Enabling Ensemble Learning for Heterogeneous Large Language Models with Deep Parallel Collaboration

Anonymous ACL submission

Abstract

Large language models (LLMs) have shown complementary strengths in various tasks and instances, motivating the research of ensembling LLMs to push the frontier leveraging the wisdom of the crowd. Existing work achieves this objective via training the extra reward model or fusion model to select or fuse all candidate answers. However, these methods pose a great challenge to the generalizability of the trained models. Besides, existing methods use the textual responses as communication media, ignoring the rich information in the inner representations of neural networks. Therefore, we propose a training-free ensemble framework **DEEPEN**, averaging the probability distributions outputted by different LLMs. A key challenge in this paradigm is the vocabulary discrepancy between heterogeneous LLMs, which hinders the operation of probability distribution averaging. To address this challenge, DEEPEN maps the probability distribution of each model from the probability space to a universe *relative space* based on the relative representation theory, and performs aggregation. Then, the result of aggregation is mapped back to the probability space of one LLM via a search-based inverse transformation to determine the generated token. We conduct experiments on the ensemble of various LLMs of 6B to 70B. Experimental results show that DEEPEN achieves consistent improvements across six popular benchmarks involving subject examination, reasoning and knowledge-QA, proving the effectiveness of our approach[1].

1 Introduction

With the scaling of model capacities and data volumes, generative large language models (LLMs) have shown impressive language understanding and generation abilities, shedding light for artificial general intelligence (Zhao et al., 2023; OpenAI, 2023; Jiang et al., 2023a; Touvron et al., 2023). Due to diversities of data sources, model architectures, and training recipes, LLMs have different strengths and weaknesses in various tasks and cases. Therefore, recent research has explored the ensemble of LLMs to exploit the complementary potential (Jiang et al., 2023b; Lu et al., 2023).

Existing work employs a two-stage process for LLM ensemble, where each LLM first generates candidate answers independently, and then the final answer is derived by either selecting or fusing these candidates, corresponding to selection-based and fusion-based ensemble approaches. Selection-based ensemble leverages an extra trained reward model to score all candidates and select the best one (Jiang et al., 2023b; Wang et al., 2024; Shnitzer et al., 2024; Lu et al., 2023), which fails to achieve collaboration between LLMs. Fusion-based ensemble trains a fusion model to generate a better answer according to all candidates (Jiang et al., 2023b) or adopt the LLM to perform fusion (Du et al., 2023). However, these methods either pose a great challenge to the generalizability of the reward model and the fusion model or incur a non-trivial computation cost and inference latency for communication. Besides, existing research enables collaboration via conveying the textual responses between LLMs while ignoring the rich information (*e.g.,* confidence) in the inner representations.

One ideal solution to this dilemma is prediction fusion (Bisk et al., 2020; Garmash and Monz, 2016). For LLM ensemble, prediction fusion works at each decoding step, averaging the probability distributions from different LLMs to determine the generated token. This paradigm is not only training-free, making it more general, but also leverages the inner representations (*i.e.,* probability distributions) as communication media between LLMs. An essential assumption for prediction fusion is that the ensemble models share a common vocabulary. However, this assumption often does not hold due

[1] Our code will be released soon.
✉ means corresponding author.

to the heterogeneity of LLMs caused by different vocabulary sizes and tokenizer models.

In this work, we tackle this key challenge via drawing upon the cross-model invariance of relative representation, which represents each token using the embedding similarities of this token to a set of anchor tokens (Moschella et al., 2023). Specifically, we propose an ensemble framework **DeepEn** (**Dee**p **P**arallel **En**semble), which enables prediction fusion for heterogeneous LLMs. DeepEn transforms the probability distribution from the heterogeneous probability space to a homogeneous relative space, using a matrix formed by the relative representation of all tokens. We refer to this matrix as a relative representation matrix. Next, DeepEn aggregates the relative representations of all probability distributions in the relative space, to obtain the representation containing the comprehensive decision on the tokens to generate. Finally, the result of aggregation are transformed back to the probability space of one of the ensemble LLMs using a search-based inverse transformation to determine the generated token.

Although our approach can apply on the ensemble of any number of LLMs, we mainly prove its effectiveness on the 2-model and 3-model ensemble. Our experiments involve various LLMs of 6B to 70B, and consider dense models and the MoE model Mixtral. Experimental results demonstrate that DeepEn achieves consistent improvement on six widely-used benchmarks.

2 Theoretical Analysis

In this section, we first introduce relative representation and then illustrate the theoretical support for our methodology.

