Dynamic Collaboration of Multi-Language Models based on Minimal Complete Semantic Units

Chao Hao[1,2], **Zezheng Wang**[2,†], **Yanhua Huang**[2], **Ruiwen Xu**[2], **Wenzhe Niu**[2], **Xin Liu**[3], **Zitong Yu**[1,4,5,†]

[1]Great Bay University [2]Xiaohongshu Inc. [3]Lappeenranta University of Technology
[4]Guangdong Provincial Key Laboratory of Intelligent Information Processing & Shenzhen Key Laboratory of Media Security, Shenzhen University
[5]Dongguan Key Laboratory for Intelligence and Information Technology

Abstract

This paper investigates the enhancement of reasoning capabilities in language models through token-level multi-model collaboration. Our approach selects the optimal tokens from the next token distributions provided by multiple models to perform autoregressive reasoning. Contrary to the assumption that more models yield better results, we introduce a distribution distance-based dynamic selection strategy (DDS) to optimize the multi-model collaboration process. To address the critical challenge of vocabulary misalignment in multi-model collaboration, we propose the concept of minimal complete semantic units (MCSU), which is simple yet enables multiple language models to achieve natural alignment within the linguistic space. Experimental results across various benchmarks demonstrate the superiority of our method. The code will be available at https://github.com/Fanye12/DDS.

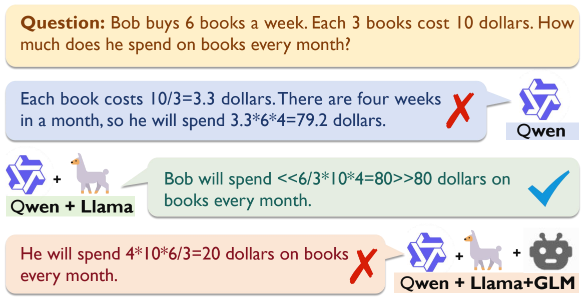

Figure 1: **Motivation of the proposed DDS.** For the same mathematical problem, Single LLM Qwen gave the wrong answer, token-level collaboration between Qwen and Llama produced the correct answer, whereas collaboration among Qwen, Llama, and GLM resulted in an incorrect answer. This demonstrates that multi-model collaboration can improve answer but having more models in collaboration does not necessarily improve outcomes; selecting the appropriate models for collaboration is essential.

1 Introduction

With the rapid development of large language models (LLMs), numerous impressive works such as GPT4 (Achiam et al., 2023), Llama3 (Dubey et al., 2024), and Qwen2 (Yang et al., 2024) have emerged. People are increasingly accustomed to seeking answers from LLMs when encountering problems, and even researchers consult LLMs during their scientific work. Although LLMs have demonstrated remarkable capabilities in many areas of natural language processing (NLP), they often show their inability to perform complex reasoning tasks (Fu et al., 2022). Therefore, how to further improve the performance of LLMs in complex reasoning tasks has become a hot topic (Kojima et al., 2022; Liang et al., 2023). Enhancing model performance from training side is very costly, as training a language model requires significant resources. Furthermore, performance improvements have begun to plateau due to the slowing impact of scaling laws (Kaplan et al., 2020; Touvron et al., 2023). Therefore, more and more research (Wei et al., 2022; Kojima et al., 2022; Madaan et al., 2024; Liang et al., 2023; Xu et al., 2024) has begun to focus on improving model performance with some simple and low-cost methods.

Ensembling is a highly promising approach that has been extensively studied since the early days of deep learning. Recent studies have also confirmed that ensembling multiple large language models (LLMs) can further enhance their capabilities (Yao et al., 2024; Shen et al., 2024). Different LLMs usually have different knowledge boundaries and their own strengths (Wan et al., 2024; Jiang et al., 2023). If their capabilities can be well synergized, it will certainly enhance the reasoning ability of the language model and break through the performance bottleneck of a single LLM (Khan et al., 2024; Du

† Corresponding authors. yuzitong@gbu.edu.cn, zzwang1993@foxmail.com.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 12894–12911
November 4-9, 2025 ©2025 Association for Computational Linguistics

et al., 2023), which is also the expected goal of multi-model collaboration. Previous multi-model collaboration methods (Khan et al., 2024; Liang et al., 2023; Du et al., 2023) mostly conduct majority voting or discuss at the level of the overall answer. For the same question, each LLM puts forward its own point of view and tries to convince other models, so as to finally reach a consensus among multiple models. The debate process usually involves selecting the final answer based on certain rules or introducing a new referee model. This may require many rounds of debate and relies heavily on one of the models to give a strong correct answer and convince the other debaters to get the final answer.

It is generally believed that the knowledge of LLM is stored in its massive parameters, but the output next token distribution is the specific external manifestation of its knowledge [1] (Hinton, 2015; Wan et al., 2024; Radford et al., 2019). **Therefore, a straightforward idea is to combine the knowledge of multiple LLMs by combining the next token distributions given by these models**. Token-level multi-model collaboration is based on this approach to enhance the reasoning capabilities of LLMs. It not only avoids the need for complex interaction rules among multiple models but can also exhibit emergent abilities to a certain extent, offering correct answers when individual models cannot do so independently.

Contrary to the assumption that more models yield better results, we find that not all model additions will have a positive impact on the final results. As shown in Fig. 1, for three similarly capable LLMs, when Qwen and Llama collaborate to provide the correct answer, the addition of GLM actually results in an incorrect final outcome. This demonstrates that simply increasing the number of models does not necessarily lead to positive results. The key is to select the appropriate models for collaboration. Based on the principle that "there is typically only one correct answer, whereas incorrect answers can be numerous and varied", we propose a distribution distance-based dynamic selection strategy (DDS) to solve this problem. Specifically, we calculate the pairwise distances between the next token distributions provided by multiple LLMs. We filter out the outlier distributions that are far from the majority, retaining those that are closer together, which are considered to be near the correct answer.

In addition, due to inherent differences in model architecture, training data, and training processes among different LLMs, they typically exhibit vocabulary discrepancies. For instance, the word "Llama" might be tokenized into "Lla" and "ma" by model A's tokenizer, while model B's tokenizer could split it into "Ll" and "ama". Such discrepancies prevent us from performing integration by simply averaging multiple probability vectors, as is traditionally done in ensemble learning. Existing methods (Yu et al., 2024; Huang et al., 2024; Yao et al., 2024) typically attempt to resolve this issue by aligning different vocabularies. However, this alignment process often necessitates additional computation and inevitably introduces certain noise, which can affect the final outcomes.

In this paper, we propose the concept of "minimal complete semantic units (MCSU)" to achieve natural alignment across different LLMs in natural language. Specifically, we use MCSU to replace token as the smallest semantic unit in the LLM autoregressive generation process. For instance, the word "apple" is encoded as a single token, which we consider to have complete semantic meaning, while "Llama" might be split into "Lla" and "ma", where "Lla" and "ma" do not constitute a complete semantic unit. For these exceptional cases, we allow the LLM to continue generating tokens until the generated tokens can be combined into MCSU. We then use the product of joint probabilities to represent the probability score for this MCSU. Moreover, we find that about 90% of common English words are encoded as a single token (see Appendix D), meaning that most commonly used tokens are already MCSUs. Therefore, the introduction of MCSUs does not result in significant additional computation, providing a low-cost solution to the vocabulary misalignment issue during token-level multi-model collaboration.

2 Related Works

2.1 Output-level Model Emsembling

Output-level model emsembling is usually done by imitating some group collaboration behaviors of humans to conduct multi-model collaboration (Khan et al., 2024; Liang et al., 2023; Du et al., 2023; Yin et al., 2023; Sun et al., 2023). The goal is to combine the advantages of multiple models and introduce external feedback from other models

[1] LLM's output is obtained by autoregressive sampling from this distribution, so LLM's knowledge determines the quality of its output.

