Choice is what matters after Attention

Chenhan Fu[1] **Guoming Wang**[1] **Juncheng Li**[1] **Rongxing Lu**[2] **Siliang Tang**[1]
[1]Zhejiang University [2]University of New Brunswick

Abstract

The decoding strategies widely used in large language models (LLMs) today are Top-p Sampling and Top-k Sampling, both of which are methods situated between greedy decoding and random sampling. Inspired by the concept of loss aversion from prospect theory in behavioral economics, and the endowment effect as highlighted by Richard H. Thaler, the 2017 Nobel Memorial Prize in Economic Sciences — particularly the principle that "the negative utility of an equivalent loss is approximately twice the positive utility of a comparable gain" — we have developed a new decoding strategy called Loss Sampling. We have demonstrated the effectiveness and validity of our method on several LLMs, including Llama-2, Llama-3 and Mistral. Our approach improves text quality by 4-30% across four pure text tasks while maintaining diversity in text generation. Furthermore, we also extend our method to multimodal large models (LMs) and Beam Search, demonstrating the effectiveness and versatility of Loss Sampling with improvements ranging from 1-10%.

1 Introduction

Decoding strategies have been widely employed by current large language models (LLMs) to select output words (tokens) at each time step during text sequence generation. These strategies play a crucial role in determining the quality, diversity, and efficiency of the generated text. The most basic decoding strategies are Greedy Decoding (Brown et al., 1990; Sutskever et al., 2014; Vaswani et al., 2017) and Random Sampling (Mikolov et al., 2010; Radford et al., 2018; Holtzman et al., 2019) : Greedy Decoding, which selects the highest probability word at each step, is fast and efficient but often results in repetitive and dull text; conversely, Random Sampling selects words based on the probability distribution, enhancing diversity but potentially reducing quality with semantically incoherent or irrelevant words. To improve the quality and diversity of the generated text while reducing repetition and unnatural occurrences, the academic community has proposed Top-k Sampling (Fan et al., 2018; Holtzman et al., 2018; Radford et al., 2019, 2018) and Top-p Sampling (Nucleus Sampling) (Holtzman et al., 2019; Radford et al., 2019; Dathathri et al., 2019), both of which are intermediate methods between Greedy Decoding and Random Sampling: Top-k Sampling restricts the sampling space to only the highest probability k words at each step, enhancing text coherence and quality while maintaining a degree of diversity (Fan et al., 2018); Top-p Sampling selects a minimum set of words at each step that cumulatively exceed a predefined probability threshold p, and samples randomly within this set, allowing dynamic adjustment of the sampling space size to better balance text diversity and quality (Holtzman et al., 2019).

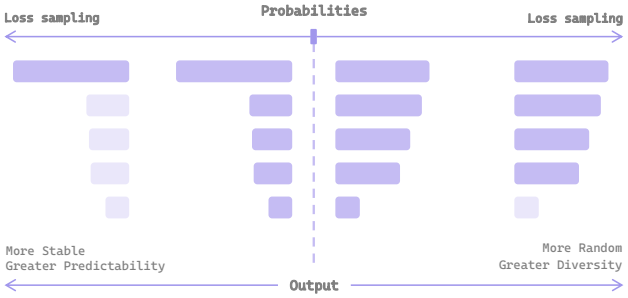

Figure 1: Two examples of highly peaked distribution and flat distribution under Loss Sampling.

However, these methods also have certain drawbacks. For example, in a highly peaked distribution (as shown in the left part of Figure 1), Top-k Sampling will supplement lower probability tokens into the candidate pool when the number of high-probability candidate tokens is far less than the k value; conversely, in a flat distribution (as shown in the right part of Figure 1), Top-p

Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258.

Sampling may overlook some tokens with probabilities close to the highest probability, especially when p is set to a low value. Inspired the concept of Loss Aversion (Kahneman and Tversky, 2013; Kahneman et al., 1991; Tversky and Kahneman, 1991) from Prospect Theory (Kahneman and Tversky, 2013; Tversky and Kahneman, 1992) of behavioral economics (Kahneman, 2003) and the Endowment Effect from the 2017 Nobel Prize in Economics laureate Richard H. Thaler — where the negative effect of a loss is significantly greater than the positive effect of an equivalent gain (not a simple 1:1 ratio, as detailed in Section 2) — we have introduced a new training-free, plug-and-play, lightweight sampling strategy called Loss Sampling, which also lies between Greedy Decoding and Random Sampling.

The key intuition behind Loss Sampling is that the positive expected utility of choosing a token to enter the candidate pool (i.e., being selected by the final multinomial sampling in the candidate pool) must outweigh its negative expected utility (i.e., not being selected in the final sampling). In Loss Sampling, we normalize the entire probability distribution using the highest token probability value, and then sample based on a given threshold, enabling dynamic expansion and contraction of the token candidate pool. As shown in Figure 1 (introduced in detail in Section 3), our Loss Sampling selects candidates based on the relative probabilities of tokens. It neither overselects tokens with excessively low probabilities nor underselects tokens with similar probabilities, thereby addressing the shortcomings of both Top-k and Top-p Sampling.

We compare our method with Top-p and Top-k Sampling, analyzing experimental results across eight datasets within four different task domains. By integrating statistical data from the Type Token Ratio (TTR) (Johnson, 1944; Richards, 1987; McCarthy and Jarvis, 2010) and Self-BLEU (Zhu et al., 2018), we find that Loss Sampling not only maintains the diversity of the generated text (no more than $\pm 4\%$) but also enhances its quality (the performance improvements for different tasks ranged from 4% to 30%). We further extended it to multimodal large models and Beam Search (Koehn et al., 2003; Sutskever et al., 2014), which also demonstrated its effectiveness.

