SHARING BEYOND DECISION: Deep Collaboration between Large Language Models via Representation Ensemble

Yichong Huang[1] **Xiaocheng Feng**[1]✉ **Jinlan Fu**[2]✉ **Xiachong Feng**[3]
Baohang Li[1] **Zekai Ye**[1] **Libo Qin**[4] **Hao Fei**[2] **See-Kiong Ng**[2] **Bing Qin**[1]
[1]Harbin Institute of Technology, China
{ychuang, xcfeng, baohangli, zkye, tliu, qinb}@ir.hit.edu.cn
[2]National University of Singapore, Singapore
{haofei37, seekiong}@nus.edu.sg jinlanjonna@gmail.com
[3]The University of Hong Kong, China
fengxc@hku.hk
[4]Central South University, China
lbqin@csu.edu.cn

Abstract

Large Language Models (LLMs) exhibit unique strengths arising from differences in model architecture, training data, and strategies. Ensemble learning has been explored to leverage these complementary strengths through decision-level sharing (*i.e.,* **Decision Ensemble**), which combines the predictions from multiple LLMs. However, such methods integrate only shallow decisions and overlook the exchange of deeper levels of information within the internal representations of LLMs, such as problem understanding, world knowledge, and latent reasoning patterns. In this work, we propose **Representation Ensemble** (RISE), a novel ensemble framework that enables cross-LLM representation sharing for richer information exchange. To address challenges of representation-level interaction caused by layer misalignment and latent-space incompatibility across LLMs, we introduce a representation alignment method based on relational similarity measures and an orthogonal latent-space transformation. Experimental results show that (1) RISE achieves performance competitive with existing decision ensemble methods, and (2) RISE is strongly complementary to decision ensemble, with their combination boosting collaboration gains by 14%–41%. Finally, we further compare ensemble of small LLMs to a single larger LLM and to model merging and composition approaches, and find that ensemble learning consistently generalizes well without additional training.

✉ Corresponding authors.

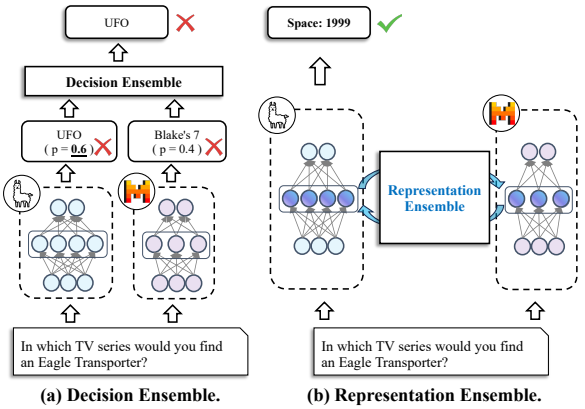
1 Introduction

As model capacities and data volumes continue to scale, LLMs have demonstrated remarkable understanding and generation capabilities, offering new prospects for artificial general intelligence (Zhao et al., 2023; OpenAI, 2023; Jiang et al., 2023a; Touvron et al., 2023a). Due to variations in data sources, model architectures, and training methodologies, LLMs exhibit distinct strengths across tasks and scenarios. Consequently, recent research has explored LLM ensembling to harness their complementary potential, thereby yielding more comprehensive predictions (Jiang et al., 2023b; Lu et al., 2023), even without extra training or access to the source data (Huang et al., 2024; Xu et al., 2024).

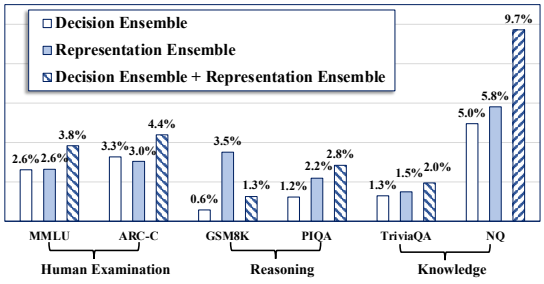
Existing work leverages LLMs' complementary strengths by sharing decisions at different levels. Instance-level ensembling selects the optimal answer or fuses candidate responses. Specifically, LLM-Blender (Jiang et al., 2023b) introduces extra training modules to select and fuse candidate responses. Token-level ensembling, at each decoding step, averages probability distributions over next-token predictions from multiple LLMs, reducing errors in the generation process. To address vocabulary discrepancies across LLMs, DeePEn (Huang et al., 2024) projects the heterogeneous token distributions into a unified space using relative representation, and EVA (Xu et al., 2024) learns a mapping matrix to achieve alignment.

Despite significant progress, existing studies focus solely on decision-level ensembling, overlooking the exchange of deeper information in LLMs' internal representations, such as complex problem

Transactions of the Association for Computational Linguistics, vol. 14, pp. 1765–1786, 2026. https://doi.org/10.1162/TACL.a.773
Action Editor: Jacob Andreas. Submission batch: 3/2025; Revision batch: 9/2025; Published 7/2026.



(a) Decision Ensemble. (b) Representation Ensemble.



(c) Relative improvements achieved by different ensemble learning methods.

Figure 1: Representation ensemble (ours) vs. decision ensemble (DeePEn).

understanding (Liu et al., 2019), factual knowledge (Gromov et al., 2025), and latent reasoning patterns (Hao et al., 2024). This limitation often leads to suboptimal collaboration: as illustrated in Figure 1(a), restricting information exchange to the decision level biases the ensemble toward high-confidence candidates. When none of the individual models provides the correct prediction, the decision ensemble consequently struggles.

To overcome this limitation, we propose a novel LLM ensemble framework, **Representation Ensemble** (**RISE**), as shown in Fig. 1(b). RISE enables cross-LLM representation sharing, allowing richer information exchange and deeper collaboration. Specifically, RISE selects one LLM as the central model and the others as assistant models. During inference, the assistant models share their internal representations with the central model to form a consensus representation. This consensus integrates the collective understanding and complementary insights of all LLMs, making it more likely to solve problems no single model can handle.

Unfortunately, interaction across LLM representations faces two key challenges: intricate **layer misalignment** and inherent **latent space mismatch**. To address these challenges, we propose a representation alignment method. First, to tackle layer misalignment, we perform layer-wise alignment using relational representation similarity measures (Klabunde et al., 2024). Second, we identify suitable layers for cross-LLM representation fusion through layer clustering. Third, to overcome latent space mismatch, we derive a representation projection matrix via Procrustes analysis, computing an optimal orthogonal (rotational) transformation (Maiorca et al., 2023). As projection distortions can still arise from inherent latent space differences, we further introduce an inference-stage representation-verification strategy to detect and remove such projection errors.

On six benchmarks (human examination, reasoning, and knowledge QA), RISE consistently improves over individual LLMs and attains performance competitive with decision ensemble. Moreover, combining RISE with decision ensemble yields the best results, boosting collaboration gains across tasks (Fig. 1(c)). To understand when each paradigm helps, we introduce the *Ensemble Effect Quadrant* , an analysis framework that reveals their complementary regimes (Sec. 6.1). The main contributions are summarized:

- We propose **RISE**, a novel LLM ensemble framework that enables deeper collaboration via cross-model representation sharing.
- We develop a **representation alignment** method to mitigate layer misalignment and latent-space incompatibility, enabling effective integration of representations.
- Through extensive experiments on six benchmarks, we show that combining RISE with decision ensemble yields **substantial collaboration gains of 14%–41%**.

2 Preliminaries

2.1 Ensemble Learning for LLMs

This work follows the token-level decision ensemble paradigm due to its strong performance (Huang et al., 2024; Xu et al., 2024; Yao et al., 2025). Given N LLMs to ensemble, token-level ensemble selects one of them as the central model θ^C, which usually is the best-performing model. The other LLMs serve as assistant models $\{\theta^{A_i}\}_{i=1}^{N-1}$. At the t-th decoding step, each model (θ^C or θ^{A_i}) first outputs a probability distribution ($\mathbf{p}^C$ or $\mathbf{p}^i$) over its vocabulary. Then, all assistant models convey their

distributions to the central model to perform aggregation: $\hat{\mathbf{p}}^C = \phi(\mathbf{p}^C, \mathbf{p}^1, ..., \mathbf{p}^{N-1})$, where $\phi(\cdot)$ indicates the aggregation function. For LLMs with the same vocabulary, this function could be simply implemented as weighted averaging. For LLMs with different vocabularies, Huang et al. (2024) proposed DeePEn to map these heterogeneous distributions into a universal relative space to perform aggregation. Finally, the next token is determined based on the aggregated distribution $\hat{\mathbf{p}}^C$. This process is iterated in an auto-regressive manner until outputting the complete answer.

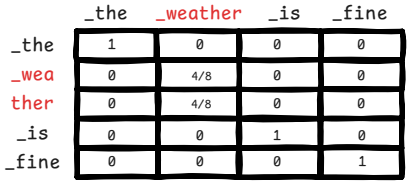


Figure 2: Character-based token alignment.

