

# 000 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 INFEERENCE SCALING OF LLM ENSEMBLING: BRIDGING TOKEN SPACES WITH TOKEN TRANSLATION

Anonymous authors

Paper under double-blind review

## ABSTRACT

Large language models (LLMs) exhibit diverse strengths and weaknesses across tasks, motivating recent efforts to ensemble multiple models to harness their complementary capabilities, boosting test-time performance. While model diversity and capability are known to influence ensemble effectiveness, a persistent challenge in LLM ensembling arises from mismatched tokenizer vocabularies. Existing alignment strategies typically rely on token-level embeddings or string-level heuristics of tokens, overlooking the tokenizer priors embedded during LLM pretraining. Specifically, tokenizers such as Byte-Pair Encoding (BPE) and Unigram are constructed by statistically analyzing large pretraining corpora to identify frequent subword units, and they tokenize text using greedy or probabilistic algorithms that reflect these learned subword distributions. In this work, we propose a novel and remarkably simple *Token Translation* (ToT) method that explicitly leverages these tokenizer priors to bridge heterogeneous token spaces. Our method is lightweight, requiring only a few lines of code, pre-computable, and highly efficient at inference. To further enhance robustness, we incorporate token-level model uncertainty to dynamically reweight each model’s contribution during decoding. Extensive evaluations across diverse model combinations and tasks demonstrate that our method consistently outperforms existing ensembling baselines.

## 1 INTRODUCTION

Large language models (LLMs) (Touvron et al., 2023; OpenAI, 2023) have demonstrated strong performance across tasks such as question answering (Kamalloo et al., 2023), summarization (Zhang et al., 2024), and reasoning (Jaech et al., 2024; Guo et al., 2025). Owing to differences in data, architectures, and training objectives, LLMs exhibit complementary strengths and weaknesses (Jiang et al., 2023b; Yao et al., 2025). This motivates *test-time ensembling*, a practical approach to combine multiple pretrained LLMs for improved performance and adaptability without additional fine-tuning.

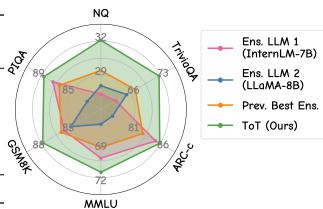
Test-time ensembling methods generally fall into two categories: response-level (Feng et al., 2024; Jiang et al., 2023b; Lu et al., 2023) and logit-level (or token-level) ensembling (Yao et al., 2025; Yu et al., 2024; Huang et al., 2024; Phan et al., 2025). Response-level approaches select among complete outputs or route inputs to a model, offering a simple but coarse way to combine LLMs. Logit-level approaches instead operate directly on token (or sub-token) distributions during generation, enabling finer-grained integration of model predictions. However, they face a key challenge: *tokenizer mismatch*, as models often adopt heterogeneous tokenization schemes and vocabularies, making the alignment of token-level predictions nontrivial.

LLMs adopt distinct tokenization schemes (e.g., BPE (Sennrich et al., 2015), Unigram (Kudo, 2018)), each trained on a separate pretraining corpus. These schemes differ in segmentation granularity, vocabulary, and subword decomposition, resulting in heterogeneous token spaces. Such mismatches prevent direct alignment of token-level distributions across models and can lead to sparse or unstable mappings, especially when vocabularies only partially overlap.

Prior works attempted to align tokens across models by comparing token embedding similarities (Huang et al., 2024) or using string-level heuristics such as string distance (Wan et al., 2024a). They fail due to inconsistent token boundaries and lack of grounded semantic correspondence. However, these strategies often overlook a critical source of information: *tokenizer priors*—the statistical

054  
 055  
 056  
 057  
 058 Table 1: Comparisons of LLM ensemble methods. **Left panel:** Comparison across five methodological  
 059 dimensions. **Right panel:** Radar plot comparing ToT with the two base LLMs and the strongest ensemble  
 060 baseline across six benchmarks.