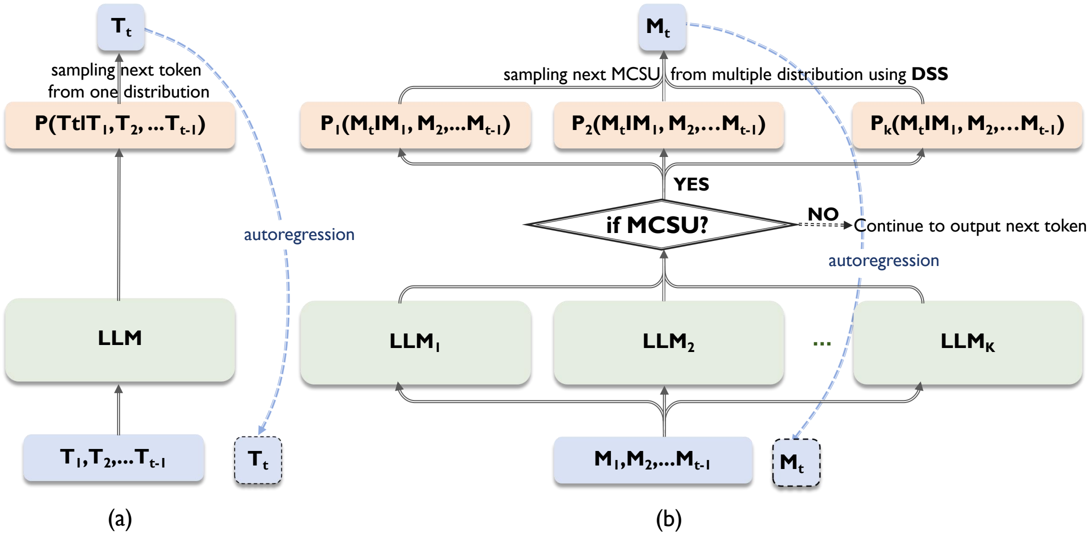

Figure 2: From the implementation point of view, the proposed DDS is an adjustment based on single model autoregression. (a) A single LLM samples the next token from its output next token distribution and generates the response autoregressively. (b) The proposed DDS selects the optimal next MCSU from multiple next MCSU distributions given by multiple LLMs and generates the response autoregressively.

to break through the performance bottleneck of a single model (Liang et al., 2023). The simplest method of multi-model collaboration is to perform majority voting based on the answers given by multiple LLMs. Jiang et al. (2023) developed an auxiliary ranking model to evaluate and select the best candidate output from multiple LLMs' responses. Similarly, Shnitzer et al. (2023) designed a router that determines the optimal candidate model based on the given question.

Moreover, many methods begin to allow LLMs to interact with each other. Liang et al. (2023) and Du et al. (2023) enhanced the performance of LLM in specific tasks by allowing multiple LLMs to debate on the same problem and finally reach a consensus. Yin et al. (2023) proposed a cross-model exchange based on network topology to obtain feedback from other LLMs to improve their own output. Inspired by human behavior, Sun et al. (2023) proposed multiple collaboration modes, including discussion, review, and retrieval, to jointly work towards enhancing inference performance.

Output-level model emsembling methods often require introducing additional components that need to be trained or an extra referee LLM to help select the final result. Some methods even necessitate designing complex interaction rules, all of which increase the complexity. Furthermore, the accuracy of the final result heavily depends on one of the models providing a strong correct answer. These issues have brought great limitations to the output-level model emsembling methods.

2.2 Token-level Model Emsembling

Unlike output-level ensembling methods that integrate at the final answer stage, token-levelensembling methods perform integration at each step of the LLM generation process. These methods select the optimal next token by integrating the next token probability distributions output by multiple LLMs. Shen et al. (2024) developed a trainable classifier to help determine which LLM should be utilized for completing the current step in the autoregressive generation process. Wan et al. (2024) leveraged output probability vectors from various models during the training process, using these vectors as labels to distill the knowledge.

Owing to the vocabulary discrepancies among different LLMs mentioned previously, numerous approaches (Yu et al., 2024; Xu et al., 2024; Huang et al., 2024; Yao et al., 2024) concentrate on aligning their tokenizers to achieve improved ensemble outcomes. Xu et al. (2024) proposed a method that directly learns the projection matrices between different vocabularies, using overlapping tokens as anchors to bridge the gap between heterogeneous LLMs. Similarly, Huang et al. (2024) utilized anchors to calculate the relative representations to different vocabularies, thereby enabling the vocabulary projection indirectly. In another approach, Yu et al. (2024) also relied on anchors to calculate the relative representations, achieving a similar indirect vocabulary projection. Yao et al. (2024) introduced the UNITE, a novel approach that efficiently combines models by focusing on the union of the top-k tokens from each model, thereby avoiding the need for full vocabulary alignment and reducing computational overhead.

However, this vocabulary alignment operation is bound to introduce noise, which can affect the final performance. The proposed MCSU aims to mitigate this impact.

3 Method

Given the same question, different LLMs usually give different answers. This is because they have learned different knowledge due to the difference in network architecture, training data and training process (Raiaan et al., 2024). As mentioned earlier, it is generally believed that the knowledge is stored in the huge parameters of LLMs (Radford et al., 2019), and the output next token probability distribution is the specific external manifestation of their knowledge (Hinton, 2015; Wan et al., 2024). Therefore, our starting point is to combine the knowledge of different LLMs by combining these distributions:

$$\begin{aligned} P_t &= Combine(P_t^1, P_t^2, ..., P_t^K), \\ KN_t &= Combine(KN_t^1, KN_t^2, ..., KN_t^K), \end{aligned} \tag{1}$$

where P_t represents distributions, KN represents knowledge and $Combine$ represents a certain combination.

Fig. 2 provides an overview of the proposed method, which mainly consists of two parts. The first part achieves a natural alignment of different LLMs through the MCSU, thus avoiding the complex operations previously introduced for vocabulary alignment. In the second part, we introduce DDS. We hypothesize that not all probability vectors output by LLMs are beneficial to the final result. Therefore, at each step of autoregression, we dynamically select several appropriate probability distributions for integration. We will introduce them in detail in the following sections.

3.1 Minimal Complete Semantic Units

In LLMs, a token is the smallest semantic unit for calculation. Natural language is segmented into individual tokens by a tokenizer, and outputs are generated autoregressively by predicting the next token until the end token is encountered. However, some tokens may not always convey complete semantic meaning. To address this, we introduce the concept of a minimal complete semantic unit (MCSU), which is defined as a word, punctuation mark, or number representing the smallest unit of complete meaning. Tokens that only represent part of a word lack complete semantic meaning and contribute to vocabulary misalignment across different LLMs. [2]

For alphabetic languages like English, the tokenization process often splits some words into multiple subword tokens, which may result in individual tokens lacking complete semantic meaning. This is precisely why we proposed the Minimum Complete Semantic Unit (MCSU) concept. In languages such as English or French, whitespace can serve as a reliable delimiter for identifying MCSUs.

In contrast, for logographic languages like Chinese, each token inherently represents either a single character or a complete word, thus constituting an MCSU by definition without requiring additional segmentation criteria. We believe these two cases adequately represent the majority of language types. The experiments in this paper were mostly conducted on English datasets, but the results in Table 4 also demonstrate its good performance on Chinese datasets.

During integration, if the next token produced by an LLM does not form an MCSU, the model continues generating tokens until a sequence of consecutive tokens can form an MCSU. Since English words are typically separated by spaces or punctuation marks, it is relatively straightforward to determine whether a sequence constitutes an MCSU. We represent the probability of an MCSU using the product of the joint probabilities of its constituent tokens.

For an MCSU M consisting of several tokens, $M = T_1, T_2, \ldots, T_n$, where T denotes a token, its probability is expressed as:

$$P_M = \prod_{i=1}^{n} P(T_i | T_1, T_2, \ldots, T_{i-1}). \tag{2}$$

We use MCSUs and their probabilities to replace the original next token probability distribution for subsequent multi-model collaboration, addressing

[2]For example, "Llama" might be split into the tokens "Lla" and "ma" by model A, and into "Ll" and "ama" by model B. These tokens are not MCSUs and are a primary reason for the misalignment of vocabularies across different LLMs.

the issue of inconsistent tokens across different LLM vocabularies. Additionally, to reduce computational complexity, we employ Top-k (Fan et al., 2018) sampling to select the top K MCSUs with the highest probabilities.