2.1 Relative Representation

Previous study discovers that despite the misalignment between latent spaces of different neural networks, the embedding similarity between samples do not change across models (Moschella et al., 2023; He and Ozay, 2022; Park et al., 2019). Specifically, Moschella et al. (2023) propose *relative representation*, which represents each sample $x^{(i)}$ by the embedding similarities to a set of anchor samples $\mathbb{A}$. Formally:

$$\mathbf{r}_{x^{(i)}} = (cos(e_{x^{(i)}}, e_{a^{(1)}}), ..., cos(e_{x^{(i)}}, e_{a^{(|\mathbb{A}|)}})), \tag{1}$$

where $e_{(*)}$ denotes the embedding of samples, also is *absolute representation*. Note that $x^{(i)}$ and anchors $\mathbb{A}$ are identically distributed.

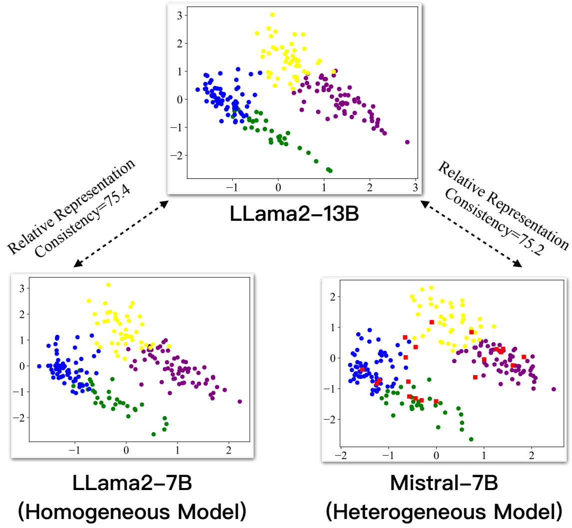

Figure 1: Visualizations for relative representations between homogeneous models and between and heterogeneous models. PCA is applied only for visualization. Red block indicate the representation of tokens that only appear in the vocabulary of Mistral. Relative representation consistency is obtained by calculating the cosine similarity between the relative representations of the same token in different models.

It is empirically discovered that relative representation possesses **cross-model invariance**, *i.e.,* the relative representation of the same sample keeps invariant across different models, which lays the theoretical foundation for our work to ensemble heterogeneous LLMs.

2.2 Theoretical Support for DeepEn

Averaging probability distribution (*i.e.,* prediction fusion) is an effective method to achieve the coordination between various neural networks. The underlying mechanism is to interpolate the output semantics (*i.e.,* the semantic that model intends to express) in the probability space. However, vocabulary discrepancy isolates these target semantics in semantic spaces with different basis vectors, hindering the interpolation. To tackle this challenge, we aim to enable the cross-model alignment for target semantics, *i.e.,* find a transformation to transform the target semantic into a universal space. To this effect, we propose to represent the target semantic with the convex combination of relative representations of all tokens where the weight is the probability assigned to the token.

Definition of target semantics in relative space. Formally, given the absolute representation of the

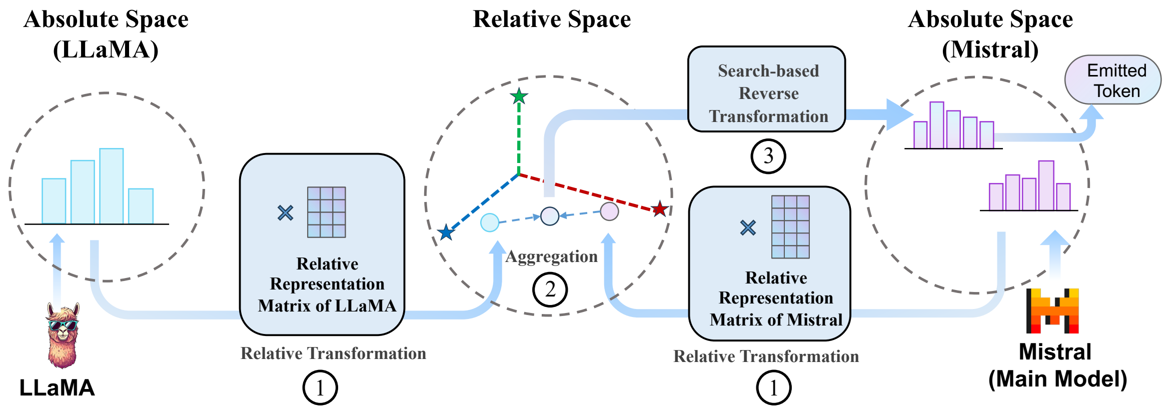

Figure 2: Overview of DEEPEN.