Method	Mismatch Handling	Minimal Overhead	Tokenizer Priors	Uncertainty Aware	Bidirectional
LLM-Blender (Jiang et al., 2023b)	✗	✗	✗	✗	✗
GAC (Yu et al., 2024)	✗	✓	✗	✗	✗
DeepEn (Huang et al., 2024)	✓	✗	✗	✗	✗
CES (Phan et al., 2025)	✓	✗	✗	✗	✗
UniTE (Yao et al., 2025)	✓	✓	✗	✗	✗
ToT (Ours)	✓	✓	✓	✓	✓



064 patterns captured during tokenizer training on large corpora. Widely deployed tokenizers like BPE  
 065 and Unigram segment text by greedily selecting high-frequency subword units, and this behavior  
 066 implicitly encodes valuable knowledge about how linguistic input is represented in each LLM.

067 To address the challenge of tokenizer mismatch in token-level LLM ensembling, we propose a  
 068 simple and highly effective alignment strategy that leverages *tokenizer priors*, i.e., the statistical  
 069 patterns implicitly captured by tokenizers during training on large corpora. Our key idea is to align  
 070 heterogeneous token spaces through an operation we term ***token translation***: for any token from  
 071 a source model, we first decode it into its original text form using the source tokenizer, and then  
 072 re-encode this text using the target model’s tokenizer. This reveals how the target tokenizer interprets  
 073 the same content and provides a grounded correspondence between token spaces. Intuitively, it  
 074 approximates the token sequence the target model is likely to generate when expressing the same  
 075 underlying content, capturing both interpretation and generation preferences. To enhance alignment  
 076 quality, we further introduce a bidirectional translation scheme that integrates alignments in both  
 077 directions to capture complementary structures. This approach is easy to implement, pre-computable  
 078 for efficiency, and enables robust token-level fusion across models with diverse vocabularies. Detailed  
 079 comparisons of existing methods are provided in Table 1, with the left panel showing the method-level  
 080 characteristics and the right panel reporting quantitative performance. As highlighted, our proposed  
 081 ToT method consistently checks all the desired properties and achieves the best overall results.

082 After aligning token predictions into a shared space, we address the challenge of aggregating outputs  
 083 from models with varying reliability through an ***uncertainty-aware ensembling*** mechanism. In  
 084 practice, different models may exhibit varying levels of confidence depending on the context, and  
 085 treating them equally can dilute stronger signals. To address this, we dynamically reweigh each  
 086 model’s token distribution based on prediction entropy: models with lower uncertainty get greater  
 087 influence, while uncertain ones are downweighted. This adaptive weighting allows the ensemble to  
 088 emphasize more reliable token-level signals and improves reliability across diverse inputs.

089 We validate our method through extensive experiments across multiple model combinations, and  
 090 downstream tasks. Our results show that the proposed approach consistently outperforms existing  
 091 LLM ensembling methods in terms of accuracy. These results further demonstrate that ensembling  
 092 performance improves as more capable LLMs are incorporated, under optimal configurations using  
 093 our proposed method. Our main contributions are summarized as follows:

- We highlight the untapped potential of tokenizer priors for bridging heterogeneous token spaces and addressing tokenizer mismatch, a key bottleneck in token-level LLM ensembling.
- We introduce a bidirectional token alignment method ToT based on *token translation*, a simple yet effective operation that aligns tokens across models by leveraging their tokenizer behaviors.
- We propose an uncertainty-aware ensembling strategy that adaptively reweights model contributions at each token step based on prediction confidence, enhancing output quality.
- We demonstrate that our method consistently improves performance across diverse tasks and model combinations, achieving an average absolute gain of 5.95 points, offering a scalable, well-founded solution for training-free LLM ensembling.

## 2 RELATED WORKS

104  
 105 Ensembling large language models (LLMs) has emerged as a promising direction for improving  
 106 prediction accuracy, robustness, and calibration. Existing methods can be broadly grouped into three  
 107

108 categories: *response-level*, *logit-level*, and *training-time* ensembling. Our work focuses on token-level  
 109 fusion and introduces a lightweight *token translation* strategy to address tokenizer mismatch while  
 110 preserving fine-grained generation behavior.

