAG outperforms the tree, which in turn outperforms the list. This indicates that the DAG better captures consensus information among samples, leading to a higher acceptance rate. Additionally, we find that even when using a list, our method still achieves competitive performance, confirming that our approach generates high-quality draft tokens. Removing fuzzy matching leads to a performance drop, highlighting its role in enhancing the robustness of the draft pool.

Effect of Temperature We investigate the impact of temperature on accept length. As shown in Figure 4 (left), accept length exhibits a trend of first increasing and then decreasing as temperature rises. This is because, at lower temperatures, the diversity among samples is low, resulting in an insufficient

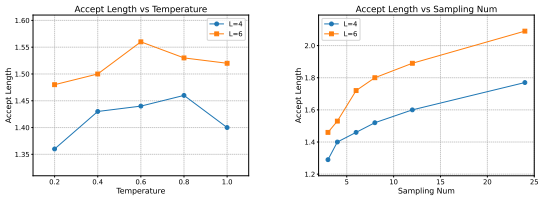

Figure 4: The effect of sampling temperature (left) and size (right) on accept length on Qwen2.5-7B-Instruct for GSM8K.

number of candidates in the draft pool. Conversely, at higher temperatures, the diversity among samples increases, reducing the success rate of suffix matching. Therefore, the optimal temperature lies in the middle range.

Effect of Sampling Number We further examine the impact of sampling size on accept length under the same draft length. As shown in Figure 4 (right), accept length increases as the sampling size grows. This is because a larger parallel sampling size provides a richer set of candidates in the draft pool, leading to a higher acceptance rate for the selected draft tokens.

5 Conclusion

We present a speculative decoding method that leverages consensus patterns from parallel reasoning paths to accelerate multi-sample inference, eliminating external dependencies. By dynamically synthesizing draft tokens through probabilistic aggregation and graph-based selection, our approach achieves higher acceptance rates and lower latency than baselines. This work bridges speculative decoding with sampling-based reasoning, establishing a resource-efficient paradigm for accelerating LLMs while preserving their output distribution. This approach could pave the way for the implementation of efficient multi-sample strategies in real-world applications.

Limitations

Our method has three key limitations:

- Our method is not combined with speculative decoding methods optimized for batch processing, so its acceleration effect on time requires further exploration.

- The probabilistic aggregation mechanism incurs non-negligible overhead when processing a large number of parallel paths, potentially diminishing latency gains in extreme-scale scenarios.

Ethics Statement

All of the datasets used in this study were publicly available, and no annotators were employed for our data collection. We confirm that the datasets we used did not contain any harmful content and was consistent with their intended use (research). We have cited the datasets and relevant works used in this study.

References

Pranjal Aggarwal, Aman Madaan, Yiming Yang, and Mausam. 2023. Let's sample step by step: Adaptive-consistency for efficient reasoning and coding with LLMs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12375–12396, Singapore. Association for Computational Linguistics.

Jacob Austin, Augustus Odena, Maxwell I. Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie J. Cai, Michael Terry, Quoc V. Le, and Charles Sutton. 2021. Program synthesis with large language models. *CoRR*, abs/2108.07732.

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D. Lee, Deming Chen, and Tri Dao. 2024. Medusa: Simple LLM inference acceleration framework with multiple decoding heads. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. 2023a. Accelerating large language model decoding with speculative sampling. *CoRR*, abs/2302.01318.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya

Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code. *CoRR*, abs/2107.03374.

Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. 2023b. Universal self-consistency for large language model generation. *CoRR*, abs/2311.17311.

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

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

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Yichao Fu, Peter Bailis, Ion Stoica, and Hao Zhang. 2024. Break the sequential dependency of LLM inference using lookahead decoding. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Zhenyu He, Zexuan Zhong, Tianle Cai, Jason Lee, and Di He. 2024. REST: Retrieval-based speculative decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1582–1595, Mexico City, Mexico. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Denis Kocetkov, Raymond Li, Loubna Ben Allal, Jia Li, Chenghao Mou, Yacine Jernite, Margaret Mitchell, Carlos Muñoz Ferrandis, Sean Hughes, Thomas Wolf, Dzmitry Bahdanau, Leandro von Werra, and Harm de Vries. 2023. The stack: 3 TB of permissively licensed source code. *Trans. Mach. Learn. Res.*, 2023.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 19274–19286. PMLR.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. 2024a. Numinamath. [https://huggingface.co/AI-MO/NuminaMath-CoT](https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf).

Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. 2023. Making language models better reasoners with step-aware verifier. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5315–5333, Toronto, Canada. Association for Computational Linguistics.

