Harnessing Consistency for Robust Test-Time LLM Ensemble

Zhichen Zeng[1]*, **Qi Yu**[1]*, **Xiao Lin**[1], **Ruizhong Qiu**[1], **Xuying Ning**[1], **Tianxin Wei**[1], **Yuchen Yan**[1], **Jingrui He**[1], **Hanghang Tong**[1],

[1]University of Illinois Urbana-Champaign
{zhichenz,qiyu6,htong}@illinois.edu

Abstract

Different large language models (LLMs) exhibit diverse strengths and weaknesses, and LLM ensemble serves as a promising approach to integrate their complementary capabilities. Despite substantial progress in improving ensemble quality, limited attention has been paid to the robustness of ensembles against potential erroneous signals, which often arise from heterogeneous tokenization schemes and varying model expertise. Our analysis shows that ensemble failures typically arise from both the token level and the model level: the former reflects severe disagreement in token predictions, while the latter involves low confidence and pronounced disparities among models. In light of this, we propose CoRE, a plug-and-play technique that harnesses model consistency for robust LLM ensemble, which can be seamlessly integrated with diverse ensemble methods. *Token-level consistency* captures fine-grained disagreements by applying a low-pass filter to downweight uncertain tokens with high inconsistency, often due to token misalignment, thereby improving robustness at a granular level. *Model-level consistency* models global agreement by promoting model outputs with high self-confidence and minimal divergence from others, enhancing robustness at a coarser level. Extensive experiments across diverse benchmarks, model combinations, and ensemble strategies demonstrate that CoRE consistently improves ensemble performance and robustness. Our code is available at https://github.com/zhichenz98/CoRE-EACL26.

1 Introduction

Large language models (LLMs) (Brown et al., 2020; Team et al., 2023; Touvron et al., 2023; Achiam et al., 2023; Guo et al., 2025) have demonstrated remarkable performance in natural language processing tasks. Due to the difference in model architectures, training algorithms, and datasets, different LLMs expertize in different areas, and it is important to ensemble various LLMs to integrate their complementary knowledge (Yao et al., 2024; Abdulaal et al.; Huang et al., 2024).

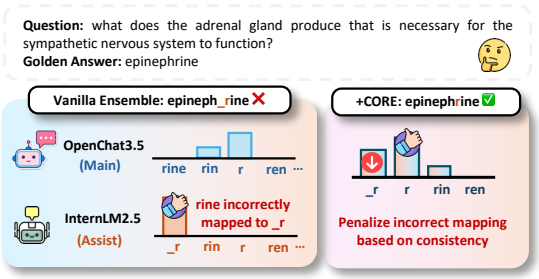


Figure 1: Motivation of CoRE. **Left:** vanilla ensemble yields an incorrect prediction because the token `_r`, misaligned from the assistant token `rine`, dominates the ensemble. **Right:** CoRE penalizes inconsistent tokens, rendering correct prediction `r`.

Extensive efforts on test-time LLM ensemble can be broadly categorized into two categories: token-level and response-level ensemble. Token-level ensemble aligns and fuses the token probabilities of different LLMs at each decoding step (Yu et al., 2024; Yao et al., 2024; Huang et al., 2024; Xu et al., 2024c), enabling fine-grained real-time correction for each token generation. Response-level ensemble offers a more coarse-level ensemble by selecting either a complete response (Jiang et al., 2023b; Lv et al., 2024a; Tekin et al., 2024) or a span (Xu et al., 2024b; Lv et al., 2024b; Liu et al., 2024a) from candidate outputs. Despite their success in improving ensemble quality, existing approaches largely overlook ensemble robustness against noisy or erroneous signals. For instance, incorrect token alignments can result in faulty probability fusion, while errors in model predictions may further compromise the correctness of ensemble outputs. Therefore, it is crucial for ensemble methods to detect and mitigate such potential errors during inference to ensure reliable performance.

*Equal contribution.

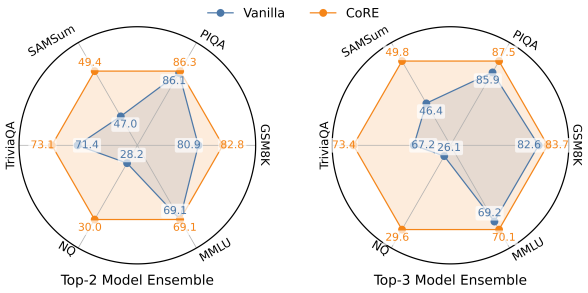


Figure 2: Ensemble performance across six benchmarks. We report the average performance of different ensemble methods with (CoRE) and without CoRE (Vanilla).

To bridge this gap, we propose a plug-and-play technique named CoRE to enhance both robustness and performance of LLM ensemble. Our preliminary observations reveal that ensemble failures are closely tied to large discrepancies between the token distributions of different LLMs. Motivated by this, we propose to harness both token and model consistency to achieve a robust ensemble. At the token level, significant disparities in output probabilities for a specific token indicate fine-grained uncertainty, often stemming from token space misalignment. To address this, we introduce token consistency, which measures the disparity between each model's token probability and a reference probability, serving as a low-pass filter to amplify reliable tokens and suppress inconsistent ones. At the model level, large divergence among full token distributions reflects model conflicts. To capture this, we define model consistency that promotes model outputs with high self-confidence and low divergence from peers, thereby strengthening consistent models and downweighting unreliable ones. A key advantage of the proposed CoRE is that it is orthogonal to various token-level ensemble methods, and thus can be seamlessly integrated with *no additional inference cost*. We summarize the main contributions as follows.

- We systematically analyze and improve the robustness of LLM ensembles.
- We assess consistency at token and model levels to enhance ensemble performance and robustness. CoRE can be seamlessly integrated with various ensemble methods to enable test-time correction with no additional cost.
- We conduct experiments across diverse benchmark tasks, model combinations, and ensemble methods. As shown in Figure 2, CoRE consistently improves baseline ensemble methods, achieving an average performance gain of 1.3% and 2.8% on Top-2 and Top-3 model ensemble, respectively.

2 Related Works

2.1 Test-time LLM Ensemble

Ensembling multiple large language models (LLMs) at test time offers a practical way to harness their diverse strengths and mitigate individual weaknesses. Existing methods can be broadly categorized into token and response level ensembles.

Token-level ensemble *fuses* next-token predictions across models at each decoding step, enabling fine-grained correction during generation. Early works (Fu et al., 2023; Wan et al., 2024; Mavromatis et al., 2024) align token sequences via minimum edit distance, capturing structural difference but incurring high computational cost. GAC (Yu et al., 2024) and UniTE (Yao et al., 2024) bridge disparate vocabularies by aligning tokens through exact or prefix matches in text space. Recent works, such as DeepEn (Huang et al., 2024) and EVA (Xu et al., 2024c), learn projection functions that map heterogeneous token distributions into a shared representation space, enabling direct output fusion.

Response-level Ensemble *selects* the most promising response from the multiple outputs to ensemble LLMs at a coarser-grained level. One line of works select or synthesize a *full response* among model outputs by either training-free approaches, such as perplexity scoring or majority voting (Jiang et al., 2023b; Lv et al., 2024a; Tekin et al., 2024), or training-based approaches that learn to rank responses (Si et al., 2023). Another line of works (Xu et al., 2024b; Lv et al., 2024b; Liu et al., 2024a; Cui et al., 2026) propose ensembling at the span level, aiming to strike a balance between fine-grained correction and context-aware decision-making.

2.2 Model Consistency

Recent research has explored consistency-based strategies to boost LLM performance by either internal self-consistency or cross-model agreement.

Self-Consistency enhances answer reliability by aggregating diverse reasoning paths from a single model via frequency (Wang et al., 2022; Li et al., 2024a; Aggarwal et al., 2023), entropy (Kadavath et al., 2022; Lin et al., 2023; Kang et al., 2025), or confidence signals (Chen et al., 2023; Taubenfeld et al., 2025), but it incurs notable computational overhead due to repeated sampling.

Multi-model Consistency combines outputs from different LLMs through majority voting (Trad

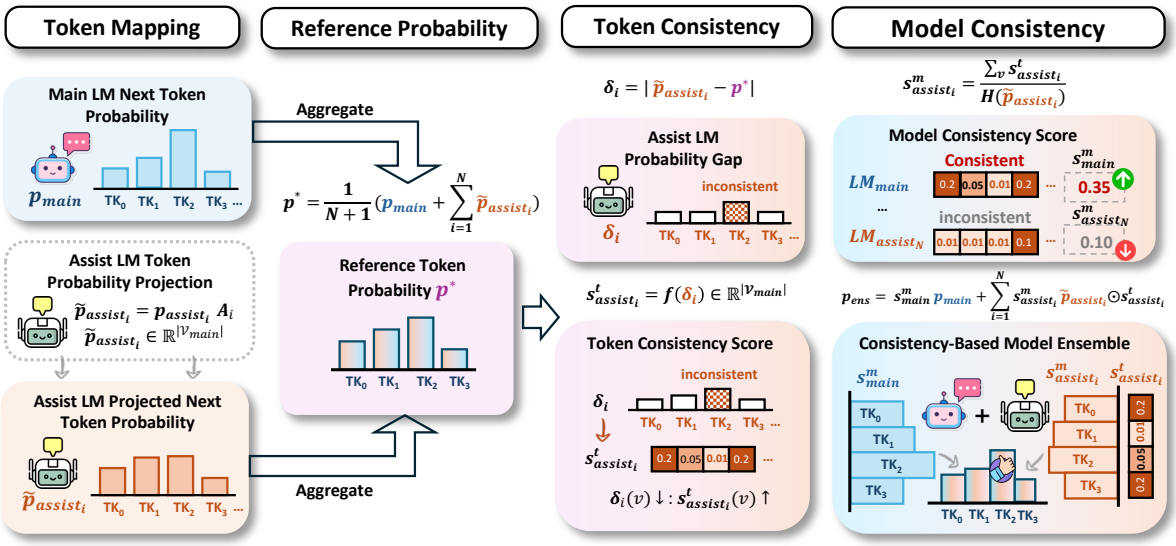


Figure 3: Overview of CoRE. Token probabilities from different models are first aligned through token mapping, and a reference probability distribution is constructed for consistency evaluation. Token consistency mitigates the influence of inconsistent tokens, while model consistency highlights reliable and self-consistent models. Combining both yields a robust and improved ensemble prediction.

and Chehab, 2024; Niimi, 2025) or collaborative reasoning (Wang et al., 2024; Liang et al., 2023; Li et al., 2024b). Despite the improved robustness, most focus on response-level agreement, overlooking fine-grained token-level consistency essential for detecting subtle errors in real time.