3.2 Distribution Distance-based Dynamic Selection Strategy

As mentioned earlier, not all probability distributions generated by LLMs have a positive impact on the final integration. This is understandable because the model may inherently struggle with answering the given question, leading to potentially erroneous probability distributions. Based on the principle that "there is typically only one correct answer, whereas incorrect answers can be numerous and varied", we propose the distrition distance-based dynamic selection strategy to help filter out the probability distributions used for the final integration.

According to this principle, we believe that if the distance between two distributions is closer, they are nearer to the correct answer, as incorrect answers typically exhibit greater variability. We use the probability distribution representing the next MCSU instead of the next token's probability distribution because different LLMs are aligned on MCSUs. The probability distributions representing MCSUs can be integrated, and they are naturally aligned semantically.

Given the large size of the vocabulary, for each LLM, we adopt a Top-k approach, retaining only the K MCSUs with the highest probabilities while ignoring the rest, which have lower probabilities. This significantly reduces computational complexity, especially considering that the vocabulary size is in the tens of thousands. We calculate the KL divergence between different probability distributions, if the distance between two distributions is small, we consider them closer and retain them, discarding those with larger distances. During each autoregressive step, this method dynamically selects the distributions most beneficial for the final integration. The specific process is as follows:

For each LLM i, define the probability distribution of the next MCSU as P_i.

For each P_i, retain only the top K items with the highest probabilities, setting the rest to zero:

$$P_i^{\text{top-k}}(x) = \begin{cases} P_i(x) & \text{if } x \in \text{top-k items} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Find the union U of all $P_i^{\text{top-k}}$. For each $P_i^{\text{top-k}}$, set the probabilities of items not in U to $1e^{-9}$ (the minimum value close to 0 is convenient for subsequent calculations):

$$P_i^*(x) = \begin{cases} P_i^{\text{top-k}}(x) & \text{if } x \in U \\ 1e^{-9} & \text{otherwise} \end{cases} \tag{4}$$

For any two probability distributions P_i^* and P_j^*, calculate the KL divergence between them:

$$D_{\text{KL}}(P_i^* \parallel P_j^*) = \sum_{x \in U} P_i^*(x) \log \frac{P_i^*(x)}{P_j^*(x)} \tag{5}$$

If $D_{\text{KL}}(P_i^* \parallel P_j^*) < \epsilon$ (where ϵ is a predefined threshold), retain these distributions. Note that if none of these distributions are close to each other, then all of them should be retained, as there is no reason to discard any of them.

We determine ϵ by computing the global mean through statistical methods, and set its value to 0.1 in this paper (the detailed procedure can be found in Appendix A).

Finally, we usually average the retained distributions and select the MCSU with the largest probability for subsequent iterations.

4 Experiments

4.1 Experiment Setup

Tasks ans datasets. We evaluate the performance of our proposed method on the following benchmarks: **Arithmetic Reasoning**. For this task, we selecte four datasets of different difficulty, including SVAMP (Patel et al., 2021), GSM8K (Cobbe et al., 2021), AddSub (Hosseini et al., 2014) and AQuA (Ling et al., 2017). **Commonsense Reasoning**. We select four datasets to evaluate the performance of the prpoposed method, including CommonseQA (Talmor et al., 2018), StrategyQA (Geva et al., 2021), OpenBookQA (Mihaylov et al., 2018) and ARC-c (Clark et al., 2018). **Symbolic Reasoning**. We select four datasets from BigBench (Srivastava et al., 2022) for testing, including Date Understanding, Penguin, Colored Objects and Logical Deduction.

Baselines. We compare the proposed method with three sets of widely used baselines: (1) source LLMs, including Qwen-2 7B (Yang et al., 2024), Llama-3 8B (Dubey et al., 2024) and GLM-4 9B (GLM et al., 2024); (2) Output-level emsembling: majority voting, a widely used collaboration

Methods / Datasets		SVAMP	GSM8K	AddSub	AQuA	Avg.
Single LLM	Qwen-2-7B	90.0	82.3	90.8	**65.5**	82.1
	Llama-3-8B	85.2	79.6	86.5	54.2	76.3
	GLM-4-9B	88.6	79.6	87.5	58.1	78.4
Emsembling	Majority Voting	90.8	79.9	91.0	65.2	81.6
	LLM-Blender	90.2	81.3	91.0	65.0	81.9
	GAC	89.8	81.9	91.0	63.9	81.7
	DEEPEN	89.9	82.1	91.3	64.9	82.1
Ours	DDS	**91.6**	**85.1**	**91.4**	**65.5**	**83.4**

Table 1: Comparison of accuracy on four mathematical reasoning datasets using DDS and strong baselines. The best results are highlighted in **bold**. All results are expressed as a percentage of accuracy, with the $\%$ symbol omitted.

Methods / Datasets		CSQA	StrategyQA	OpenBookQA	ARC-c	Avg.
Single LLM	Qwen-2-7B	71.9	73.2	81.0	81.0	76.7
	Llama-3-8B	67.9	70.2	75.3	76.3	72.4
	GLM-4-9B	67.3	71.8	79.5	79.4	74.5
Emsembling	Majority Voting	72.3	72.0	80.5	83.2	77.0
	LLM-Blender	72.0	72.8	80.1	83.0	76.9
	GAC	73.1	73.0	79.9	83.2	77.3
	DEEPEN	72.5	74.1	79.5	**84.1**	77.6
Ours	DDS	**76.0**	**75.5**	**83.5**	84.0	**79.8**

Table 2: Comparison of accuracy on four commonsense reasoning datasets using DDS and strong baselines.

method, the answers of the three models are subjected to majority voting to select the most consistent answer. LLM-Blender (Jiang et al., 2023), sorts the answers given by multiple models and then selects the one with the highest ranking. (3) Token-level emsembling: DEEPEN (Huang et al., 2024) and GAC (Yu et al., 2024), mainly achieve multi-model collaboration by aligning vocabulary.

Implementation details. We use the CoT (Wei et al., 2022) method and greedy decoding strategy to obtain the final results. The top 5 MCSUs are sampled in the Top-k sampling algorithm. We use regular expressions to extract the answers from LLM's answers to calculate the accuracy. The whole method is training-free, and most experiments are completed on one Nvidia H800 GPU.

4.2 Main Results

Mathematical Reasoning. The quantitative results on four datasets are shown in Table 1. It can be seen that the proposed DDS achieves significant performance improvements in most cases. Comparing the three single models, Qwen performs the best in mathematics, while Llama performs the worst. Furthermore, naive majority voting does not guarantee performance improvement over the best-performing single model. Finally, the other three collaborative methods also do not show clear superiority over the Qwen model in this benchmark.

Commomsense Reasoning. The quantitative results on four datasets are shown in Table 2. It can be clearly seen that the proposed DDS achieves significant performance improvements. Comparing the three single models, GLM performs the best in this task, while Llama still performs the worst. In addition, majority voting brings certain performance improvements, which are generally better than the best performance of a single model. Finally, all other three collaborative methods are improved compared to the single model.

Symbolic Reasoning. The quantitative results on four datasets are shown in Table 3. The proposed DDS still achieves the best performance in most cases. Comparing the three single models, GLM performs the best in this task, while Llama still performs the worst. Moreover, it can be seen that majority voting brings certain performance improvements, which are generally better than the best performance of a single model. Finally, all other three collaborative methods are improved compared to the single model.

Analysis: From the above experiments, it can be seen that compared with the performance of single models and other baseline methods, the proposed DDS performs relatively well and has improved. Moreover, we can see that although majority voting

Methods / Datasets		Date	Penguin	Colored Objects	Logical Deduction	Avg.
Single LLM	Qwen-2-7B	66.5	79.2	73.2	77.3	74.1
	Llama-3-8B	**69.6**	76.1	66.4	70.1	70.6
	GLM-4-9B	58.2	81.3	76.9	82.0	74.6
Emsembling	Majority Voting	65.1	82.2	79.4	81.2	77.0
	LLM-Blender	65.5	82.5	79.3	81.4	77.2
	GAC	65.0	**83.1**	79.2	81.3	77.1
	DEEPEN	64.5	82.9	**79.4**	80.2	76.8
Ours	DDS	68.8	83.0	**79.4**	**83.0**	**78.5**

Table 3: Comparison of accuracy on four symbolic reasoning datasets using DDS and strong baselines.