output semantic $\mathbf{p}^{(t)}$ and the relative representation matrix $R \in \mathbb{R}^{|V| \times |A|}$ where V is the vocabulary and $A \subseteq V$ is the anchor token set. The i-th row of R is the relative representation of word $w^{(i)}$:

$$R[i] = (cos(e_{w^{(i)}}, e_{a^{(1)}}), ..., cos(e_{w^{(i)}}, e_{a^{(|\mathbb{A}|)}})), \tag{2}$$

and the relative representation of the target semantic $\mathbf{p}^{(t)}$ is defined as:

$$\mathbf{r}^{(t)} = \mathbf{p}^{(t)} \cdot R \tag{3}$$

Model-invariance of relative representation of target semantic. Next, we illustrate why this representation scheme could align the output semantics isolated in heterogeneous absolute spaces. First, considering two LLMs θ_A and θ_B with the same vocabulary (*e.g.,* LLaMA2-7B and LLaMA2-13B). When expressing the same target semantic, these models output the same probability distribution (*i.e.,* absolute representation) $\mathbf{p}_A^{(t)}$ and $\mathbf{p}_B^{(t)}$. Besides, they have the same (highly similar in practice) relative representation matrix due the vocabulary consistency and cross-model invariance of relative representation. Therefore, the induced relative representations of output semantics are also the same:

$$\mathbf{r}_A^{(t)} = \mathbf{p}_A^{(t)} \cdot R_A = \mathbf{p}_B^{(t)} \cdot R_B = \mathbf{r}_B^{(t)}. \tag{4}$$

Then, let's consider a language model θ_C with a different vocabulary (*e.g.,* Mistral). Based on the fact that different LLMs share mass tokens in their vocabularies, the vocabulary of model θ_C is identical to adding and removing partial tokens to the vocabulary of θ_B, which leads to $\mathbf{p}_B^{(t)} \neq \mathbf{p}_C^{(t)}$ and $R_B \neq R_C$. However, in our study, we discover that this change to vocabularies has not incurred significant influence on the relative representation of the unchanged tokens (*i.e.,* the common tokens between θ_B and θ_C), as shown in Fig. 1. Therefore, we assume that the *local change in the vocabulary could hardly influence the relative space*. This hypothesis is strict but reasonable as the semantic does not change with the tokenizer intuitively.

3 Methodology

In this section, we first introduce the overall process of our ensemble framework DEEPEN and then describe the three parts of DEEPEN in detail.

3.1 Overview

Given N models to ensemble, DEEPEN first constructs their transformation matrices (*i.e.,* relative transformations) mapping the probability distributions from the heterogeneous absolute spaces into the relative space (§3.2). At each decoding step, all models perform the prediction and output N probability distributions. These distributions are transformed into the relative space and aggregated (§3.3). Finally, the result of aggregation is transformed back into the absolute space of one of the ensembling models, and is used to determine the generated token (§3.4).

3.2 Construction of Relative Transformation

Given N models to ensemble, DEEPEN first finds out the intersection of vocabularies of all models, *i.e.,* common token set C, and samples a subset or uses the full set of common tokens as the anchor token set $A \subseteq C$. Next, for each model, DEEPEN calculates embedding similarities of each token to the anchor words, obtaining the relative representation matrix R (as shown in Eq.4). Finally, to overcome the relative representation degeneration of outlier words, which will be introduced later, we perform normalization on the relative representation of all tokens by a softmax operation so that it becomes a probability distribution. We denote the normalized representation matrix $\hat{R}$:

$$\hat{R}[i] = softmax(R[i]). \tag{5}$$

Anchor Selection. The choice of anchor tokens is crucial for the relative representation capability. Previous research discovers that the capability improves as the number of anchor words increases (Moschella et al., 2023). Therefore, we employ the full set of common words between LLMs as the anchor words. It is also empirically proved that this method achieves more stable performance on downstream tasks.

Normalization of relative representation matrix. In DEEPEN, the relative representation of each token is normalized by the softmax operation to avoid the relative representation degeneration of outlier words, which are referred to as words that are far away from other words (including the anchors) and become distinguishable in relative space since for being zero vectors. The softmax operation effectively resolves this problem by making each relative representation a probabilistic distribution instead of a zero vector.

