Asking a Language Model for Diverse Responses

Sergey Troshin*
University of Amsterdam
s.troshin@uva.nl

Irina Saparina*
University of Edinburgh
i.saparina@sms.ed.ac.uk

Antske Fokkens
Vrije Universiteit Amsterdam
antske.fokkens@vu.nl

Vlad Niculae
University of Amsterdam
v.niculae@uva.nl

Abstract

Large language models increasingly rely on explicit reasoning chains and can produce multiple plausible responses for a given context. We study the candidate sampler that produces the set of plausible responses contrasting the ancestral (parallel) sampling against two alternatives: enumeration, which asks the model to produce n candidates in one pass, and iterative sampling, which proposes candidates sequentially while conditioning on the currently generated response set. Under matched budgets, we compare these samplers on quality, lexical and computation flow diversity, and efficiency. Our empirical results demonstrate that enumeration and iterative strategies result in higher diversity at comparable quality. Our findings highlight the potential of simple non-independent sampling strategies to improve response diversity without sacrificing generation quality.

1 Introduction

Large language models (LLMs) have shown strong performance across a wide range of applications (OpenAI et al., 2024; DeepSeek-AI et al., 2025). In particular, the ability to generate explicit reasoning chains that guide planning and decision-making has become a cornerstone of recent progress (Wei et al., 2022; Yao et al., 2023; Zhu et al., 2025; Zhang et al., 2024). Many of these applications benefit from access to multiple plausible responses for a given context, including test-time control (Mudgal et al., 2024; Deng and Raffel, 2023; Troshin et al., 2025), majority voting or best-of-n (Stiennon et al., 2020; Nakano et al., 2022), conformal generative modeling (Kladny et al., 2025), reasoning with diverse decoding paths (Wang et al., 2024) and ambiguity resolution (Kobalczyk et al., 2025; Chen et al., 2025; Saparina and Lapata, 2025).

A necessary component of these pipelines is a *candidate sampler* that returns a set of n responses in context. The candidates are commonly obtained by ancestral sampling from the model distribution, or from variations such as temperature, top-p, top-k (Holtzman et al., 2020; Basu et al., 2021; Hewitt et al., 2022; Minh et al., 2025; Vilnis et al., 2023). Beyond being in some sense the natural approach, ancestral sampling also benefits from being simple to implement and readily parallelizable across devices, as each response is sampled independently of the others. Nevertheless, ancestral sampling suffers from repetitions of high-probability sequences, which motivated researchers to propose non-independent algorithms, including arithmetic sampling (Vilnis et al., 2023), diverse, stochastic, and determinantal beam search modifications (Vijayakumar et al., 2018; Kool et al., 2019; Meister et al., 2021). These approaches, well-studied in the literature, are based on search-style algorithms on top of a language model's output probability, which still scores each sample separately, possibly with the help of a separate dissimilarity function. In this work, we take a substantially different approach and ask whether we can use the standard LLM generation pipelines to enable efficient non-independent sampling, by processing multiple candidates at the same time.

In particular, we are interested in a candidate sampler that:

(i) produces high-quality samples;

(ii) promotes response diversity;

(iii) scales efficiently as the number of responses increases;

(iv) is simple to use and relies on standard LLM decoding primitives.

We compare the commonly used **parallel** sampling strategy (ancestral sampling) with two alternative sampling strategies, which we define as **enumera-**

*These authors contributed equally to this work

Proceedings of the 2nd Workshop on Uncertainty-Aware NLP (UncertaiNLP 2025), pages 66–72

tion and **iterative** approaches, and study them from the perspective of quality, diversity, and efficiency.

Our main finding is that the enumeration and iterative strategies are simple and promising alternatives to the standard parallel approach. We find that our non-independent iterative and enumeration strategies result in higher lexical and computational flow diversity. Such approaches can be seen in a way as upper-bound oracles to diverse generation, in the sense that they fully model the joint distribution over samples and are only limited by the instruction-following performance of the LLM. Our implementation is released as open-source.[1]

2 Methodology

We consider tasks for which there are multiple valid responses. In the context of this work, we consider a valid response to contain both a derivation and a final answer, so different derivations leading to the same answer are valid responses. Given a model p_θ and a prompt c, our goal is to produce a set $\mathcal{S} = \{y^{(1)}, \ldots, y^{(n)}\}$ of n responses. We keep all decoding hyperparameters fixed across methods and vary only the sampling protocol.