2.2 Representational Similarity Measure

To measure the similarity between two heterogeneous latent spaces, this work adopts Nearest-Neighbor-based measure (Klabunde et al., 2024; Huh et al., 2024). First, given B samples, the representations in their latent spaces are denoted as $\mathbf{H}^1 \in \mathbb{R}^{B\times d^1}$, $\mathbf{H}^2 \in \mathbb{R}^{B\times d^2}$. We formally define the k nearest neighbors of i-th instance in the latent space $\mathbf{H}^j$ as the set $\mathcal{N}^k_{\mathbf{H}^j}(i)$. Then, the similarity of these two latent spaces is computed as:

$$sim(\mathbf{H}^1, \mathbf{H}^2) = \frac{1}{B}\sum_{i=1}^{B}\frac{|\mathcal{N}^k_{\mathcal{H}^1}(i) \cap \mathcal{N}^k_{\mathcal{H}^2}(i)|}{k}. \tag{1}$$

We empirically set $k = 10$, and provide an analysis of the effect of different k values in §B.1.

2.3 Token Alignment

Since different LLMs usually have different tokenizers (Wan et al., 2024), the input $\mathbf{x}$ usually is split into different sequences of tokens: $\mathbf{s}^1$ and $\mathbf{s}^2$. To align these two sequences, we devise an intuitive strategy of token alignment based on the character correspondence, which is illustrated in Fig. 2. The alignment weight between i-th token in $\mathbf{s}^1$ and j-th token in $\mathbf{s}^2$ equals the fraction of overlapped characters between their spans.

3 Representation Alignment across LLMs

To address the critical challenges of layer misalignment and latent space mismatch between different LLMs, this work achieves representation alignment across LLMs. We illustrate this process in Fig. 3. Specifically, we first establish the layer correspondence between the central model and each assistant model (§3.1). Next, we identify the transition layers in the central model as fusion layers(§3.2). Then, to tackle the latent space mismatch, the hidden state pairs of the central model and each assistant model (§3.3) are collected to train or derive a projection from the latent space of the assistant model to that of the central model (§3.4).

3.1 Layer Alignment

Since LLMs have varying layers with distinct functions, the first challenge in representation alignment is establishing layer correspondence. This work tackles this challenge using the representation similarity measure. Considering the central model θ^C with L^C layers and the assistant model θ^{A_i} with L^{A_i} layers, we measure the representation similarity between each layer and establish a layer alignment matrix $\mathbf{M}^{(C,A_i)} \in \mathbb{R}^{L^C\times L^{A_i}}$. Formally, the (m, n)-th entry of the matrix is calculated as:

$$\mathbf{M}^{(C,A_i)}[m][n] = sim(\mathbf{H}^C_m, \mathbf{H}^{A_i}_n), \tag{2}$$

where $\mathbf{H}^C_m \in \mathbb{R}^{B\times d^C}$ denotes the m-th-layer hidden states in the central model, and $\mathbf{H}^{A_i}_n(x) \in \mathbb{R}^{d^{A_i}}$ denotes the n-th-layer hidden states in the assistant model. These hidden states are obtained by sampling $B = 100,000$ tokens from a high-quality corpus MiniPile (Kaddour, 2023). sim indicates the representational similarity measure (§2.2).

3.2 Fusion Layer Selection

Sharing representations across all layers is computationally expensive. Prior work shows that LLM layers exhibit a stage-wise structure, with adjacent layers being highly similar (Sun et al., 2025). We therefore select a small set of *transition layers* that mark boundaries between stages.

Given a model with L layers, we compute the layer similarity matrix $\mathbf{Q} \in \mathbb{R}^{L\times L}$, where $\mathbf{Q}[i][j] = sim(\mathbf{H}_i, \mathbf{H}_j)$. We define the *layer separability* of layer l as

$$\omega(l) = \mathbf{Q}[l][l-1] - \mathbf{Q}[l][l+1], \tag{3}$$

which captures the asymmetry of representational similarity across neighboring layers.

According to prior work and our pilot study in Fig. 4, we typically observe four prominent representational transitions over depth. These transitions empirically correspond to the last layers

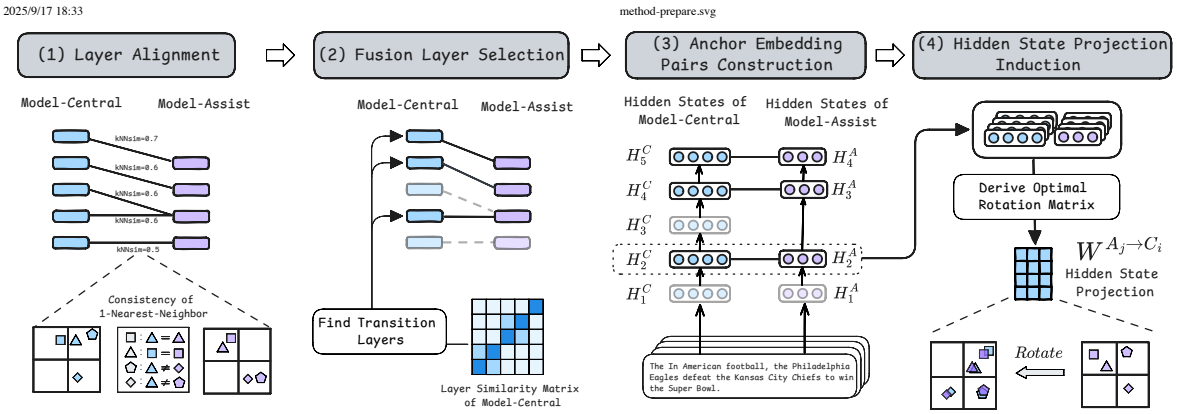


Figure 3: Illustration of representation alignment from the assistant model to the central model.

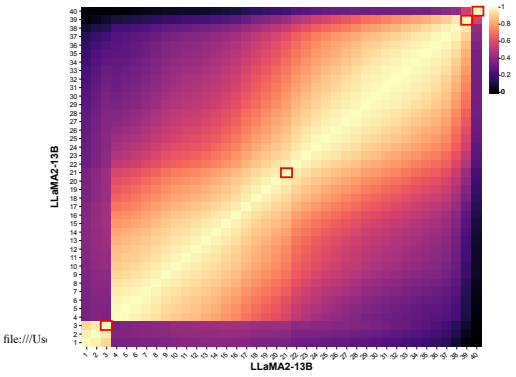


Figure 4: Illustration of layer similarity across different layers of LLaMA2-13B. This work divides all layers of LLaMA2-13B into: bottom layers (1-3), middle layers (4-21), high layers (22-39), and the top layer (40). Red boxes indicate the transition layers.

of four stages, commonly referred to as bottom, middle, high, and top layers. We hypothesize that the first three transition layers lie within the following depth ranges, respectively: $[1, \lfloor L/4 \rfloor]$, $[\lfloor L/4 \rfloor + 1, \lfloor 3L/4 \rfloor]$, and $[\lfloor 3L/4 \rfloor + 1, L-1]$. In each range $\mathcal{I}$, the transition layer is selected as

$$l^* = \arg\max_{l \in \mathcal{I}} \omega(l). \tag{4}$$

We find that sharing lower-layer representations brings negligible gains, so we discard this variant and keep only the remaining three:

- **RISE-M:** aggregating the representations of the *middle* layers, sharing their understanding of the question across LLMs.
- **RISE-H:** aggregating the representations of the *high* layers to share abstract thoughts.
- **RISE-T:** aggregating the representations of the *top* layer to share output semantics.

3.3 Construction of Anchor Embedding Pairs

Considering the selected fusion layer l^C of central model and its aligned layer l^{A_i} of assistant model, this step aims to collect the anchor embedding pairs $\{\mathbf{H}^C_{l^C} \in \mathbb{R}^{B\times d^C}, \mathbf{H}^{A_i}_{l^{A_i}} \in \mathbb{R}^{B\times d^{A_i}}\}$, to derive the projection from the latent space of l^{A_i} to that of l^C. These anchor embedding pairs record a set of hidden state pairs of the central model and assistant model on same tokens in the corresponding layer.

3.4 Derivation of Representation Projection

This step aims to estimate the projection $\mathcal{F}$ from $\mathbf{H}^{A_i}_{l^{A_i}}$ to $\mathbf{H}^C_{l^C}$. For clarity, we simplify these notations as $\mathbf{H}^C$, and $\mathbf{H}^{A_i}$. Then, the estimation is formalized: $\mathbf{H}^C = \mathcal{F}(\mathbf{H}^{A_i})$. We preprocess $\mathbf{H}^C$ and $\mathbf{H}^{A_i}$ by standardizing each feature to have zero mean and unit variance. Then, we explored three approaches to estimate this projection:

Gradient Descent. The most intuitive resolution to estimate $\mathcal{F}$ is to train a feed-forward network with gradient descent:

$$\mathcal{F}(\mathbf{h}) = \mathbf{W}_{out} LN(\mathbf{W}_{in}\mathbf{h}) + \mathbf{b}, \tag{5}$$

where $\mathbf{W}_{in} \in \mathbb{R}^{d' \times d^{A_i}}$, $\mathbf{W}_{out} \in \mathbb{R}^{d^C \times d'}$, $\mathbf{b} \in \mathbb{R}^{d^C}$. d' is empirically set to 16384. $LN(\cdot)$ denotes the layer normalization.