111 **Response-level ensembling** aggregates complete outputs from different models. Early methods such  
 112 as PairRanker (Jiang et al., 2023b) and routing-based ensembles (Feng et al., 2024; Lu et al., 2023)  
 113 select the best response based on reranking or input-conditioned selection. More recent techniques  
 114 like SweetSpan (Xu et al., 2024a) attempt to fuse spans across responses. While conceptually simple,  
 115 these approaches ignore token-level information and often struggle to generalize across tasks and  
 116 model combinations.

117 **Logit-level ensembling** operates at a finer granularity by aggregating token logits during generation.  
 118 We use this term broadly to include variants that operate on tokens, sub-tokens (bytes), or token  
 119 spans. UniTE (Yao et al., 2025) uses top- $k$  token union for alignment, while GAC (Yu et al., 2024)  
 120 and DeePen (Huang et al., 2024) rely on dense mapping matrices between vocabularies to perform  
 121 projection-based fusion. Span-level ensembling (Xu et al., 2025) merges consecutive tokens into  
 122 larger segments, but substantially increases computation cost. Sub-token-level ensembling (Gu et al.,  
 123 2024; Phan et al., 2025) such as CES, instead decomposes tokens into smaller units, but suffers  
 124 from a lack of *tokenizer priors*, often requiring expensive search. Our method avoids these issues by  
 125 leveraging tokenizer priors to construct sparse, grounded token mappings via a translation mechanism.

126 **Training-time ensembling** focuses on distilling multiple model outputs into a student model.  
 127 FusELL (Wan et al., 2024a) and FuseChat (Wan et al., 2024b) perform logit-based distillation,  
 128 while EVA (Xu et al., 2024b) and EnsW2S (Agrawal et al., 2024) learn vocabulary projection lay-  
 129 ers to integrate predictions. Ent (Ruan et al., 2022) also explore weighted ensembling, but their  
 130 weights are learned during training, unlike our test-time aggregation of frozen heterogeneous LLMs.  
 131 While effective for model compression, these approaches require model retraining and full access to  
 132 parameters, making them inapplicable in test-time scenarios where models are frozen or proprietary.

### 134 3 METHODOLOGY

135 In this section, we present our test-time ensembling framework for token-level fusion across LLMs  
 136 with heterogeneous tokenizers. We first define the *token alignment* problem arising from vocabulary  
 137 mismatch, then introduce *token translation* to align token spaces, along with a *bidirectional* variant  
 138 for added robustness. Finally, we describe an *uncertainty-aware* ensembling strategy that reweights  
 139 model outputs by confidence. An overview is shown in Figure 1.

#### 142 3.1 PROBLEM FORMULATION: TOKEN ALIGNMENT ACROSS HETEROGENEOUS TOKENIZERS

143 Token-level ensembling across LLMs requires the ability to align token distributions across models  
 144 with heterogeneous tokenizers. We follow the setting adopted by prior work such as UniTE (Yao  
 145 et al., 2025), GAC (Yu et al., 2024), and DeePen (Huang et al., 2024), where a designated *main model*  
 146 serves as the reference, and one or more *assist models* provide auxiliary signals. At each decoding  
 147 step, each model outputs a probability distribution over its own vocabulary:  $\mathbf{p}^{(\text{main})} \in \Delta^{|\mathcal{V}_{\text{main}}|}$  of  
 148 tokenizer  $\text{Tok}_{\text{main}}$  and  $\mathbf{p}^{(\text{assist})} \in \Delta^{|\mathcal{V}_{\text{assist}}|}$  of tokenizer  $\text{Tok}_{\text{assist}}$ , where  $\Delta^{|\mathcal{V}|}$  denotes the probability  
 149 simplex, i.e., the set of non-negative vectors in  $\mathbb{R}^{|\mathcal{V}|}$  that sum to one.