2 Background: Prospect Theory in Behavioral Economics

Prospect Theory was introduced in 1979 by psychologists Daniel Kahneman and Amos Tversky as an economic theory (Kahneman and Tversky, 2013). This theory is primarily used to explain how people make choices in the decision-making process when faced with risks and uncertainties, especially when these choices deviate from the predictions of Expected Utility Theory (Levy, 1992).

In Expected Utility Theory, it is assumed that individuals are rational and will weigh the utility of all possible outcomes and their probabilities of occurring in order to make decisions that maximize expected utility. Rational individuals calculate the product of the utility of each outcome and its probability for each option, then sum these products to choose the option that maximizes expected utility as follows:

$$EU = \sum_i p_i \cdot U(x_i) \tag{1}$$

where p_i represents the probability of outcome i, $U(x_i)$ represents the utility of outcome i, x_i represents the monetary value of outcome i, and EU stands for the expected utility of the option.

However, in reality, people's behavior in economic decisions could be irrational, and their decisions are often driven by some invisible forces rather than by calm, rigorous logic and calculations. Prospect Theory offers a credible explanation, delving into the psychological processes of decision-making under risk and uncertainty and establishing a model framework that can quantitatively describe the irrational factors in human decision-making (Kahneman and Tversky, 2013). The theory includes two main components: the value function and the weighting function.

2.1 Value Function $V(x)$

The value function $V(x)$ describes how people assess the value of gaining or losing something, that is, converting monetary amounts into utility (or psychological value). It typically exhibits the following characteristics: concave for gains, convex for losses, and steeper in the loss domain than in the gain domain (reflecting loss aversion) (Kahneman and Tversky, 2013). The mathematical form is as follows:

$$V(x) = \begin{cases} x^{\alpha} & \text{if } x \geq 0 \ , \\ -\lambda(-x)^{\beta} & \text{if } x < 0 \ . \end{cases} \tag{2}$$

where, the value function $V(x)$ represents the psychological value of a monetary amount x, where α and β are sensitivity parameters for gains and losses respectively, and λ is the loss aversion parameter, typically greater than 1. This indicates that the psychological impact of losses is greater than that of equivalent gains, as shown by the slopes in Figure 2.

2.2 Weighting Function $w(p)$

The weighting function $w(p)$ describes how people assign weights to events of different probabilities. Even

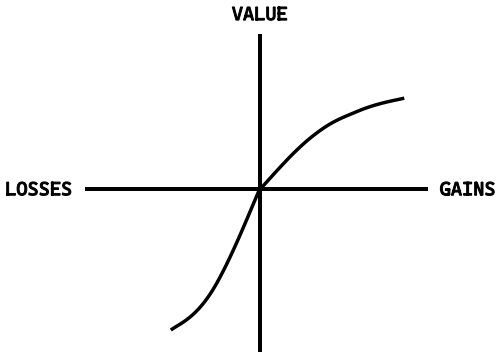

Figure 2: The Value Function from Kahneman and Tversky's experiments.

when probabilities are known, people often do not make decisions based on the true probabilities but rather on some psychological weighting. This means that in assessing probabilities, people do not always adhere to the objective probabilities; they tend to overestimate the significance of low-probability events and underestimate that of moderate to high-probability events (Levy, 1992). This probability weighting can partially explain why individuals tend to be overly optimistic when facing extremely low probability but high reward scenarios (such as buying lottery tickets), and overly pessimistic when facing moderate probability risks.

$$w(p) = \frac{p^\gamma}{(p^\gamma + (1-p)^\gamma)^{1/\gamma}} \tag{3}$$

where $w(p)$ represents the psychological weighting of an event, where p denotes the event's actual probability, and γ describes the non-linear perception of probability by individuals.

Therefore, within Prospect Theory, the overall expected utility is calculated as the sum of the products of the utility of each outcome and its corresponding weight as follows:

$$EU = \sum_i w(p_i) \cdot V(x_i) \tag{4}$$

where, EU represents the expected utility, $w(p_i)$ denotes the weight assigned to outcome i, and $V(x_i)$ represents the value of the outcome i.

From this discussion, it is evident that Prospect Theory emphasizes the concept of "Loss Aversion", which suggests that people's reactions to losses are disproportionately stronger than their pleasure from equivalent gains. In 2017, Nobel laureate Richard Thaler further developed this idea through the concept of the Endowment Effect (Kahneman et al., 1991), which is based on the foundational work of Daniel Kahneman and Amos Tversky in Prospect Theory. The core psychological mechanism here is loss aversion, which is specifically manifested by the pain of losing a possessed item being significantly greater than the joy of acquiring an item of the same value. In this paper, We will develop a new decoding strategy, named Loss Sampling, which is situated between greedy decoding and random sampling.

Algorithm 1 Pseudocode of Loss Sampling Strategy

procedure LOSS SAMPLING($probs, threshold$)
  $max\_prob \leftarrow probs.max()$
  $norm\_probs \leftarrow probs / max\_prob$
  $mask \leftarrow norm\_probs < threshold$
  $probs\_sort[mask] = 0.0$
  $norm\_probs.div\_(norm\_probs.sum)$
  $next\_token \leftarrow torch.multinomial(norm\_probs, 1)$
end procedure

3 Loss Sampling

Kahneman and Tversky, through their experiments (Kahneman and Tversky, 2013), discovered that for small or medium amounts of gains and losses, the negative utility brought about by a loss is approximately twice the positive utility generated by an equivalent gain, as illustrated in Figure 2.