2.1 Sampling Strategies

Parallel sampling. We sample n times independently with different random seeds; samples do not condition on one another:

$$y^{(i)} \sim p_\theta(\cdot \mid c;) \quad \text{for } i = 1..n \tag{1}$$

Enumeration sampling. We prompt the model to generate multiple different outputs in one pass; later outputs condition on earlier ones:

$$y^{(k)} \sim \prod_{i=1}^{k} p_\theta\left(y^{(i)} \mid c,\, y^{(1:i-1)}\right). \tag{2}$$

The number of desired samples is not specified in the prompt, but rather implicitly predicted. To the best of our knowledge, the enumeration approach has not been studied in the literature. However, due to its simplicity, we speculate it it is used in practice, for example, Ilia and Aziz (2024) prompt ChatGPT (OpenAI, 2022) to enumerate 40 responses in context as a complementary strategy to ancestral sampling; Saparina and Lapata (2024) prompt models to enumerate all possible interpretations of ambiguous questions.

[1] https://github.com/serjtroshin/ask4diversity

Iterative sampling. We generate one candidate at a time, and we re-prompt the model to extend an already generated list of responses with a new response. Namely, for $k = 1$, we generate as:

$$y^{(1)} \sim p_\theta(\cdot \mid c), \tag{3}$$

and for $k > 1$, we pass the generated solutions:

$$y^{(k)} \sim p_\theta\left(\cdot \mid c(y^{(1)}, \cdots, y^{(k-1)})\right). \tag{4}$$

In practice, the conditioning is achieved with a templated prompt; refer to Appendix A for the specific prompts used for all strategies.

3 Experimental Setup

We evaluate on GSM8K (Cobbe et al., 2021), a grade school math problem-solving benchmark. Each problem has a single gold answer, but multiple valid solutions may lead to it. Therefore, a candidate is $y^{(i)} = (r^{(i)}, a^{(i)})$, with $r^{(i)}$ the solution (reasoning) and $a^{(i)}$ the final extracted answer.

3.1 Models

In our work, we rely on the Qwen3 family of models (Yang et al., 2025), chosen for their high reasoning performance, diverse range of model sizes. In our preliminary investigation, we observe that Qwen3 models are able to follow our zero-shot instructions, and they show high accuracy in following the required output format. For our experiments, we use `Qwen3-{4B,8B,14B}` models with thinking generation mode on; and we use `Qwen3-4B-{Instruct/Thinking}-2507` released solely for non-thinking/thinking use-cases.

We use the hyperparameters suggested by the model developers: `temperature` $= 0.6$, `top-`$k =$ 20, `top-`$p = 0.95$, `repetition_penalty` $= 1.0$.

3.2 Metrics

Quality. We define the quality metrics as the average accuracy over response sets given a golden answer for a problem. We calculate the accuracy of a response set by taking the minimum, mean, and maximum statistics over the answers within the set and averaging these statistics over the dataset.

Lexical diversity. We follow Li et al. (2016) and report **averaged distinct N-gram diversity** metric as the proportion of distinct N-grams in the set of responses relative to the total number of N-grams.

