
When to Ensemble: Identifying Token-Level Points for Stable and Fast LLM Ensembling

Heecheol Yun, Kwangmin Ki, Junghyun Lee, Eunho Yang

ICLR 2026

What is LLM Ensemble?

- Each LLM exhibits unique strengths derived from its specific training recipe, with no single model achieving universal dominance across all domains
- Ensembling effectively combines the complementary strengths of multiple models during inference!
 - **Highly Efficient** compared to training a new oracle model
 - **Superior performance** compared to an individual model

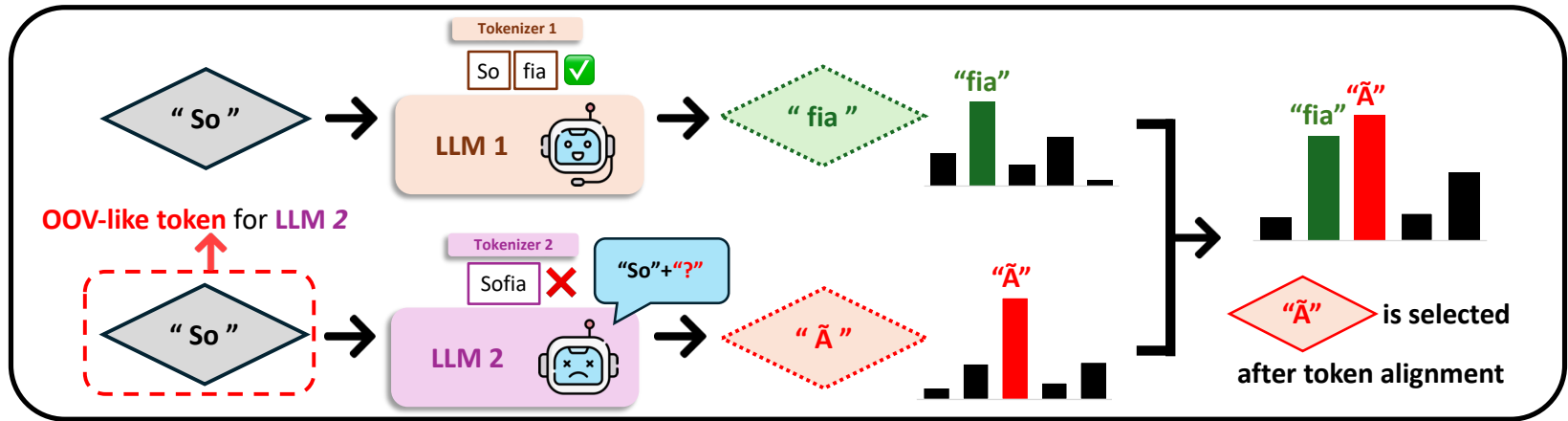
What is LLM Ensemble?

- Each LLM exhibits unique strengths derived from its specific training recipe, with no single model achieving universal dominance across all domains
- Ensembling effectively combines the complementary strengths of multiple models during inference!
 - **Highly Efficient** compared to training a new oracle model
 - **Superior performance** compared to an individual model
- We focus on ***probability-level ensemble***, which aggregates the next-token probability distributions of multiple LLMs to select the most confident token
 - Facilitates collaboration across **diverse model architectures**
 - Leverages the collective knowledge **embedded within their probability distributions**

Motivation

- In **long-form generation**, we observe that the effectiveness of ensembling critically depends on *when to ensemble*
 - Ensembling at every token position leads to unstable outputs and diminishing generation speeds, especially for long-form generation
- To address this, two key factors should be considered:
 1. **Tokenization mismatch** among the participating models
 2. **Consensus in their next-token probability distributions**

Factor 1: Tokenization mismatch is crucial for stability



- Ensemble operations can select **“OOV-like tokens”** that conflict with the tokenization schemes of participating models
- These conflicts force models into out-of-distribution (OOD) states, which distorts their next-token probability distributions and hinders *probability-level ensembling*

Factor 1: Tokenization mismatch is crucial for stability

Problem

Terry eats 2 yogurts a day. They are currently on sale at 4 yogurts for \$5.00. How much does he spend on yogurt over 30 days?

Answer

To determine how much Terry spends on yogurt over 30 days, we can break down the problem into several steps:

1. Calculate the daily yogurt consumption in terms of the sale unit: - Terry eats 2 yogurts per day. - The sale is 4 yogurts for \$5.00. - Therefore, Terry consumes ($2/4 = 0.5$) of a sale unit per day.
 2. Determine the daily cost based on the sale price: - Since 4 yogurts cost \$5.00, 1 yogurt costs ($5/4 = 1.25$). - **Therefore, 0.5 of a sale unit (which is equivalent to 2 yogurts) costs ($0.5 \times 1.25 = 0.625$).** - So, Terry spends \$0.625 per day on yogurt.
 3. Calculate the total cost over 30 days: - Terry spends \$0.625 per day. - Over 30 days, the total cost is ($0.625 \times 30 = 18.75$).
- Therefore, the total amount Terry spends on yogurt over 30 days is `\\boxed{18.75}`.