Therefore, we set the value of λ in Equation (2) to 2. In our decoding sampling strategy, Loss Sampling, the utility weight of the token is selected x^α and the utility weight of the token is not selected $(-x)^\beta$ are both simplified to 1. Hence, the final value of $V(x)$ is as follows:

$$V(x) = \begin{cases} 1 & \text{if token is chosen ,} \\ -2 & \text{if token is not chosen .} \end{cases} \tag{5}$$

In Equation (3), the weighting function $w(p)$ represents an individual's psychological weighting of event outcomes. We consider that the model's perception of probability when sampling the next token is not linear, unsimilar to how humans perceive probabilities. Additionally, the transformation from logits to probabilities already involves a non-linear perception (i.e., the softmax transformation). Therefore, we simply set γ to 1, resulting in Equation (6).

$$w(p) = \begin{cases} p & \text{if token is chosen ,} \\ 1-p & \text{if token is not chosen .} \end{cases} \tag{6}$$

Therefore, the expected utility of a token being selected by the model as a candidate token, calculated using Equation (5) and (6), is as follows:

$$EU = 1 \cdot p + (-2) \cdot (1-p) = 3p - 2 \tag{7}$$

Table 1: The eight benchmarks across the four tasks in the experiment.

Task	Benckmark	Evaluation method
Text Translation	WMT17 (Bojar et al., 2017)	zero-shot
	Opus-100 (Zhang et al., 2020)	zero-shot
Reading Comprehension	SQuAD 2.0 (Rajpurkar et al., 2018)	zero-shot
	TriviaQA (Joshi et al., 2017)	zero-shot
Commonsense Reasoning	HellaSWAG (Zellers et al., 2019)	zero-shot
	CommonsenseQA (Talmor et al., 2018)	one-shot
Mathematical Reasoning	GSM8K (Cobbe et al., 2021)	one-shot-CoT
	SVAMP (Patel et al., 2021)	zero-shot-CoT

Table 2: Parameters of the experimental models and the number of their pretrained tokens.

Model	Size	Pretrained tokens amount
Llama-2	7B	2 trillion
	13B	2 trillion
Mistral	7B	$\sim$ 8 trillion
Llama-3	8B	15 trillion

In our Loss-Sampling strategy, similar to Top-k Sampling and Top-p Sampling , we add tokens that satisfy $EU > 0$ (i.e., $p > \frac{2}{3}$) to the candidate list, followed by multinomial random sampling. However, before filtering with $p > \frac{2}{3}$, it is necessary to preprocess the probabilities of each token in the model's output vocabulary. We have noted that after the softmax transformation of the model's output logits, it is not certain that any token's probability will exceed $\frac{2}{3}$, and in the Top-p and Top-k Sampling strategies, at least one token is guaranteed to be selected for the candidate list (for Top-k, $k = 1$ corresponds to greedy decoding), which is the token with the highest probability. Following the same concept, we take the highest probability and normalize it to the maximum value to ensure that a token will definitely be selected. Subsequent selection can be understood as: compared to the token with the highest probability, which will definitely be selected, other tokens with relatively higher probabilities are more capable of competing with the highest probability token. When these exceed the Loss Aversion threshold, the model expects to include them in the candidate list, meaning that the potential positive utility expectation of choosing that token (being ultimately sampled) outweighs the potential negative utility expectation (not being ultimately sampled).

In Figure 1, we illustrate two scenarios: (1) On the left, in the example of a highly peaked distribution, there is one token whose probability significantly surpasses that of the other tokens, leading to its exclusive addition to the candidate list for subsequent multinomial random sampling in our Loss Sampling method, thereby making the selection of the next token more stable; (2) On the right, in the example of a flat distribution, multiple tokens have similar probabilities (the ratio between them being greater than $\frac{2}{3}$), and they are all chosen to be added to the candidate list, thereby making the selection of the next token more random. However, for Top-p and Top-k, the highly peaked distribution example on the left can be well sampled by Top-p, but for Top-k, it might add some tokens with lower probabilities or lesser relevance to the candidate list. Conversely, the flat distribution example on the right might see Top-p missing some tokens that are close to the highest probability. Our Loss Sampling exhibits greater predictability in selecting tokens under a Highly Peaked Distribution and greater diversity under a Flat Distribution, thereby achieving a balance between quality and diversity in text generation. The pseudocode for Loss Sampling is as shown in Algorithm 1.

4 Experimental Protocols

Our goal is to empirically test the effectiveness of our sampling strategy, Loss Sampling, in large language models, specifically its ability to enhance the quality of text generation while maintaining diversity. Our approach involves evaluating various sampling strategies, which lie between greedy decoding and random sampling, on a variety of pretrained LLMs of different sizes and architectures. Next, we will provide detailed descriptions of our models, sampling strategies, and testing benchmarks.

4.1 Models

Our experiments are primarily conducted on the pretrained base models of the Llama2 (Touvron et al., 2023), Llama3, and Mistral (Jiang et al., 2023) model families, with detailed information provided in Table 2.

Table 3: Results of the effectiveness of Loss Sampling in Llama2-7B and Mistral-7B on WMT17.

Model	Method	WMT17				
		BLEU	ROUGE-1	ROUGE-2	ROUGE-L	METEOR
Llama2-7B	Top-p	10.46	45.02	21.87	38.12	40.28
	Top-k	9.81	44.33	21.04	37.48	38.61
	Loss	**13.92**	**54.63**	**28.03**	**46.92**	**48.78**
Mistral-7B	Top-p	10.8	47.88	23.61	40.57	42.70
	Top-k	9.62	45.41	21.60	38.51	40.19
	Loss	**12.74**	**50.22**	**26.30**	**43.20**	**45.78**

Focusing on Base Model: During the development of LLMs, there are typically two stages: the pretraining stage and the alignment stage (Devlin et al., 2018; Vaswani et al., 2017). In these stages, the model is referred to as the base model and the fine-tuned model, respectively. It is important to note that the fine-tuned model is no longer a general-purpose compressor, as it does not model the next token distribution for arbitrary text but only for structured (query-response) data (Huang et al., 2024). Therefore, for evaluating next token sampling strategies, the base model, which models the next token distribution for any text, is more appropriate. Hence, in this work, we focus only on the base model.