3 Methodology

In this section, we present our proposed CoRE for a trustworthy LLM ensemble. We begin by introducing the notation and formally defining the LLM ensemble problem in Section 3.1. Then, in Section 3.2, we reveal a strong correlation between ensemble performance and consistency scores at both the token and model levels. Motivated by these findings, we introduce our token-level consistency measure in Section 3.3 and our distribution-level consistency measure in Section 3.4, both designed to improve ensemble performance and reliability.

3.1 Problem Formulation

Formally, we are given a *main model* with vocabulary set $\mathcal{V}_{\text{main}}$ and N *assistant models* with vocabulary set $\mathcal{V}_{\text{assist}_i}$ with $i \in \{1, 2, ..., N\}$. We denote the predicted token distributions of main and assistant models as $\boldsymbol{p}_{\text{main}}$ and $\boldsymbol{p}_{\text{assist}_i}$, respectively. LLM ensemble first learns a token alignment matrix $\boldsymbol{A}_i \in \mathbb{R}^{|\mathcal{V}_{\text{assist}_i}| \times |\mathcal{V}_{\text{main}}|}$ between the main model and the i-th assistant model, where $\boldsymbol{A}_i(v, u) = 1$ indicates that token $v \in \mathcal{V}_{\text{assist}_i}$ is aligned with token $u \in \mathcal{V}_{\text{main}}$. Token prediction probabilities can be further projected into the main model token space via $\boldsymbol{p}_{\text{assist}_i}\boldsymbol{A}_i \in \mathbb{R}^{|\mathcal{V}_{\text{main}}|}$. For simplicity, we denote the aligned probability as $\tilde{\boldsymbol{p}}_{\text{assist}_i} = \boldsymbol{p}_{\text{assist}_i}\boldsymbol{A}_i$. The aligned probabilities can be further ensembled via

$$\boldsymbol{p}_{\text{ens}} = w_{\text{main}}\boldsymbol{p}_{\text{main}} + \sum_{i=1}^{N} w_{\text{assist}_i}\tilde{\boldsymbol{p}}_{\text{assist}_i}$$

where $[w_{\text{main}}, w_{\text{assist}_1}, ..., w_{\text{assist}_N}] \in \Delta^{N+1}$ are the normalized model weights. For the special case where all models are assigned uniform weights, we denote the resulting ensembled probability distribution as $\boldsymbol{p}^*$, that is

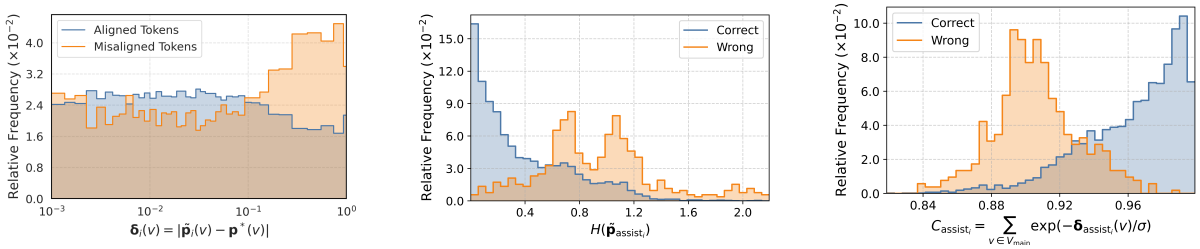
$$\boldsymbol{p}^* = \frac{1}{N+1}\left(\boldsymbol{p}_{\text{main}} + \sum_{i=1}^{N}\tilde{\boldsymbol{p}}_{\text{assist}_i}\right), \quad (1)$$

In this paper, we use the above average probability as the reference $\boldsymbol{p}^*$ for consistency computation.

3.2 Observations

We first study the distinct patterns in both token and model levels, which signal the inherent factors affecting the ensemble performance in Figure 4.

First, we examine how token disparity, measure by the difference between the aligned probability and the reference probability on a specific token v, i.e., $\boldsymbol{\delta}_i(v) = |\tilde{\boldsymbol{p}}_{\text{assist}_i}(v) - \boldsymbol{p}^*(v)|$, reflects the alignment correctness. We experiment on 100 randomly-sampled questions from NQ dataset, and as shown in Figure 4a, aligned tokens concentrate at lower disparities, while misaligned tokens



(a) Token Probability Disparity. (b) Entropy of Token Probability. (c) Sum of Token Consistency.

Figure 4: Observations. **(a) Token probability disparity:** Aligned tokens (blue) exhibit smaller disparities than misaligned ones (orange), indicating higher consistency. **(b) Entropy of token probability:** Correct answers (blue) exhibit lower entropy than wrong ones (orange), indicating higher confidence. **(c) Sum of token consistency:** Correct answers (blue) exhibit higher sum of token consistency score than wrong ones (orange).

exhibit a heavier right tail. A one-sided test of $H_0 : \mathbb{E}_{v\in\text{misalign}}\boldsymbol{\delta}_i(v) \leq \mathbb{E}_{v\in\text{align}}\boldsymbol{\delta}_i(v)$ yields a p-value of 2.7×10^{-44}, indicating that aligned tokens indeed have smaller disparity. We summarize this in Observation 1.

Observation 1 (Token Consistency)**.** *Large token probability disparity signals token misalignment.*

For example, if token $\mathsf{a} \in \mathcal{V}_{\text{assist}_i}$ is misaligned to token $\mathsf{b} \in \mathcal{V}_{\text{main}}$, the difference in their probabilities $|\boldsymbol{p}_{\text{assist}_i}(\mathsf{a}) - \boldsymbol{p}^*(\mathsf{b})| = |\tilde{\boldsymbol{p}}_{\text{assist}_i}(\mathsf{b}) - \boldsymbol{p}^*(\mathsf{b})|$ is expected to be large.

Secondly, we evaluate how model confidence, measured by the entropy of the token probability $H(\tilde{\boldsymbol{p}}_{\text{assist}_i})$, relates to the answer correctness. We experiment on PIQA dataset, and as shown in Figure 4b, correct answers exhibit lower entropy than wrong ones. A one-sided test of $H_0 : \mathbb{E}[H(\tilde{\boldsymbol{p}})|\text{wrong}] \leq \mathbb{E}[H(\tilde{\boldsymbol{p}})|\text{correct}]$ yields a p-value of 7.7×10^{-108}, confirming that wrong answers have larger entropy than correct ones. We summarize this into Observation 2.

Observation 2 (Model Confidence)**.** *Small model confidence measure by the entropy of token probability signals correct answer.*

Thirdly, we evaluate how model consistency, measured by the sum of RBF-transformed token disparities, i.e., $C_{\text{assist}_i}(v) = \sum_{v\in\mathcal{V}_{\text{main}}} \exp(-\boldsymbol{\delta}_{\text{assist}_i}(v)/\sigma)$, varies between correct and wrong answers. We experiment on PIQA dataset, and as shown in Figure 4c, correct answers exhibit higher model consistency than wrong ones. A one-sided test of $H_0 : \mathbb{E}[C_{\text{assist}_i}|\text{wrong}] \geq \mathbb{E}[C_{\text{assist}_i}|\text{correct}]$ yields a p-value of 5.0×10^{-222}, indicating that correct answers have greater consistency. We summarize this into Observation 3.

Observation 3 (Model Consistency)**.** *Large model consistency measured by the sum of RBF-transformed token disparity signals correct answer.*

Motivated by these observations, we propose to harness both token and model consistency for robust LLM ensemble.

3.3 Token Consistency

We first introduce our proposed token consistency as a fine-grained measure. A key challenge in LLM ensemble is the disparate token spaces due to different tokenization schemes adopted by different LLMs. Despite extensive efforts in aligning token spaces, they inevitably encounter alignment errors that impair ensemble performance.

To address this limitation, it is essential to distinguish the misaligned tokens and rectify their contributions. Motivated by Observation 1, where misaligned tokens exhibit large probability disparities, we propose token consistency as a low-pass filter to downweight their influence. Adopting the average distribution in Eq. (1) as the reference probability, we quantify the i-th assist model's token consistency $\boldsymbol{s}^{\text{t}}_{\text{assist}_i}$ as follows

$$\begin{aligned} &\boldsymbol{s}^{\text{t}}_{\text{assist}_i} = f(\boldsymbol{\delta}_i) \in \mathbb{R}^{|\mathcal{V}_{\text{main}}|}, \\ &\text{where } \boldsymbol{\delta}_i = |\tilde{\boldsymbol{p}}_{\text{assist}_i} - \boldsymbol{p}^*| \in \mathbb{R}^{|\mathcal{V}_{\text{main}}|}, \end{aligned} \tag{2}$$

where different functions, e.g., RBF kernel $f_{\text{rbf}}(\boldsymbol{\delta}) = \exp(-\boldsymbol{\delta}/\sigma)$, power function $f_{\text{pow}}(\boldsymbol{\delta}) = \alpha(1-\boldsymbol{\delta})^\beta$, and sigmoid function $f_{\text{sig}}(\boldsymbol{\delta}) = 1 - \text{Sigmoid}(\gamma(\boldsymbol{\delta}_i - 0.5))$, can be adopted. Intuitively, large $\boldsymbol{\delta}$ induces small token consistency that signals large inconsistency with the reference probability. By multiplying token consistency with the aligned distribution, i.e., $\boldsymbol{s}^{\text{t}}_{\text{assist}_i} \odot \tilde{\boldsymbol{p}}_{\text{assist}_i}$, token consistency acts as a low-pass filter: penalizing inconsistent tokens with large disparities while promoting consistent tokens that are widely agreed upon.