Wrong Information ✘

Problem

Sofia and Tess will each randomly choose one of the 10 integers from 1 to 10. What is the probability that neither integer chosen will be the square of the other?

Answer

To determine the probability that neither integer chosen by Sofia and Tess will be the square of the other, we need to follow these steps: ... 2. Determine the total number of possible outcomes: Each of Sofia and Tess ...

Unnatural Token Repetition ✘

Answer

****Option A**:** **Incorret** because the coronary arteries However, the **gallbladde r** does not **se crete insulin** ... **nourishment** or feeding, ... which involves the **digesive** system ...

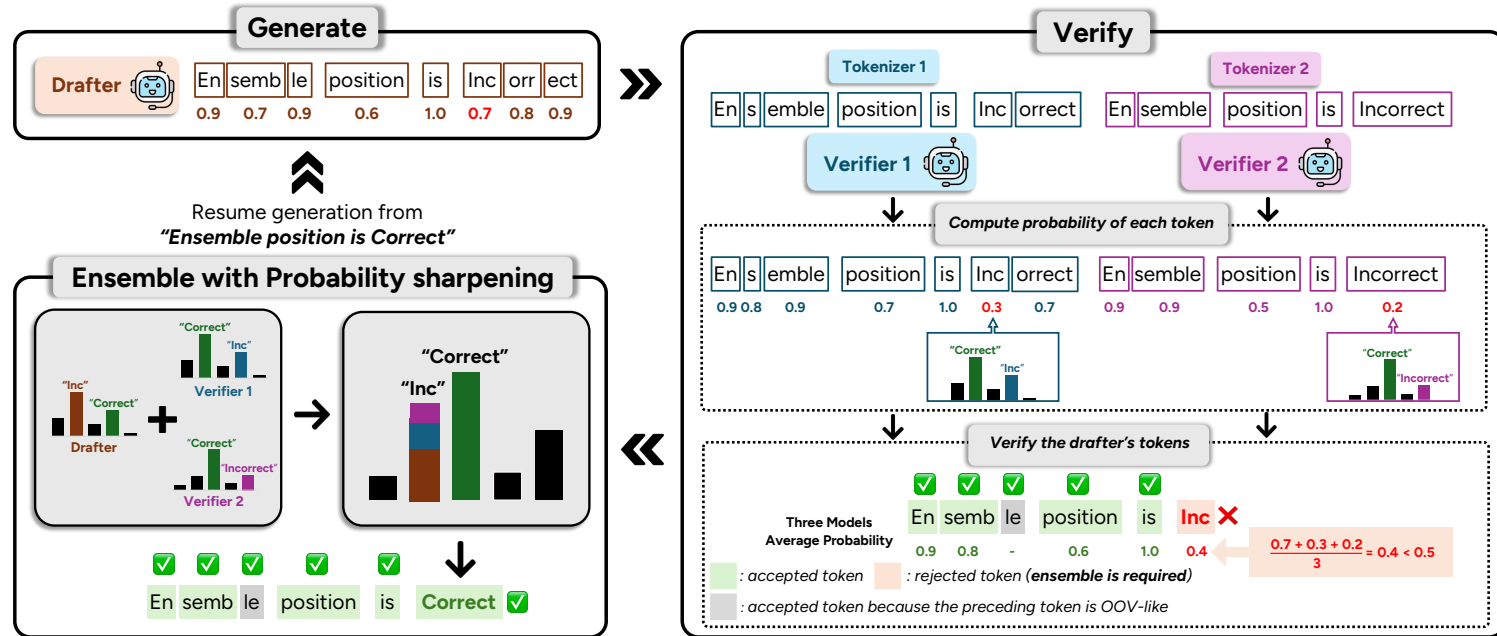
Typo ✘

- Repeated introduction of OOV-like tokens destabilizes ensemble outputs, introducing severe artifacts such as **unnatural token repetition** or **typos**

Factor 2: Consensus in models' next-token distributions relates to efficiency

- Since each LLM has heterogeneous vocabulary spaces, ensemble operations require **aligning these different vocabularies into a shared vocabulary space**
- Given the massive vocabulary sizes of modern LLMs, this vocabulary alignment **imposes latency as sequence length increases**
- However, **when individual next-token probability distributions exhibit sufficient consensus, we determine the most confident token directly**, bypassing expensive alignment operations