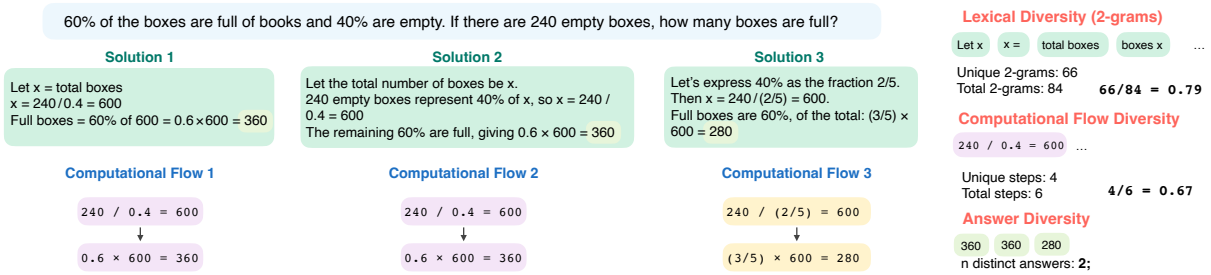

Figure 1: Example of a math problem with three responses, their computation flows, and the resulting metrics: lexical, computational flow and answer diversity.

Computation flow diversity. To complement the lexical diversity metric, we extract a computation flow of each solution by mapping it to sequences of normalized arithmetic steps (*e.g.*, "Janet sells 9 eggs at $2 each, which gives 18" maps to $9 \times 2 = 18$). We obtain flows with a one-shot prompt to Qwen-3-32B (see Appendix C). We report the proportion of unique steps relative to the total number of steps in the set. To compute this metric, we estimate the distinct 1-grams over the simple arithmetical steps, namely $9 \times 2 = 18$ is considered to be a single 1-gram. This approach can collapse steps that are arithmetically identical but occur in different parts of a solution; however, we found this to be rare in our experiments. If needed, repeated occurrences can be distinguished by indexing them within a flow (e.g., (1) $9 \times 2 = 18$, (2) $9 \times 2 = 18$).

Final answer variability. For some applications, it might be useful to have samples with different answers (*e.g.* to have both positive and negative demonstrations), and we measure the answer variability as the number of unique answers among the response set. For GSM8K, high answer variability means that some answers are parsed as incorrect.

Figure 1 illustrates an input math problem, three different responses, the corresponding computation flows, and the resulting metrics. The first and second responses differ in phrasing, but follow the same computation; the third differs in wording and computation but yields an incorrect result.

4 Results

4.1 Quality and Diversity

In Table 1, we report the evaluation results on the GSM8K dataset.

Parsing the solutions. We parse the responses from the generated outputs by searching for the required solution tags, *i.e.*, `<Solution>...</Solution>`. For the *parallel* and *iteration* strategies, we obtain more than 4 successfully parsed responses on average (out of 5 required). For the *enumeration* strategy, we do not specify the required number of responses and obtain between 2 and 4 parsed responses on average. Overall, Qwen3 models demonstrate a satisfactory ability to follow our instructions for output formatting.

Diversity of the responses. We observe that in all cases the diversity of samples from the *parallel* strategy is lower compared to the diversity of the two non-independent strategies, both for the lexical and computational flow diversity. We observe that often higher lexical diversity does not imply higher compute diversity, and we think these metrics can provide complementary signals to the developers.

Quality of the answers. In most cases, our models demonstrate good zero-shot task performance with an accuracy of around 90%. Parallel sampling shows the most stable high quality (lowest quality variation), probably because it is the most standard approach, and it is easier for a language model to adapt to the corresponding prompt requirements.

Variability of the answers. Additionally, we report the answer variability and the average minimum and maximum accuracy over the responses. We observe that overall models exhibit low answer variability with less than 1.3 distinct answers on average. Enumeration strategy results in the highest quality difference (i.e., the gap between maximum and minimum accuracy), while the parallel and iteration are on par with each other. We note that under diversity requirements, we do not expect a model to always produce a parsable or even correct answer, and part of the quality loss can be attributed to answer parser failures.