Least Squares. The classic Least Squares estimates $\mathcal{F}$ as a matrix: $\mathbf{W} = \mathbf{H}^C(\mathbf{H}^{A_i})^{-1}$, where $(\mathbf{H}^{A_i})^{-1}$ is the pseudo inverse of $\mathbf{H}^{A_i}$.

Rotation. As previous work reveals the rotational relationship between the latent spaces of similar models, we estimate $\mathcal{F}$ by deriving the rotation matrix $\mathbf{W}$ across latent spaces of different LLMs,

i.e., solving an orthogonal Procrustes problem:

$$\begin{aligned} \arg\min_{\mathbf{W}} \quad & ||\mathbf{W}\mathbf{H}^{A_i} - \mathbf{H}^{C}||_F, \\ \text{subject to} \quad & \mathbf{W}^T\mathbf{W} = I, \end{aligned} \tag{6}$$

where $||\cdot||_F$ denotes the Frobenius norm, and the constraint is used to ensure the solution $\mathbf{W}$ to be an orthogonal transformation (*i.e.,* a rotation). This problem can be efficiently resolved by Procrustes analysis (Schönemann, 1966).

Gradient descent is the most time-consuming due to hyperparameter tuning, whereas least squares and rotation offer efficient closed-form projections. We adopt ***Rotation*** as the default implementation for its superior effectiveness.

Finally, in addition to computing the projection $\mathcal{F}^{A_i \to C}$, RISE also computes the projection $\mathcal{F}^{C \to A_i}$ from the space of the central model to the space of the assistant model in the same way. This reverse projection is used to broadcast the representation to the assistant models (§4.4).

4 Representation Ensemble

We illustrate the inference process of RISE in Fig. 5. At the t-th decoding step, given the input $\mathbf{x}$, which is the combination of the input query $\mathcal{P}$ and the previously generated tokens $\mathbf{y}_{1:t-1}$, RISE aims to collaboratively determine the next token y_t, utilizing the central model and all assistant models with a consensus representation.

4.1 Representation Projection

Firstly, RISE asks the central model forward pass until the selected fusion layer l^C, and lets each assistant model A_i to forward pass until its corresponding aligned layer l^{A_i}. The hidden states are recorded as $\mathbf{H}^C$ and $\mathbf{H}^{A_i}$. Note that the subscripts of l^C and l^{A_i}, which denote the layer, are omitted.

Since different LLMs usually have different tokenizers, the input $\mathbf{x}$ usually is split into different sequences of tokens: $\mathbf{H}^C \in \mathbb{R}^{s^C \times d^C}$ and $\mathbf{H}^{A_i} \in \mathbb{R}^{s^{A_i} \times d^{A_i}}$. To tackle this problem, we utilize the token alignment strategy (§2.3) to compute a token alignment matrix $\mathbf{T}^{A_i \to C} \in \mathbb{R}^{s^C \times s^{A_i}}$. Through left multiplying this matrix, RISE aligns $\mathbf{H}^{A_i}$ with $\mathbf{H}^C$ in terms of token sequence.

Next, RISE aligns $\mathbf{H}^{A_i}$ with $\mathbf{H}^C$ in terms of latent space utilizing the derived hidden state projection $\mathcal{F}(\cdot)$. Formally:

$$\widetilde{\mathbf{H}}^{A_i} = \mathcal{F}(\mathbf{T}^{A_i \to C}\mathbf{H}^{A_i}). \tag{7}$$

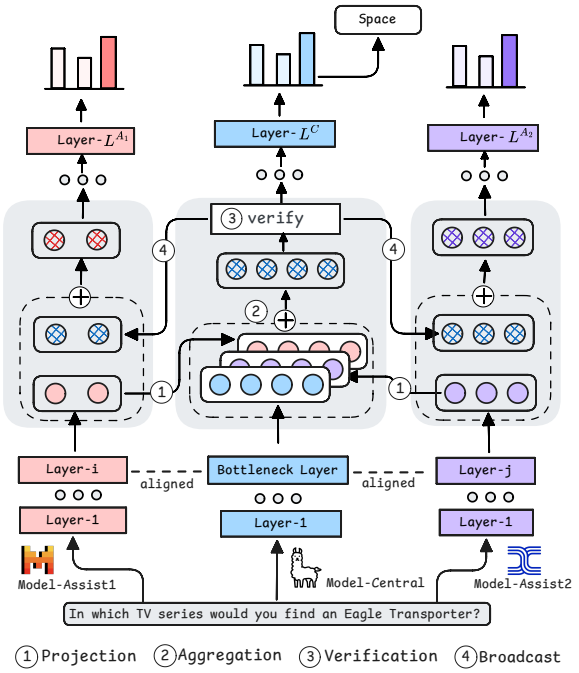


Figure 5: Illustration of Representation Ensemble.

Specially, when $\mathcal{F}$ is implemented with the rotation transformation, it is required to normalize the source hidden states $\mathbf{H}^{A_i}$ and de-normalize the transformed hidden states $\widetilde{\mathbf{H}}^{A_i}$.

4.2 Representation Aggregation

In this step, RISE aggregates the hidden states of all models in a manner of weighted averaging:

$$\hat{\mathbf{H}} = \alpha_0 \mathbf{H}^C + \sum_{i=1}^{N} \alpha_i \widetilde{\mathbf{H}}^{A_i}, \tag{8}$$

where α_0 and α_i denotes the ensemble weight of each model, which satisfies: $\alpha_0 + \sum_{i=1}^{N-1} \alpha_i = 1$.

4.3 Representation Verification

Due to inherent differences in the latent spaces of different LLMs, the projection process may introduce distortions, leading to projection errors. Therefore, RISE identifies the projected representation $\widetilde{\mathbf{H}}^{A_i}$ and sets an ensemble gate to determine whether to continue the forward pass based on the aggregated representation $\hat{\mathbf{H}}$. This work adopts two strategies of representation verification:

Representation Scale Verification. Our case study reveals that projection errors often manifest as out-of-range activation values. Therefore, we record the scale range of each dimension, *i.e.,* the minimum and maximum activation values. During inference, RISE verifies whether all dimensions of

$\widetilde{\mathbf{H}}^{A_i}$ fall within the per-dimension min–max range. If not, RISE sets $\alpha_i = 0$ at the current step of representation aggregation. We find that this simple strategy plays an important role in effective representation ensemble.

Decoding Plausible Constraint (*DPC*). Additionally, we incorporate the technology of decoding plausible constraint (Li et al., 2023b), to ensure plausible decoding. Concretely, in our work, we record the original decoding result and its probability of the central model as:

$$\begin{aligned} y^C &= \arg\max_y p(y|\mathbf{x}, \theta^C), \\ p^C &= \max_y p(y|\mathbf{x}, \theta^C). \end{aligned} \tag{9}$$

Then, RISE makes a trial to decode the aggregation result $\hat{\mathbf{H}}$, *i.e.,* performing the forwarding pass from the next layer of the fusion layer to the top layer, and records the generated token as $\hat{y}$. Finally, RISE judges whether $\hat{y}$ is a plausible decoding via calculating its acceptance with respect to the central model $\hat{p} = p(\hat{y}|\mathbf{x}, \theta^C)$. The final decoding token of this step is determined as:

$$y_t = \begin{cases} \hat{y}, & \hat{p} \geq p^C \times \eta, \\ y^C, & \text{otherwise}, \end{cases} \tag{10}$$

where $\eta \in [0, 1]$ is a hyperparameter. Tab. 2 presents the ablation study of DPC. We omit the ablation of scale verification since the model cannot produce valid outputs without it.

4.4 Representation Broadcast

In scenarios where we combine representation ensemble with decision ensemble (§4.5), the assistant models must also generate their own predictions. To enable this, we devise this step of representation broadcast, which aims to convey the aggregation results $\hat{\mathbf{H}}$ from the central model to each assistant model. The process of broadcast is similar to that of hidden state projection (§4.1). Formally:

$$\begin{aligned} \hat{\mathbf{H}}^{A_i} =& \beta \times \mathcal{F}^{C \to A_i}(\mathbf{T}^{C \to A_i} \hat{\mathbf{H}}^C) \\ &+ (1 - \beta) \times \mathbf{H}^{A_i}, \end{aligned} \tag{11}$$

where $\mathbf{T}^{C \to A_i}$ is the token alignment matrix indicating the correspondence across tokens from the central model to the assistant model. $\mathcal{F}^{C \to A_i}$ is the hidden state projection from the latent space of the central model to the one of the assistant model. β is set to 0.3 empirically. After broadcasting, each assistant model receives the aggregated representations of other models via the central model. We also present an ablation study in §6.4.