Yiwei Li, Peiwen Yuan, Shaoxiong Feng, Boyuan Pan, Bin Sun, Xinglin Wang, Heda Wang, and Kan Li. 2024b. Turning dust into gold: Distilling complex reasoning capabilities from llms by leveraging negative data. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 18591–18599. AAAI Press.

Yiwei Li, Peiwen Yuan, Shaoxiong Feng, Boyuan Pan, Xinglin Wang, Bin Sun, Heda Wang, and Kan Li. 2024c. Escape sky-high cost: Early-stopping self-consistency for multi-step reasoning. In *The Twelfth*

International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024. OpenReview.net.

Yiwei Li, Ji Zhang, Shaoxiong Feng, Peiwen Yuan, Xinglin Wang, Jiayi Shi, Yueqi Zhang, Chuyi Tan, Boyuan Pan, Yao Hu, and Kan Li. 2025. Revisiting self-consistency from dynamic distributional alignment perspective on answer aggregation. *Preprint*, arXiv:2502.19830.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. 2024d. EAGLE-2: Faster inference of language models with dynamic draft trees. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7421–7432, Miami, Florida, USA. Association for Computational Linguistics.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. 2024e. EAGLE: speculative sampling requires rethinking feature uncertainty. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Pan Lu, Liang Qiu, Wenhao Yu, Sean Welleck, and Kai-Wei Chang. 2023. A survey of deep learning for mathematical reasoning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14605–14631, Toronto, Canada. Association for Computational Linguistics.

Xupeng Miao, Gabriele Oliaro, Zhihao Zhang, Xinhao Cheng, Zeyu Wang, Rae Ying Yee Wong, Zhuoming Chen, Daiyaan Arfeen, Reyna Abhyankar, and Zhihao Jia. 2023. Specinfer: Accelerating generative LLM serving with speculative inference and token tree verification. *CoRR*, abs/2305.09781.

Haifeng Qian, Sujan Kumar Gonugondla, Sungsoo Ha, Mingyue Shang, Sanjay Krishna Gouda, Ramesh Nallapati, Sudipta Sengupta, Xiaofei Ma, and Anoop Deoras. 2024. BASS: batched attention-optimized speculative sampling. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 8214–8224. Association for Computational Linguistics.

Ashutosh Sathe, Divyanshu Aggarwal, and Sunayana Sitaram. 2024. Improving self consistency in llms through probabilistic tokenization. *CoRR*, abs/2407.03678.

Qidong Su, Christina Giannoula, and Gennady Pekhimenko. 2023. The synergy of speculative decoding and batching in serving large language models. *CoRR*, abs/2310.18813.

Guangya Wan, Yuqi Wu, Jie Chen, and Sheng Li. 2024. Dynamic self-consistency: Leveraging reasoning paths for efficient LLM sampling. *CoRR*, abs/2408.17017.

Xinglin Wang, Shaoxiong Feng, Yiwei Li, Peiwen Yuan, Yueqi Zhang, Boyuan Pan, Heda Wang, Yao Hu, and Kan Li. 2024a. Make every penny count: Difficulty-adaptive self-consistency for cost-efficient reasoning. *CoRR*, abs/2408.13457.

Xinglin Wang, Yiwei Li, Shaoxiong Feng, Peiwen Yuan, Boyuan Pan, Heda Wang, Yao Hu, and Kan Li. 2024b. Integrate the essence and eliminate the dross: Fine-grained self-consistency for free-form language generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11782–11794, Bangkok, Thailand. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

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

Yuxi Xie, Kenji Kawaguchi, Yiran Zhao, James Xu Zhao, Min-Yen Kan, Junxian He, and Michael Qizhe Xie. 2023. Self-evaluation guided beam search for reasoning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024. Qwen2.5 technical report. *CoRR*, abs/2412.15115.

Jun Zhang, Jue Wang, Huan Li, Lidan Shou, Ke Chen, Gang Chen, and Sharad Mehrotra. 2024a. Draft& verify: Lossless large language model acceleration via self-speculative decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11263–11282, Bangkok, Thailand. Association for Computational Linguistics.

Lefan Zhang, Xiaodan Wang, Yanhua Huang, and Ruiwen Xu. 2024b. Harmonized speculative sampling. *CoRR*, abs/2408.15766.