3.4 Model Consistency

In addition to fine-grained token consistency, it is crucial to quantify the trustworthiness of the model. Prior works typically assign model weights based on heuristics, such as uniform weighting (Yao et al., 2024; Yu et al., 2024; Xu et al., 2024c) or prior knowledge (Huang et al., 2024), or self-confidence metrics like perplexity (Mavromatis et al., 2024; Liu et al., 2024a). However, the inter-model consistency remains largely underexplored.

Motivated by Observation 2 and 3, we argue that models exhibiting both high inter-model consistency and strong self-confidence should be prioritized. To capture this, we define the model consistency by aggregating token consistency over the main token space, and regularizing it by the entropy of the output distribution serving as a proxy for confidence. Formally, the model consistency of an assistant model is given by:

$$s^{\mathrm{m}}_{\mathrm{assist}_i} = \frac{\sum_{v \in \mathcal{V}_{\mathrm{main}}} \boldsymbol{s}^{\mathrm{t}}_{\mathrm{assist}_i}(v)}{H(\tilde{\boldsymbol{p}}_{\mathrm{assist}_i})} \in \mathbb{R}, \tag{3}$$

where $H(\cdot)$ denotes the entropy of a distribution. Here, the numerator rewards agreement with the reference model, while the denominator penalizes high uncertainty, thus favoring outputs that are both consistent and confident. A similar definition applies for the main model, denoted as $s^{\mathrm{m}}_{\mathrm{main}}$.

By using token consistency as a token-wise filter and model consistency as model-level weights, the final ensembled distribution is computed as:

$$\boldsymbol{p}_{\mathrm{ens}} = s^{\mathrm{m}}_{\mathrm{main}} \boldsymbol{p}_{\mathrm{main}} + \sum_{i=1}^{N} s^{\mathrm{m}}_{\mathrm{assist}_i} \boldsymbol{s}^{\mathrm{t}}_{\mathrm{assist}_i} \odot \tilde{\boldsymbol{p}}_{\mathrm{assist}_i}, \tag{4}$$

where $[s^{\mathrm{m}}_{\mathrm{main}}, s^{\mathrm{m}}_{\mathrm{assist}_1}, \ldots, s^{\mathrm{m}}_{\mathrm{assist}_N}] \in \Delta^{N+1}$ are the normalized model consistency serving as the model weights. Note that we apply token consistency only to assistant models to mitigate potential misalignment, as token misalignment arises solely when mapping tokens from assistant models to the main model's token space.

4 Experiments

We carry out extensive experiments to answer the following research questions:

- **RQ1:** How does CoRE enhance ensemble performance? (Section 4.2)
- **RQ2:** To what extent does CoRE enhance ensemble robustness? (Section 4.3)
- **RQ3:** What are the respective roles of token and model consistency, and how do their designs affect performance? (Section 4.4)
- **RQ4:** How does the performance scale with more models w/ and w/o CoRE? (Section 4.4)

4.1 Experiment Setup

Base Models. We conduct our experiments on the following widely used models, including `Llama-3-8B-Instruct` (Dubey et al., 2024), `Mistral-7B-Instruct-v0.1` (Jiang et al., 2023a), `Qwen2.5-3b-Instruct` (Team, 2024), `InternLM2.5-7b-Chat` (Team, 2023) and `openchat-3.5-0106` (Wang et al., 2023).

Ensemble Methods. We consider four baseline methods that align token spaces for LLM ensemble for benchmark evaluation, including:

- MinED (Fu et al., 2023; Mavromatis et al., 2024) searches for textually closest tokens based on minimum edit distance.
- GaC (Yu et al., 2024) merges disparate token spaces into a union token space for ensemble;
- UniTE (Yao et al., 2024) utilizes tokenizer to map tokens to their prefix counterparts;
- EVA (Xu et al., 2024c) learns a mapping by aligning overlapping token embeddings.

Note that our proposed CoRE mainly operates on the token space and is not specifically designed for methods like DeePEn (Huang et al., 2024) that ensemble in the latent embedding space. While being outside our main scope, we nonetheless conduct a separate analysis by slightly adapting CoRE to integrate with DeePEn, in order to explore its potential benefits in this alternative setting.

Datasets and Metrics. We evaluate six benchmarks covering four different categories, including: (1) **Reasoning:** `GSM8K` (Cobbe et al., 2021) (4-shot with CoT) covering grade school math problems, and `PIQA` (Bisk et al., 2020) (0-shot) with commonsense reasoning choice problems; (2) **Summarization:** `SAMSum` (Gliwa et al., 2019) (0-shot) on dialogue summarization; (3) **Knowledge:** `TriviaQA` (Joshi et al., 2017) (5-shot) and `NaturalQuestions (NQ)` (Kwiatkowski et al., 2019) (5-shot); (4) **Comprehensive Examination**: `MMLU` (Hendrycks et al., 2009) (5-shot) covering 57 subjects that humans typically learn. We adopt Exact Match for `PIQA, TriviaQA, NQ` and `MMLU`, Accuracy for `GSM8K`, and Rouge-1 score for `SAMSum`.

Method		GSM8K		PIQA		SAMSum		TriviaQA		NQ		MMLU		Average	
		Top-2	Top-3	Top-2	Top-3	Top-2	Top-3	Top-2	Top-3	Top-2	Top-3	Top-2	Top-3	Top-2	Top-3
MINED	Vanilla	79.51	81.65	85.47	82.32	43.79	42.18	67.50	59.63	25.07	22.22	68.38	66.66	61.62	59.11
	CORE	82.41	83.78	85.75	87.49	49.25	49.55	72.17	72.85	29.83	29.61	68.42	69.77	64.64	65.51
	Δ	+2.90	+2.13	+0.28	+5.17	+5.46	+7.37	+4.67	+13.22	+4.76	+7.39	+0.04	+3.11	+3.02	+6.40
UNITE	Vanilla	81.26	82.79	85.64	86.89	48.03	47.99	70.93	64.90	28.17	25.84	68.61	69.56	63.78	63.00
	CORE	82.71	83.70	85.96	87.38	49.41	49.77	73.41	73.61	30.39	29.94	68.63	69.91	65.09	65.72
	Δ	+1.45	+0.91	+0.32	+0.49	+1.38	+1.78	+2.48	+8.71	+2.22	+4.10	+0.02	+0.35	+1.31	+2.72
EVA	Vanilla	82.09	83.02	87.81	87.43	48.16	47.83	73.47	73.02	30.00	29.09	70.62	71.07	65.36	65.24
	CORE	83.40	83.24	87.81	87.76	49.43	50.27	73.13	73.23	29.53	29.09	70.62	70.91	65.65	65.75
	Δ	+1.31	+0.22	+0.00	+0.33	+1.27	+2.44	-0.34	+0.21	-0.47	+0.00	+0.00	-0.16	+0.29	+0.51
GAC	Vanilla	80.74	83.02	85.67	86.89	48.08	47.63	73.67	71.07	29.61	27.15	68.61	69.56	64.40	64.22
	CORE	82.49	84.00	85.75	87.32	49.41	49.73	73.72	73.97	30.22	29.81	68.63	69.96	65.04	65.80
	Δ	+1.75	+0.98	+0.08	+0.43	+1.33	+2.10	+0.05	+2.90	+0.61	+2.66	+0.02	+0.40	+0.64	+1.58

Table 1: Benchmark results. We report ensemble performance using the Top-2 and Top-3 base models on each dataset, with (CORE) and without (Vanilla) our method. The Δ rows report the performance gain from applying CORE, with blue cells indicating improvement and red cells indicating degradation.

Model	GSM8K	PIQA	SAMSum	TriviaQA	NQ	MMLU
Llama3	74.91	76.10	43.57	63.08	22.13	63.49
Mistral7b	41.55	62.62	44.80	52.47	15.01	52.17
Qwen2.5	36.76	80.03	43.77	43.50	12.96	64.80
InternLM2.5	82.69	86.91	42.21	63.23	26.62	71.27
OpenChat3.5	76.35	62.62	50.05	72.13	28.92	62.11

Table 2: Base model performance. We use Blue, Yellow and Red to denote Top-1, Top-2 and Top-3 models.

Experiment Pipeline. We first evaluate the performance of base models on each dataset, and then select the Top-2 and Top-3 models for benchmark ensemble evaluation. We adopt RBF function as the default consistency score for benchmark results, and set $\sigma = 0.5$. All experiments are conducted on NVIDIA A100 80GB GPUs.

4.2 Benchmark Results

We present the ensemble results in Table 1 and base models' performance in Table 2, from which we draw the following observations:

(1) CORE achieves consistent improvements on different methods, datasets, and base model combinations. Specifically, on reasoning datasets (GSM8K, PIQA), CORE enhances vanilla methods by 1.01 on Top-2 and 1.33 on Top-3 ensemble. On the summarization dataset (SAMSum), CORE improves vanilla baselines by an average of 2.35 and 3.42 on Top-2 and Top-3 ensembles, respectively. For knowledge-intensive datasets (TriviaQA, NQ), it yields average gains of 1.75 (Top-2) and 4.90 (Top-3). On the comprehensive exam benchmark (MMLU), CORE provides smaller but consistent improvements of 0.03 (Top-2) and 0.94 (Top-3).