4.2 Sampling Policy Setup

As is well-known, in generating text, the parameters p and k in Top-p and Top-k are control parameters that influence quality and diversity. Therefore, we conducted a grid search for the optimal balance parameters in Section 6, aiming to achieve a balance between text quality and diversity. For Top-p and Top-k Sampling, common settings are $k = 5$ to 50 (Fan et al., 2018) and $p = 0.7$ to 0.95 (Holtzman et al., 2019). Detailed experimental results can be found in Section 6. Ultimately, we opted for Top-k Sampling with $k = 10$ and Top-p Sampling with $p = 0.85$ or 0.9 (text translation and commonsense reasoning tasks use $p = 0.85$, while the other two tasks use $p = 0.9$).

4.3 Benchmarks

We evaluate the effectiveness of our Loss Sampling across four task domains, totaling eight benchmarks. All models are evaluated using few-shot in-context learning (Brown et al., 2020) or in a zero-shot manner (Larochelle et al., 2008; Palatucci et al., 2009). Additionally, mathematical reasoning task employs the "Let's think step by step" prompt for chain of thought (CoT) (Wei et al., 2022) reasoning. Detailed information is shown in Table 1.

5 Main Results

Text Translation: Our experimental results for the text translation task are shown in Table 3, which includes only the results for the Llama2-7B and Mistral-7B on the WMT17 dataset. Comprehensive and detailed results can be found in Supplementary Material S.1 . It is observed that our Loss Sampling method has led to performance improvements across all metrics, and it remains effective for Mixture of Experts (MoE) models as well. Moreover, according to the results in Table 4, our method performs comparably to Top-p and Top-k Sampling under experimental parameters in terms of generating diversity. It is noteworthy that the performance of the 8B-sized Llama3 model is comparable to, and even surpasses, that of the 13B-sized Llama2 on the Opus-100 dataset, likely due to the fact that the number of pretrained tokens for Llama3 is 7.5 times that of Llama2.

Reading Comprehension: The specific experimental results for the reading comprehension question-answering task are shown in Figure 3. It can be observed that our Loss Sampling method has improved the accuracy of answers on both the SQuAD2.0 and TriviaQA datasets, with the greatest enhancements noted on the Llama3-8B and Llama2-13B models respectively.

Commonsense Reasoning: Our experimental results for the commonsense reasoning task are shown in Figure 4. Similar to the performance improvements observed in the first two tasks, enhancements can also be seen on both the HellaSWAG and CommonsenseQA datasets, demonstrating the broad effectiveness of our Loss Sampling method.

Mathematical Reasoning: The experimental results for the mathematical reasoning task are shown in Figure 5. We find that on the GSM8K dataset, Loss Sampling did not enhance performance in two models of Llama2 and even resulted in a decline; however, it significantly improved performance on Llama3-8B, likely due to its inherently stronger capabilities, and

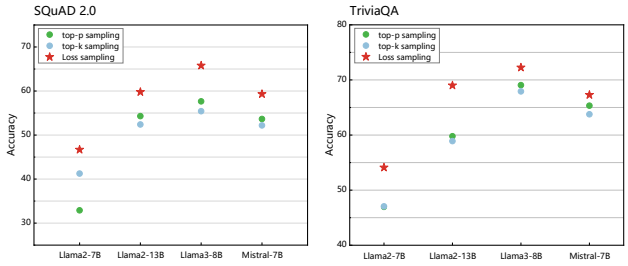

Figure 3: Results on the effectiveness of Loss Sampling in reading comprehension tasks.

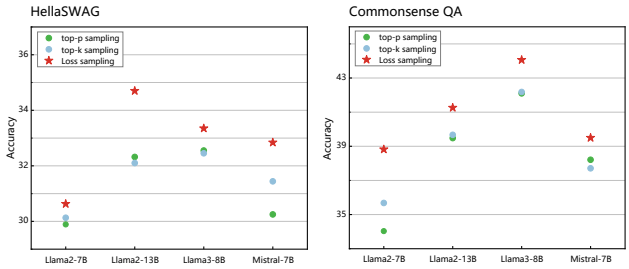

Figure 4: Results on the effectiveness of Loss Sampling in commonsense reasoning tasks.

also yielded a moderate improvement on Mistral-7B. On the SVAMP dataset, performance improvements were only observed in Llama2-13B and Llama3-8B, while in Llama2-7B and Mistral-7B, performance was only comparable to the best Top-k Sampling.

Overall, our Loss Sampling has demonstrated its effectiveness across multiple task domains, not only maintaining the diversity of the generated text but also producing text of higher quality than that generated by Top-p and Top-k Sampling.

6 Grid Search for Balanced p & k

As previously mentioned, while Greedy Decoding is simple and efficient, it can lead to generated texts being monotonous and repetitive. On the other hand, Random Sampling can increase the diversity of the generated texts but may result in incoherence and lower quality. Although Top-p and Top-k Sampling lie between Greedy Decoding and Random Sampling, the values of p and k still significantly affect the balance between the quality and diversity of the generated texts. Therefore, in this section, we conduct a grid search within individual datasets across four task domains to find the optimal p and k values that best balance the quality and diversity of the generated texts.

The grid search results on the Text Translation task using the WMT17 dataset are shown in Figure 6, and the results for the other three tasks are presented in

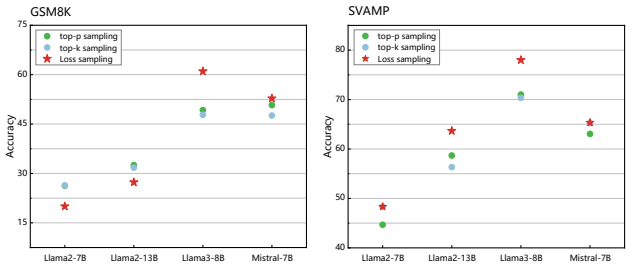

Figure 5: Results on the effectiveness of Loss Sampling in mathematical reasoning tasks.