Model	Strategy	# Parsed Solutions	Min Quality	Mean Quality	Max Quality	Lexical Diversity	Compute Diversity	# Distinct Answers
Qwen3-4B	parallel	4.77	0.86	0.91	0.95	42.8	33.1	1.13
	enumeration	3.90	0.88	0.90	0.91	68.1	56.1	1.04
	iteration	4.75	0.83	0.87	0.90	61.8	60.0	1.07
Qwen3-8B	parallel	4.23	0.89	0.91	0.93	44.5	34.7	1.06
	enumeration	2.81	0.89	0.90	0.91	73.1	64.1	1.03
	iteration	4.87	0.89	0.91	0.92	63.4	79.8	1.03
Qwen3-14B	parallel	4.90	0.92	0.94	0.96	38.4	31.5	1.05
	enumeration	3.58	0.90	0.92	0.94	70.2	57.1	1.05
	iteration	4.96	0.60	0.73	0.83	70.1	59.3	1.25
Qwen3-4B-Instruct	parallel	4.98	0.88	0.92	0.94	33.1	47.7	1.10
	enumeration	3.09	0.88	0.90	0.91	72.8	61.2	1.04
	iteration	5.00	0.86	0.89	0.90	60.3	55.6	1.08
Qwen3-4B-Thinking	parallel	4.67	0.81	0.89	0.94	47.9	30.7	1.24
	enumeration	2.27	0.64	0.73	0.79	66.2	64.5	1.19
	iteration	4.17	0.78	0.87	0.92	68.0	62.0	1.16

Table 1: Main results for *parallel*, *enumeration*, and *iteration* sampling strategies. For enumeration, we let the model decide the number of solutions, for parallel and iteration, we expect 5 solutions, and report the average number of parsed solutions. Min and max quality denote the average minimum and maximum accuracy over the response sets. # distinct answers denote the average number of distinct answers among the set of parsed responses.

4.2 Compute Efficiency

An important question when developing the sampling strategies is to understand how efficient it is to generate the set of n responses. We distinguish the total number of generation calls that we need to do in order to generate n responses, and the support for parallelization. We compare the three strategies *w.r.t.* the compute they require.

From the perspective of parallel-time computation, the *parallel* approach is most time-efficient by design, and this sort of parallelization is well optimized and supported in LLM codebases, but its time efficiency is tied to the access to parallel computation (*e.g.*, a multi-GPU setup). As we observe from the diversity results, the independence assumption results in lower diversity (a higher degree of repetitions).

Both enumeration and iteration are most suited for single-GPU generation. For *enumeration*, we need a single call to the model to enumerate the generations in the response; in thinking mode, the model shares the computation to produce n responses: it generates a single thinking chain first, and then it enumerates the responses. A limitation of this strategy is that this approach requires a larger context length to produce multiple responses in one go, which in turn slows down the decoding for the standard quadratic-time attention implementation.

For *iteration*, we need n full sequential calls: the generated responses are reused, but not any other internals. Iteration is less time-efficient than enumeration, since the former requires multiple sequential generation calls; on the other hand, iteration sampling allows for easy and more explicit control of the number of responses, and may be more compatible with other probabilistic modeling strategies for subset selection without sacrificing the expressiveness of enumeration sampling.

The main difference between parallel and the two serial approaches (enumeration and iteration) is the degree to which information is shared and efficiently reused across the set when generating responses. We see promise in further study of information conditioning and compression, specifically, quantifying the extent of this sharing and reuse. In particular, the enumeration strategy can potentially approach the efficiency of a single parallel call while processing the responses quasi-independently, which in turn affects the diversity of the responses.

5 Conclusion

We study the problem of generating a diverse set of responses. We propose two non-independent approaches for sampling responses from a language model, namely enumeration and iteration strategies, and compare them against parallel algorithms based on ancestral sampling. On GSM8k, we find that our non-independent approaches can provide higher diversity of the samples, while maintaining simplicity and overall quality of the generations. Compute efficiency analysis shows that enumera-

tion and iteration are well-suited to a single GPU and can reduce redundancy without specialized search machinery. We hope our work will motivate further investigation of simple non-independent strategies for diverse candidate sampling.

6 Limitations

One of the main limitations of our work is a narrow evaluation scope. We focus on a single dataset with verifiable rewards and a room for diversity of answers and reasoning chains. Future work can evaluate these methods on tasks that inherently benefit from diverse generations, such as creative writing, code generation, or ambiguous question answering. We do not compare the results to established diverse decoding methods such as beam search variants, as we limit our scope to sampling from the model output distribution rather than modifying it through specialized decoding algorithms. Ippolito et al. (2019) provide an extensive survey and evaluation methodology for the established methods.