3.3 Aggregation in Relative Space

At each decoding step, once each model θ_i outputs the probability distribution $\mathbf{p}_i$, DEEPEN transforms $\mathbf{p}_i$ into the relative representation $\mathbf{r}_i$ using the normalized relative representation matrix:

$$\mathbf{r}_i = \mathbf{P}_i \cdot \hat{R}_i, \tag{6}$$

and aggregate all relative representations to obtain the aggregated relative representation:

$$\overline{\mathbf{r}} = \sum_{i=1}^{N} \mathbf{r}_i. \tag{7}$$

Note that we adopt the most simple method of aggregation (*i.e.,* averaging) to make our framework more general in this work. But it could be integrated with other sophistic technologies of output fusion (*e.g.,* stacking (Wolpert, 1992))

3.4 Inverse Transformation of Relative Representations

To decide the emitted token according to the result of aggregation, DEEPEN aims to transform it from the relative space back to the absolute space of one of ensembling models (main model). In practice, we select the model with the best performance on the validation set as the main model, which often achieves better ensemble performance. To enable this inverse transformation, we adopt a research strategy, which finds out the absolute representation that the relative representation of which is identical to the aggregated relative representation. This search problem is formulated as:

$$\overline{\mathbf{p}}_{(i)} = \underset{\mathbf{p}'_{(i)} \in \mathbb{P}_{(i)}}{\arg\min}\, \ell(\mathbf{p}'_{(i)} \times \hat{R},\, \overline{\mathbf{r}}), \tag{8}$$

where $\mathbb{P}_{(i)}$ denotes the absolute space of model θ_i, and $\ell(\cdot)$ is the loss function to measure the distance between two relative representations. In this work, we adopt the KL-divergence since for its stability in our experiments.

This search is conducted under the guidance of the gradient of the divergence between relative representations with respect to the absolute representation. Specifically, the start point is initialized with the main model's original absolute representation $\mathbf{p}_{(i)}$, and use the gradient of the loss in Eq.8 with respect to $\mathbf{P}'_{(i)}$:

$$\mathbf{p}'_{(i)} = \mathbf{p}'_{(i)} - \eta \times \frac{\partial \ell}{\partial \mathrm{P}'_{(i)}}, \tag{9}$$

where η is an important hyperparameter named as the relative ensemble learning rate. This search process is iterated for $T = 5$ steps, where T is named as the number of relative ensemble learning steps.

Finally, we use the updated absolute representation $\mathbf{p}'_{(i)}$ to determine the emitted token.

4 Experiments

4.1 Experimental Setup

Benchmarks. Although DEEPEN is a task-agnostic approach, which could apply on any NLP task or even beyond NLP, we mainly conduct experiments on six benchmarks, which could be categorized as three kinds of tasks:

- **Comprehensive Examination:** (1) MMLU (5-shot) (Hendrycks et al., 2021), which covers 57 subjects that humans learn across STEM, the humanities, and the social sciences and ranges in difficulty from an elementary level to an advanced professional level, and (2) ARC-C (0-shot) (Clark et al., 2018), which consists of a

Models	MMLU		ARC-C	
	dev	test	dev	test
Individual Models				
Yi-6B	61.19	63.24	72.72	73.33
Mistral-7B	60.80	62.07	73.88	74.10
SkyWork-13B	58.65	61.21	67.08	66.50
Top-2 Ensemble				
DeepEn	63.61	65.69	77.73	75.89
Δ	+2.42	+2.45	+3.85	+1.79
Top-3 Ensemble				
PairRanker	–	63.77	–	70.85
GenFuser	–	37.59	–	58.21
DeepEn	63.25	65.25	78.09	77.09
Δ	+2.06	+2.01	+4.21	+2.99

Table 1: Results on comprehensive examination.

Models	GSM8K		PIQA	
	dev	test	dev	test
Individual Models				
LLaMA2-70B	68.67	65.73	70.54	71.27
Mixtral-8×7B	66.67	63.84	70.90	71.88
DeepEn	69.67	67.33	73.54	75.10
Δ	+1.00	+1.60	+2.64	+3.22

Table 2: Results on reasoning tasks.

collection of natural science questions authored for standardized tests.

- **Reasoning Capabilities:** (1) GSM8K (Cobbe et al., 2021) (4-shot), which is a dataset of high quality problems at the grade school math level, and (2) PIQA (Bisk et al., 2020), which is a commonsense reasoning dataset.

- **Knowledge Capacities:** (1) TrivialQA (5 shot) (Joshi et al., 2017), which consists of Trivia enthusiast authored question-answer pairs, and (2) NaturalQuestions (5-shot) (Kwiatkowski et al., 2019), which is a QA corpus consists of queries issued to the Google search engine.