4.5 Other Variants

Multi-Stage RISE. The current RISE aggregates representations at a single fusion layer but can extend to multiple layers. To explore this, we introduce multi-stage RISE, which sequentially performs representation aggregation at the middle, high, and top layers.

RISE + Decision Ensemble. This variant aims to combine RISE and the method of decision ensemble, not only sharing the understanding, but also sharing the decisions across models. Specifically, RISE enhances the representation of each model utilizing the consensus representation. After that, we adopt existing technology of decisions ensemble (*e.g.,* DeePEn (Huang et al., 2024)) to fuse the probability distributions of all models and determine the next token.

5 Experiments

5.1 Experimental Setup

To make our experiments more convincing, we follow previous research (Huang et al., 2024) to select the benchmarks, evaluation metrics, and models.

Benchmarks. Our experiments involves six benchmarks categorized into:

- **Human Examination:** (1) MMLU (5-shot) (Hendrycks et al., 2021), covering 57 subjects that humans learn, and (2) ARC-C (0-shot) (Clark et al., 2018), collected from standardized natural science tests.
- **Reasoning Capabilities:** (1) GSM8K (Cobbe et al., 2021) (4-shot), a dataset of grade school math problems, and (2) PIQA (Bisk et al., 2020) (0-shot), a commonsense reasoning benchmark.
- **Knowledge Capacities:** (1) TriviaQA (5-shot) (Joshi et al., 2017) and (2) NQ (5-shot) (Kwiatkowski et al., 2019), which are two generative question answering benchmarks.

Evaluation. We follow the test scripts of OpenCompass leaderboard[1]. On multiple-choice tasks (MMLU, ARC-C, and PIQA), the option with the highest likelihood is selected to calculate accuracy.

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

Models	Examination		Reasoning		Knowledge	
	MMLU	ARC-C	GSM8K	PIQA	TriviaQA	NQ
Individual Models						
LLaMA2-13B	55.07	59.32	29.80	59.68	74.32	29.11
InternLM-20B	59.94	75.9	53.83	64.78	66.88	26.09
Skywork-13B	61.16	66.50	53.90	74.04	58.65	19.75
Tigerbot-13B	51.95	57.44	48.82	68.28	66.22	22.71
Mistral-7B	62.13	73.33	47.50	65.61	73.18	27.62
Yi-6B	63.25	73.33	37.91	76.15	59.02	18.98
Top-2 Ensemble						
★①BLENDER	63.85 (+0.60)	75.73 (-0.17)	54.89 (+0.99)	78.31 (+2.16)	74.10 (-0.22)	28.61 (-0.50)
★②DEEPEN	64.42 (+1.17)	77.52 (+1.62)	55.88 (+1.98)	78.87 (+2.72)	75.77 (+1.45)	30.64 (+1.53)
▲③**RISE-T**	63.98 (+0.73)	77.39 (+1.49)	56.25 (+2.35)	76.21 (+0.06)	75.53 (+1.21)	30.22 (+1.11)
▲④**RISE-H**	64.65 (+1.40)	77.49 (+1.59)	55.27 (+1.37)	78.09 (+1.94)	75.32 (+1.00)	30.55 (+1.44)
▲⑤**RISE-M**	64.19 (+0.94)	76.41 (+0.51)	55.50 (+1.60)	78.70 (+2.55)	74.97 (+0.65)	29.81 (+0.70)
■③+②	64.93 (+1.68)	78.21 (+2.31)	55.34 (+1.44)	79.09 (+2.94)	76.08 (+1.76)	30.22 (+1.11)
■④+②	64.90 (+1.65)	78.03 (+2.13)	55.12 (+1.22)	79.42 (+3.27)	76.30 (+1.98)	30.72 (+1.61)
■⑤+②	64.91 (+1.66)	77.69 (+1.79)	55.19 (+1.29)	79.53 (+3.38)	76.17 (+1.85)	30.69 (+1.58)
Top-4 Ensemble						
★①BLENDER	61.44 (-1.81)	71.03 (-4.87)	43.37 (-10.53)	71.16 (-4.99)	67.87 (-6.45)	24.18 (-4.93)
★②VOTING	64.91 (+1.66)	77.82 (+1.92)	63.76 (+9.86)	76.82 (+0.67)	–	–
★③MBR	–	–	58.83 (+4.93)	–	74.10 (-0.22)	30.11 (+1.00)
★④DEEPEN	64.90 (+1.65)	78.38 (+2.48)	54.13 (+0.23)	77.09 (+0.94)	75.28 (+0.96)	30.55 (+1.44)
▲⑤**RISE-T**	63.87 (+0.62)	78.21 (+2.31)	56.03 (+2.13)	77.42 (+1.27)	75.43 (+1.11)	30.06 (+0.95)
▲⑥**RISE-H**	64.92 (+1.67)	78.03 (+2.13)	56.71 (+2.81)	77.76 (+1.61)	75.15 (+0.83)	30.80 (+1.69)
▲⑦**RISE-M**	64.48 (+1.23)	77.61 (+1.71)	55.27 (+1.37)	77.81 (+1.66)	75.23 (+0.91)	30.08 (+0.97)
■⑤+④	65.35 (+2.10)	78.72 (+2.82)	54.81 (+0.91)	77.81 (+1.66)	75.77 (+1.45)	30.94 (+1.83)
■⑥+④	65.40 (+2.15)	79.23 (+3.33)	55.50 (+1.60)	78.31 (+2.16)	75.48 (+1.16)	31.94 (+2.83)
■⑦+④	65.68 (+2.43)	78.63 (+2.73)	55.19 (+1.29)	78.26 (+2.11)	75.30 (+0.98)	31.80 (+2.69)
■⑥+④+②	65.57 (+2.32)	79.18 (+3.28)	65.50 (+11.60)	77.26 (+1.11)	75.27 (+0.95)	32.16 (+3.05)

Table 1: Main results. The best individual model is highlighted in red, and the best ensemble method is highlighted in green. The top-4 models on each benchmark are underlined. '—' indicates that the method does not apply to the task. ★ indicates decision ensemble. ▲ indicates representation ensemble. ■ indicates their combination.

On free-form generation tasks (GSM8K, TriviaQA, and NQ), we calculate exact match accuracy.

Individual models. We follow Huang et al. (2024) to select six well-performing LLMs: LLaMA-2-13B (Touvron et al., 2023b), Mistral-7B-v0.1 (Jiang et al., 2023a), InternLM-20B (Team, 2023), Yi-6B (AI et al., 2025), Skywork-13B (Wei et al., 2023), and Tigerbot-13b-v2 (Chen et al., 2023). For each benchmark, we select the top-2 and top-4 models on the validation set for ensembling (Huang et al., 2024). The top-1 model is empirically chosen as the central model, which is analyzed in §B.3. The results of ensembling various number of models are shown in §B.4.

Hyperparameters. We search the optimal ensemble weights α_i on the development set for each task. In the setting of top-2-model ensemble, the search space is $\alpha_0 \in \{0.5, 0.6, 0.7, 0.8\}, \alpha_1 = 1 - \alpha_0$. In the setting of top-4-model ensemble, search space is defined as $\alpha_0 \in \{0.25, 0.3, 0.4, 0.5\}, \alpha_i = (1-\alpha_0)/(N-1)$. For each central–assistant model pair, we collect 100k anchor samples to derive the representation projection, as analyzed in §B.2.

Comparative methods. We compare RISE with (1) **DEEPEN** (Huang et al., 2024), which is a representative method of token-level decision ensemble, and (2) **BLENDER** (Jiang et al., 2023b), which comprises a reward model PAIRRANKER to score each response of LLMs and a fusion model GEN-FUSER to fuse candidate responses. As Huang et al. (2024) have evidenced the poor generalization of BLENDER on unseen distribution, we directly report the results of BLENDER from their paper. In the setting top-4 ensemble, we introduce two additional ensemble methods: (3) **VOTING**,

selecting the choice favored by most models, and (4) **MBR** (Freitag et al., 2023), which selects the answer with the highest textual similarity to other candidate answers.

5.2 Main Results

Tab. 1 summarizes the results and suggests three observations:

(1) RISE consistently improves over individual models. All three variants of RISE yield positive gains over the best single model on all six benchmarks, demonstrating that cross-model representation sharing is effective. Among the variants, RISE-H tends to achieve the strongest average performance, suggesting that higher-layer representations are particularly informative.

(2) RISE is competitive with the decision ensemble baseline. Compared with the strong decision ensemble DEEPEN, in the top-2 setting RISE can outperform on MMLU and GSM8K, but slightly trails on other 4 out of 6 tasks. In the top-4 setting, RISE can surpass DEEPEN on 5 out of 6 tasks, while DEEPEN remains better on ARC-C. These mixed outcomes highlight the complementary nature of representation and decision ensembling.