Method: Stable And Fast LLM Ensembling (SAFE)



- SAFE iterates a three-step cycle:
 1. **Generate:** The drafter produces a lookahead sequence of tokens
 2. **Verify:** The verifiers evaluate the drafted tokens against two criteria to pinpoint ensembling positions:
 - i) **OOV-like Token Verification:** Ensuring no OOV-like tokens are introduced
 - ii) **Ensemble Distribution Verification:** Detecting insufficient agreement across verifiers
 3. **Ensemble:** Ensembling is applied only at the tokens validated in the Verify step with probability sharpening

Experiments: Experimental Settings

- **Models**

- Interlm3-8B-Instruct, Qwen2.5-7B-Instruct, EXAONE-3.5-7.8B-Instruct
- These three LLMs have heterogeneous tokenization schemes (40-60% tokenization agreement rates on Oxford 5000 words)

- **Benchmarks**

- **General Reasoning:** MMLU-redux, ARC-Challenge, BBH
- **Mathematical Reasoning:** MATH500, GSM8K

- **Baselines**

- We apply our method to two SOTA probability-level ensemble methods: **GaC (EMNLP'24)** and **UniTE (ICLR'25)**
- “**X + SAFE**” means that the ensemble method **X** is applied only at the token positions that SAFE identifies as requiring ensembling

- **Metrics**

- Accuracy (%)
- E/T (%): Ensemble frequency (# Ensemble / # Token)

Experiments: Results

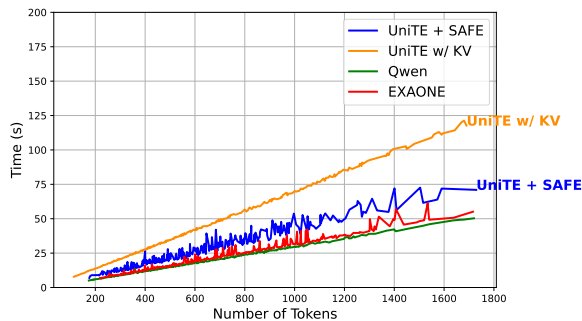
Method	MMLU-redux		MATH500		GSM8K		BBH		ARC-C		Avg.
	Accuracy	E/T	Accuracy	E/T	Accuracy	E/T	Accuracy	E/T	Accuracy	E/T	Accuracy
Internlm3-8B	76.89	-	74.8	-	90.14	-	82.26	-	90.27	-	82.87
Qwen2.5-7B	74.88	-	72.4	-	91.81	-	79.15	-	88.74	-	81.40
EXAONE3.5-7.8B	73.25	-	72.8	-	90.45	-	78.75	-	90.44	-	81.14
<i>Two-model ensembling (Internlm3 + Qwen2.5)</i>											
GaC	77.00 (+0.11)	8.43	74.2 (-0.6)	1.04	91.28 (-0.53)	0.82	82.34 (+0.08)	5.69	90.61 (+0.34)	10.22	83.09 (+0.22)
GaC + SAFE	77.11 (+0.22)	5.23	76.0 (+1.2)	0.71	91.36 (-0.45)	0.67	82.34 (+0.08)	3.73	91.13 (+0.86)	6.22	83.59 (+0.72)
UniTE	73.39 (-3.5)	100	59.6 (-15.2)	100	75.06 (-16.75)	100	79.58 (-2.68)	100	87.97 (-2.30)	100	75.12 (-7.75)
UniTE + SAFE	77.81 (+0.92)	12.59	77.4 (+2.6)	3.82	92.04 (+0.23)	5.16	82.97 (+0.71)	10.35	90.78 (+0.51)	14.47	84.20 (+1.33)
<i>Two-model ensembling (Qwen2.5+ EXAONE3.5)</i>											
GaC	76.01 (+1.13)	13.42	75.4 (+2.6)	2.31	92.65 (+0.84)	2.60	79.61 (+0.46)	8.15	90.27 (-0.17)	14.66	82.79 (+1.39)
GaC + SAFE	76.79 (+1.91)	7.52	76.4 (+3.6)	1.09	92.57 (+0.76)	1.26	79.66 (+0.51)	4.51	90.78 (+0.34)	8.31	83.24 (+1.84)
UniTE	53.75 (-21.13)	100	43.4 (-29.4)	100	77.03 (-14.78)	100	67.45 (-11.70)	100	72.61 (-17.83)	100	62.85 (-18.55)
UniTE + SAFE	76.54 (+1.66)	17.24	76.4 (+3.6)	4.69	92.72 (+0.91)	5.60	81.69 (+2.54)	14.03	90.78 (+0.34)	19.24	83.63 (+2.23)
<i>Two-model ensembling (Internlm3 + EXAONE3.5)</i>											
GaC	76.36 (-0.53)	8.71	75.8 (+1.0)	1.14	90.75 (+0.30)	0.88	81.57 (-0.69)	6.32	90.78 (+0.34)	10.07	83.05 (+0.18)
GaC + SAFE	77.21 (+0.32)	5.94	77.2 (+2.4)	0.84	90.67 (+0.22)	0.72	81.54 (-0.72)	4.38	91.72 (+1.28)	6.92	83.67 (+0.80)
UniTE	72.51 (-4.38)	100	73.6 (-1.2)	100	89.31 (-1.14)	100	78.04 (-4.22)	100	88.23 (-2.21)	100	80.34 (-2.53)
UniTE + SAFE	76.08 (-0.81)	15.84	77.0 (+2.2)	4.72	90.75 (+0.30)	5.55	81.37 (-0.89)	13.75	90.27 (-0.17)	17.89	83.09 (+0.22)
<i>Three-model ensembling (Internlm3 + Qwen2.5 + EXAONE3.5)</i>											
UniTE	73.92 (-2.97)	100	76.0 (+1.2)	100	91.28 (-0.53)	100	77.47 (-4.79)	100	87.20 (-3.24)	100	81.17 (-1.70)
UniTE + SAFE	77.60 (+0.71)	16.18	79.0 (+4.2)	4.12	92.04 (+0.23)	5.14	82.77 (+0.51)	12.74	91.55 (+1.11)	18.60	84.59 (+1.72)