Datasets	Qwen	Llama	GLM	Majority Voting	DEEPEN	DDS
HumanEval	79.9	62.2	71.8	73.2	79.7	**81.1**
C-Eval	77.2	45.9	75.6	70.1	74.2	**78.3**

Table 4: Comparison of proposed DDS with other baseline methods on the HumanEval and C-Eval datasets.

Model/Metric		ROUGE	BLEU	BLEURT
Single LLM	Qwen-2 7B	0.548	0.532	0.659
	Llama-3 8B	0.555	0.517	0.634
	GLM-4 9B	0.587	0.536	0.660
Ours	DDS	**0.601**	**0.549**	**0.663**

Table 5: Performance of DDS in reliability evaluation. We report three metrics on the TruthfulQA benchmark, where larger values indicate more truthful results.

is a very simple way of collaboration, it can still bring certain performance improvements compared to single models most of the time, which is also consistent with the experience of human collective wisdom. Other collaboration methods are relatively complex but do not bring further significant performance improvements.

4.3 Evaluation on Cross-Task and Cross-Lingual Scenarios

we have conducted additional evaluations in two key domains: code generation and Chinese knowledge quiz tasks. Specifically, we tested on the HumanEval (Chen et al., 2021) and C-Eval (Huang et al., 2023) datasets, with the detailed results presented in the table 4.

From the experimental results, it can be seen that DDS performs well on these two tasks. This further demonstrates the potential and robustness of our proposed DDS method, which shows promise for application in broader domains to enhance LLMs' reasoning capabilities. Moreover, the good performance on the Chinese dataset C-Eval also indicates that proposed MCSU is applicable to other languages, not just English.

Model	SVAMP	CSQA	Penguin
Qwen	90.0	71.9	79.2
Llama	85.2	67.9	76.1
GLM	88.6	67.3	81.5
Qwen+Llama	90.2	73.7	81.3
Qwen+GLM	91.0	74.1	81.4
Llama+GLM	87.3	71.7	81.7
Qwen+Llama+GLM	90.8	74.9	81.0
DDS (Qwen+Llama+GLM)	**91.6**	**76.0**	**83.0**

Table 6: Ablation on the number of used LLMs. The multi-model collaboration represented by the "+" sign in the second part does not use the DSS strategy, but directly performs weighted averaging.

Dataset / ϵ	0.05	0.08	0.1	0.15	0.2	0.3	0.4	0.5
GSM8K	83.5	85.2	85.1	85.0	84.5	84.2	82.9	82.7
CSQA	75.1	75.5	76.0	75.7	75.8	75.5	75.3	74.9

Table 7: Ablation experiment of threshold ϵ. We present the performance of DDS across varying ϵ on the GSM8K and CSQA datasets.

4.4 Reliability Evaluation

The validity of LLM outputs is crucial, and their reliability is equally indispensable. The main experiments in Section 4.2 have already demonstrated that our proposed DDS can significantly enhance the reasoning performance of language models. Therefore, we are also interested in exploring whether DDS can improve the reliability of language models. For this purpose, we select TruthfulQA (Lin et al., 2021), a classic and widely used benchmark for evaluating the truthfulness of language model outputs, which can reflect the reliability of LLM outputs to some extent.

The results are shown in Table 5. The three metrics in the table indicate higher truthfulness of the model outputs when they are larger. It can be clearly seen that the metrics for the three single models are relatively close, while our proposed DDS achieves significantly higher metrics, indicating that DDS enhances the reliability of language model outputs to some extent.

	GSM8K	CSQA
Question	*The book is printed on paper that, when stacked, is 100 pages to the inch. Each paper is printed on both sides, with one page of the book printed on each side. How many pages are in the book, if it is 1.5 inches thick?*	*Kramer wrote a self-referential book. What might that book be about? (A) counter (B) coffee table (C) school room (D) backpack (E) bedside table*
Qwen	To find out how many pages are in the book, we need to understand that each inch of the book's thickness represents 100 pages (since the paper is 100 pages per inch). Given that the book is 1.5 inches thick, we can calculate the total number of pages as follows: Total pages = Thickness of the book in inches × Pages per inch Total pages = 1.5 × 100 = 150 Therefore, the book has **150 pages**.	Analysis: A self-referential book is one that refers to itself, often discussing its own structure, content, or writing process. Given this context, the book would likely delve into themes related to literature, writing, or the act of creating a book. Answer: (E) bedside table
Llama	The book is 1.5 inches thick. Each inch has 100 pages. So 1.5 inches will have 1.5 x 100 = 150 pages. The answer is 150.	Analysis: The question is asking what Kramer might write a self-referential book about. A self-referential book is a book that refers to itself, often in a humorous or ironic way. Kramer is a character from the TV show "Seinfeld", known for his eccentricities and witty remarks. Correct answer: (A) The Kramer
GLM	If the paper is 100 pages to the inch and each paper is printed on both sides, then each inch of thickness corresponds to 50 pages (since one side of the paper is one page). If the book is 1.5 inches thick, then the number of pages in the book would be: 1.5 inches * 50 pages per inch = 75 pages. So, the book has 75 pages.	Among the options, "bedside table" (E) is the closest to a book, as it is a piece of furniture that could be associated with a book (books are often placed on bedside tables). Given the limitations of the options, the most appropriate answer would be: (E) bedside table
DDS	If the book is 1.5 inches thick and each paper is 1/100 of an inch thick, then the number of papers in the book is 1.5 inches * 100 papers/inch = 150 papers. Since each paper has two sides, and each side is a page, the total number of pages in the book is 150 papers * 2 sides/paper = 300 pages.	Coffee table could be a possibility, as a coffee table is a common piece of furniture that Kramer might have written about in a self-referential manner. Answer: (B) coffee table

Table 8: Examples where DDS is able to answer correctly even though all three models answer incorrectly. Incorrect answers are shown in red and correct answers are shown in blue. Since some answers are too long, we have made some adjustments to facilitate presentation, but the core content remains unchanged.

4.5 Ablation Study

Ablation on the number of used LLMs. We show the results of using different numbers of LLMs in Table 6. It can be seen that as the number of LLMs increases from one to three, the overall performance gradually improves. The magnitude of this change is different in different tasks. We also notice that in some cases, the effect of using two models is better than that of using three. We think this is reasonable. When human groups collaborate, it is not always the case that more participants lead to better outcomes, the ability of the participants is a key factor, which is also the motivation for us to propose DDS. We hope to improve the quality of the results by selecting participants with better abilities. But in general, using more models will expand the knowledge boundary and thus improve performance.

It can be clearly seen that DDS performs best on the three datasets, better than single models and those directly integrated methods, which also proves the effectiveness of the strategy and achieves our expected goal. In addition, we also conducted experiments using more LLMs, which can be found in the Appendix B.

Ablation on the KL Divergence threshold ϵ. As shown in the Table 7, the optimal results were achieved when ϵ approached 0.1. DDS is highly sensitive to the threshold ϵ. Both excessively large or small values of ϵ can render the DDS strategy ineffective. A larger threshold weakens the filtering effectiveness, allowing harmful distributions to influence decisions, and a smaller threshold reduces the number of qualified distributions, forcing the retention of all distributions (equivalent to DDS being inactive).

4.6 Examples of Emergent Capabilities

The goal of DDS is to combine the capabilities of multiple LLMs to break through the performance bottlenecks of a single LLM. Traditional multi-model collaboration methods rely on most models providing the same correct answer for majority voting, or rely on one model providing a highly confident answer to convince the others to reach a consensus. However, we are surprised to find that our DDS can provide the correct answer even when each individual model answers incorrectly, demonstrating a certain level of emergent ability, akin to the saying "two heads are better than one".

Typical examples are shown in Table 8. It can be clearly seen that for the same math problem, the three LLMs give wrong answers of 150, 150 and 75 respectively, ignoring the fact that a piece of paper has two sides or confusing the logical relationship, while DDS gives the correct answer of 300; for another question, the three single LLMs fail to figure out the meaning of “a self-refine book” and thus can not give a correct answer, but DDS understands and gives the correct answer. We speculate that the token-level collision of ideas in DDS has brought some different inspirations, and may sometimes achieve the effect of $1+1$ being greater than 2. More examples and discussion can be seen in Appendix E.

We also present and analyze some failure cases, which can be found in Appendix F.