(2) CORE achieves more stable ensemble. As more LLMs are included, baseline ensembles may suffer from negative ensemble, where performance degrades compared to the best single model. In contrast, augmenting with CORE leads to robust and consistent improvements. Notably, CORE successfully mitigates 17 negative ensemble cases encountered by the baseline ensemble methods.

Method	PIQA		NQ		MMLU		Average	
	Top-2	Top-3	Top-2	Top-3	Top-2	Top-3	Top-2	Top-3
Vanilla	87.52	87.35	28.03	29.23	70.56	70.49	62.04	62.36
CORE	87.74	87.35	28.14	29.53	70.57	70.5	62.15	62.46
Δ	+0.22	+0.00	+0.11	+0.30	+0.01	+0.01	+0.11	+0.10

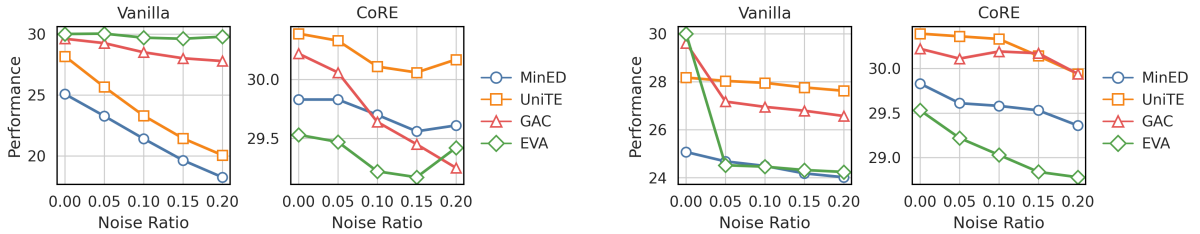
Table 3: Ensemble performance with DEEPEN.

Though not directly compatible with DEEPEN, which performs ensemble in the latent embedding space rather than the token space, we adapt CORE to compute consistency over token embeddings and report the results in Table 3. Though less pronounced than token-space baselines, augmenting with CORE still yields consistent improvements. We attribute this to two main factors: (1) DEEPEN constructs its latent space using overlapping tokens that are identical, which inherently reduces token misalignment, and (2) the learned latent space already promotes cross-model agreement by design. Interestingly, though some vanilla baselines, e.g., MINED and UNITE, may underperform DEEPEN, they achieve better performance when augmented with CORE, validating the effectiveness of CORE.

4.3 Robustness Results

4.3.1 Robustness against Noises

We first evaluate ensemble performance against noises. We consider two types of noises, including: (1) alignment noise, where 5%, 10%, 15% and 20%



(a) Token noises. (b) Probability noises.

Figure 5: Robustness against noises. CoRE maintains stable performance with minimal degradation under noises.

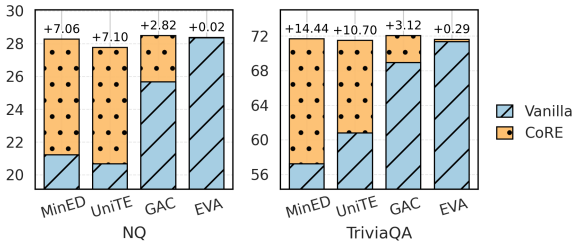


Figure 6: Robustness against large performance gap. We ensemble the best and worst performing models on NQ and TriviaQA. Numbers above the bars indicate the performance gains achieved by CoRE.

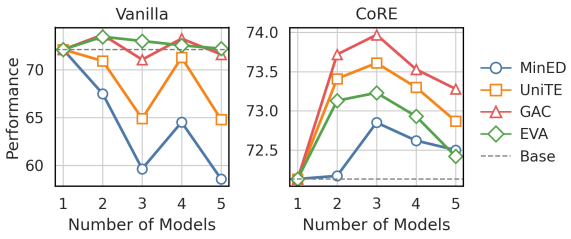


Figure 7: Scaling results on TriviaQA. **Left:** Vanilla ensemble methods suffer from negative ensembling, with performance degrading as more models are added. **Right:** CoRE enables stable scaling, consistently outperforming the best single model across ensemble sizes.

of the rows in the token mapping matrix are perturbed, and (2) probability noise, where Gaussian noise with standard deviations of 0.05, 0.10, 0.15, and 0.20 is added to the token probabilities. The results are shown in Figures 5.

Under both noise types, vanilla ensemble methods are highly sensitive, with average performance drops of 4.25 and 2.60 points as noise ratios increase from 0 to 0.2 for token and probability noise, respectively. In contrast, CoRE demonstrates strong robustness, exhibiting only marginal degradation of 0.38 and 0.49 under the same conditions. Notably, CoRE is particularly effective in mitigating the impact of token alignment noise. This improvement can be attributed to the token consistency mechanism that identifies and rectifies misaligned tokens, thereby preserving ensemble accuracy even under severe perturbations.

4.3.2 Robustness against Performance Gap

Previous methods (Yao et al., 2024) may encounter negative ensemble, i.e., model performance drops after ensemble, when facing significant performance discrepancies among base models. To evaluate CoRE's robustness against performance gaps, we ensemble the best and worst performing models on NQ and TriviaQA with and without CoRE, and the results are shown in Figure 6.

It is shown that CoRE consistently improves all baseline ensemble methods on two datasets. In particular, for vanilla methods that suffer severe degradation, e.g., MinED, UniTE, and GAC, augmenting with CoRE yields substantial average gains of +5.66 on NQ and +9.42 on TriviaQA. Moreover, while vanilla methods exhibit widely varying performance, their performance with CoRE converges to comparable levels, highlighting CoRE's ability to mitigate performance gaps and deliver stable improvements across diverse ensemble strategies.

4.4 Studies

4.4.1 Scaling Results

We evaluate how CoRE enhances ensemble scalability as more LLMs are incorporated. As shown in Figure 7, vanilla ensembles suffer from negative ensembling, with performance often degrading as more models are added, whereas CoRE enables stable scaling and consistently outperforms the best single model. This performance degradation of the vanilla ensemble arises because increased model diversity introduces conflicting signals that, without consistency control, overwhelm the ensemble. Furthermore, token space alignment becomes more challenging when more LLMs are involved, and being unaware of the potential misalignment can lead to unstable and even counterproductive ensembling. However, CoRE enforces consistency and filters out unreliable outputs, transforming diversity into complementary information rather than noise, and thereby enabling robust and effective scaling.

4.4.2 Ablation Study

On cosistency scores. We conduct ablation studies on token and model consistency on NQ and

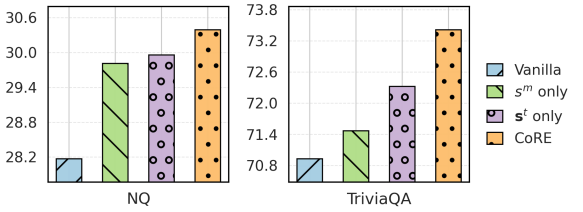


Figure 8: Ablation study. Both token and model consistency benefit the ensemble performance.

TriviaQA, and the results are shwon in Figure 8. A clear trend emerges: CoRE achieves the best performance, followed by applying only token consistency $\boldsymbol{s}^t$, then only model consistency s^m, while the vanilla ensemble performs the worst. This indicates that both consistency components contribute positively to the performance: token consistency enhances fine-grained alignment across tokens, model consistency improves global agreement among models, and their joint integration yields the best overall gains.

On entropy weight. We ablate the entropy term in the model consistency in Eq. (3), and the results are presented in Figure 9. From these results, we draw two key observations. First, ablating the entropy term consistently leads to performance degradation across all baselines. As shown by the hatched regions in the figure, reintroducing the entropy term yields performance gains ranging from +0.27 to +1.28, effectively validating its contribution to the model's overall accuracy. Notably, the UniTE model benefits most significantly, with a gain of +1.28. Second, even when the entropy term is ablated (colored bars), the performance remains robust compared to the vanilla ensemble, validating the effectiveness of the underlying consistency design independent of the entropy regularization.

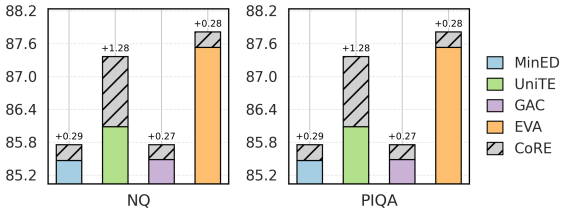


Figure 9: Ablation study of the entropy term. Colored bars indicate model consistency without the entropy term, while hatched bars indicate model consistency with the entropy term (CoRE).

On token consistency design. The token consistency design in Eq. (2) is asymmetric, where only assist models are penalized while the main model remains original. To validate this design, we compare it against a symmetric variant where token consistency is applied to both main and assist models. The results are reported in Figure 10. We observe that the symmetric application (colored bars) consistently underperforms the proposed assistant-only design (hatched bars, CoRE). As indicated by the annotated gains in the figure, removing the penalty from the main model leads to significant improvements, such as +1.78 on PIQA with GAC and +1.66 on NQ with EVA. This confirms that the primary inconsistencies stem from assistant-to-main misalignment; therefore, penalizing the main model introduces unnecessary constraints that hurt the overall ensemble performance.

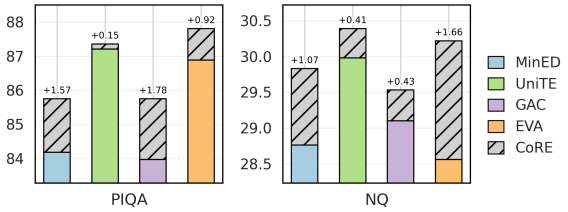


Figure 10: Ablation study of the token consistency. Colored bars indicate applying token consistency on both main and assist models, while hatched bars indicate only applying token consistency on assist models (CoRE).