(3) Combining representation and decision ensembling yields the best overall results. Across both top-2 and top-4 settings, the combined methods (RISE + DEEPEN) attain the strongest scores on most settings, confirming that the two paradigms are complementary. For clarity, we define the *collaboration gain* as the relative improvement of an ensemble over the best single model. In top-2-model ensemble, DeePEn achieves a +3% relative improvement on average, while the combined method achieves +3.43%. In top-4-model ensemble, DeePEn achieves a +2.32% relative improvement on average, while the combined method achieves up to +4%.

5.3 Results of Different Implementations of Representation Projection

We evaluate representation projections obtained from different implementations in the top-2-model ensemble setting. To clearly illustrate the effectiveness of each projection method, we fix the ensemble weights to $(0.5, 0.5)$.

As shown in Tab. 2, the poor performance of gradient descent suggests that this method struggles with aligning latent spaces across different LLMs. We hypothesize that this is because gradient descent prioritizes capturing point-to-point correspondences rather than establishing a global alignment across feature spaces.

Models	MMLU	TriviaQA	NQ
Individual Models			
Rank-1st	63.25	74.32	29.11
Rank-2nd	62.13	73.18	27.95
***Top**-Layer Representation Ensemble*			
GradientDescent	63.25	73.88	28.86
LeastSquare	63.57	74.27	29.20
Rotation	**63.60**	**74.50**	**29.31**
Rotation *w. DPC*	63.61	75.32	30.22
***High**-Layer Representation Ensemble*			
GradientDescent	63.10	73.87	28.34
LeastSquare	**64.67**	74.78	29.31
Rotation	64.65	**74.85**	**29.42**
Rotation *w. DPC*	64.61	75.28	30.53
***Middle**-Layer Representation Ensemble*			
GradientDescent	63.25	74.15	29.11
LeastSquare	**64.17**	74.78	29.17
Rotation	64.13	**74.87**	**29.28**
Rotation *w. DPC*	64.13	74.97	29.5
***Multi-Stage** Representation Ensemble*			
GradientDescent	63.11	73.68	27.76
LeastSquare	63.45	75.28	27.87
Rotation	63.88	**75.32**	**29.53**
Rotation *w. DPC*	63.94	75.48	30.19

Table 2: Results of different implementations of representation projection.

We further observe that the rotation transformation outperforms the least squares approach, suggesting a rotational relationship between the latent spaces of different LLMs. Furthermore, in generative question-answering tasks (TriviaQA and NQ), the decoding plausible constraint (DPC) strategy effectively improves RISE's performance by mitigating the negative impact of projection errors.

Finally, multi-stage RISE does not yield significant gains over the single-stage version. We attribute this to more frequent communication, which may accumulate projection errors and thus reduce potential benefits.

6 Analysis

6.1 Quadrant of Ensemble Effects

Definition. To better understand why RISE works and how it complements decision ensemble, we

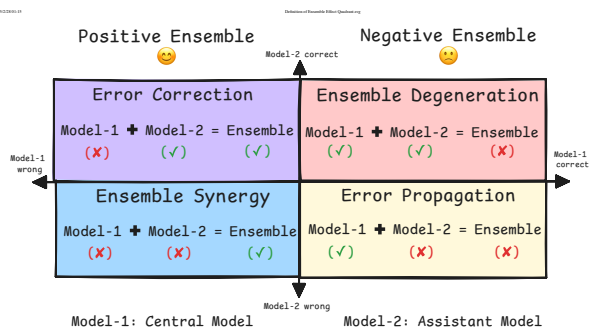


Figure 6: Definition of Ensemble Effect Quadrant.

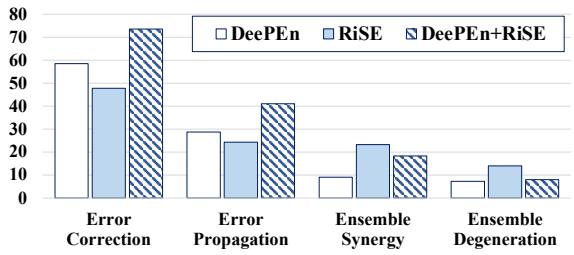


Figure 7: Results of EEQ on the development set of TriviaQA (1900 samples). Y-axis indicates the number of the corresponding samples.

devise an evaluation framework called the *Ensemble Effect Quadrant* (EEQ), which groups the test samples into four categories according to their behavior under ensemble learning, as illustrated in Fig. 6. Ensemble synergy reflects how the two models together outperform either one alone, whereas ensemble degeneration reveals the interference that collaboration may introduce.

Results. The EEQ statistics for RISE and the decision ensemble baseline (DEEPEN) are shown in Fig. 7. **(1) RISE produces more *Ensemble Synergy* cases, whereas DEEPEN achieves more *Error Correction* cases**. This difference highlights their complementary mechanisms: DEEPEN aggregates model outputs *after* each model has produced a final prediction, whereas RISE shares hidden representations *before* decoding. This pre-decoding information exchange enables the central model to form richer internal representations and to solve problems that neither individual model could solve alone. **(2) The combined approach inherits both strengths**, showing high counts in both Ensemble Synergy and Error Correction, which explains why it achieves the best overall collaboration gains.

6.2 Effect of Ensemble Weights

Ensemble weights significantly impact ensemble learning (Shahhosseini et al., 2020). We analyze

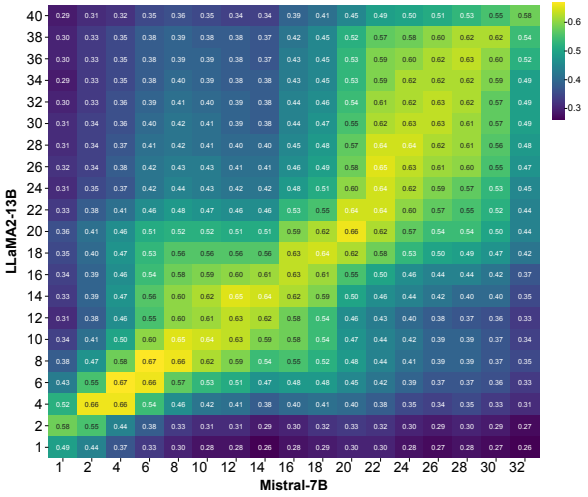


Figure 8: Illustration of layer alignment across LLaMA2-13B and Mistral-7B.

α_0	0.5	0.6	0.7	0.8	*Avg.*	(%)
ACC	74.89	75.21	75.26	75.21	–	–
#Correction	51	51	48	41	**48**	(44%)
#Synergy	24	25	23	21	**23**	(21%)
#Error Prop	31	26	23	17	**24**	(22%)
#Degenerate	16	16	13	11	**14**	(13%)

Table 3: Effect of different values of ensemble weight of the central model on the TriviaQA development set. The ACC row reports accuracy across different ensemble weight values. The 2nd row to the 5th row indicate the number of samples with different ensemble effect. Rows 2 to 5 show the number of samples exhibiting different ensemble effects. The last column (%) represents the proportion of each category relative to the total number of affected samples.

their effect using EEQ, with results shown in Tab. 3. As the central model's weight (α_0) decreases from 0.8 to 0.5, the assistant model's influence grows, initially improving performance before causing a decline. Further analysis reveals that all four phenomena increase as α_0 decreases, suggesting that while a stronger assistant model enhances error correction and synergy, it also amplifies error propagation and degeneration, leading to performance fluctuations.

6.3 Latent Space Alignment

Visualization of Layer Alignment across LLMs. This section illustrates the layer alignment relationship, which is measured by neighbor consistency (§2.2). As shown in Fig. 8, the intermediate layers of different LLMs exhibit high similarity, which lays foundation for representation sharing. And the

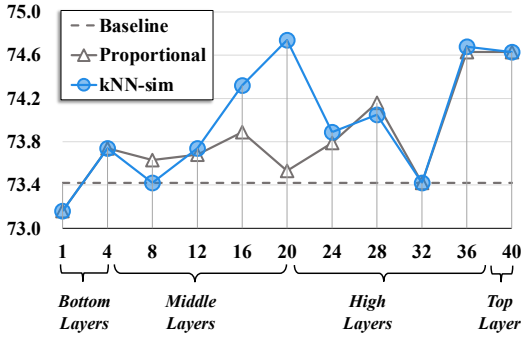


Figure 9: Results of using different layers in the central model (LLaMA2-13B) as fusion layers on the TriviaQA development set. "kNN-sim" aligns the assistant model's layer using a neighbor-consistency-based measure, while "Proportional" selects it proportionally based on the layer ratio.

Methods	MMLU		PIQA		NQ	
	ACC	Δ	ACC	Δ	ACC	Δ
Rank-1st	63.25	–	76.15	–	29.11	–
①DeePEn	64.90	+1.65	77.09	+0.94	30.55	+1.44
②RISE	64.92	+1.67	77.76	+1.61	30.80	+1.69
①+②	65.55	+2.30	78.31	+2.16	31.94	+2.83
w/o. ***Broadcast***	65.40	+2.15	77.76	+1.61	31.50	+2.39

Table 4: Ablation of broadcast in top-4-model ensemble setting. Δ indicates the improvement over 'Rank-1st'.

alignment pattern deviates from a strict diagonal trend, due to differences in training paradigms.