Supplementary Material S.2. It can be observed that, in the text translation task, the optimal balance is achieved with k value of 10 and p value of 0.85; in the reading comprehension task, the optimal balance is with k value of 10 and p value of 0.9; in the commonsense reasoning task, the best balance is found with k value of 10 and p value of 0.85; and in the mathematical reasoning task, the optimal balance is with k value of 10 and p value of 0.9.

Generative Diversity: Loss Sampling vs. Top-p & Top-k Next, we also compared the performance in diversity of our Loss Sampling method with the optimally balanced Top-p and Top-k Sampling. The specific experimental results are shown in Table 4. It can be observed that Loss Sampling, with a threshold of $\frac{2}{3}$, performs comparably to Top-p and Top-k in terms of generating text diversity.

7 Extended Experiment

7.1 Does Loss Sampling work on multimodal large language model?

We have also extended Loss Sampling to multimodal tasks, not just pure natural language tasks. We conduct experiments using the multimodal LLM LLaVA-V1.5-13B (Liu et al., 2023a), and selected datasets from visual question answering tasks — TextQA (Singh et al., 2019) and MMBench (Liu et al., 2023b), as well as Flickr8k (Hodosh et al., 2013) from the image caption task, as benchmarks. The settings for Top-p and Top-k were configured at 0.9 and 10, respectively. The experimental results, as shown in Table 5, indicate that our method has led to performance improvements across all three datasets in two task domains. It is worth noting that the improvement varies between the TextQA and MMBench within the same task. A possible reason is that MMBench involves multiple choice where selecting a specific token (A,B,C,D) is more critical.

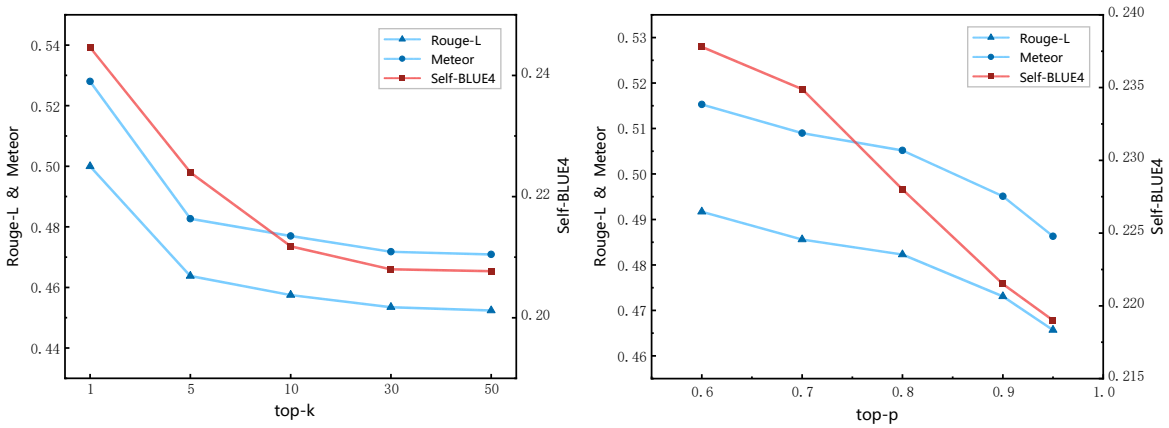

Figure 6: Grid search was performed on Top-k and Top-p for text translation (WMT17).

Table 4: Comparison of three sampling strategies in terms of text generation diversity, where **bold** indicates the best performance of TTR, and underline indicates the best performance of Self-BLEU4.

Method	Metric	WMT17	Hellaswag	SVAMP
Top-p Sampling	TTR ↑	0.1475	0.0651	**0.4247**
	Self-BLUE4 ↓	0.2216	0.4387	0.1364
Top-k Sampling	TTR ↑	0.1507	0.0664	0.4053
	Self-BLUE4 ↓	0.2119	0.4202	0.1346
Loss Sampling	TTR ↑	**0.1765**	**0.0855**	0.3626
	Self-BLUE4 ↓	0.2096	0.4377	0.1365

7.2 Can Loss Sampling work with Beam seach?

We have also extended our Loss Sampling to the Beam search (Shaham and Levy, 2021) decoding strategy. Beam search is an improvement over the greedy strategy; at each time step, the model no longer retains only the highest scoring output but instead keeps *num_beams* outputs, resulting in *num_beams* generated sequences. When $num\_beams = 1$, Beam Search degenerates into greedy decoding. However, for the original Beam Search, it extends new candidate solutions from each candidate at each time step, then selects the top *num_beams* most probable to continue the search, discarding the rest. Therefore, we integrated our Loss Sampling method into the variant of Beam Search known as Beam Sampling, which uses Top-p (or Top-k) sampling to improve the process of selecting the highest probability solutions in Beam Search. We continue using LLaVA-V1.5-13B, conducting experiments on the image caption task with the Flickr8k dataset, setting *num_beams* to 3, with other settings consistent as described in Section 7.1. The experimental results, shown in Table 6, demonstrate that our method is effective on Beam Sampling as well, leading to significant performance improvements.

8 Related Work

In natural language generation tasks, the quality of the model's output is closely related to the sampling strategy. Common generation strategies include Greedy Decoding (Brown et al., 1990; Sutskever et al., 2014; Vaswani et al., 2017), Top-k Sampling (Fan et al., 2018; Holtzman et al., 2018; Radford et al., 2019, 2018), and Top-p Sampling (also known as Nucleus Sampling) (Holtzman et al., 2019; Radford et al., 2019; Dathathri et al., 2019).

Greedy Decoding is an efficient and deterministic generation strategy where the model selects the highest probability token from the current distribution at each step. This approach ensures locally optimal decisions at each step, providing a predictable output path and eliminating randomness in sequence generation.

However, greedy decoding has notable drawbacks. It focuses exclusively on local optimality, often neglecting the global diversity of the text and leading to inflexibility. Specifically, it can result in repetitive and verbose outputs, especially in longer texts, as the model tends to select the same words or phrases repeatedly (Holtzman et al., 2019).