4.5 Hyperparameter Study

We conduct a sensitivity analysis on the hyperparameters governing the consistency functions: σ in the RBF kernel, α and β in the power function, and γ in the sigmoid function. As shown in Figure 11, across a broad range of values, CoRE is fairly robust to these hyperparameters and does not require heavy fine-tuning.

Conceptually, these hyperparameters control the *sharpness* of the consistency score, i.e., how aggressively inconsistent tokens and models are penalized.High values of α, β, γ (or small σ) yield "peaked" scores that strictly downweight inconsistencies, effectively filtering noise when assistant models are weak. Conversely, lower values (or large σ) produce smoother scores that tolerate more diversity, which is beneficial when the ensemble consists of strong, reliable models.

4.5.1 Consistency Score Functions

We explore alternative designs for consistency score to evaluate the generalization of CoRE. Specifically, we consider three consistency functions: CoRE-RBF with RBF kernel $f_{\text{rbf}}(\boldsymbol{\delta}) = \exp(-\boldsymbol{\delta}/\sigma)$, CoRE-pow with power function

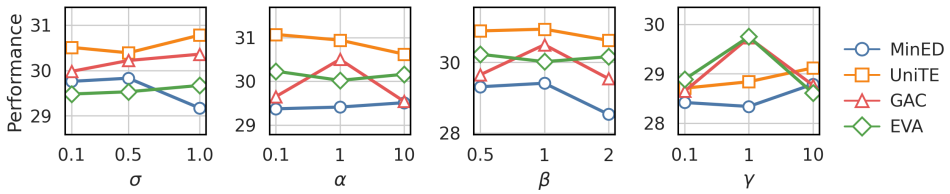


Figure 11: Hyperparameter study on σ in RBF function, α and β in power function, and γ in sigmoid function.

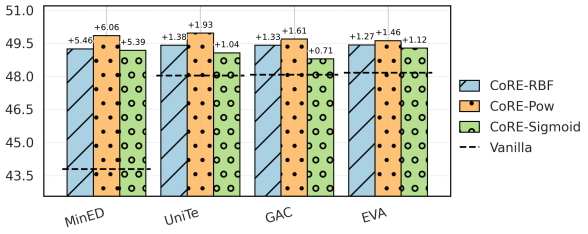


Figure 12: Ensemble performance with different consistency score functions on SAMSum. Numbers above the bars indicate the performance gains achieved.

$f_{\text{pow}}(\boldsymbol{\delta}) = \alpha(1-\boldsymbol{\delta})^{\beta}$, and CoRE-SIG with sigmoid function $f_{\text{sig}}(\boldsymbol{\delta}) = 1 - \text{Sigmoid}(\gamma(\boldsymbol{\delta} - 0.5))$. Experiments are conducted on the SAMSum dataset, with results presented in Figure 12.

In general, different consistency score designs consistently enhance ensemble methods, demonstrating the broad applicability of CoRE. On average, CoRE-Pow achieves the best performance with a 2.76 gain, followed by CoRE-RBF (2.35) and CoRE-Sigmoid (2.06). The performance differences stem from how each function maps token disparity $\boldsymbol{\delta}$ to consistency weights. RBF sharply favors well-aligned tokens, offering stable but conservative gains. Power applies smoother decay, better balancing selectivity and inclusiveness, thus achieving the highest gain. Sigmoid filters noise via soft thresholding but its limited smoothness dampens overall improvement.

4.5.2 Case Study

Example 1

Question: what does the adrenal gland produce that is necessary for the sympathetic nervous system to function?
Gold Answer: epinephrine
OpenChat Response: adrenaline ✗
InternLM Response: epinephrine and norepinephrine ✗
Vanilla Response: epineph_rine ✗
CoRE Response: epinephrine ✓

We present a case study on ensembling OpenChat3.5 and InternLM2.5 offering an intuitive illustration of how CoRE operates. As shown in Example 1 and Figure 13, both individual models and vanilla ensemble fail to generate the correct answer, whereas augmenting with CoRE

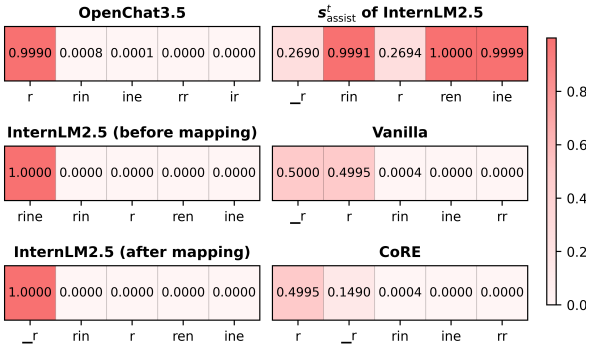


Figure 13: The Top-5 next token probability and the corresponding token consistency s^t_{assist} given generated text 'epineph' in Example 1.

yields the correct response. The key difference lies in the fourth token following the generated text "epineph": the vanilla ensemble predicts `_r`, while CoRE correctly predicts `r`. Analyzing the token probabilities of `InternLM2.5` *before* and *after* token mapping reveals that the correct token `rine` is incorrectly aligned to `_r`, leading the vanilla ensemble to an erroneous fusion. In contrast, CoRE identifies the misaligned token `_r` given its low consistency and downweights its influence, thereby penalizing unreliable tokens and models to produce the correct output. More case studies are provided in Appendix B.

5 Conclusion

In this paper, we study the robustness of LLM ensembles, a critical yet underexplored aspect in prior work. Our analysis shows that ensemble failures often arise from inconsistencies at both the token and model levels. To address this, we introduce CoRE, a lightweight and plug-and-play framework that enforces consistency across multiple dimensions without incurring additional inference cost. Extensive experiments show that CoRE significantly enhances both the performance and robustness of existing ensemble methods. Our findings highlight consistency as a key principle for building reliable LLM ensembles and open new directions for robustness-oriented ensemble in future research.

Limitations

CoRE requires access to token-level logits of ensembled LLMs to enforce consistency at both token and model level, preventing its use with closed-source or black-box LLM APIs. Moreover, determining *when to ensemble* remains an open question: if the main model is already confident or the assistant model exhibits low confidence, skipping ensembling may prevent unreliable outputs from degrading performance. Finally, identifying *which models to ensemble* is also worth exploring, while our model weights provide a soft balancing mechanism, future work could study more principled criteria for selecting beneficial model combinations.

Acknowledgements

This work is supported by NSF (2433308) and AFOSR (FA9550-24-1-0002). The content of the information in this document does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

Ahmed Abdulaal, Chen Jin, Nina Montaña-Brown, Aryo Pradipta Gema, Daniel C Castro, Daniel C Alexander, Philip Alexander Teare, Tom Diethe, Dino Oglic, and Amrutha Saseendran. Balancing act: Diversity and consistency in large language model ensembles. In *The Thirteenth International Conference on Learning Representations*.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Pranjal Aggarwal, Aman Madaan, Yiming Yang, and 1 others. 2023. Let's sample step by step: Adaptive-consistency for efficient reasoning and coding with llms. *arXiv preprint arXiv:2305.11860*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, and 1 others. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, and 1 others. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. 2023. Universal self-consistency for large language model generation. *arXiv preprint arXiv:2311.17311*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Chengming Cui, Tianxin Wei, Ziyi Chen, Ruizhong Qiu, Zhichen Zeng, Zhining Liu, Xuying Ning, Duo Zhou, and Jingrui He. 2026. Adafuse: Adaptive ensemble decoding with test-time scaling for llms. *arXiv preprint arXiv:2601.06022*.

Alexey Dosovitskiy. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Yao Fu, Hao Peng, Litu Ou, Ashish Sabharwal, and Tushar Khot. 2023. Specializing smaller language models towards multi-step reasoning. In *International Conference on Machine Learning*, pages 10421–10430. PMLR.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. Samsum corpus: A human-annotated dialogue dataset for abstractive summarization. *arXiv preprint arXiv:1911.12237*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2009. Measuring massive multitask language understanding, 2021. *URL https://arxiv. org/abs*, page 20.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

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. *Advances in Neural Information Processing Systems*, 37:119838–119860.

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. *Preprint*, arXiv:2310.06825.

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

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, and 1 others. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Zhewei Kang, Xuandong Zhao, and Dawn Song. 2025. Scalable best-of-n selection for large language models via self-certainty. *arXiv preprint arXiv:2502.18581*.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, and 1 others. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Yiwei Li, Peiwen Yuan, Shaoxiong Feng, Boyuan Pan, Xinglin Wang, Bin Sun, Heda Wang, and Kan Li. 2024a. Escape sky-high cost: Early-stopping self-consistency for multi-step reasoning. *arXiv preprint arXiv:2401.10480*.

Yunxuan Li, Yibing Du, Jiageng Zhang, Le Hou, Peter Grabowski, Yeqing Li, and Eugene Ie. 2024b. Improving multi-agent debate with sparse communication topology. *arXiv preprint arXiv:2406.11776*.

Zihao Li, Xiao Lin, Zhining Liu, Jiaru Zou, Ziwei Wu, Lecheng Zheng, Dongqi Fu, Yada Zhu, Hendrik Hamann, Hanghang Tong, and 1 others. 2025a. Language in the flow of time: Time-series-paired texts weaved into a unified temporal narrative. *arXiv preprint arXiv:2502.08942*.

Zihao Li, Lecheng Zheng, Bowen Jin, Dongqi Fu, Baoyu Jing, Yikun Ban, Jingrui He, and Jiawei Han. 2025b. Can graph neural networks learn language with extremely weak text supervision? In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11138–11165.