Yongchao Zhou, Kaifeng Lyu, Ankit Singh Rawat, Aditya Krishna Menon, Afshin Rostamizadeh, San-

jiv Kumar, Jean-François Kagy, and Rishabh Agarwal. 2024. Distillspec: Improving speculative decoding via knowledge distillation. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

A Algorithm

Algorithm 1 Self-Consistency Draft Token Generation

Input: Reasoning paths $\{y_1, y_2, \ldots, y_N\}$ at t step, suffix length k, maximum draft token length L, tolerance ϵ
Output: Optimal draft token sequence $\mathcal{D}$
Initialization:
  Construct the initial draft pool $\mathcal{C}_0 \leftarrow \emptyset$
for each path y_i in $\{y_1, y_2, \ldots, y_N\}$ **do**
  Find the k-token suffix $y_i^{t-k+1:t}$
  for each path y_j where $j \neq i$ **do**
    Search for fuzzy matching prefixes in y_j and add subsequent tokens to candidate pool $\mathcal{C}_t$
  end for
end for
Construct Directed Acyclic Graph (DAG) $\mathcal{G} = (V, E)$:
for each token pair u, v in the candidate pool **do**
  Compute edge weight $w(u, v) \leftarrow \alpha \cdot p_\theta(v|u) + (1 - \alpha) \cdot \text{Freq}(u \rightarrow v)$
end for
Greedily select tokens for optimal sequence:
  Initialize current sequence $\mathcal{D} \leftarrow \emptyset$
while current draft token length $len(\mathcal{D}) < L$ and DAG is not empty **do**
  Select token v_i with the highest weight $\sum_{u:(u,v)\in E} w(u, v)$ at each layer
  Add token v_i to the draft token sequence $\mathcal{D}$
end while
return $\mathcal{D}$

B Details about Experimental Settings

The raw evaluation setting for EAGLE-2 and REST both use a batch size of 1, and the inherently sequential execution of LLM leads to low hardware utilization on modern GPUs under this setting. Thus, the draft token length can be set quite large (e.g., 60). Su et al. (2023) find that as the batch size increases, the optimal draft token length decreases. We reproduce BASS (Qian et al., 2024) multi-sample speculative decoding method on EAGLE-2 and find that when the draft token length is fixed to 4, the speedup relative to vanilla decoding gradually diminishes, and when the batch size is greater than or equals to 8, it even becomes slower than vanilla decoding. Detailed results of this experiment is shown in Figure 5. In this work, we focus on constructing draft tokens with higher acceptance rates and lower latency of draft token construction, and accelerating parallel draft token verification and draft token generation are orthogonal tasks. Therefore, how to accelerate parallel draft token verification is not within the scope of our study, and different draft token construction methods can be validated using the same approach. As a result, we adopt accept length and latency of draft token construction as the evaluation metric for multi-sampling speculative decoding methods, rather than speedup. Additionally, we keep the computational load of a single multi-sample speculative decoding step constant, i.e., draft token length * batch size = 24. Furthermore, since draft token length = draft sequence length * draft sequence number for tree decoding in EAGLE-2 and REST, when the draft token length is fixed, there are different schemes to generate draft candidates (e.g., when the draft token length is 4, we can either generate 2 draft sequences of length 2 or 1 draft sequence of length 4). In this work, both EAGLE-2 and REST report the results of the best draft candidates generation scheme under the specified draft token length.

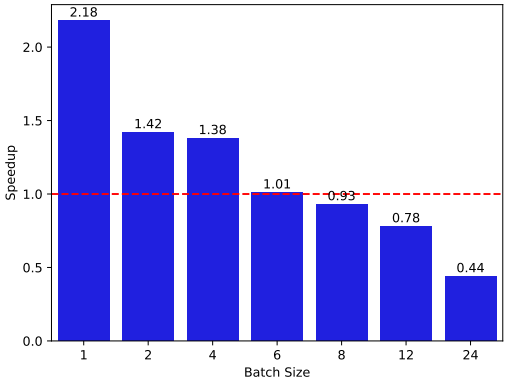

Figure 5: Speedup ratio with different batch size.