Table 5: Validation results of the effectiveness of Loss Sampling on multimodal LLM.

Method	TextQA	MMBench	Flickr8k				
	Accuracy	Accuracy	BLEU	ROUGE-1	ROUGE-2	ROUGE-L	Meteor
Top-p	57.97	76.28	14.79	57.23	31.64	53.35	54.61
Top-k	56.48	75.62	13.74	55.36	29.40	50.59	53.15
Loss	**61.05**	**76.76**	**16.51**	**60.74**	**34.93**	**56.08**	**58.64**

Table 6: Validation results of the effectiveness of Loss Sampling with Beam Sampling.

Method	Flickr8k				
	BLEU	ROUGE-1	ROUGE-2	ROUGE-L	METEOR
Beam Sampling with Top-p	17.31	60.10	34.58	56.11	58.78
Beam Sampling with Top-k	18.04	60.47	34.97	56.70	59.13
Beam Sampling with Loss	**19.30**	**62.03**	**37.76**	**57.81**	**60.40**

Top-k Sampling is a commonly used strategy in natural language generation, where the fundamental idea is to control the generated output by limiting the size of the candidate token set. During each generation step, the model assigns probability values to all possible tokens, retains the top k, and randomly samples from these based on their relative probabilities to generate the next token (Fan et al., 2018; Holtzman et al., 2018; Radford et al., 2019).

The advantage of Top-k sampling is its ability to exclude low-probability tokens, preventing less relevant or lower-quality words from affecting the output. However, by focusing only on the top k tokens, Top-k sampling has limitations, as it may eliminate creative alternatives, leading to a lack of diversity and innovation in the text, especially in tasks that require flexible and varied generation.

Top-p Sampling is a dynamic truncation strategy based on cumulative probability. Rather than fixing the size of the candidate set, it selects the smallest set of tokens whose cumulative probability reaches a predefined threshold p (Radford et al., 2019). The model sorts tokens by probability and adds them until the cumulative probability exceeds p.

Unlike Top-k, Top-p does not impose a fixed size on the candidate set but rather adjusts the set size dynamically according to the current probability distribution. This flexibility allows Top-p to adapt better to varying contexts, especially in high-uncertainty situations, generating more varied results (Keskar et al., 2019). However, Top-p sampling also has its drawbacks. Its effectiveness relies on the choice of the p threshold. If p is too low, the candidate set may be too small, resulting in less diverse outputs. If p is too high, too many low-probability tokens may be included, reducing text quality. Thus, careful tuning of p is essential for optimizing performance in specific tasks.

9 Discussion

Conclusion: Inspired by Loss Aversion from Prospect Theory, we introduce a novel decoding strategy called Loss Sampling. This approach addresses the potential shortcomings of Top-k and Top-p sampling methods, achieving both text generation diversity (as defined by TTR and Self-BLEU, with differences within 4%) and superior generation quality compared to Top-k and Top-p. In text translation tasks, Loss Sampling improve performance by 5% to 26% across metrics, including BLEU, ROUGE and METEOR. It boosts accuracy by 5% to 19% in reading comprehension and commonsense reasoning, and shows an 11% gain in math for strong models. Additionally, we extend its effectiveness to multimodal tasks and combined it with Beam Search.

Limitations: However, our work is still not complete, and future research directions could include: (1) Broader experiments: Apply Loss Sampling to a wider range of NLP tasks to validate its generalizability. (2) Combining multiple strategies: Combining Loss Sampling with other decoding methods to further improve the quality and flexibility of text generation. (3) Optimize Loss Sampling by adjusting loss aversion parameters based on experimental feedback. (4) Applications on other modalities and multimodal: Explore the effectiveness of Loss Sampling in other modalities and multimodal tasks through further studies.

References

Bojar, O. r., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huang, S., Huck, M., Koehn, P., Liu, Q., Logacheva, V., Monz, C., Negri, M., Post, M., Rubino, R., Specia, L., and Turchi, M. (2017). Findings of the 2017 conference on machine translation (wmt17). In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.

Brown, P. F., Cocke, J., Della Pietra, S. A., Della Pietra, V. J., Jelinek, F., Lafferty, J., Mercer, R. L., and Roossin, P. S. (1990). A statistical approach to machine translation. *Computational linguistics*, 16(2):79–85.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., et al. (2021). Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Dathathri, S., Madotto, A., Lan, J., Hung, J., Frank, E., Molino, P., Yosinski, J., and Liu, R. (2019). Plug and play language models: A simple approach to controlled text generation. *arXiv preprint arXiv:1912.02164*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Fan, A., Lewis, M., and Dauphin, Y. (2018). Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833*.

Hodosh, M., Young, P., and Hockenmaier, J. (2013). Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47:853–899.

Holtzman, A., Buys, J., Du, L., Forbes, M., and Choi, Y. (2019). The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Holtzman, A., Buys, J., Forbes, M., Bosselut, A., Golub, D., and Choi, Y. (2018). Learning to write with cooperative discriminators. *arXiv preprint arXiv:1805.06087*.

Huang, Y., Zhang, J., Shan, Z., and He, J. (2024). Compression represents intelligence linearly. *arXiv preprint arXiv:2404.09937*.

Jiang, A., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al. (2023). Mistral 7b (2023). *arXiv preprint arXiv:2310.06825*.

Johnson, W. (1944). Studies in language behavior: A program of research. *Psychological Monographs*, 56(2):1–15.

Joshi, M., Choi, E., Weld, D. S., and Zettlemoyer, L. (2017). Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

Kahneman, D. (2003). Maps of bounded rationality: Psychology for behavioral economics. *American economic review*, 93(5):1449–1475.

Kahneman, D., Knetsch, J. L., and Thaler, R. H. (1991). Anomalies: The endowment effect, loss aversion, and status quo bias. *Journal of Economic perspectives*, 5(1):193–206.