5 Conclusion

We propose DDS, a distribution distance-based token-level multi-model collaborative dynamic integration strategy. By selecting the most suitable models for integration at each step of autoregression, DDS effectively enhances the reasoning capabilities of language models. Additionally, we introduce the concept of the “minimum complete semantic unit”, which provides a simple and effective solution to the vocabulary misalignment issue between different LLMs, laying the groundwork for effective collaboration among multiple models. We hope that our approach can bring some inspiration to the community and further promote the use of some low-cost ways to enhance the reasoning ability of LLM.

Limitations

Since the forward process of multiple models needs to be calculated during integration, the reasoning time and amount of calculation will increase compared to single model reasoning. How to further improve efficiency is an issue that needs attention. Moreover, compared to answer-level ensemble methods that can directly invoke multiple APIs to perform asynchronous inference, our method requires loading multiple models locally to obtain the probability distributions of their intermediate outputs, which imposes certain hardware requirements and limits its deployment on resource-constrained devices. Notably, we also must pay attention to whether this multi-model collaboration approach might bypass some of the restrictions of single-model generated answers, leading to the production of unethical or harmful content. This is an area that requires further research in our future work.

Ethical Statement

This paper adheres to the ACL Code of Ethics. Firstly, we ensure thatthe datasets used do not contain sensitive personalinformation and pose no harm to society. Secondlyany pre-trained models employed have been dulylicensed.Furthermore, our code will be releasedunder a suitable license. Lastly, the proposed multi-model collaborative approach aims to enhance the reliability and performance of LLMs, contributing positively to the robustness and integrity of natural language processing applications, thereby supporting a safer and more trustworthy computational environment for users worldwide.

Acknowledgement

This work is sponsored by CCF-Tencent Rhino-Bird Open Research Fund, and supported by Guangdong Basic and Applied Basic Research Foundation (Grant No. 2023A1515140037), Guangdong Provincial Key Laboratory (Grant 2023B1212060076), Guangdong Key Laboratory of Information Security Technology, Sun Yat-sen University, Guangdong Research Team for Communication and Sensing Integrated with Intelligent Computing (Project No. 2024KCXTD047). The computational resources are supported by Song-Shan Lake HPC Center (SSL-HPC) in Great Bay University.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

01. AI, :, Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Tao Yu, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. 2024. Yi: Open foundation models by 01.ai. *Preprint*, arXiv:2403.04652.

Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, Xiaoyi Dong, Haodong Duan, Qi Fan, Zhaoye Fei, Yang Gao, Jiaye Ge, Chenya Gu, Yuzhe Gu, Tao Gui, Aijia Guo, Qipeng Guo, Conghui He, Yingfan Hu, Ting Huang, Tao Jiang, Penglong Jiao, Zhenjiang Jin, Zhikai Lei, Jiaxing Li, Jingwen Li, Linyang Li, Shuaibin Li, Wei Li, Yining Li, Hongwei Liu, Jiangning Liu, Jiawei Hong, Kaiwen Liu, Kuikun Liu, Xiaoran Liu, Chengqi Lv, Haijun Lv, Kai Lv, Li Ma, Runyuan Ma, Zerun Ma, Wenchang Ning, Linke Ouyang, Jiantao Qiu, Yuan Qu, Fukai Shang, Yunfan Shao, Demin Song, Zifan Song, Zhihao Sui, Peng Sun, Yu Sun, Huanze Tang, Bin Wang, Guoteng Wang, Jiaqi Wang, Jiayu Wang, Rui Wang, Yudong Wang, Ziyi Wang, Xingjian Wei, Qizhen Weng, Fan Wu, Yingtong Xiong, and et al. 2024. Internlm2 technical report. *CoRR*, abs/2403.17297.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde 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, Josh 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.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

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

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

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833*.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2022. Complexity-based prompting for multi-step reasoning. In *The Eleventh International Conference on Learning Representations*.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *Preprint*, arXiv:2406.12793.

Geoffrey Hinton. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Mohammad Javad Hosseini, Hannaneh Hajishirzi, Oren Etzioni, and Nate Kushman. 2014. Learning to solve arithmetic word problems with verb categorization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 523–533.

Yichong Huang, Xiaocheng Feng, Baohang Li, Yang Xiang, Hui Wang, Bing Qin, and Ting Liu. 2024. Enabling ensemble learning for heterogeneous large language models with deep parallel collaboration. *arXiv preprint arXiv:2404.12715*.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, Yao Fu, Maosong Sun, and Junxian He. 2023. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. In *Advances in Neural Information Processing Systems*.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. *arXiv preprint arXiv:2306.02561*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Akbir Khan, John Hughes, Dan Valentine, Laura Ruis, Kshitij Sachan, Ansh Radhakrishnan, Edward Grefenstette, Samuel R Bowman, Tim Rocktäschel, and Ethan Perez. 2024. Debating with more persuasive llms leads to more truthful answers. *arXiv preprint arXiv:2402.06782*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Zhaopeng Tu, and Shuming Shi. 2023. Encouraging divergent thinking in large language models through multi-agent debate. *arXiv preprint arXiv:2305.19118*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. *arXiv preprint arXiv:1705.04146*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2024. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are nlp models really able to solve simple math word problems? *arXiv preprint arXiv:2103.07191*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Mohaimenul Azam Khan Raiaan, Md Saddam Hossain Mukta, Kaniz Fatema, Nur Mohammad Fahad, Sadman Sakib, Most Marufatul Jannat Mim, Jubaer Ahmad, Mohammed Eunus Ali, and Sami Azam. 2024. A review on large language models: Architectures, applications, taxonomies, open issues and challenges. *IEEE Access*.

Shannon Zejiang Shen, Hunter Lang, Bailin Wang, Yoon Kim, and David Sontag. 2024. Learning to decode collaboratively with multiple language models. *arXiv preprint arXiv:2403.03870*.

Tal Shnitzer, Anthony Ou, Mírian Silva, Kate Soule, Yuekai Sun, Justin Solomon, Neil Thompson, and Mikhail Yurochkin. 2023. Large language model routing with benchmark datasets. *arXiv preprint arXiv:2309.15789*.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

Qiushi Sun, Zhangyue Yin, Xiang Li, Zhiyong Wu, Xipeng Qiu, and Lingpeng Kong. 2023. Corex: Pushing the boundaries of complex reasoning through multi-model collaboration. *arXiv preprint arXiv:2310.00280*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2018. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Fanqi Wan, Xinting Huang, Deng Cai, Xiaojun Quan, Wei Bi, and Shuming Shi. 2024. Knowledge fusion of large language models. *arXiv preprint arXiv:2401.10491*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Yangyifan Xu, Jinliang Lu, and Jiajun Zhang. 2024. Bridging the gap between different vocabularies for llm ensemble. *arXiv preprint arXiv:2404.09492*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Yuxuan Yao, Han Wu, Mingyang Liu, Sichun Luo, Xiongwei Han, Jie Liu, Zhijiang Guo, and Linqi Song. 2024. Determine-then-ensemble: Necessity of top-k union for large language model ensembling. *arXiv preprint arXiv:2410.03777*.

Zhangyue Yin, Qiushi Sun, Cheng Chang, Qipeng Guo, Junqi Dai, Xuan-Jing Huang, and Xipeng Qiu. 2023. Exchange-of-thought: Enhancing large language model capabilities through cross-model communication. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15135–15153.

Yao-Ching Yu, Chun-Chih Kuo, Ziqi Ye, Yu-Cheng Chang, and Yueh-Se Li. 2024. Breaking the ceiling of the llm community by treating token generation as a classification for ensembling. *arXiv preprint arXiv:2406.12585*.

A Determination of the KL Divergence Threshold

The Kullback-Leibler (KL) divergence ranges from zero to infinity. In statistics:

1) A value of 0 indicates identical distributions;

2) Values between 0 and 1 typically suggest high similarity between distributions;

3) Values exceeding 1 generally indicate significant divergence.

We determined this threshold ϵ through data-driven statistical analysis by:

The reference KL divergence values were computed by measuring distributional discrepancies in the validation dataset, implementing the following procedure:

a) Collected large number of sample pairs (P, Q) and calculated their KL divergence.

b) Plotted the distribution of KL values (histogram / CDF).

c) Selected the mean value as threshold based on distribution characteristics.