Evaluation. For all benchmarks, we follow the test scripts of OpenCompass[2]. Specifically, on the multiple-choice tasks (MMLU, ARC-C, and BBH), the option with the highest likelihood is selected to calculate the accuracy. On the free-form generation

[2] https://opencompass.org.cn/

Models	TriviaQA		NQ	
	dev	test	dev	test
Individual Models				
LLaMA2-13B	72.74	73.57	21.56	28.67
Mistral-7B	70.47	72.58	19.94	27.67
InternLM-20B	64.11	65.90	19.50	26.09
Top-2 Ensemble				
DeepEn	74.79	75.13	21.81	30.17
Δ	+2.05	+1.56	+0.25	+1.50
Top-3 Ensemble				
PairRanker	–	70.27	–	26.9
GenFuser	–	1.87	–	0.17
DeepEn	74.10	74.92	23.50	30.81
Δ	+1.36	+1.35	+1.94	+2.14

Table 3: Results on knowledge-intensive QA Tasks. NQ refers to NaturalQuestion.

tasks (GSM8K, TrivialQA and NaturalQuestions), we calculate the exact match (EM) accuracy.

Individual models. Based on the key principles of ensemble learning (Sagi and Rokach), the synergy is created only in cases where individual models possess comparable performance to each other. Therefore, on each task type, we select three LLMs that have achieved promising results on the OpenCompass leaderboard and whose performance is as close to each other as possible. On the challenge reasoning task, we conduct experiments for the ensemble of two large-scale LLMs: LLaMA2-70B and Mixtral-8×7B.

Hyperparameters. In this work, we select all of the common tokens between LLMs as the anchor tokens to build the relative spaces, *i.e.*, $A = C$. In the decoding of relative representation, we search the optimal relative learning rate (Eq. 9) from 0.0 to 0.3 with an interval of 0.05. We set the number of relative ensemble learning steps $T = 5$.

Comparative methods. We also compare our ensemble framework DeepEn with the selection-based ensemble method PairRanker, which is a reward model to score each response of LLMs and the fusion-based ensemble method GenFuser, which is a generative model to fuse multiple candidate responses. Both models are from LLM-Blender framework Jiang et al. (2023b) and are trained on the constructed instruction tuning dataset MixInstruct.

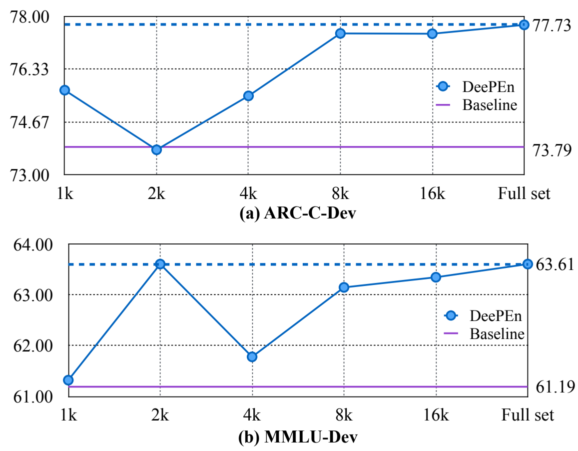

Figure 3: Effect of the number of anchor words. The x-axis indicates the number of anchor words randomly sampled from the common words.

Methods	MMLU-Dev		TrivialQA-Dev	
	ACC	Δ	ACC	Δ
Baseline	61.19	–	72.74	–
DEEPEN	63.61	+2.42	74.79	+2.05
w/o. **Rel-Norm**	60.73	-0.46	72.95	+0.21

Table 4: Ablation study of normalization on the relative representation matrix to the ensembling performance on the development sets. **Baseline** refers to as the best single model on each benchmark. **DEEPEN** refers the performance of ensembling top-2 models in the benchmark.

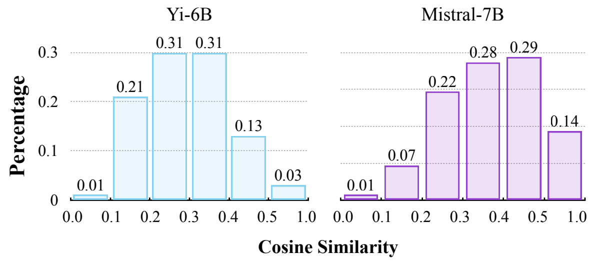

Figure 4: Distance distribution to nearest neighbor words. The distance is measured by calculating the cosine similarity between words.

4.2 Results

The main results are shown in Tab 1,2, and 3, from which we have drawn the following observations:

DEEPEN achieves consistent improvements over the individual models across all benchmarks. These results prove that our framework indeed enables collaboration between LLMs via aggregating the target semantics behind the output probability distributions. Specifically, the ensemble of top-2 models achieves the improvements of +1.5(NQ)∼+3.22(PIQA) on test sets and +0.25(NQ)∼+3.85(ARC-C) on validation sets in terms of accuracy. Through the ensemble of top-3 models, DEEPEN gains a further improvement over ensembling top-2 models. As shown in Tab. 1, the improvement is increased from +1.79 to +2.99 after the joining of the third-ranked model on ARC-C.