Fusion Layer Selection. This work selects the "transition layer" as the fusion layer. To understand the effectiveness of this choice, we first illustrate the transition layer in Fig. 4. Next, we test RISE with using different layer as the fusion layer, as shown in Fig. 9. The results demonstrate that using the last layer of each stage (*i.e.,* transition layers) as the fusion layer performs significantly better. Besides, we introduce a baseline method of layer alignment, which aligns layers across LLMs proportional to model layer number. The results show the effectiveness of our layer alignment method based on the representational similarity measure.

6.4 Ablation of Broadcast Mechanism.

To understand the effect of broadcast mechanism (§4.4), we conduct an ablation study by removing the broadcast when combining representation ensemble (RISE-H) and decision ensemble (DeePEn) in top-4-ensemble setting. As shown in Tab. 4, without broadcasting the aggregated representation, performance degrades significantly. These results illustrate that effective collaboration requires all members to hold consensus representations.

7 Further Investigation of Ensemble Learning

This section aims to place ensemble learning in a broader context by addressing the questions: (**1**) Which is better: ensembling small LLMs or training a larger LLM? (**2**) How does ensemble learning compare with model merging? (**3**) How does ensemble learning compare with model composition?

7.1 Ensemble of Small LLMs vs. Single Larger LLM

Although ensemble learning is well established in machine learning (Hansen and Salamon, 1990; Zhou, 2012), it increases *inference* cost. This practical trade-off raises the question: *Can an ensemble of two small LLMs match a single larger LLM under a comparable inference budget?* We examine Qwen3-8B and Qwen3-14B as the small models and Qwen3-32B as the larger model (Yang et al., 2025). As shown in Tab. 5 and Fig. 10, we observe:

(1) Scaling parameters does not guarantee uniform gains across tasks. While scaling laws suggest that larger models trained with more compute usually achieve higher accuracy and lower error (Kaplan et al., 2020; Hoffmann et al., 2022), we see substantial per-benchmark variability: Qwen3-32B *underperforms* Qwen3-14B on TriviaQA, and Qwen3-14B underperforms Qwen3-8B on PIQA.

(2) Ensembling small LLMs yields performance competitive with a larger model at similar inference cost. On average, ensembling Qwen3-14B with Qwen3-8B improves Qwen3-14B by **+2.19** points, narrowing the gap to Qwen3-32B from -2.57 to -0.39. The ensemble *matches* the 32B model on ARC-C and GSM8K, is close on PIQA, and *surpasses* it on NQ while further strengthening Qwen3-14B's advantage on TriviaQA.

(3) Deployment efficiency. Unlike parameter scaling, ensembling imposes no extra pretraining and high-quality data, and only adds inference-time overhead. *This advantage is especially valuable in scenarios where acquiring extra high-quality training data is impractical, yet one still seeks to improve model performance.*

Models	Examination		Reasoning		Knowledge	
	MMLU	ARC-C	GSM8K	PIQA	TriviaQA	NQ
Individual LLMs						
Qwen3-32B	84.18	92.74	93.40	89.41	55.90	22.66
Qwen3-14B	78.74	89.15	89.64	84.53	59.15	21.75
Qwen3-8B	73.92	88.55	88.02	86.02	53.33	18.84
Model Composition of Qwen3-14B and Qwen3-8B						
CALM	78.58 (-0.16)	90.22 (+1.07)	78.22 (-11.32)	83.97 (-2.05)	51.50 (-7.65)	19.11 (-2.64)
Ensemble Learning of Qwen3-14B and Qwen3-8B						
①**DeePEn**	78.56 (-0.18)	92.05 (+2.90)	91.43 (+1.89)	88.24 (+2.22)	59.83 (+0.68)	22.99 (+1.24)
②**RISE**	79.30 (+0.56)	92.22 (+3.07)	92.57 (+3.03)	87.74 (+1.72)	59.77 (+0.62)	23.46 (+1.71)
①+②	79.37 (+0.63)	92.22 (+3.07)	92.34 (+2.80)	88.41 (+2.39)	60.50 (+1.35)	23.13 (+1.38)

Table 5: Ensemble learning of small LLMs (Qwen3-8B and Qwen3-14B) vs. one larger LLM (Qwen3-32B). Numbers in parentheses show the absolute improvement achieved by ensemble learning over Qwen3-14B. Red indicates the best results of individual models. Green indicate the best results of ensemble learning. CALM (Bansal et al., 2024) is a training-based model composition method.

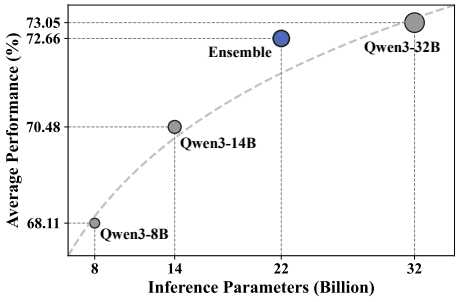


Figure 10: Ensemble of small LLMs (Qwen3-8B and Qwen3-14B) vs. single larger LLM (Qwen3-32B). "Ensemble" indicates the results of combining RISE and DeePEn. For clarity, we omit the result of DeePEn (72.18) and RISE (72.30).

7.2 Ensemble Learning vs. Model Merging

Model merging fuses the *parameters* of several fine-tuned LLMs, typically of the same architecture and often connected through mode connectivity (Garipov et al., 2018), into a single model (Li et al., 2023a; Khan et al., 2025). Ensemble learning, in contrast, aggregates *predictions* from multiple models (Wortsman et al., 2022). While ensembling pushes the performance frontier and improves generalization by integrating even heterogeneous models, model merging aims to combine different capabilities into one deployable model, providing better inference efficiency (Lu et al., 2024).

Setup. Following prior work (Yu et al., 2024; Li et al., 2025; Yadav et al., 2023), we fine-tune LLaMA3-8B separately on the NuminaMath-CoT corpus (LI et al., 2024) and the CodeAlpaca dataset (Chaudhary, 2023), obtaining **Math-LLaMA3-8B** and **Code-LLaMA3-8B**. We evaluate both paradigms on GSM8K (Cobbe et al., 2021) and HumanEval (Chen et al., 2021). More details of the fine-tuning are provided in §A.1.

Merging methods. We compare three typical approaches using MERGEKIT (Goddard et al., 2024): (1) Weight Averaging (Wortsman et al., 2022), (2) Ties (Yadav et al., 2023), locating and resolving the parameter interference heuristically, and (3) Dare (Yu et al., 2024), sparsifying fine-tuning deltas to reduce interference.

Results. Table 6 shows that on GSM8K, merging methods suffer a noticeable drop (e.g., Weight Averaging 61.41, –3.41 vs. Math-LLaMA3-8B), whereas ensemble learning with DeePEn+RISE reaches 65.2 (+0.38). On HumanEval, both merging and ensembling outperform the best single model (35.98), with DeePEn+RISE achieving 46.95 (+10.97). Overall, consistent with earlier studies, **ensemble learning yields higher accuracy**, while **model merging offers a single deployable model and thus better inference efficiency**. All results in this table are obtained using the base version of LLaMA3-8B. Additional results using instruction-tuned models (LLaMA3-8B-Instruct) are reported in §B.5 and exhibit the same relative trends.

7.3 Ensemble vs. Model Composition

Model composition combines multiple models into a single architecture and trains it end-to-

Models	GSM8K	HumanEval
Individual LLMs		
LLama3-8B	22.06	6.10
Math-LLaMA3-8B	64.82	31.71
Code-LLaMA3-8B	51.86	35.98
Model Merging		
Weight Averaging	61.41 (-3.41)	44.50 (+8.52)
Ties	55.28 (-9.54)	42.07 (+6.09)
Dare	59.59 (-5.23)	23.17 (-12.81)
Ensemble Learning		
①**DeePEn**	64.59 (-0.23)	45.12 (+9.14)
②**RISE**	64.52 (-0.30)	45.73 (+9.75)
DeePEn + RISE	65.20 (+0.38)	46.95 (+10.97)

Table 6: Results of LLaMA3-8B (Base) and its fine-tuned versions on math and code datasets. Numbers in parentheses indicate the absolute improvement over the best *individual* model in the corresponding column.

end (Bansal et al., 2024; Fu et al., 2024). In contrast, RISE applies Procrustes analysis to obtain a *closed-form* transformation across latent spaces, avoiding additional end-to-end training.

Implementation. We re-implement CALM to compose Qwen3-14B and -8B via a cross-attention (see Bansal et al. (2024) for details). Using the same MiniPile corpus as RISE, we tune lr ($1e{-4}$), batch size (256), and weight decay (0.01) on a held-out set, and train for 2k steps.

Results. Table 5 shows that CALM provides no improvement on most benchmarks, likely because end-to-end training overfits the training data and limits generalization. We conjecture that CALM may suit cases where train and test distributions closely match (Bansal et al., 2024; Fu et al., 2024). By contrast, RISE uses about 5% of CALM's training data to yet generalizes more robustly. We plan to explore methods for improving the generalization ability of model composition in future work.