- SAFE improves performance with less ensembling **under CoT**

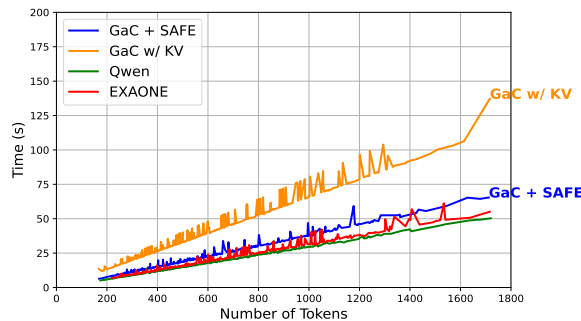
Experiments: Results

Method	MMLU-redux	MATH500	GSM8K
Qwen2-7B	69.25	49.8	85.90
Llama3.1-8B	68.51	47.6	82.56
<i>Two-model ensembling (Qwen2 + Llama3.1)</i>			
GaC	69.50 (+0.25)	52.4 (+2.6)	85.37 (-0.53)
GaC + SAFE	69.99 (+0.74)	59.4 (+9.6)	86.66 (+0.76)
UniTE	68.90 (-0.35)	54.0 (+4.2)	79.98 (-5.92)
UniTE + SAFE	69.71 (+0.46)	55.6 (+5.8)	84.08 (-1.82)

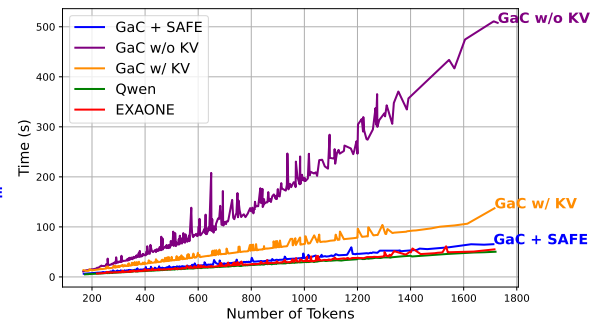
- SAFE further improves performance when **ensembling models with similar tokenization (e.g., Qwen + Llama)**



(a) Latency comparison of UniTE with and w/o SAFE



(b) Latency comparison of GaC with and w/o SAFE



(c) Latency of GaC w/o our KV caching

- SAFE can be as fast as individual models**
- SAFE closely matches the latency of individual models when generating hundreds of tokens

Conclusion

- We showed that probability-level ensembling should occur at appropriate token positions, especially when generating long sequences
- We proposed SAFE, a method that determines these positions by jointly considering the two key factors
- SAFE offers the following three advantages
 1. **Efficiency:** SAFE improves efficiency through a combination of selective ensembling and a speculative strategy
 2. **Stability:** SAFE guarantees generation stability by preventing OOV-like tokens from being fed into the participating models
 3. **Plug-and-Play:** SAFE can be seamlessly integrated with existing ensemble methods

Thank you!