C Effect of Hyperparameters

We conducted additional experiments to examine the sensitivity of our method to two important hyperparameters: α (the balance between matching frequency and transition probability) and tolerance ϵ (the maximum edit distance in suffix matching). Figure 6 shows the accept length under varying values of each parameter:

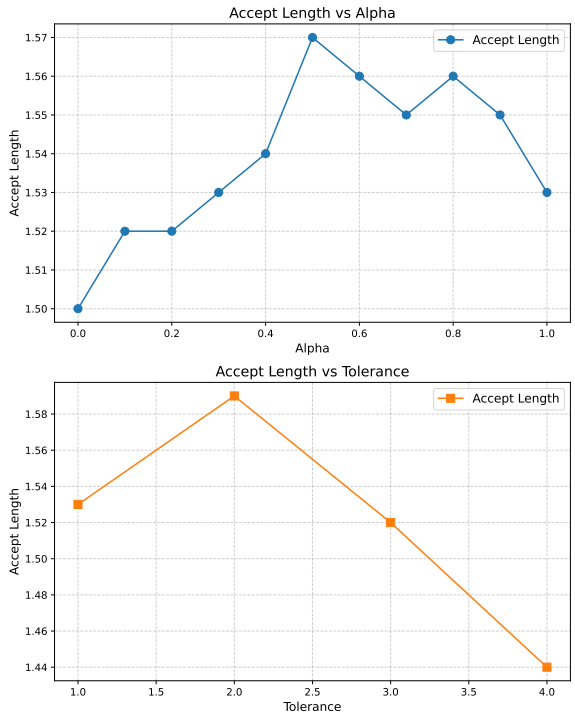

Figure 6: Effect of Hyperparameters on α and tolerance ϵ.

- For α, performance improves as it decreases from 1.0 to 0.5, then slightly degrades towards 0.0, suggesting that a balanced weighting yields the best results.
- For tolerance ϵ, mild fuzziness (e.g., $\epsilon = 2$) improves match coverage, while overly permissive matching (e.g., $\epsilon = 4$) introduces noise and degrades performance.

D More results

Table 2 shows more results to prove the effectiveness of proposed method.

E Related Work

Multi-Sample Inference Multi-sample inference strategies have become pivotal for enhancing the performance and reliability of LLMs in complex reasoning tasks. Techniques like self-consistency (Wang et al., 2023) and Best-of-N sampling (Cobbe et al., 2021a) leverage parallel generation of multiple candidate outputs to mitigate individual errors and improve accuracy in math reasoning and many other tasks (Chen et al., 2023b; Wang et al., 2024b). One line of work aims to improve the performance of multiple sampling methods, for example, through weighted voting (Li et al., 2023, 2024b), enhancing the diversity of reasoning chains (Sathe et al., 2024; Li et al., 2025), or step-wise guidance (Xie et al., 2023). However, the computational cost of generating multiple samples remains a critical bottleneck. Recent work has explored optimizing multi-sample efficiency on controlling the sampling size (Li et al., 2024c; Aggarwal et al., 2023; Wang et al., 2024a; Wan et al., 2024), yet the fundamental trade-off between accuracy gains and costs persists. Our method aims to accelerate multiple sampling, significantly reducing inference latency while ensuring no performance degradation.

Speculative Decoding Speculative decoding (Leviathan et al., 2023; Fu et al., 2024) has emerged as a key paradigm for accelerating LLM inference by reducing auto-regressive steps without altering output quality. However, these methods still struggle to fully align drafts with the original model's distribution (Zhang et al., 2024a; Cai et al., 2024) and require significant cost for training (Li et al., 2024e,d; Zhou et al., 2024; Zhang et al., 2024b) or storage (He et al., 2024). Our work diverges by exploiting the inherent redundancy in multi-sample outputs, obviating external resources while achieving superior alignment through consensus-driven drafts.

Table 2: Accepted length under different draft lengths (L) for REST, EAGLE-2, and our method on MATH and HumanEval.

Model	Dataset	Qwen2.5-7B-Instruct				Qwen2.5-14B-Instruct				LLaMA3-8B-Instruct			
		L=6	4	3	2	L=6	4	3	2	L=6	4	3	2
REST	MATH	0.77	0.59	0.44	0.23	0.57	0.21	0.18	0.15	0.57	0.50	0.40	0.24
EAGLE-2	MATH	0.98	0.87	0.69	0.68	0.89	0.79	0.63	0.62	1.48	1.18	1.09	0.94
Ours	MATH	**1.89**	**1.63**	**1.47**	**1.17**	**1.78**	**1.63**	**1.42**	**1.17**	**2.95**	**2.15**	**1.76**	**1.26**
REST	HumanEval	0.18	0.18	0.18	0.17	0.26	0.21	0.18	0.17	0.36	0.27	0.26	0.21
EAGLE-2	HumanEval	0.78	0.76	0.78	0.71	0.69	0.70	0.69	0.66	1.52	1.13	**1.19**	**1.00**
Ours	HumanEval	**0.92**	**0.91**	**0.88**	**0.79**	**0.90**	**0.92**	**0.91**	**0.81**	**1.59**	**1.30**	**1.19**	0.93