8 Related Works

Ensemble learning (Sagi and Rokach, 2018; Jiang et al., 2023b; Lu et al., 2024) exploits the complementary strengths of multiple models. Work on LLMs mainly centers on *decision sharing* at three granularities: (i) **Instance-level**, which aggregates complete answers from each LLM or uses a router to pick the most suitable model (Jiang et al., 2023b; Lu et al., 2023; Wang et al., 2024); (ii) **Token-level**, which merges next-token probability distributions to correct errors during generation (Huang et al., 2024; Xu et al., 2024; Yao et al., 2025; Mavromatis et al., 2024); and (iii) **Span-level**, which aggregates predicted spans for a middle ground between the two (Liu et al., 2024; Xu et al., 2025). Unlike these decision-based approaches, our method also shares *representations* across LLMs, enabling deeper information fusion and stronger collaboration.

Our work builds on extensive studies of representation analysis and alignment, falling into four perspectives: (1) representation similarity across language models (Wu et al., 2020; Huh et al., 2024); (2) the semantics of different layers' representations, capturing linguistic (Liu et al., 2019) and factual knowledge (Sajjad et al., 2023, 2022); (3) the stages of inference (Sun et al., 2025; Lad et al., 2024), and (4) representation alignment (Alvarez-Melis and Jaakkola, 2018; Grave et al., 2019; Artetxe et al., 2019; Lample et al., 2018). These studies greatly motivate our exploration of representation sharing in ensemble settings.

9 Conclusion

This work proposes a novel ensemble framework, Representation Ensemble RISE, aiming to sharing representation across LLMs and achieving deep collaboration. To address the obstacles of layer misalignment and latent space mismatch, this work invents a method of representation alignment based on representational similarity measure and latent space transformation. Extensive experiments show the effectiveness of RISE and its complementary strengths with decision ensemble. Further results of combining decision ensemble and representation ensemble show that this combination significantly pushes the frontier of model collaboration.

10 Limitations

Our approach shares some common limitations of ensemble methods: (1) it inherits the relatively high inference cost. Our future work will explore more adaptive ensembling to mitigate this overhead, (2) it requires task-specific tuning of the ensemble weights. In addition, our method relies on representational alignment between models; as this alignment degrades, the benefit of representation sharing correspondingly diminishes.

11 Acknowledgments

We are grateful to the anonymous reviewers and the Action Editor, Jacob Andreas, for their thorough

reviews and valuable guidance, which greatly improved the quality of this work. This research/project is supported by the National Research Foundation, Singapore under its National Large Language Models Funding Initiative, (AISG Award No: AISG-NMLP-2024-002), the National Natural Science Foundation of China (NSFC) (grant 6252200908, 62276078, U22B2059), the Fundamental Research Funds for the Central Universities (XNJKKGYDJ2024013) and the State Key Laboratory of Micro-Spacecraft Rapid Design and Intelligent Cluster (MS01240122). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

References

01. AI, :, Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Guoyin Wang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yanpeng Li, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. 2025. Yi: Open foundation models by 01.ai.

David Alvarez-Melis and Tommi Jaakkola. 2018. Gromov-Wasserstein alignment of word embedding spaces. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1881–1890, Brussels, Belgium. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2019. An effective approach to unsupervised machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 194–203, Florence, Italy. Association for Computational Linguistics.

Rachit Bansal, Bidisha Samanta, Siddharth Dalmia, Nitish Gupta, Sriram Ganapathy, Abhishek Bapna, Prateek Jain, and Partha Talukdar. 2024. LLM augmented LLMs: Expanding capabilities through composition. In *The Twelfth International Conference on Learning Representations*.

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.

Sahil Chaudhary. 2023. Code alpaca: An instruction-following llama model for code generation. https://github.com/sahil280114/codealpaca.

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.

Ye Chen, Wei Cai, Liangmin Wu, Xiaowei Li, Zhanxuan Xin, and Cong Fu. 2023. Tigerbot: An open multilingual multitask llm.

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.

Markus Freitag, Behrooz Ghorbani, and Patrick Fernandes. 2023. Epsilon sampling rocks: Investigating sampling strategies for minimum Bayes

risk decoding for machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9198–9209, Singapore. Association for Computational Linguistics.

Chengpeng Fu, Xiaocheng Feng, Yichong Huang, Wenshuai Huo, Baohang Li, Hui Wang, Bin Qin, and Ting Liu. 2024. Relay decoding: Concatenating large language models for machine translation.

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry Vetrov, and Andrew Gordon Wilson. 2018. Loss surfaces, mode connectivity, and fast ensembling of dnns. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 8803–8812, Red Hook, NY, USA. Curran Associates Inc.

Charles Goddard, Shamane Siriwardhana, Malikeh Ehghaghi, Luke Meyers, Vladimir Karpukhin, Brian Benedict, Mark McQuade, and Jacob Solawetz. 2024. Arcee's MergeKit: A toolkit for merging large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 477–485, Miami, Florida, US. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez,

Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Sub-

ramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The llama 3 herd of models.

Edouard Grave, Armand Joulin, and Quentin Berthet. 2019. Unsupervised alignment of embeddings with wasserstein procrustes. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 1880–1890. PMLR.

Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Dan Roberts. 2025. The unreasonable ineffectiveness of the deeper layers. In *The Thirteenth International Conference on Learning Representations*.

L.K. Hansen and P. Salamon. 1990. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. 2024. Training large language models to reason in a continuous latent space.

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*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack W. Rae, and Laurent Sifre. 2022. Training compute-optimal large language models. NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Yichong Huang, Xiaocheng Feng, Baohang Li, Yang Xiang, Hui Wang, Ting Liu, and Bing Qin. 2024. Ensemble learning for heterogeneous large language models with deep parallel collaboration. In *Advances in Neural Information Processing Systems*, volume 37, pages 119838–119860. Curran Associates, Inc.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. 2024. Position: The platonic representation hypothesis. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 20617–20642. PMLR.

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.

Jean Kaddour. 2023. The minipile challenge for data-efficient language models.

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.

Arham Khan, Todd Nief, Nathaniel Hudson, Mansi Sakarvadia, Daniel Grzenda, Aswathy Ajith, Jordan Pettyjohn, Kyle Chard, and Ian Foster. 2025. Deep model merging: The sister of neural network interpretability – a survey.

Max Klabunde, Tobias Schumacher, Markus Strohmaier, and Florian Lemmerich. 2024. Similarity of neural network models: A survey of functional and representational measures.

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.

Vedang Lad, Wes Gurnee, and Max Tegmark. 2024. The remarkable robustness of LLMs: Stages of inference? In *ICML 2024 Workshop on Mechanistic Interpretability*.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations*.

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

Weishi Li, Yong Peng, Miao Zhang, Liang Ding, Han Hu, and Li Shen. 2023a. Deep model fusion: A survey.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023b. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12286–12312, Toronto, Canada. Association for Computational Linguistics.

Zijian Li, Xiaocheng Feng, Huixin Liu, Yichong Huang, Ting Liu, and Bing Qin. 2025. From: Frobenius norm-based data-free adaptive model merging. In *The 2025 Conference on Empirical Methods in Natural Language Processing*.

Cong Liu, Xiaojun Quan, Yan Pan, Liang Lin, Weigang Wu, and Xu Chen. 2024. Cool-fusion: Fuse large language models without training.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.

Jinliang Lu, Ziliang Pang, Min Xiao, Yaochen Zhu, Rui Xia, and Jiajun Zhang. 2024. Merge, ensemble, and cooperate! a survey on collaborative strategies in the era of large language models.

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.

Valentino Maiorca, Luca Moschella, Antonio Norelli, Marco Fumero, Francesco Locatello, and Emanuele Rodolà. 2023. Latent space translation via semantic alignment. In *Advances in Neural Information Processing Systems*, volume 36, pages 55394–55414. Curran Associates, Inc.

Costas Mavromatis, Petros Karypis, and George Karypis. 2024. Pack of llms: Model fusion at test-time via perplexity optimization.

OpenAI. 2023. Gpt-4 technical report.

Omer Sagi and Lior Rokach. 2018. Ensemble learning: A survey. *WIREs Data Mining and Knowledge Discovery*, 8(4):e1249.

Hassan Sajjad, Fahim Dalvi, Nadir Durrani, and Preslav Nakov. 2023. On the effect of dropping layers of pre-trained transformer models. *Comput. Speech Lang.*, 77(C).

Hassan Sajjad, Nadir Durrani, Fahim Dalvi, Firoj Alam, Abdul Khan, and Jia Xu. 2022. Analyzing encoded concepts in transformer language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3082–3101, Seattle, United States. Association for Computational Linguistics.

Peter H. Schönemann. 1966. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10.

Mohsen Shahhosseini, Guiping Hu, and Hieu Pham. 2020. Optimizing ensemble weights and hyperparameters of machine learning models for regression problems.