Collaboration with models significantly worse could lead to degeneration. As the results in Tab 3, the ensemble of top-3 models underperforms than the one of top-2 models on TrivialQA. This degeneration is reasonable due to the significant performance gap (-6.68) between the third-ranked model (InternLM-20B) and the second-ranked models (Mistral-7B) on this benchmark.

DEEPEN shows excellent generalizability. As we can see, Both PAIRRANKER and GENFUSER perform poorly on each benchmark, which is caused by the serious distribution shift from their training data to the test data. Instead, DEEPEN works well under the training-free setting.

5 Analysis

Towards better understanding DEEPEN, we conduct a series of analyses on the relative transformation and the algorithm of relative representation decoding.

5.1 Analysis on Relative Transformation

Effect of anchor selection. We demonstrate the impact of different numbers of anchor words through experiments with the top-2 ensemble models on the MMLU and ARC-C benchmarks. As shown in Fig. 3, we observe that an increased number of anchor words does not necessarily translate into improved performance for LLMs in downstream tasks. Fortunately, selecting the full set of common words as anchors could achieve stable promising performance.

Effect of normalization on relative representation matrix. To demonstrate the importance of normalization on the relative representation matrix to the ensemble performance, we conduct an ablation analysis. The result is shown in Tab. 4, the ensemble struggles to achieve improvements due to the ineffective representation of outlier words, *i.e.*, words distant to other words. The proportion of outlier words can be derived from the distribu-

RELR (η)	0.05	0.10	0.15	0.20	0.25	0.30
MMLU	+2.42	+1.57	+1.77	+1.96	+1.31	+1.31
TrivialQA	+1.31	+2.05	+1.63	+1.94	+1.82	+1.26

Table 5: Sensitivity analysis of relative ensemble learning rate (**RELR**). We report the improvements of ensembling top-2 models over the best individual models.

	Baseline	TrivQA	NQ	ARC-C	MMLU
TrivialQA	72.74	72.90	72.13	69.35	70.70
NQ	21.56	21.56	21.88	21.81	21.62
ARC-C	59.32	69.32	71.97	73.76	73.26
MMLU	55.10	59.98	61.02	61.86	61.42

Table 6: Cross-distribution validation of relative ensemble learning rate (η). We report the performance of ensembling LLaMA2-13B and Mistral-7B. Each row indicates the test set used to evaluate performance. Each column indicates the development set used to search the optimal value of RELR.

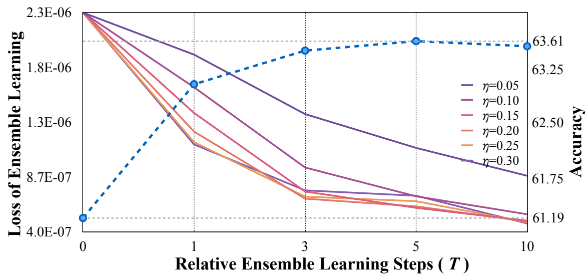

Figure 5: Effect of different number of relative ensemble learning steps.

Models	**MMLU-Dev**		**ARC-C-Dev**	
	INDIV	DEEPEN	INDIV	DEEPEN
Yi-6B	61.19	**63.61** (+2.42)	72.72	**77.55** (+4.83)
Mistral-7B	60.80	**64.46** (+3.66)	73.88	**77.73** (+3.85)

Table 7: Performance of DEEPEN with choosing different main models on the development sets. INDIV refers to as individual models. The result of DeePEn indicates the performance of using the model of this row as the main model.

tion of distance to nearest neighbor words, which is illustrated in Fig. 4. As illustrated, a remarkable proportion ($> 30\%$) of words are distant from other words, *i.e.,* cosine similarity to its nearest neighbor word is less than 0.3. Through the normalization operation, the target semantics that intend to emit outlier words could be prevented from becoming zero vectors by relative transformation.

5.2 Analysis of Reverse Transformation

As described in §3.4, the relative representation of the aggregated target semantic is transformed back to the absolute space of the main model via a search-based algorithm. There are two important factors (relative ensemble learning rate and number of search steps) in this reverse transformation, which are comprehensively analyzed, respectively.

Analysis of relative ensemble learning rates. As shown in Tab. 5, the performance of DEEPEN is sensitive to the value of relative ensemble learning rate (η), which is abbreviated by RELR. This observation motivates us to measure the generality of this hyperparameter. Specifically, we illustrate the cross-distribution performance of the searched optimal value of η in Tab. 6. As observed, the optimal value of RELR varies across different datasets, which suggests that the inverse transformation from relative space to absolute space requires adaptive mapping modes.