150 The goal is to construct a *mapping matrix*  $\mathbf{M} \in \mathbb{R}^{|\mathcal{V}_{\text{assist}}| \times |\mathcal{V}_{\text{main}}|}$  for each assist model, where  
 151  $\mathbf{M}_{i,j}$  indicates the alignment between assist token  $v_i^{(\text{assist})}$  and main token  $v_j^{(\text{main})}$ . By per-  
 152 forming row-normalization on  $\mathbf{M}$ , we obtain the transition matrix  $\mathbf{T} \in \mathbb{R}^{|\mathcal{V}_{\text{assist}}| \times |\mathcal{V}_{\text{main}}|}$ , where  
 153  $\mathbf{T}_{i,j} = \Pr(v_j^{(\text{main})} | v_i^{(\text{assist})})$  is the probability of transiting assist token  $v_i^{(\text{assist})}$  to main token  $v_j^{(\text{main})}$ .  
 154 Using  $\mathbf{T}$ , the assist model’s probability is projected into the main vocabulary space via  $\mathbf{T}^\top \mathbf{p}^{(\text{assist})}$ .  
 155 Finally, these aligned distributions are ensembled by their linear combination.

#### 158 3.2 TOKEN TRANSLATION VIA DECODE → ENCODE

159 Given the problem setting above, the key challenge lies in constructing a meaningful and efficient  
 160 transition matrix  $\mathbf{T}$ , as the vocabularies  $\mathcal{V}_{\text{main}}$  and  $\mathcal{V}_{\text{assist}}$  from different LLM families are often

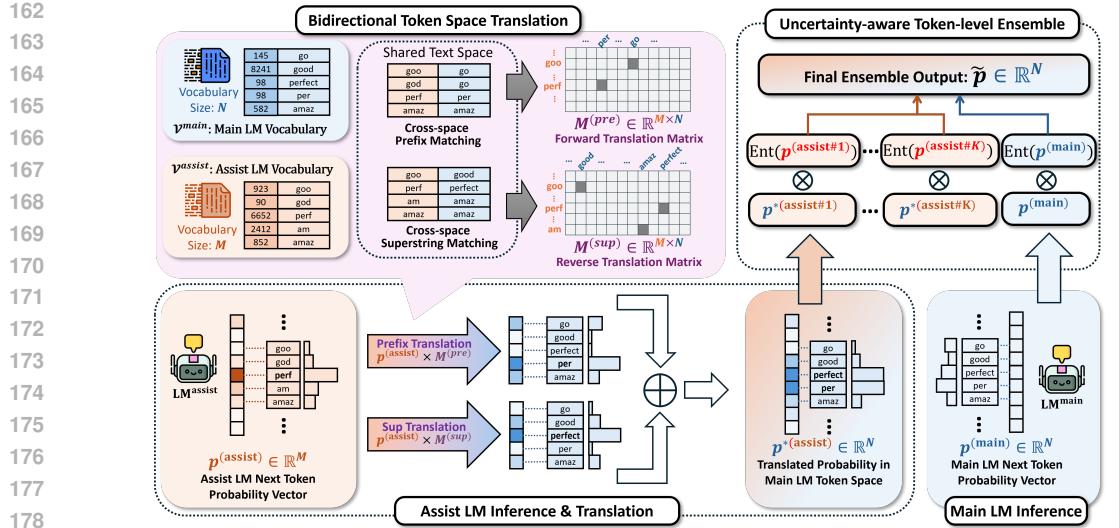


Figure 1: Illustration of our **ToT** framework for token-level ensembling across heterogeneous LLMs. Assist model predictions are translated into the main model’s token space via bidirectional token translation using tokenizer priors. Final predictions are fused using uncertainty-aware weighting for robust ensemble decoding.

only partially overlapping. Traditional approaches based on string matching or embedding similarity frequently fail due to inconsistent token boundaries and the absence of semantic correspondence. To address this, we propose an alignment strategy based on *tokenizer priors*, which leverages the statistical segmentation behavior inherently learned during tokenizer training, and further introduce a simple yet effective mechanism, *token translation*, that uses these priors to construct  $\mathbf{T}$  and establish semantic correspondences across heterogeneous vocabularies.

Tokenizers such as BPE and Unigram are trained on large corpora to segment text into frequent and meaningful subword units. These statistical segmentation behaviors encode valuable priors about how each tokenizer decomposes natural language. Unlike model weights that are optimized for downstream objectives, tokenizer behavior is fixed and deterministic once trained. Therefore, it provides a stable and semantically grounded signal for reasoning about how different models represent and generate language.