Qi Sun, Marc Pickett, Aakash Kumar Nain, and Llion Jones. 2025. Transformer layers as painters.

InternLM Team. 2023. Internlm: A multilingual language model with progressively enhanced capabilities. `https://github.com/InternLM/InternLM-techreport`.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, and et al. 2023b. Llama 2: Open foundation and fine-tuned chat models.

Fanqi Wan, Xinting Huang, Deng Cai, Xiaojun Quan, Wei Bi, and Shuming Shi. 2024. Knowledge fusion of large language models. In *The Twelfth International Conference on Learning Representations*.

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*.

Tianwen Wei, Liang Zhao, Lichang Zhang, Bo Zhu, Lijie Wang, and at el. 2023. Skywork: A more open bilingual foundation model.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 23965–23998. PMLR.

John Wu, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 2020. Similarity analysis of contextual word representation models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4638–4655, Online. Association for Computational Linguistics.

Yangyifan Xu, Jianghao Chen, Junhong Wu, and Jiajun Zhang. 2025. Hit the sweet spot! span-level ensemble for large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 8314–8325, Abu Dhabi, UAE. Association for Computational Linguistics.

Yangyifan Xu, Jinliang Lu, and Jiajun Zhang. 2024. Bridging the gap between different vocabularies for llm ensemble. In *North American Chapter of the Association for Computational Linguistics*.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. 2023. Ties-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems*, volume 36, pages 7093–7115. Curran Associates, Inc.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei

Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. 2025. Qwen3 technical report.

Yuxuan Yao, Han Wu, Mingyang LIU, Sichun Luo, Xiongwei Han, Jie Liu, Zhijiang Guo, and Linqi Song. 2025. Determine-then-ensemble: Necessity of top-k union for large language model ensembling. In *The Thirteenth International Conference on Learning Representations*.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2024. Language models are super mario: absorbing abilities from homologous models as a free lunch. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

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.

Zhi-Hua Zhou. 2012. *Ensemble methods: foundations and algorithms*. CRC press.

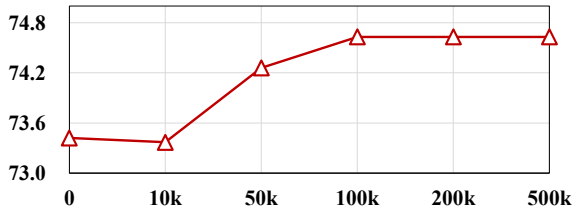


Figure 11: Effect of using different number of anchor samples to induct hidden state projection.

Models	GSM8K	HumanEval
Individual LLMs		
LLama3-8B-Instruct	75.44	50.00
Math-LLaMA3-8B-Instruct	71.80	43.07
Code-LLaMA3-8B-Instruct	69.22	46.34
Model Merging		
Weight Averaging	71.95 (+0.15)	49.39 (+3.05)
Ties	70.20 (–1.60)	48.78 (+2.44)
Dare	72.02 (+0.22)	50.61 (+4.27)
Ensemble Learning		
DeePEn	72.10 (+0.30)	51.22 (+4.88)
RISE	72.18 (+0.38)	51.22 (+4.88)
DeePEn+RISE	72.55 (+0.75)	51.82 (+5.48)

Table 7: Results of LLaMA3-8B-Instruct and its fine-tuned versions on math and code datasets. Numbers in parentheses indicate the absolute improvement over the best *individual* fine-tuned model on each task.

A Experimental Details

A.1 Fine-tuning of LLaMA3-8B

In §7.2 and §B.5, we compare ensemble learning with model fusion on fine-tuned variants of LLaMA3-8B. Below we provide the detailed fine-tuning configurations used in our experiments.

We fine-tune both the base and instruction-tuned variants of LLaMA3-8B using parameter-efficient fine-tuning with LoRA. Unless otherwise specified, all experiments share the same training settings.

Specifically, we adopt the AdamW optimizer with a batch size of 64 and a warmup ratio of 0.1. For the learning rate, we set it to 1e-4 for the base LLaMA3-8B model and 1e-5 for the LLaMA3-8B-instruct model, which we found to be stable across tasks. All models are fine-tuned using LoRA with rank $r = 16$, while keeping the remaining model parameters frozen.

B Complementary Results

B.1 Representation Similarity Analysis

In this section, we report the representation similarity measured with different values of k (Eq. 1). Concretely, given two LLMs θ_A with L_A layers and θ_B with L_B layers, we first calculate their layer alignment matrix $\mathbf{M}^{(A,B)} \in \mathbb{R}^{L_A \times L_B}$, which is illustrated in Section 3.1. Then, we perform max-pooling operations for each row of $\mathbf{M}^{(A,B)}$ and average these layer-wise similarities to obtain the final representation similarity. As the results shown in Tab. 8, we observe higher alignment across LLMs when $k = 10$.

B.2 Effect of Anchor Words for Hidden States Projection Derivation

We evaluate RISE-T with representation projections derived with varying number of anchors in top-2-model ensemble on TriviaQA development set. The ensemble weight is fixed to $\{\alpha_0 = 0.5, \alpha_1 = 0.5\}$. As shown in Fig. 11, using fewer anchors (e.g., 10k) degrades performance, whereas 100k anchors suffice to derive a well-performing hidden state projection.

B.3 Choice of Central Model

In RISE, one of the collaborating LLMs is designated as the central model, and the hidden states of other models are projected into the representation space of this central model. Due to architectural and training differences across LLMs, representation projection is inherently imperfect, leading to asymmetric information transmission.

As a result, the central model tends to play a dominant role during representation aggregation, making the choice of the central model an important factor for ensemble performance. To analyze this effect, we evaluate RISE under different central model selections.

As shown in Tab. 9, consistently across multiple benchmarks, selecting the model with the best validation performance as the central model leads to better ensemble performance. Based on this observation, we adopt and recommend the strategy of choosing the strongest validation model as the central model in all experiments.

B.4 Ensembling Various Number of LLMs

We report the performance from top-2-model ensembling to top-6-model ensembling. As shown in Fig. 12, the performance of DEEPEN degrades significantly when introducing more poor-performing model as suffering from serious interference. For-

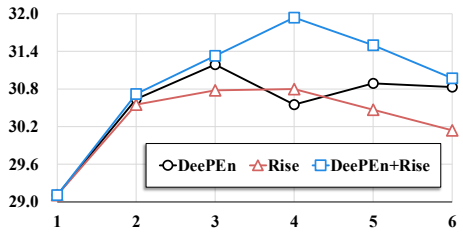


Figure 12: Effect of ensembling varying number of models.

tunately, the combination of DEEPEN and RISE effectively alleviates this issue by sharing representation for all models.

B.5 Model Fusion vs. Ensemble for Instruction Models

To address concerns regarding the relatively low GSM8K performance of base LLaMA models, we additionally conduct experiments using instruction-tuned variants of LLaMA3-8B (LLaMA3-8B-Instruct). Specifically, we fine-tune LLaMA3-8B-Instruct to obtain Math-LLaMA3-8B-Instruct and Code-LLaMA3-8B-Instruct, and compare model merging and ensemble learning on GSM8K and HumanEval.

Tab. 7 reports the results. Compared to base models, instruction-tuned models substantially improve absolute performance across both benchmarks. We also observe that the task-specific fine-tuned variants perform slightly worse than the vanilla LLaMA3-8B-Instruct model. A plausible explanation is that LLaMA3-8B-Instruct has already been extensively post-trained on math and code data (Grattafiori et al., 2024), leaving limited room for further task-specific fine-tuning.

Importantly, the relative trends remain consistent with our main findings in §7.2: (1) model merging methods exhibit less consistent improvements, and (2) ensemble learning achieves the strongest performance across both benchmarks.

These results indicate that our conclusions are not tied to weaker base models, and that the proposed representation ensemble remains effective when applied to stronger instruction-tuned LLMs.

Model Pairs	k=10	k=50	k=100	k=500	k=1000
Qwen3-14B with Qwen3-8B	**0.51**	0.47	0.33	0.39	0.45
LLaMA2-13B with Mistral-7B	**0.58**	0.57	0.55	0.54	0.55

Table 8: Cross-LLM Representation similarity measured by the Nearest-Neighbor overlap with different k values.

Models	MMLU		GSM8K		PIQA		TriviaQA	
	INDIV	RISE	INDIV	RISE	INDIV	RISE	INDIV	RISE
Qwen3-14B	78.74	79.30 (+0.56)	89.54	92.57 (+3.03)	84.53	87.19 (+2.66)	59.15	59.77 (+0.62)
Qwen3-8B	73.92	76.93 (+3.01)	88.02	90.14 (+2.12)	86.02	87.74 (+1.72)	53.33	55.70 (+2.37)

Table 9: Effect of selecting different central models for RISE. INDIV denotes the performance of the individual model, while RISE reports the ensemble performance when the model in each row is chosen as the central model. Highlighted cells indicate the stronger individual model and the corresponding better-performing central model across benchmarks.