Mingfu Liang, Xi Liu, Rong Jin, Boyang Liu, Qiuling Suo, Qinghai Zhou, Song Zhou, Laming Chen, Hua Zheng, Zhiyuan Li, and 1 others. 2025. External large foundation model: How to efficiently serve trillions of parameters for online ads recommendation. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 344–353.

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

Haokun Lin, Haobo Xu, Yichen Wu, Jingzhi Cui, Yingtao Zhang, Linzhan Mou, Linqi Song, Zhenan Sun, and Ying Wei. 2024a. Duquant: Distributing outliers via dual transformation makes stronger quantized llms. *Advances in Neural Information Processing Systems*, 37:87766–87800.

Haokun Lin, Haobo Xu, Yichen Wu, Ziyu Guo, Renrui Zhang, Zhichao Lu, Ying Wei, Qingfu Zhang, and Zhenan Sun. 2025a. Quantization meets dllms: A systematic study of post-training quantization for diffusion llms. *arXiv preprint arXiv:2508.14896*.

Xiao Lin, Philip Li, Zhichen Zeng, Tingwei Li, Tianxin Wei, Xuying Ning, Gaotang Li, Yuzhong Chen, and Hanghang Tong. 2026. Alert: Zero-shot llm jailbreak detection via internal discrepancy amplification. *arXiv preprint arXiv:2601.03600*.

Xiao Lin, Zhining Liu, Dongqi Fu, Ruizhong Qiu, and Hanghang Tong. 2024b. Backtime: Backdoor attacks on multivariate time series forecasting. *Advances in Neural Information Processing Systems*, 37:131344–131368.

Xiao Lin, Zhichen Zeng, Tianxin Wei, Zhining Liu, Hanghang Tong, and 1 others. 2025b. Cats: Mitigating correlation shift for multivariate time series classification. *arXiv preprint arXiv:2504.04283*.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2023. Generating with confidence: Uncertainty quantification for black-box large language models. *arXiv preprint arXiv:2305.19187*.

Cong Liu, Xiaojun Quan, Yan Pan, Liang Lin, Weigang Wu, and Xu Chen. 2024a. Cool-fusion: Fuse large language models without training. *arXiv preprint arXiv:2407.19807*.

Xiaolong Liu, Zhichen Zeng, Xiaoyi Liu, Siyang Yuan, Weinan Song, Mengyue Hang, Yiqun Liu, Chaofei Yang, Donghyun Kim, Wen-Yen Chen, and 1 others. 2024b. A collaborative ensemble framework for ctr prediction. *arXiv preprint arXiv:2411.13700*.

Zhining Liu, Ze Yang, Xiao Lin, Ruizhong Qiu, Tianxin Wei, Yada Zhu, Hendrik Hamann, Jingrui He, and Hanghang Tong. 2025. Breaking silos: Adaptive model fusion unlocks better time series forecasting. *arXiv preprint arXiv:2505.18442*.

Bo Lv, Chen Tang, Yanan Zhang, Xin Liu, Ping Luo, and Yue Yu. 2024a. Urg: A unified ranking and generation method for ensembling language models. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 4421–4434.

Bo Lv, Chen Tang, Yanan Zhang, Xin Liu, Yue Yu, and Ping Luo. 2024b. Specfuse: Ensembling large language models via next-segment prediction. *arXiv preprint arXiv:2412.07380*.

Costas Mavromatis, Petros Karypis, and George Karypis. 2024. Pack of llms: Model fusion at test-time via perplexity optimization. *arXiv preprint arXiv:2404.11531*.

Junichiro Niimi. 2025. A simple ensemble strategy for llm inference: Towards more stable text classification. In *International Conference on Applications of Natural Language to Information Systems*, pages 189–199. Springer.

Xuying Ning, Dongqi Fu, Tianxin Wei, Wujiang Xu, and Jingrui He. 2025. Graph4mm: Weaving multimodal learning with structural information. *arXiv preprint arXiv:2510.16990*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 1715–1725.

Chenglei Si, Weijia Shi, Chen Zhao, Luke Zettlemoyer, and Jordan Boyd-Graber. 2023. Getting more out of mixture of language model reasoning experts. *arXiv preprint arXiv:2305.14628*.

Amir Taubenfeld, Tom Sheffer, Eran Ofek, Amir Feder, Ariel Goldstein, Zorik Gekhman, and Gal Yona. 2025. Confidence improves self-consistency in llms. *arXiv preprint arXiv:2502.06233*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

InternLM Team. 2023. Internlm: A multilingual language model with progressively enhanced capabilities.

Qwen Team. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2.

Selim Furkan Tekin, Fatih Ilhan, Tiansheng Huang, Sihao Hu, and Ling Liu. 2024. Llm-topla: Efficient llm ensemble by maximising diversity. *arXiv preprint arXiv:2410.03953*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Fouad Trad and Ali Chehab. 2024. To ensemble or not: Assessing majority voting strategies for phishing detection with large language models. In *International Conference on Intelligent Systems and Pattern Recognition*, pages 158–173. Springer.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

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

Guan Wang, Sijie Cheng, Xianyuan Zhan, Xiangang Li, Sen Song, and Yang Liu. 2023. Openchat: Advancing open-source language models with mixed-quality data. *arXiv preprint arXiv:2309.11235*.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Zou. 2024. Mixture-of-agents enhances large language model capabilities. *arXiv preprint arXiv:2406.04692*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. 2023. Jailbroken: How does llm safety training fail? *Advances in Neural Information Processing Systems*, 36:80079–80110.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, and 1 others. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Haobo Xu, Yuchen Yan, Dingsu Wang, Zhe Xu, Zhichen Zeng, Tarek F Abdelzaher, Jiawei Han, and Hanghang Tong. 2024a. Slog: An inductive spectral graph neural network beyond polynomial filter. In *Forty-first International Conference on Machine Learning*.

Yangyifan Xu, Jianghao Chen, Junhong Wu, and Jiajun Zhang. 2024b. Hit the sweet spot! span-level ensemble for large language models. *arXiv preprint arXiv:2409.18583*.

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

Yuchen Yan, Yuzhong Chen, Huiyuan Chen, Xiaoting Li, Zhe Xu, Zhichen Zeng, Lihui Liu, Zhining Liu, and Hanghang Tong. 2024a. Thegcn: Temporal heterophilic graph convolutional network. *arXiv preprint arXiv:2412.16435*.

Yuchen Yan, Yongyi Hu, Qinghai Zhou, Lihui Liu, Zhichen Zeng, Yuzhong Chen, Menghai Pan, Huiyuan Chen, Mahashweta Das, and Hanghang Tong. 2024b. Pacer: Network embedding from positional to structural. In *Proceedings of the ACM Web Conference 2024*, pages 2485–2496.

Yuchen Yan, Aakash Kolekar, Sahika Genc, Wenju Xu, Edward W Huang, Anirudh Srinivasan, Mukesh Jain, Qi He, and Hanghang Tong. 2025. To answer or not to answer (taona): A robust textual graph understanding and question answering approach. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 6360–6376.

Yuchen Yan, Lihui Liu, Yikun Ban, Baoyu Jing, and Hanghang Tong. 2021a. Dynamic knowledge graph alignment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 4564–4572.

Yuchen Yan, Si Zhang, and Hanghang Tong. 2021b. Bright: A bridging algorithm for network alignment. In *Proceedings of the web conference 2021*, pages 3907–3917.

Yuchen Yan, Qinghai Zhou, Jinning Li, Tarek Abdelzaher, and Hanghang Tong. 2022. Dissecting cross-layer dependency inference on multi-layered inter-dependent networks. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 2341–2351.

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

Hyunsik Yoo, Zhichen Zeng, Jian Kang, Ruizhong Qiu, David Zhou, Zhining Liu, Fei Wang, Charlie Xu, Eunice Chan, and Hanghang Tong. 2024. Ensuring user-side fairness in dynamic recommender systems. In *Proceedings of the ACM Web Conference 2024*, pages 3667–3678.

Qi Yu, Zhichen Zeng, Yuchen Yan, Zhining Liu, Baoyu Jing, Ruizhong Qiu, Ariful Azad, and Hanghang Tong. 2025a. Planetalign: A comprehensive python library for benchmarking network alignment. *arXiv preprint arXiv:2505.21366*.

Qi Yu, Zhichen Zeng, Yuchen Yan, Lei Ying, R Srikant, and Hanghang Tong. 2025b. Joint optimal transport and embedding for network alignment. In *Proceedings of the ACM on Web Conference 2025*, pages 2064–2075.

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

Zhichen Zeng, Wenxuan Bao, Xiao Lin, Ruizhong Qiu, Tianxin Wei, Xuying Ning, Yuchen Yan, Chen Luo, Monica Xiao Cheng, Jingrui He, and 1 others. 2026. Subspace alignment for vision-language model test-time adaptation. *arXiv preprint arXiv:2601.08139*.

Zhichen Zeng, Boxin Du, Si Zhang, Yinglong Xia, Zhining Liu, and Hanghang Tong. 2024a. Hierarchical multi-marginal optimal transport for network alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 16660–16668.

Zhichen Zeng, Mengyue Hang, Xiaolong Liu, Xiaoyi Liu, Xiao Lin, Ruizhong Qiu, Tianxin Wei, Zhining Liu, Siyang Yuan, Chaofei Yang, and 1 others. 2025a. Hierarchical lora moe for efficient ctr model scaling. *arXiv preprint arXiv:2510.10432*.

Zhichen Zeng, Xiaolong Liu, Mengyue Hang, Xiaoyi Liu, Qinghai Zhou, Chaofei Yang, Yiqun Liu, Yichen Ruan, Laming Chen, Yuxin Chen, and 1 others. 2025b. Interformer: Effective heterogeneous interaction learning for click-through rate prediction. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pages 6225–6233.