Previous alignment strategies often rely on post-hoc heuristics such as string similarity or embedding proximity. However, such approaches fail to capture generation preferences intrinsic to each tokenizer. **Our key insight is that tokenizer segmentation behavior have already encoded a deterministic causal structure for next-token generation, and leveraging these priors leads to more reliable cross-model alignment.** A formal justification and analysis are provided in Appendix C.

#### Example: Failure of Similarity-based Alignment

Suppose the assist tokenizer produces the token `good`, while the main tokenizer includes `go`, `goo`, and `nice`. The intended continuation is `goods`. A surface-similarity heuristic might align `good` to `goo` or even `nice` due to lexical or semantic resemblance. However, this ignores how the main tokenizer actually segments and generates: in practice it may begin with `go`, then continue with `ods` (e.g., `go+od+s`). Mapping `good` to `goo` or `nice` biases decoding toward irrelevant continuations such as `goose` or `nicely`, deviating from the desired `goods`. By respecting tokenizer priors, token translation aligns `good` to the main prefix `go`, preserving the ideal generative priors.

**Method.** To account for such discrepancies, we propose to align tokens through their tokenizer-internal decoding preferences. Specifically, for assist token id  $i$ , we **decode** it into its raw text form  $v_i^{(\text{assist})}$  using the assist tokenizer. We then **re-encode**  $v_i^{(\text{assist})}$  using the main tokenizer to obtain a sequence  $[t_{i,1}^{(\text{main})}, \dots, t_{i,m}^{(\text{main})}]$  of token ids from  $\mathcal{V}_{\text{main}}$ . We map the assist token id  $i$  to the first token

216 id  $t_{i,1}^{(\text{main})}$  in this sequence:

$$217 \quad M_{i,j} = \begin{cases} 1 & \text{if } t_{i,1}^{(\text{main})} = j, \\ 218 & 0 \quad \text{otherwise.} \end{cases}$$

219 Intuitively, this token captures the initial prefix the main model is most likely to generate for the given  
220 content, providing a generation-compatible alignment.

221 This procedure is repeated for all tokens in  $\mathcal{V}_{\text{assist}}$ , yielding a sparse mapping matrix  $M \in$   
222  $\mathbb{R}^{|\mathcal{V}_{\text{assist}}| \times |\mathcal{V}_{\text{main}}|}$ . Each row has a single nonzero entry:  $M_{i,j} = 1$  if  $v_i^{(\text{assist})}$  aligns with  $v_j^{(\text{main})}$ . This  
223 sparsity makes the alignment efficient to store and apply, and it reflects a semantically informed,  
224 generation-aware projection from the assist to the main vocabulary spaces.

### 225 3.3 BIDIRECTIONAL TRANSLATION FOR COMPLEMENTARY ALIGNMENT

226 While unidirectional token translation offers a simple and effective means of token alignment, it  
227 might be limited by the asymmetry. In particular, it may favor shorter prefixes or overlook longer and  
228 semantically richer tokenizations that arise in the reverse direction. To enhance alignment robustness  
229 and capture complementary structures, we propose a *bidirectional translation* scheme.

230 The key idea is to consider both translation directions between the assist and main vocabularies and  
231 combine the outcomes. Specifically, in addition to the forward mapping  $M^{(\text{pre})}$  where each assist  
232 token is mapped to the first token generated by the main tokenizer (as done in Section 3.2), we also  
233 obtain a reverse mapping  $M^{(\text{sup})}$  where each main token is decoded and then re-tokenized using the  
234 assist tokenizer. Intuitively,  $M^{(\text{pre})}$  maps the assist token to its *prefix* main token, while  $M^{(\text{sup})}$  maps  
235 the assist token to its *superstring* main tokens. These complementary mappings are further fused to  
236 yield a more robust bidirectional alignment.