Effect of iteration steps in relative ensemble learning. To give a deep view of the dynamics of the inverse transformation in DEEPEN, we report the performance change along with different numbers of relative ensemble learning steps (T). Besides, the dynamics of loss of relative ensemble learning (ℓ in Eq. 8)is also reported. As shown in Fig. 5, on the one hand, more steps of relative ensemble learning significantly lead to lower losses. However, the loss is hard to reach zero, *i.e.,* underfitting. On the other hand, increasing the number of steps of relative ensemble learning will cause the performance to increase first and then decrease. The reason behind the performance drop could be that in the early stage of optimization, the focus of optimization is on updating the high-probability tokens. In the later stage of optimization, since the probabilities of all words will be adjusted equally, the low-probability tokens will be interfered with, thus affecting the performance. Therefore, it is recommended to set a modest value of step number (*e.g.,* $T = 5$).

Choice of main model. In the process of inverse transformation, DEEPEN maps the relative aggregated representation to the absolute space of the main model. Ideally, we expected the results of inverse transformation to keep invariant with the choice of main model. However, this objective is hard to achieve due to the underfitting observed in

Models	MMLU-Dev			MMLU-Test		
	INDIV	VANIL	DEEPEN	INDIV	VANIL	DEEPEN
LLaMA1	43.26	45.48	44.37	43.70	45.01	44.22
LLaMA2	42.28		**45.94**	42.99		**45.31**

Table 8: Comparison to vanilla prediction average (VANIL) on the ensemble of LLMs with the same vocabulary.

the search process. Therefore, we illustrate the performance gap of choosing different main models in Tab. 7. As the results shown on ARC-C, changing the main model from the first-ranked Mistral-7B to the second-rank Yi-6B, the ensemble performance is decreased slightly from 77.73 to 77.55. Interestingly, changing the main model from the rank-1 Yi-6B to the rank-2 Mistral-7B on **MMLU**, the performance is actually improved from 63.63 to 64.46, which indicates that Mistral-7B benefits more than Yi-6B from collaboration. Even so, **choosing different main models does not significantly affects the ensemble performance**.

5.3 Comparison to Vanilla Prediction Average

To compare our DEEPEN with vanilla prediction average, we conduct an experiment for ensembling two LLMs with the same vocabulary and comparable performance on MMLU, *i.e.,* LLaMA2-7B and LLaMA1-13B. As shown in Tab. 8, the performance of DEEPEN is comparable, even better than, that of the vanilla prediction average. Theoretically, the performance of the vanilla prediction average is the performance upper-bound of DEEPEN. The reason that DEEPEN could excel over the vanilla one on MMLU is the under-fitting in the inverse transformation process, which leads to the weights to aggregate the target semantics of different models not being a uniform distribution (*i.e.,* $(0.5, 0.5)$). For example, in Tab. 8, the weights for LLaMA1 and LLaMA2 could be $(0.6, 0.4)$, where the weight of the main model is larger than the other model.

6 Related Work

Existing work of LLM ensemble could be divided into selection-based and fusion-based ensemble.

Selection-based ensemble. *Rerank* is an intuitive solution to utilize multi-model strengths. Specifically, Jiang et al. (2023b) takes the first step towards ensembling LLMs, proposing PAIRRANKER for pairwise comparison on candidate outputs and achieving improvements on the self-constructed instruction dataset. To overcome the huge computation costs of multi-LLM inference, several works have explored to train a *router* to predict the best-performing model out of a fixed set of LLMs for the given input (Wang et al., 2024; Shnitzer et al., 2024; Lu et al., 2023).

Fusion-based ensemble. Towards a synergy between LLMs, Jiang et al. (2023b) propose GENFUSER, trained to generate an improved response to capitalize on the strengths of multiple candidates. Furthermore, Du et al. (2023) design a debate-like prompt strategy to ask the LLM to refine its answer after reading peer models' answers. However, this prompt-based method incurs substantial computation costs and inference latency for LLM communication. Specifically, considering the ensemble of N LLMs, the communication leads to the encoding of $N \times L$ tokens and decoding of L tokens, where L denotes the average length of answers.

Different from the training-dependent ensemble methods which pose a great challenge to the generalizability of the trained reward model or fusion model, our DEEPEN is training-free, making it more general. Compared to debate-like prompt strategy, DEEPEN has higher computation efficiency and achieves more stable improvements.We place the comparison experiments with Debate in Appendix A.1.

A contemporaneous work to us is FuseLLM, which propose a method of vocabulary mapping to enable knowledge distillation between heterogeneous LLMs. Even though FuseLLM works for knowledge distillation, it hardly achieve improvement under the training-free setting of ensemble learning, as the experimental results shown in A.2.