Zhichen Zeng, Ruizhong Qiu, Wenxuan Bao, Tianxin Wei, Xiao Lin, Yuchen Yan, Tarek F Abdelzaher, Jiawei Han, and Hanghang Tong. 2025c. Pave your own path: Graph gradual domain adaptation on fused gromov-wasserstein geodesics. *arXiv preprint arXiv:2505.12709*.

Zhichen Zeng, Ruizhong Qiu, Zhe Xu, Zhining Liu, Yuchen Yan, Tianxin Wei, Lei Ying, Jingrui He, and Hanghang Tong. 2024b. Graph mixup on approximate gromov–wasserstein geodesics. In *Forty-first International Conference on Machine Learning*.

Zhichen Zeng, Si Zhang, Yinglong Xia, and Hanghang Tong. 2023a. Parrot: Position-aware regularized optimal transport for network alignment. In *Proceedings of the ACM web conference 2023*, pages 372–382.

Zhichen Zeng, Ruike Zhu, Yinglong Xia, Hanqing Zeng, and Hanghang Tong. 2023b. Generative graph dictionary learning. In *International Conference on Machine Learning*, pages 40749–40769. PMLR.

Yuheng Zhang, Dian Yu, Tao Ge, Linfeng Song, Zhichen Zeng, Haitao Mi, Nan Jiang, and Dong Yu. 2025. Improving llm general preference alignment via optimistic online mirror descent. *arXiv preprint arXiv:2502.16852*.

Lecheng Zheng, Baoyu Jing, Zihao Li, Hanghang Tong, and Jingrui He. 2024. Heterogeneous contrastive learning for foundation models and beyond. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pages 6666–6676.

Kaijie Zhu, Jindong Wang, Jiaheng Zhou, Zichen Wang, Hao Chen, Yidong Wang, Linyi Yang, Wei Ye, Yue Zhang, Neil Gong, and 1 others. 2023. Promptrobust: Towards evaluating the robustness of large language models on adversarial prompts. In *Proceedings of the 1st ACM workshop on large AI systems and models with privacy and safety analysis*, pages 57–68.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*.

Appendix

A Experiments

A.1 Examples of Token Misalignment

We provide intuitive examples of token misalignment caused by different ensemble methods in Table 4, validating that existing ensemble method may generate suboptimal or even irrelevant token mappings. Capable of handling token misalignment, CoRE consistently improves the performance of existing ensemble methods.

Table 4: Examples of misaligned tokens from source LLM (`OpenChat`) to mapped LLM (`InternLM`).

Method	Source Token	Mapped Token
MinED	`'riv'`	`'_IV'`
UniTE	`'_ind'`	`'indows'`
EVA	`'equation'`	`'align'`

A.2 Adapting CoRE to DeePEn

To adapt CoRE for DeePEn that ensembles models in a latent embedding space, we normalize the unified token embeddings and transform them into probability distributions which CoRE can operate on before ensemble. This adaptation is natural because DeePEn, rather than projecting tokens of assist LLMs into the vocabulary space of the main LLM, essentially projects tokens of ensembled LLMs into a joint latent embeddings space on top of common tokens across different LLMs. By interpreting the normalized embeddings as token probability vectors, CoRE can be integrated seamlessly into embedding-based ensemble methods such as DeePEn.

B Additional Studies

B.1 Observation on Model Consistency

In addition to observations in 3.2, we directly evaluate how model score, measured by the quotient of model consistency and confidence, i.e., $s^{\mathrm{m}}_{\mathrm{assist}_i} = \sum_{v \in V_{\mathrm{main}}} \boldsymbol{s}^{\mathrm{t}}_{\mathrm{assist}_i}(v) / H(\tilde{\boldsymbol{p}}_{\mathrm{assist}_i})$, indicates the correctness of responses. As shown in Figure 14, correct answers exhibit higher model score than the wrong ones. A one-sided test of $H_0 : \mathbb{E}[s^{\mathrm{m}}_{\mathrm{assist}_i}|\mathrm{wrong}] \geq \mathbb{E}[s^{\mathrm{m}}_{\mathrm{assist}_i}|\mathrm{correct}]$ yields a p-value of 2.5×10^{-79}, confirming that correct answers have larger model scores than the wrong ones. We summarize this into Observation 4, which further validates our design of the model score.

Observation 4 (Model Score)**.** *Large model score measured by the quotient of model consistency and confidence signals correct answer.*

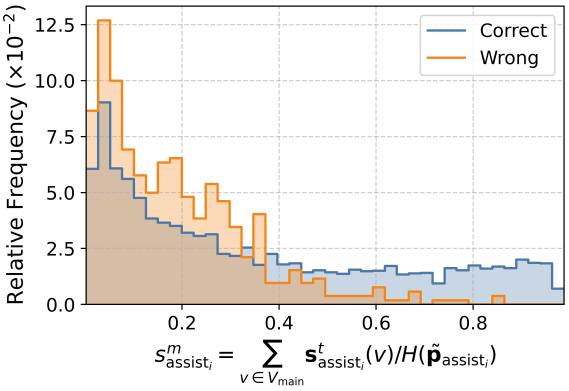


Figure 14: Model score visualization.

B.2 Additional Case Studies

We present additional case studies on the Comprehensive Examination dataset (`MMLU`), Summarization dataset (`SAMSum`), and Reasoning dataset (`GSM8K`) to provide a more intuitive understanding of how CoRE operates. As illustrated in Figure 15, correct answers are highlighted in green and incorrect ones in red. CoRE successfully produces the correct answers where both individual base models and the vanilla ensemble fail.

C Datasets

GSM8K (Cobbe et al., 2021) is a multi-step arithmetic reasoning dataset consisting of 1,319 linguistically diverse grade-school math word problems authored by humans. Each problem requires explicit intermediate reasoning to reach a numeric answer. Models are prompted with four chain-of-thought exemplars, and the final prediction is deemed correct only if the predicted numeric answer exactly matches the gold label. We use accuracy as the evaluation metric.

PIQA (Bisk et al., 2020) is a benchmark for physical commonsense reasoning. Each of its 1,838 examples presents a naturalistic physical situation and two possible solutions, requiring models to choose the more plausible one. It evaluates a model's ability to reason about everyday physical interactions. We use exact match as the evaluation metric.

SAMSum (Gliwa et al., 2019) is a dialogue summarization dataset composed of multi-turn conversations between fictitious participants. The task requires models to generate concise and coherent summaries that capture key information. No in-context examples are used. We use rouge-1 score as the evaluation metric.

TriviaQA (Joshi et al., 2017) contains 11,313 open-domain factoid questions authored by trivia enthusiasts, paired with Wikipedia-based ground-truth answers. Following prior work, each example includes five in-context QA exemplars, and model accuracy is measured by exact match between predicted and reference answers. We use exact match as the evaluation metric.

NQ (Kwiatkowski et al., 2019) consists of 3,610 real anonymized Google search queries paired with short answers from Wikipedia articles. We follow prior work using 5-shot in-context prompting. We use exact match as the evaluation metric.

MMLU (Hendrycks et al., 2020) (Massive Multitask Language Understanding) is a 57-subject multiple-choice benchmark spanning STEM, humanities, and social sciences. Each question has four answer choices, and models are evaluated under 5-shot settings with 5,000 test examples. We use exact match as the evaluation metric.

D More Related Works

Pre-trained Foundation Models. The landscape of NLP has been revolutionized by the emergence of LLMs trained on massive corpora (Achiam et al., 2023; Touvron et al., 2023; Jiang et al., 2023a), with wide applications in reasoning (Lin et al., 2024a, 2025a), language understanding (Team et al., 2023; Team, 2024), and many more (Li et al., 2025a,b). However, due to differences in training data, architectures, and tokenization schemes, these models exhibit diverse strengths and weaknesses across different tasks (Wang et al., 2023; Team, 2023). For instance, some models may excel at mathematical reasoning (Guo et al., 2025; Team et al., 2023) while others specialize in dialogue or summarization (Achiam et al., 2023). The heterogeneity among foundation models serves as the primary motivation for test-time ensemble strategies, which aim to integrate their complementary capabilities to achieve superior performance and reliability.

Model Ensemble and Alignment. In the era of big data and AI, machine learning models are inherently heterogeneous (Zheng et al., 2024; Zeng et al., 2025a,b), characterized by diverse model architectures (Hochreiter and Schmidhuber, 1997; Vaswani et al., 2017; Dosovitskiy, 2020), data modalities (Yan et al., 2021b,a, 2022; Zeng et al., 2023b, 2024b, 2025c), and distinct tokenization frameworks (Sennrich et al., 2016; Kudo and Richardson, 2018; Wu et al., 2016). Model ensemble is a well-established paradigm to combine various model capabilities for improved performance, with wide applications in multi-modal learning (Zeng et al., 2026; Ning et al., 2025), language models (Huang et al., 2024; Yao et al., 2024; Yu et al., 2024), time-series analysis (Liu et al., 2025; Lin et al., 2024b, 2025b), recommendation (Liu et al., 2024b; Liang et al., 2025; Yoo et al., 2024), social analysis (Xu et al., 2024a; Yan et al., 2024b,a, 2025) and many more. A prerequisite for ensembling heterogeneous models is to bridge their gaps via alignment (Zeng et al., 2023a, 2024a; Yu et al., 2025b,a). Specifically, token alignment ensures that probability distributions from heterogeneous LLMs are mapped onto a unified space before fusion. Early approaches align token spaces via minimum edit distance (Fu et al., 2023; Wan et al., 2024; Mavromatis et al., 2024), which captures structural differences but often incurs high computational overhead. Subsequent methods focus on vocabulary mapping in the text space (Yu et al., 2024; Yao et al., 2024) by constructing mappings based on exact or prefix matches between token strings. Recent works utilize token embeddings to exploit semantic information that text matching may fail to capture (Huang

et al., 2024; Xu et al., 2024c), learning projection functions or cross-model embeddings to transform heterogeneous token distributions into a shared representation space.