Kahneman, D. and Tversky, A. (2013). Prospect theory: An analysis of decision under risk. In *Handbook of the fundamentals of financial decision making: Part I*, pages 99–127. World Scientific.

Keskar, N. S., McCann, B., Varshney, L. R., Xiong, C., and Socher, R. (2019). Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase-based translation. In *2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Langauge Technology (HLT-NAACL 2003)*, pages 48–54. Association for Computational Linguistics.

Larochelle, H., Erhan, D., and Bengio, Y. (2008). Zero-data learning of new tasks. In *AAAI*, volume 1, page 3.

Levy, J. S. (1992). An introduction to prospect theory. *Political psychology*, pages 171–186.

Liu, H., Li, C., Li, Y., and Lee, Y. J. (2023a). Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*.

Liu, Y., Duan, H., Zhang, Y., Li, B., Zhang, S., Zhao, W., Yuan, Y., Wang, J., He, C., Liu, Z., et al. (2023b). Mmbench: Is your multi-modal model an all-around player? *arXiv preprint arXiv:2307.06281*.

McCarthy, P. M. and Jarvis, S. (2010). Mtld, vocd-d, and hd-d: A validation study of sophisticated approaches to lexical diversity assessment. *Behavior research methods*, 42(2):381–392.

Mikolov, T., Karafiát, M., Burget, L., Cernockỳ, J., and Khudanpur, S. (2010). Recurrent neural network based language model. In *Interspeech*, volume 2, pages 1045–1048. Makuhari.

Palatucci, M., Pomerleau, D., Hinton, G. E., and Mitchell, T. M. (2009). Zero-shot learning with semantic output codes. *Advances in neural information processing systems*, 22.

Patel, A., Bhattamishra, S., and Goyal, N. (2021). Are nlp models really able to solve simple math word problems? *arXiv preprint arXiv:2103.07191*.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. (2018). Improving language understanding by generative pre-training.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Rajpurkar, P., Jia, R., and Liang, P. (2018). Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*.

Richards, B. (1987). Type/token ratios: What do they really tell us? *Journal of child language*, 14(2):201–209.

Shaham, U. and Levy, O. (2021). What do you get when you cross beam search with nucleus sampling? *arXiv preprint arXiv:2107.09729*.

Singh, A., Natarajan, V., Shah, M., Jiang, Y., Chen, X., Batra, D., Parikh, D., and Rohrbach, M. (2019). Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Talmor, A., Herzig, J., Lourie, N., and Berant, J. (2018). Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Tversky, A. and Kahneman, D. (1991). Loss aversion in riskless choice: A reference-dependent model. *The quarterly journal of economics*, 106(4):1039–1061.

Tversky, A. and Kahneman, D. (1992). Advances in prospect theory: Cumulative representation of uncertainty. *Journal of Risk and uncertainty*, 5:297–323.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. (2019). Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

Zhang, B., Williams, P., Titov, I., and Sennrich, R. (2020). Improving massively multilingual neural machine translation and zero-shot translation. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639, Online. Association for Computational Linguistics.

Zhu, Y., Lu, S., Zheng, L., Guo, J., Zhang, W., Wang, J., and Yu, Y. (2018). Texygen: A benchmarking platform for text generation models. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 1097–1100.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Not Applicable]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Not Applicable]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Not Applicable]
   (b) Complete proofs of all theoretical results. [Not Applicable]

(c) Clear explanations of any assumptions. [Not Applicable]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Not Applicable]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Not Applicable]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [No]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Yes]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Yes]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Choice is what matters after Attention
Supplementary Materials

1 More Results of Text Translation

Table 1: Experimental results of Llama2-13B and Llama3-8B models on the Opus-100 dataset

Model	Method	WMT17				
		BLEU	ROUGE-1	ROUGE-2	ROUGE-L	METEOR
Llama2-13B	Top-p	12.77	53.69	27.11	45.97	47.98
	Top-k	10.51	48.26	23.61	40.81	42.23
	Loss	**14.75**	**56.57**	**29.82**	**49.00**	**51.19**
Llama3-8B	Top-p	13.88	55.41	29.05	47.31	49.51
	Top-k	12.60	53.86	27.37	45.75	47.70
	Loss	**16.05**	**57.88**	**32.01**	**49.92**	**52.57**

Table 2: Experimental results of Llama2,Llama3 and Mistral models on the Opus-100 dataset.

Model	Method	Opus-100				
		BLEU	ROUGE-1	ROUGE-2	ROUGE-L	METEOR
Llama2-8B	Top-p	3.82	21.44	9.90	18.55	26.69
	Top-k	3.61	21.31	9.48	18.33	26.12
	Loss	**4.29**	**21.64**	**10.44**	**19.22**	**28.15**
Llama2-13B	Top-p	13.97	51.62	27.26	45.69	45.35
	Top-k	12.89	50.47	26.01	44.38	44.15
	Loss	**15.84**	**53.66**	**29.52**	**47.85**	**47.58**
Llama3-8B	Top-p	16.52	55.37	31.07	49.44	49.33
	Top-k	15.29	54.05	29.58	48.03	47.58
	Loss	**17.82**	**57.17**	**32.65**	**51.24**	**50.99**
Mistral-7B	Top-p	9.70	41.71	21.31	36.67	36.33
	Top-k	8.82	39.59	19.81	34.93	34.13
	Loss	**11.74**	**44.82**	**24.11**	**39.96**	**39.46**