Acknowledgments

This work is part of the UTTER project, supported by the European Union's Horizon Europe research and innovation programme via grant agreement 101070631. This work is also supported by project VI.Veni.212.228 of the research program 'Veni', which is financed by the Dutch Research Council (NWO); and is part of 'Hybrid Intelligence: augmenting human intellect' (https://hybrid-intelligence-centre.nl) with project number 024.004.022 of the research program 'Gravitation' which is (partly) financed by the Dutch Research Council (NWO).

References

Sourya Basu, Govardana Sachitanandam Ramachandran, Nitish Shirish Keskar, and Lav R. Varshney. 2021. Mirostat: A neural text decoding algorithm that directly controls perplexity. In *ICLR*.

Maximillian Chen, Ruoxi Sun, Tomas Pfister, and Sercan O Arik. 2025. Learning to clarify: Multi-turn conversations with action-based contrastive self-training. In *The Thirteenth International Conference on Learning Representations*.

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

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 15 others. 2025. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Haikang Deng and Colin Raffel. 2023. Reward-augmented decoding: Efficient controlled text generation with a unidirectional reward model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11781–11791, Singapore. Association for Computational Linguistics.

John Hewitt, Christopher Manning, and Percy Liang. 2022. Truncation sampling as language model desmoothing. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3414–3427, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *ICLR*.

Evgenia Ilia and Wilker Aziz. 2024. Predict the next word: <humans exhibit uncertainty in this task and language models _____>. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 234–255, St. Julian's, Malta. Association for Computational Linguistics.

Daphne Ippolito, Reno Kriz, João Sedoc, Maria Kustikova, and Chris Callison-Burch. 2019. Comparison of diverse decoding methods from conditional language models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3752–3762, Florence, Italy. Association for Computational Linguistics.

Klaus-Rudolf Kladny, Bernhard Schölkopf, and Michael Muehlebach. 2025. Conformal generative modeling with improved sample efficiency through sequential greedy filtering. In *The Thirteenth International Conference on Learning Representations*.

Kasia Kobalczyk, Nicolás Astorga, Tennison Liu, and Mihaela van der Schaar. 2025. Active task disambiguation with LLMs. In *The Thirteenth International Conference on Learning Representations*.

Wouter Kool, Herke van Hoof, and Max Welling. 2019. Stochastic beams and where to find them: The gumbel-top-k trick for sampling sequences without replacement. In *ICML*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

Clara Meister, Martina Forster, and Ryan Cotterell. 2021. Determinantal beam search. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6551–6562, Online. Association for Computational Linguistics.

Nguyen Nhat Minh, Andrew Baker, Clement Neo, Allen G Roush, Andreas Kirsch, and Ravid Shwartz-Ziv. 2025. Turning up the heat: Min-p sampling for creative and coherent LLM outputs. In *ICLR*.

Sidharth Mudgal, Jong Lee, Harish Ganapathy, YaGuang Li, Tao Wang, Yanping Huang, Zhifeng Chen, Heng-Tze Cheng, Michael Collins, Trevor Strohman, Jilin Chen, Alex Beutel, and Ahmad Beirami. 2024. Controlled decoding from language models. In *ICML*.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2022. Webgpt: Browser-assisted question-answering with human feedback. *Preprint*, arXiv:2112.09332.

OpenAI. 2022. Introducing ChatGPT. `https://openai.com/blog/chatgpt`. Accessed August 15, 2025.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 24 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Irina Saparina and Mirella Lapata. 2024. Ambrosia: A benchmark for parsing ambiguous questions into database queries. In *Advances in Neural Information Processing Systems*, volume 37, pages 90600–90628. Curran Associates, Inc.