This process yielded a value near 0.1. We set this as the default threshold in our experiments, and it demonstrated satisfactory performance.

B Scaling to More Models

As shown in Table 9, it can be observed that as the number of participants in the collaboration increases, the performance improves slightly, but the extent of improvement is not significant. We believe this is reasonable because Qwen-2-7B, Llama-3-8B, and GLM-4-9B are already the three best-performing models of similar parameter scale during the same period. As pointed out in paper (Yao et al., 2024), the capabilities of the models involved in the collaboration are also an important factor, and it does not depend solely on the quantity. This also demonstrates that the DDS strategy can be applied to LLMs with diverse architectures, highlighting its strong generalizability.

On one hand, we will conduct a deeper investigation into how DDS scales with an increasing number of models. On the other hand, we will explore new strategies to better integrate the capabilities of these models.

C Theoretical Analysis of why Token-level Model Emsembling Is Effective

The knowledge of a large language model (LLM) is primarily stored in its vast number of parameters, which encode a broad understanding of language and domain-specific knowledge. However, the external manifestation of this knowledge is the next token probability distribution generated at each step of autoregressive sampling. Formally, given a context $x_{<t}$, an LLM generates a probability distribution over the next possible tokens:

$$P_{\text{LLM}}(w_t \mid x_{<t}) = \text{softmax}(f_\theta(x_{<t})),$$

where f_θ represents the internal computation of the LLM parameterized by θ. This probability distribution reflects the LLM's internal knowledge and informs the quality of the generated answers. Higher-quality knowledge results in a higher probability assigned to tokens that contribute to better answers. The final output from the model is sampled from this next token distribution, and the sequence of sampled tokens constitutes the generated text.

The answer space derived from this autoregressive process is defined by the cumulative sampling across multiple steps. Let the space of possible answers be denoted as A, with each potential answer having an associated probability based on the product of next-token probabilities. Hence, the probability of an answer $a \in A$ being generated by the LLM is:

$$P_{\text{LLM}}(a \mid x) = \prod_{t=1}^{T} P_{\text{LLM}}(w_t \mid x_{<t}),$$

where T is the length of the generated answer. Importantly, the better the model's knowledge, the higher the probability assigned to higher-quality answers in this space. Theoretically, LLM can output any answer, but since some answers have extremely low probability of occurrence, we believe that LLM is not capable of making certain answers at this time. We believe that the answer

Number	DDS (3)	DDS (4)	DDS (5)
GSM8K	85.1	84.9	85.4
CSQA	76.0	76.3	76.2

Table 9: The relationship between the performance of DSS in GSM8K and CSQA datasets and the number of participating collaborative models. DDS (3) refers to the default three models Qwen-2-7B, Llama-3-8B, and GLM-4-9B, DDS (4) is the addition of Internlm-2-7B (Cai et al., 2024), and DDS (5) is the further addition of Yi-1.5-9B (AI et al., 2024).

space only contains answers with probability reaching a certain threshold.

Combining Knowledge from Multiple LLMs.

We propose to improve the overall answer quality by combining the knowledge of multiple LLMs, leveraging their individual next token distributions. Let $P^{(i)}(w_t \mid x_{<t})$ represent the next-token probability distribution generated by the i-th LLM. By combining these distributions, we create a new, enhanced distribution that incorporates the knowledge encoded in multiple models. Specifically, we aggregate the distributions as:

$$\begin{aligned} P_{\text{combined}}(w_t \mid x_{<t}) = \text{Aggregate}(&P^{(1)}(w_t \mid x_{<t}), \\ &P^{(2)}(w_t \mid x_{<t}), \\ &\ldots, \\ &P^{(K)}(w_t \mid x_{<t})), \end{aligned}$$

where K is the number of models and the aggregation function is designed to effectively combine the distributions. The resulting answer space from this combination is strictly larger and of higher quality than any individual model's answer space, as it benefits from the union of knowledge across models. In theory, any answer contained in the single model answer space can be sampled from this answer space.

For example, consider two LLMs, M_A and M_B, answering two different questions. In some cases, M_A may provide the correct answer, while in other cases, M_B might be more accurate. By merging their answer spaces, we can capture the correct answers from both models with higher probability, ensuring that:

$$P_{\text{combined}}(a^* \mid x) \geq \max\left(P^{(A)}(a^* \mid x), P^{(B)}(a^* \mid x)\right)$$

where a^* is the optimal answer. In this way, the combined space encompasses the high-quality answers from both models and assigns them higher probabilities than the individual models would on their own. This probability is from a general perspective, because the probability that the merged answer space contains the correct answer is definitely greater than the answer space of a single LLM.

Sampling from the Combined Answer Space.

In order to efficiently sample high-quality answers from this enhanced space, we designed the DSS ensemble method, which uses distribution similarity to filter out those distributions with large differences, thereby retaining distributions with higher similarity because they are more likely to cluster near the correct answer, which is also our basic assumption, and we have demonstrated its effectiveness through a large number of experiments.

In conclusion, by combining the next-token distributions of multiple LLMs and introducing sophisticated sampling strategies, we create a richer and more reliable answer space that enhances the probability of producing high-quality answers. The combined knowledge from multiple models overcomes the limitations of any single model and significantly improves the robustness and performance of generated responses.

D Vocabulary Statistics

We show the probability of the 5000 most commonly used English words being encoded as a single token in Table 10. It can be seen that for these three models, almost 90% of the words are encoded as a single token. This also shows that most commonly used tokens are MCSUs themselves, and our introduction of MCSU does not bring too much additional computation.

Model	Qwen	Llama	GLM
Ratio	88.9	92.1	91.4

Table 10: The probability of 5000 commonly used English words being encoded as a token by different LLMs.

E More Examples

We show more specific examples in Table 11, Table 12 and Table 13, corresponding to examples where DDS correctly answered questions when only one, two, and three single models answered correctly. Combined with Table 8, we can find that DDS can effectively aggregate the capabilities of multiple models. As long as one member can give the correct answer, DDS has the ability to give the correct answer, and can even give the correct answer when all member models give incorrect answers.

F Faillure Examples and Analysis

DDS can also make mistakes. As can be seen from Table 14, DDS can also give wrong answers when some member models can answer correctly. Moreover, it cannot always exhibit emergent capabilities when none of the member models answer correctly. This also shows that DDS still has room for improvement and its potential has not been fully explored. This is also the goal of our further research.

	GSM8K	CSQA
Question	*John runs 60 miles a week. He runs 3 days a week. He runs 3 hours the first day and half as much the other two days he runs. How fast does he run?*	*Aside from water and nourishment what does your dog need? (A) bone (B) charm (C) petted (D) lots of attention (E) walked*
Qwen	John runs 3 hours on the first day, and half as much on the other two days. This means he runs 3/2 = 1.5 hours on each of the other two days. So in total, he runs 3 + 1.5 + 1.5 = 6 hours per week. He runs 60 miles in 6 hours, so his speed is 60/6 = 10 miles per hour. The answer is: 10	Analysis: Dogs, like all animals, require various needs to maintain their health and happiness. While water and nourishment are essential, dogs also need physical activity, mental stimulation, and affection. Among the given options, "walked" (Option E) is directly related to the physical exercise that dogs need. Answer: (E) walked
Llama	John runs 60 miles a week. He runs 3 days a week. So he runs 60 / 3 = 20 miles a day. He runs 3 hours the first day, and half as much the other two days. So he runs 3 hours the first day, and 1.5 hours each of the other two days. 3 + 1.5 + 1.5 = 6 hours. So he runs 20 miles in 6 hours. 20 / 6 = 3.33 miles per hour. The answer is 3.33.	Analysis: Dogs need a variety of things to stay happy and healthy, aside from water and nourishment. One of the most important things they need is exercise, which is closely related to option (E) walked. Correct answer: (E) walked
GLM	The total distance he runs in a week is 60 miles, and the total time he runs is: 3 hours (first day) + 3 hours (second day) + 3 hours (third day) = 9 hours Now, we can calculate his average speed for the week: 60 miles / 9 hours = 6.67 miles per hour So, John's average speed for the week is approximately 6.67 miles per hour.	Analysis: To determine the correct answer, we need to consider the basic needs of a dog, which include physical, emotional, and social requirements. Answer: (D) lots of attention
DDS	John runs 60 miles a week. He runs 3 days a week. On the first day, he runs 3 hours. On the other two days, he runs half as much as the first day, which is 3/2 = 1.5 hours each day. So, the total hours he runs in a week is: 3 hours (first day) + 1.5 hours (second day) + 1.5 hours (third day) = 6 hours To find out how fast he runs, we divide the total miles he runs by the total hours he runs: 60 miles / 6 hours = 10 miles per hour John runs at a speed of 10 miles per hour.	Analysis: Dogs are social animals that require attention, exercise, and affection to maintain their physical and mental health. While bones can be a treat, they are not a necessity for a dog's overall well-being. Charm is not a requirement for a dog's needs. Petting can be a form of affection, but it is not a fundamental need. Walking is essential for a dog's physical health and exercise, which is a crucial aspect of their care. Answer: (D) lots of attention

Table 11: Examples where only one of the three single LLMs answers correctly. Incorrect answers are shown in red and correct answers are shown in blue. Since some answers are too long, we have made some adjustments to facilitate presentation, but the core content remains unchanged.