LLM Robustness. Ensuring model robustness is critical for LLM deployment in real-world applications. Existing research largely focuses on safety against adversarial attacks and stability under distribution shifts. Adversarial robustness studies show that LLMs are vulnerable to jailbreak prompts (Wei et al., 2023; Lin et al., 2026) or perturbations that trigger unintended behaviors (Zou et al., 2023). Meanwhile, out-of-distribution (OOD) robustness aims to maintain performance amidst noisy inputs or domain shifts (Zhu et al., 2023). Common mitigation techniques range from safety alignment (RLHF) (Bai et al., 2022; Zhang et al., 2025) to test-time interventions like uncertainty quantification (Kadavath et al., 2022; Lin et al., 2023). While most existing works focus on individual model, we study the multi-model setting, showing that consistency constraints can serve as a robustification mechanism to mitigate the impact of unreliable signals within an ensemble.

E Potential Risks

The proposed CoRE, while enhancing robustness and reliability through token- and model-level consistency evaluation, also introduce potential risks that warrant careful consideration. CoRE relies on access to fine-grained token probability distributions from individual models to compute consistency scores, which limits its applicability to open-weight or transparent systems. When applied to closed-source or API-based models, this requirement may lead to implementation incompatibility or potential breaches of usage policies.

F Use Or Create Scientific Artifacts

Our work is built on public benchmarks and contributes new code resources to the community. For evaluation, we use widely adopted datasets including `GSM8K`, `PIQA`, `SAMSum`, `TriviaQA`, `NQ`, and `MMLU`, without making any changes to their original content. In addition, we introduce and release the codebase of CoRE at `https://github.com/zhichenz98/CoRE-EACL26`, providing a clear and well-structured repository to improve the accessibility of our research.

F.1 Cite Creators Of Artifacts

All external artifacts are properly credited to their original publications and repositories.

All benchmarks used in this work, including `GSM8K` (Cobbe et al., 2021), `PIQA` (Bisk et al., 2020), `SAMSum` (Gliwa et al., 2019), `TriviaQA` (Joshi et al., 2017), `NaturalQuestions` (Kwiatkowski et al., 2019), and `MMLU` (Hendrycks et al., 2009), are credited to their respective authors.

Each LLM model used in this work, including `Llama-3-8B-Instruct` (Dubey et al., 2024), `Mistral-7B-Instruct-v0.1` (Jiang et al., 2023a), `Qwen2.5-3b-Instruct` (Team, 2024), `InternLM2.5-7b-Chat` (Team, 2023) and `openchat-3.5-0106` (Wang et al., 2023), is referenced through its official technical report or HuggingFace model card to ensure appropriate acknowledgment of all upstream contributions.

F.2 Discuss The License For Artifacts

We comply with the licenses of all artifacts used or released in this work. The benchmarks are distributed under Creative Commons or public-domain terms, following the conditions specified by their original maintainers. Model checkpoints retain their original open-source licenses, and our own code and generated data are released under the MIT license. We will release our code upon publication.

F.3 Artifact Use Consistent With Intended Use

We confirm that our use of datasets and pre-trained models is consistent with their intended purpose. Benchmarks are used solely for inference and evaluation, which aligns with their terms of service, and no modified model weights are redistributed. The released code is for inference only and does not support fine-tuning or commercial redistribution of the original checkpoints.

F.4 Documentation Of Artifacts

We provide detailed documentation of all evaluation datasets and model combinations used in our experiments. Specifically, Appendix C describes the coverage, task type, and evaluation metric for each dataset (e.g., reasoning, summarization, and knowledge QA), while Appendix H the LLMs included in the ensemble along with their sizes and sources. Together, these materials ensure transparency regarding the domains, data characteristics, and model diversity involved in our study.

G Statistics For Data

G.1 **GSM8K** (Arithmetic Reasoning)

- Number of entries: 1319
- Average question length: 239.9 characters
- Average answer length: 272.3 characters

G.2 **PIQA** (Commonsense Reasoning)

- Number of entries: 1838
- Average question length: 36.1 characters
- Average answer length: 1.0 character

G.3 **SAMSum** (Summarization)

- Number of entries: 819
- Average question length: 521.6 characters
- Average answer length: 108.8 characters

G.4 **TriviaQA** (Knowledge)

- Number of entries: 6000
- Average question length: 78.8 characters
- Average answer length: 267.9 characters

G.5 **NaturalQuestions** (Knowledge)

- Number of entries: 3610
- Average question length: 47.7 characters
- Average answer length: 24.7 characters

G.6 **MMLU** (Comprehensive Examination)

- Number of entries: 14042
- Average question length: 274.5 characters
- Average answer length: 1.0 character

H Computational Experiments

All computational experiments in this work are fully reproducible, with details provided in the main text and the Appendix. Our code is available at https://github.com/zhichenz98/CoRE-EACL26.

H.1 Model Size And Budget

For each LLM used, we specify its total parameter count as follows: Llama-3-8B-Instruct contains around 8 billion parameters; Mistral-7B-Instruct-v0.1, InternLM2.5-7b-Chat, and openchat-3.5-0106 contains around 7 billion parameters; Qwen2.5-3b-Instruct contains around 3 billion parameters. The total compute budget for all experiments is approximately 500 A100 GPU hours.

H.2 Experiment Setup And Hyperparameters

We describe experimental settings for all experiments in Section 4.1. For hyperparameter settings, we set $\sigma = 0.5$ in the RBF kernel, $\alpha = 1.0, \beta = 1.0$ in the power function, and $\gamma = 1.0$ in the sigmoid function. We clip the normalized model weight for the main LLM to be at least 0.5 to ensure its major contribution to the ensemble results.

H.3 Descriptive Statistics

For each result in the main text and Appendix, we report the mean across multiple runs.

H.4 Parameters For Packages

The existing packages used are specified as follows. We use PyTorch (v2.8.0) as the core deep learning framework, together with HuggingFace Transformers (v4.56.1) and Tokenizers (v0.22.0) for model implementation and text pre-processing. For efficient multi-GPU and distributed training we rely on Accelerate (v1.10.1).

Example 2 (Comprehensive Examination)

Question: A person weighs 62 kg. Their drug dose is 15 mg/kg. How many grams is their dose? Choose one answer from the following:
A. 930 B. 93 C. 9.3 D. 0.93
Gold Answer: D
InternLM Response: C ✗
Qwen Response: C ✗
Vanilla Response: C ✗
CoRE Response: D ✓

Example 3 (Summarization)

Question: Generate a concise summary for the chat:
Kelly: Should I put on the red dress?
Meghan: oh yes!!!!
Mary: I'll wear a red dress too, and bright, red lipstick
Kelly: bloody Mary!
Mary: hahahaha
Gold Answer: Kelly and Mary will wear red dresses. Mary will wear a red lipstick too.
OpenChat Response: Kelly, Meghan, and Mary all decide to wear red dresses and bright red lipstick. (Rouge Score=0.55)
Mistral Response: Kelly is considering wearing a red dress and Mary suggests she wear red lipstick to go with it. (Rouge Score=0.64)
Vanilla Response: Kelly is considering wearing a red dress and Mary suggests wearing a red dress and red lipstick. Kelly responds with "bloody Mary". (Rouge Score=0.55)
CoRE Response: Kelly is considering wearing a red dress and Mary will also wear a red dress and red lipstick. (Rouge Score=0.73)

Example 4 (Reasoning)

Question: John plans to sell all his toys and use the money to buy video games. He has 13 lego sets and he sells them for $15 each. He ends up buying 8 video games for $20 each and has $5 left. How many lego sets does he still have?
Gold Answer: He spent $160 on video games because 8 x 20 = 160
He earned $165 from his sale because 160 + 5 = 165
He sold 11 lego sets because 11 x 15 = 165
He has 2 sets left because 13 - 11 = 2
Therefore, the answer is 2
InternLM Response: John sells 13 lego sets for $15 each, so he earns 13*15 = $195
He buys 8 video games for $20 each, so he spends 8*20 = $160
He has $5 left, so the total amount of money he earned is $160 + $5 = $165
Since he earned $195 from selling his lego sets, the number of lego sets he still has is $195 - $165 = $30
The answer is 30 ✗
OpenChat Response: John sells his 13 lego sets for $15 each, so he makes 13*$15=$195
He uses this money to buy 8 video games for $20 each, so he spends 8*$20=$160
He has $5 left, so he spent $195-$160-$5=$30 on lego sets
Since he sells each lego set for $15, he sells 30/15=2 lego sets
He started with 13 lego sets and sold 2, so he still has 13-2=11 lego sets
The answer is 11 ✗
Vanilla Response: John sells 13 lego sets for $15 each, so he earns 13*15 = $195
He buys 8 video games for $20 each, so he spends 8*20 = $160
He has $5 left, so he spent a total of 195-5 = $190 on lego sets
Since he sold 13 lego sets for $15 each, he must have had 190/15 = 13 lego sets
The answer is 13 ✗
CoRE Response: John sells 13 lego sets for $15 each, so he earns 13*15 = $195 from selling the lego sets.
He buys 8 video games for $20 each, so he spends 8*20 = $160 on video games.
He has $5 left, so he spent a total of 195-160-5 = $30 on other things.
Since each lego set is sold for $15, he still has 30/15 = 2 lego sets left.
The answer is 2 ✓

Figure 15: Additional case study across comprehensive examination, summarization, and reasoning benchmarks. Correct and incorrect responses are highlighted in green and red, respectively.