Irina Saparina and Mirella Lapata. 2025. Disambiguate first, parse later: Generating interpretations for ambiguity resolution in semantic parsing. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 16825–16839, Vienna, Austria. Association for Computational Linguistics.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2020. Learning to summarize with human feedback. In *NeurIPS*.

Sergey Troshin, Vlad Niculae, and Antske Fokkens. 2025. On the low-rank parametrization of reward models for controlled language generation. In *TMLR*.

Ashwin K. Vijayakumar, Michael Cogswell, Ramprasaath R. Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2018. Diverse beam search for improved description of complex scenes. In *AAAI*.

Luke Vilnis, Yury Zemlyanskiy, Patrick Murray, Alexandre Passos, and Sumit Sanghai. 2023. Arithmetic sampling: parallel diverse decoding for large language models. In *ICML*.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. 2024. MMLU-pro: A more robust and challenging multi-task language understanding benchmark. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

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*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *NeurIPS*.

Yu Zhang, Xiusi Chen, Bowen Jin, Sheng Wang, Shuiwang Ji, Wei Wang, and Jiawei Han. 2024. A comprehensive survey of scientific large language models and their applications in scientific discovery. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8783–8817, Miami, Florida, USA. Association for Computational Linguistics.

Qinglin Zhu, Runcong Zhao, Hanqi Yan, Yulan He, Yudong Chen, and Lin Gui. 2025. Soft reasoning: Navigating solution spaces in large language models through controlled embedding exploration. In *ICML*.

A Prompts

Prompt for enumeration sampling.

```
Given the following problem, reason through
it and provide multiple different solutions:

Problem: {question}

Use exactly this format (no extra text):
<Solution 1> [Your reasoning should go here]
The answer is [Answer 1]. </Solution 1>
```

```
...
<Solution N> [Your reasoning should go here]
The answer is [Answer N]. </Solution N>
```

Prompt for parallel sampling.

```
Given the following problem, reason through
it and provide a solution:

Problem: {question}

You must wrap your reasoning and answer into
<Solution> ...reasoning here... 'The answer
is [numerical value].'</Solution> format.
```

Prompt for iterative sampling.

```
Given a problem and a set of solutions,
reason through it and provide a new solution.
The new solution may result in the same
answer, but it must be different from the
ones already provided.

Problem: {question}

Existing solutions:
{solutions}

Use exactly this format (no extra text):
<New Solution> [Your reasoning should go
here]. The answer is [answer]. </New
Solution>
```

B Averaged Distinct N-gram Diversity

Given a set of responses $S = \{y^{(i)}\}_{i=1}^{n}$, for $N \in \{1, \ldots, 5\}$, we calculate the averaged distinct N-gram diversity for each set as:

$$\text{avg. dist. N-gram}\,(S) = \sum_{N=1}^{5} \frac{|\text{set}(\text{N-gram}(R_C))|}{|\text{N-gram}(R_C)|}.$$

The diversity metric is calculated as the mean avg. distinct N-gram diversity over the sets of responses.

C Prompt for Computation Flow Parsing

```
You will receive a math question and a
free-form solution. Extract the sequence of
arithmetic steps from the solution and
output them one by one.

Rules:
- Output ONLY lines made of digits 0-9,
parentheses (), the operators + - * / ^, and
optionally "=" to show each step's result.
- No words, units, currency symbols, or
extra text.
- One step per line, in the order implied by
the solution.
- Convert verbal quantities to numbers.
Replace references like "the remainder" with
the actual numeric value.
- Keep only the steps that lead to the final
answer.
- If no computable arithmetic appears,
output an empty line.

Example:

Question: Janet lays 16 eggs a day. She eats
3, uses 4 for baking, and sells the rest for
$2 each. How much money does she make?
Solution: Janet lays 16 eggs per day. She
eats 3 and uses 4 for baking, so 16 - 7 = 9
eggs left. She sells them at $2 each → 9 * 2
= $18.
Output:
3 + 4 = 7
16 - 7 = 9
9 * 2 = 18

Now, extract the arithmetic steps from the
following:

Question: {question}
Solution: {solution}
Output:
```