2 More Results of Grid Search

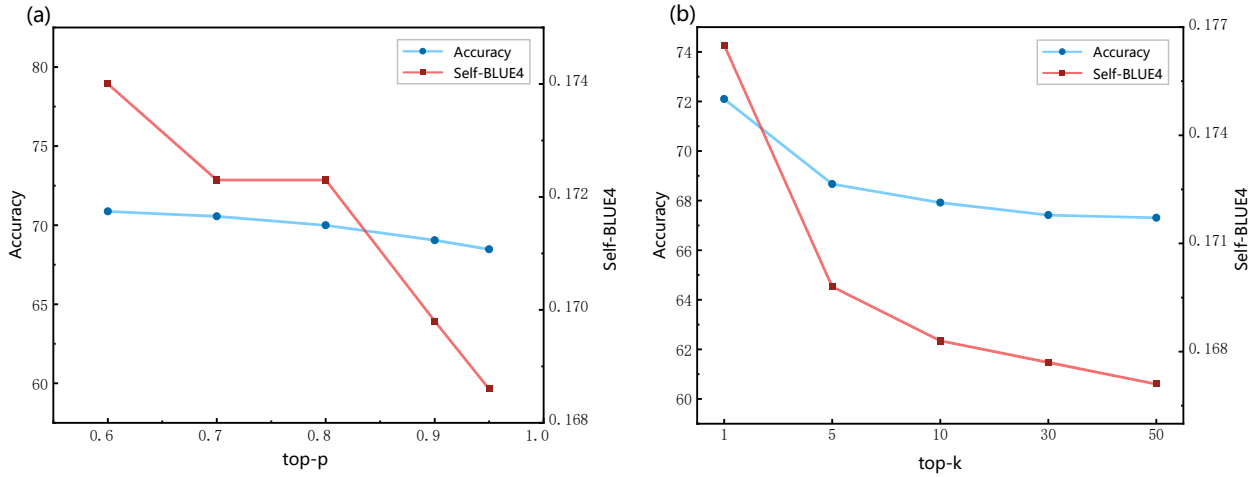

Figure 1: Grid search was performed on Top-k and Top-p for reading comprehension (TriviaQA).

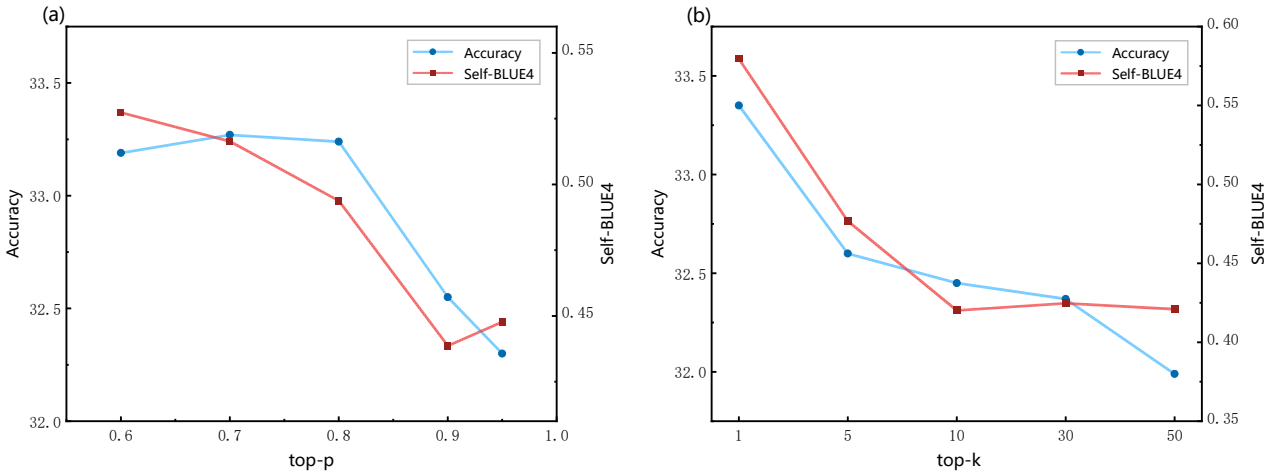

Figure 2: Grid search was performed on Top-k and Top-p for commonsense reasoning (HellaSWAG).

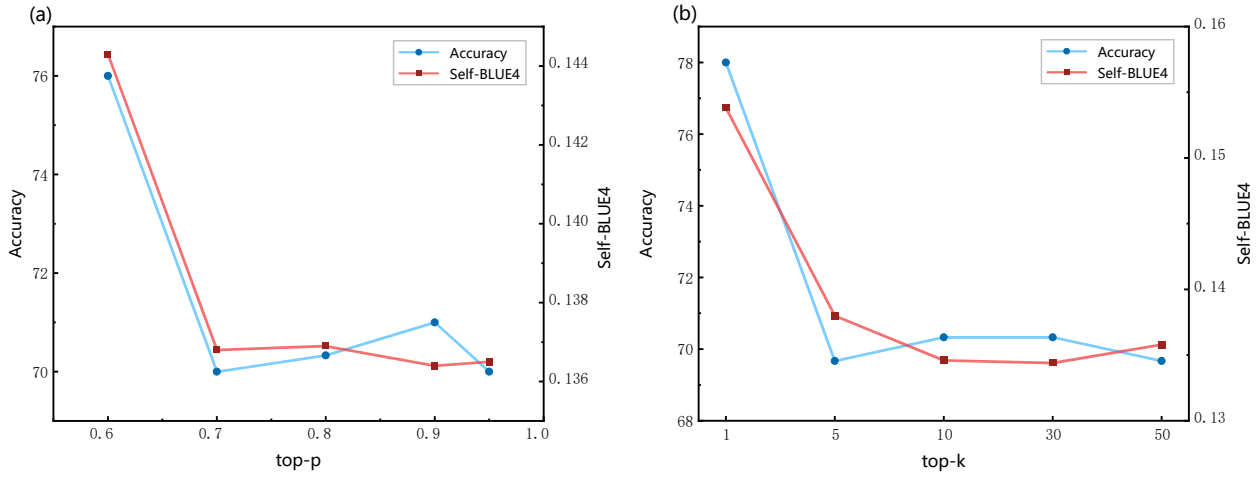

Figure 3: Grid search was performed on Top-k and Top-p for mathematical reasoning (SVAMP).