We have selected several failure cases and analyzed their internal autoregressive generation processes. We found that in most cases, the failures were due to significant distribution differences between models, especially during the critical stages of autoregression (e.g., key numbers or key options). In these instances, one model provided an MCSU with excessively high confidence for an incorrect answer, which ultimately affected the final result.

According to the article's description of DDS's dynamic selection process, if the similarity between multiple distributions during the autoregressive process is low, we retain all distributions before proceeding with ensemble integration. Therefore, failures mostly occur when there are substantial differences between model answers, and the incorrect answer happens to have the highest confidence.

The root cause, we believe, may stem from variations in response styles and reasoning processes across different models, which sometimes invalidate our assumption that "there is typically only one correct answer, whereas incorrect answers can be numerous and varied," leading to DDS failures. Nevertheless, it is undeniable that DDS remains effective in most cases, as evidenced by our experimental results.

Here's a concrete example to illustrate this scenario:

Prefix: "The final option is"

Model 1 generates option B (incorrect) with a confidence score of 0.8.

Model 2 generates option D (correct) with a confidence score of 0.7.

Model 3 produces an irrelevant adverb, "actually," with a confidence score of 0.75.

Since the three models' outputs diverge significantly (B, D, and an unrelated word "actually"), the system retains all three distributions and ultimately selects the highest-confidence answer—B (0.8), despite it being incorrect.

G Discussion

During operation, DDS requires loading all models simultaneously because each step of its autoregressive process necessitates the next MCSU distributions from every model. However, the computations across these models are mutually independent. This means the parameters and computational load required by the DDS strategy scale linearly with the number of participating models, effectively approximating the sum of all individual models' requirements, without introducing higher-order complexity. Model interactions occur exclusively during distribution similarity computation, which incurs negligible computational overhead. All other computations proceed independently across models.

DDS is a method that leverages multi-model collaboration to improve model performance from the inference side. It can break through the performance bottleneck of a single model without training and without consuming a large amount of resources. Compared to previous multi-model collaboration methods, DDS does not require manually designing prompts to enable collaboration among multiple models, nor does it involve the hassle of multiple rounds of interaction to reach a consensus. A single round of autoregression is sufficient to produce a complete and deterministic answer.

Furthermore, we believe that DDS can be used for high-quality annotation, as it can integrate the knowledge boundaries of multiple models to provide high-quality pseudo-labels for unlabeled data. The annotation quality is likely to be superior to that of a single model and could help address the current shortage of high-quality data.

	GSM8K	CSQA
Question	*Josh decides to try flipping a house. He buys a house for $80,000 and then puts in $50,000 in repairs. This increased the value of the house by 150%. How much profit did he make?*	*An underrated thing about computers is how they manage workflow, at one time it was a big deal when they could first do what? (A) share files (B) do arithmetic (C) turn on (D) cost money (E) multitask*
Qwen	Josh bought a house for $80,000 and spent $50,000 on repairs, so the total cost of the house is $80,000 + $50,000 = $130,000. The value of the house increased by 150% of its original value. The original value of the house was $80,000, so the increase in value is 150/100 * $80,000 = $120,000. This means that the new value of the house is $80,000 + $120,000 = $200,000. To find the profit, we subtract the total cost from the new value of the house: $200,000 - $130,000 = $70,000. The answer is $70000.	Analysis: The question is asking about an underrated aspect of computers that was once considered a significant achievement. Among the options, "multitasking" (option E) is often seen as an advanced feature that wasn't necessarily a big deal in the early days of computing. However, considering the context of the question, which seems to focus on early milestones, the most fitting answer would be "do arithmetic" (option B). In the early days of computers, their ability to perform basic arithmetic operations was indeed a significant achievement, as it demonstrated their computational capabilities beyond simple data processing or file management. Answer: (B) do arithmetic
Llama	Josh bought the house for 80,000. He put in 50,000 in repairs. So the total cost is 80,000 + 50,000 = 130,000. The house increased in value by 150%. So the new value is 130,000 * 2.5 = 325,000. The profit is the difference between the new value and the cost. 325,000 - 130,000 = 195,000. The answer is 195,000.	Analysis: The question is asking about an underrated thing about computers, which is their ability to manage workflow. The correct answer is the option that was a big deal when computers first started doing it. Multitasking was a significant advancement in computer technology, as it allowed computers to perform multiple tasks simultaneously, making them more efficient and productive. Correct answer: (E) multitask
GLM	To calculate the profit Josh made, we first need to determine the new value of the house after the repairs. The value of the house increased by 150%, which means the increase is 150% of the original value of the house. To find this, we calculate: 150% of $80,000 = 1.5 * $80,000 = $120,000 This $120,000 is the amount by which the value of the house increased. To find the new value of the house, we add this increase to the original value: New value of the house = Original value + Increase New value of the house = $80,000 + $120,000 New value of the house = $200,000 Now, to find the profit, we subtract the total cost (original purchase price + repair costs) from the new value of the house: Profit = New value of the house - (Original purchase price + Repair costs) Profit = $200,000 - ($80,000 + $50,000) Profit = $200,000 - $130,000 Profit = $70,000 Josh made a profit of $70,000.	Analysis: The question is asking about an "underrated thing about computers" that was a significant development when computers first had the capability to do it. Let's analyze each option: (A) share files: While sharing files is a crucial aspect of computer networking, it was not the first significant development when computers were first introduced. (B) do arithmetic: Computers have always been capable of doing arithmetic, and this was not a groundbreaking development when they were first introduced. (C) turn on: The ability to turn on a computer is a fundamental feature, but it is not something that would be considered an "underrated thing about computers" or a significant development when computers were first created. (D) cost money: The fact that computers cost money is a given, and it is not a feature that would be considered a significant development or an "underrated thing." (E) multitask: Multitasking refers to the ability of a computer to perform multiple tasks simultaneously. This was a significant development when computers first gained this capability, as it allowed for more efficient use of the system's resources and improved productivity. Based on the analysis, the correct answer is: (E) multitask
DDS	To calculate the profit Josh made, we first need to determine the new value of the house after the repairs. The value of the house increased by 150%, which means the new value is 150% of the original value plus the original value itself. The original value of the house is $80,000. The increase in value is 150% of $80,000, which is: 150% of $80,000 = 1.5 * $80,000 = $120,000 So, the new value of the house after the repairs is: $80,000 (original value) + $120,000 (increase) = $200,000 Josh spent $50,000 on repairs, so the total cost of the house after repairs is: $80,000 (original purchase price) + $50,000 (repairs) = $130,000 Now, to calculate the profit, we subtract the total cost from the new value of the house: $200,000 (new value) - $130,000 (total cost) = $70,000 Josh made a profit of $70,000.	Analysis: The question is asking about an underrated aspect of computers and a significant milestone in their development. The correct answer should be something that was a major achievement in the early days of computing. (A) Share files: While file sharing is an important feature, it's not a groundbreaking achievement that was a big deal in the early days of computers. (B) Do arithmetic: This is a fundamental capability of computers, but it's not something that was particularly groundbreaking or a big deal in the early days of computing. (C) Turn on: This is a basic function of any electronic device, not a significant achievement in the development of computers. (D) Cost money: This is not a feature or capability of computers, but rather a factor that affects their accessibility. (E) Multitask: This is a significant achievement in the development of computers, as it allows them to perform multiple tasks simultaneously, which was not possible in the early days of computing. Answer: (E) Multitask

Table 12: Examples where two of the three single LLMs answer correctly. Incorrect answers are shown in red and correct answers are shown in blue. Since some answers are too long, we have made some adjustments to facilitate presentation, but the core content remains unchanged.