7 Conclusion

In this work, we propose a training-free LLMs collaboration framework DEEPEN, which averages the probability distributions of models with heterogeneous vocabularies. To the best of our knowledge, we are the first to effectively enable the aggregation of heterogeneous probability distributions. We believe our research can motivate further research on the LLMs collaboration, model reuse, and knowledge distillation. In the future, we are planning to experiment with DEEPEN on the ensemble between LLMs and generative expert language models (*e.g.,* machine translation models) due to their complementary knowledge.

8 Limitations

Different from previous LLM ensemble methods, our DEEPEN is uniquely designed for the ensemble of white-box LLMs due to necessitating authorization of the output probability distribution. Besides, DEEPEN assigns equal weights to each individual model, which fails to deal with the interference caused by the poor-performing model. This problem motivates us to introduce ensemble strategies like stacking technology (Wolpert, 1992).

References

Yonatan Bisk, Rowan Zellers, Ronan Le bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7432–7439.

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.

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

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

Ekaterina Garmash and Christof Monz. 2016. Ensemble learning for multi-source neural machine translation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1409–1418, Osaka, Japan. The COLING 2016 Organizing Committee.

Bobby He and Mete Ozay. 2022. Feature kernel distillation. In *International Conference on Learning Representations*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023a. Mistral 7b.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023b. LLM-blender: Ensembling large language models with pairwise ranking and generative fusion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14165–14178, Toronto, Canada. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Keming Lu, Hongyi Yuan, Runji Lin, Junyang Lin, Zheng Yuan, Chang Zhou, and Jingren Zhou. 2023. Routing to the expert: Efficient reward-guided ensemble of large language models.

Luca Moschella, Valentino Maiorca, Marco Fumero, Antonio Norelli, Francesco Locatello, and Emanuele Rodolà. 2023. Relative representations enable zero-shot latent space communication. In *The Eleventh International Conference on Learning Representations*.

OpenAI. 2023. Gpt-4 technical report.

Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. 2019. Relational knowledge distillation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3967–3976.

Omer Sagi and Lior Rokach. Ensemble learning: A survey. *WIREs Data Mining and Knowledge Discovery*, 8(4):e1249.

Tal Shnitzer, Anthony Ou, Mírian Silva, Kate Soule, Yuekai Sun, Justin Solomon, Neil Thompson, and Mikhail Yurochkin. 2024. Large language model routing with benchmark datasets.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Hongyi Wang, Felipe Maia Polo, Yuekai Sun, Souvik Kundu, Eric Xing, and Mikhail Yurochkin. 2024. Fusing models with complementary expertise. In *The Twelfth International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

David H. Wolpert. 1992. Stacked generalization. *Neural Networks*, 5(2):241–259.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models.

A Comparison with More Baselines

A.1 DEEPEN vs. Debate

We compare our DEEPEN with the debate-like prompt strategy (Du et al., 2023) on the ARC-C benchmark in Tab. 9. As we can see, Debate works only on the CoT(Wei et al., 2022) setting while underperforming on the setting without CoT. Note that DEEPEN achieve a larger improvement (+1.79) over a stronger baseline (74.10), which indicate the superiority. Besides, DEEPEN incurs less computation cost than Debate. Specifically, the cost of DEEPEN is estimated as $(N+2T-1)\times|V|\times|A|$, and the cost of Debate is estimated as $N \times L \times |\theta|$. Last but not least, DEEPEN leads to less inference latency than Debate.

A.2 DEEPEN vs. FuseLLM

To achieve a fair comparison to FuseLLM, we re-implement their vocabulary mapping method with the released code under the ensemeble learning setting. Specifically, on ARC-C, the probability distribution outputted by Yi-6B is mapped in the vocabulary of Mistral-7B. Then, we averaged the mapped probability distribution and the probability distribution outputted by Mistral-7B. Finally, the averaged distribution is used to determine the emitted token. As the results in Tab. 9, FuseLLM hardly achieve improvements under the setting of ensemble learning. The most essential difference is that FuseLLM uses the literal similarity to enable the token alignment while DEEPEN leverages the relative representation to enable the probability distribution aggregation.

Models	**ARC-C-Test**			
	Acc (w/o. CoT)	Acc (w. CoT)	Communication Cost	Inference Latency
Mistral-7B	74.10	67.86	0	1x
Yi-6B	73.33	70.60	0	1x
Debate	73.25 (-0.85)	72.20 (+1.4)	14B$\times L$	2x
FuseLLM	74.10 (+0.00)	–	0.20B$\times L$	1.0x
DEEPEN	75.89 (+1.79)	–	0.85B$\times L$	1.25x

Table 9: Comparison with previous debate-like prompt strategy and FuseLLM in terms of performance (Accuracy), computation cost for communication, and inference latency. L represents the average length of generated response.