	GSM8K	CSQA
Question	*Janet's ducks lay 16 eggs per day. She eats three for breakfast every morning and bakes muffins for her friends every day with four. She sells the remainder at the farmers' market daily for $2 per fresh duck egg. How much in dollars does she make every day at the farmers' market?*	*A revolving door is convenient for two direction travel, but it also serves as a security measure at a what? (A) bank (B) library (C) department store (D) mall (E) new york*
Qwen	Janet's ducks lay 16 eggs per day. She eats 3 eggs for breakfast, so she has 16 - 3 = 13 eggs left. She uses 4 eggs to bake muffins, so she has 13 - 4 = 9 eggs left to sell. She sells each egg for $2, so she makes 9 * $2 = $18 from selling eggs every day. The answer is: $18	Analysis: Revolving doors are commonly used in public places to control access and maintain a certain level of security. They are particularly useful in areas where there is a need to control who enters and exits, such as banks, where sensitive transactions occur, and large buildings with high foot traffic. Answer: (A) bank
Llama	Janet lays 16 eggs per day. She eats 3 and bakes 4, so she has 16 - 3 - 4 = 9 eggs left. She sells these 9 eggs for 2 dollars each. 9 x 2 = 18. The answer is 18.	Analysis:A revolving door is a type of door that allows people to enter and exit through the same door, but in opposite directions. It is often used in high-security areas to control the flow of people and prevent unauthorized entry. In this context, the correct answer would be an option that is typically associated with high-security measures. Answer: (A) bank
GLM	Janet's ducks lay 16 eggs per day. She eats 3 eggs for breakfast, so she has 16 - 3 = 13 eggs left. She bakes muffins for her friends using 4 eggs, so she has 13 - 4 = 9 eggs remaining. She sells these 9 eggs at the farmers' market for $2 per egg, so she makes 9 * $2 = $18 every day at the farmers' market.	Analysis: A revolving door is designed to allow people to enter and exit in both directions smoothly, which can be particularly convenient in high-traffic areas. However, it also serves as a security measure by controlling the flow of people in and out of a building. This can be particularly important in places where security is a priority, such as financial institutions. Options Analysis: (A) Bank - Banks are known for their security measures, and controlling access is crucial. A revolving door could serve as a security measure here. (B) Library - While libraries may have security concerns, they are not typically as high as those in financial institutions. (C) Department store - Security is important, but it is not as critical as in a bank. (D) Mall - Similar to a department store, while security is important, it is not as critical as in a bank. (E) New York - This is not a specific location where a revolving door would serve as a security measure. Based on the analysis, the correct answer is: (A) bank
DDS	Janet's ducks lay 16 eggs per day. She eats 3 eggs for breakfast and uses 4 eggs for baking muffins. So, she uses a total of 3 + 4 = 7 eggs for personal use. This leaves her with 16 - 7 = 9 eggs to sell at the farmers' market. She sells each egg for $2, so she makes 9 * $2 = $18 every day at the farmers' market.	Analysis: A revolving door is a type of door that allows people to enter and exit in both directions, but it also serves as a security measure. This is because it can help to control the flow of people entering and exiting a building, making it more difficult for unauthorized individuals to enter. This is particularly useful in places where security is a concern, such as financial institutions. Answer: (A) bank

Table 13: Examples where all of the three single LLMs answer correctly. Incorrect answers are shown in red and correct answers are shown in blue. Since some answers are too long, we have made some adjustments to facilitate presentation, but the core content remains unchanged.

	GSM8K	CSQA
Question	*Elliott is trying to walk 10,000 steps a day. He finished half of his steps on his walks to and from school and did another 1,000 steps going for a short walk with his friend. He also went for a short jog around the block and realized that after he had finished his jog, he only had 2,000 steps left to take. How many steps did Elliott take during his jog?*	*What would vinyl be an odd thing to replace? (A) pants (B) record albums (C) record store (D) cheese (E) wallpaper*
Qwen	Elliott finished half of his steps on his walks to and from school, which means he took 10,000/2 = 5,000 steps on those walks. He also took 1,000 steps going for a short walk with his friend. So far, he has taken 5,000 + 1,000 = 6,000 steps. After his jog, he still had 2,000 steps left to take. This means that the number of steps he took during his jog is the total number of steps he needs to take (10,000) minus the steps he has already taken (6,000) and the steps he still needs to take (2,000). So, the number of steps he took during his jog is 10,000 - 6,000 - 2,000 = 2,000. The answer is: 2000.	Analysis: Vinyl is a type of material that was commonly used in the production of record albums and, less frequently, for certain types of wallpaper. It is not typically associated with clothing or food items like cheese. Therefore, the most unlikely option for replacing something with vinyl would be (D) cheese. Answer: (D) cheese
Llama	Elliott started with 10,000 steps. He did half of that, which is 5,000 steps. Then he did another 1,000 steps. So he did 5,000 + 1,000 = 6,000 steps. He had 2,000 steps left. So he must have done 6,000 + 2,000 = 8,000 steps. The answer is 8,000.	Analysis: The word "vinyl" is often associated with records, as vinyl records are a type of phonograph record made of polyvinyl chloride (PVC). Therefore, it would be an odd thing to replace something that is not related to records. Correct answer: (B) record albums
GLM	Elliott has a goal of 10,000 steps a day. He has already completed half of his steps on his walks to and from school, which is 10,000 / 2 = 5,000 steps. He also did another 1,000 steps going for a short walk with his friend, so in total, he has completed 5,000 + 1,000 = 6,000 steps. After his jog, he has 2,000 steps left to take, which means he has taken 10,000 - 2,000 = 8,000 steps so far. To find out how many steps he took during his jog, we subtract the steps he took before the jog from the total steps he has taken so far: 8,000 - 6,000 = 2,000 steps. Therefore, Elliott took 2,000 steps during his jog.	Analysis: The term "vinyl" is commonly associated with a few specific items. It is a type of plastic material that has been used in various applications over the years. Here's a brief analysis of each option: (A) pants - While "vinyl pants" are not common, they do exist, and vinyl can be used in the production of certain types of clothing. So, it's not an odd thing to replace pants with vinyl. (B) record albums - Vinyl is the material traditionally used for record albums, so replacing them with vinyl would not be odd. (C) record store - A record store is a place where vinyl records are sold, so replacing a record store with vinyl would be odd because vinyl is a material, not a place. (D) cheese - Cheese is a food product, and vinyl is a material. Replacing cheese with vinyl would be odd because they are completely different types of items. (E) wallpaper - Vinyl can be used to make wallpaper, so replacing wallpaper with vinyl is not an odd thing. Based on the analysis, the odd thing to replace would be: (D) cheese
DDS	Elliott has a goal of 10,000 steps a day. He has already completed half of his steps on his walks to and from school, which is 5,000 steps. He also did another 1,000 steps going for a short walk with his friend. So, he has completed 5,000 + 1,000 = 6,000 steps so far. He has 2,000 steps left to reach his goal of 10,000 steps. This means that the steps he took during his jog must be the difference between the steps he has completed and the steps he has left to complete his goal. So, the steps he took during his jog are 10,000 - 6,000 = 4,000 steps.	Analysis: Vinyl is a type of material that is often used to make records, which are used to play music. It is also used in some other applications, such as in the production of certain types of clothing and accessories. However, it is not typically used to make pants, cheese, or wallpaper. Therefore, the odd thing to replace with vinyl would be something that is not typically made from vinyl. Answer: (A) pants

Table 14: Examples where DDS answers incorrectly. Incorrect answers are shown in red and correct answers are shown in blue. Since some answers are too long, we have made some adjustments to facilitate presentation, but the core content remains unchanged.