237 Formally, we use  $\text{Pre}(v_i^{(\text{assist})})$  to denote the mapped prefix of  $v_i^{(\text{assist})}$  and  $\text{Sup}(v_i^{(\text{assist})})$  to denote  
238 the set of mapped superstrings of  $v_i^{(\text{assist})}$ . Let  $v_i^{(\text{assist})} \in \mathcal{V}_{\text{assist}}$  and  $v_j^{(\text{main})} \in \mathcal{V}_{\text{main}}$  be tokens from  
239 the assist and main vocabularies, respectively. Define the forward mapping matrix  $M^{(\text{pre})}$  such that  
240  $M_{i,j}^{(\text{pre})} = 1$  if  $v_j^{(\text{main})} = \text{Pre}(v_i^{(\text{assist})})$  under token translation. Similarly, define the reverse mapping  
241 matrix  $M^{(\text{sup})}$  such that  $M_{i,j}^{(\text{sup})} = 1$  if  $v_j^{(\text{main})} \in \text{Sup}(v_i^{(\text{assist})})$  under reverse token translation.  
242 Instead of taking the maximum of the two, we apply a weighted combination:

$$243 \quad M^{(\text{assist} \rightarrow \text{main})} = M^{(\text{pre})} + \alpha \cdot M^{(\text{sup})},$$

244 where  $\alpha \in [0, 1]$  is a tunable weight that controls the influence of reverse translation. We set  $\alpha < 1$  to  
245 prioritize prefix translation, ensuring alignment remains consistent with the main model’s generative  
246 behavior, while still allowing reverse direction signals to contribute additional coverage for longer or  
247 nested token patterns.

248 This strategy reflects the intuition that forward translation maintains the semantics most compatible  
249 with the main model’s own vocabulary, while reverse translation can introduce complementary  
250 patterns that improve the recall. This is particularly useful when the reverse tokenizer favors  
251 generating longer subwords that align with semantic units absent in the forward match.

252 To generate a valid transition matrix, the fused mapping matrix is further row-normalized to produce  
253 a transition matrix  $T^{(\text{assist} \rightarrow \text{main})}$ , that is

$$254 \quad T^{(\text{assist} \rightarrow \text{main})} = \text{diag}(M^{(\text{assist} \rightarrow \text{main})} \mathbf{1})^{-1} M^{(\text{assist} \rightarrow \text{main})}.$$

255 Based on the translation matrix transition matrix  $T^{(\text{assist} \rightarrow \text{main})}$ , we can decompose the token transla-  
256 tion process into two parts, including translating to prefix and translating to substring, as follows

$$257 \quad \begin{aligned} p^{(\text{assist} \rightarrow \text{main})}(i) &= \left( T^{(\text{assist} \rightarrow \text{main})} \top p^{(\text{assist})} \right) (i) \\ 258 &= \sum_{v_j^{(\text{assist})}} \Pr(\text{Pre}) \Pr(v_i^{(\text{main})} | v_j^{(\text{assist})}, \text{Pre}) \Pr(v_j^{(\text{assist})}) + \sum_{v_j^{(\text{assist})}} \Pr(\text{Sup}) \Pr(v_i^{(\text{main})} | v_j^{(\text{assist})}, \text{Sup}) \Pr(v_j^{(\text{assist})}) \\ 259 &= \sum_j \left\{ \underbrace{\frac{1}{1 + \alpha N} \mathbb{1}(v_i^{(\text{main})} = \text{Pre}(v_j^{(\text{assist})}))}_{\Pr(\text{Pre})} \underbrace{p^{(\text{assist})}(j)}_{\Pr(v_j^{(\text{assist})})} + \underbrace{\frac{\alpha N}{1 + \alpha N} \text{Unif}(\mathbb{1}(v_i^{(\text{main})} \in \text{Sup}(v_j^{(\text{assist})})))}_{\Pr(\text{Sup})} \underbrace{p^{(\text{assist})}(j)}_{\Pr(v_i^{(\text{main})} | v_j^{(\text{assist})}, \text{Sup})} \right\}, \end{aligned}$$

---

**270   Algorithm 1** Token Translation (ToT)

---

**272   Input:** Tokenizers  $\text{Tok}_{\text{assist}}, \text{Tok}_{\text{main}}$ ; Vocabularies  $\mathcal{V}_{\text{assist}}, \mathcal{V}_{\text{main}}$ ; weight  $\alpha$   
**273   Initialize:**  $\mathbf{M}^{(\text{pre})} \leftarrow \mathbf{0}_{|\mathcal{V}_{\text{assist}}| \times |\mathcal{V}_{\text{main}}|}, \mathbf{M}^{(\text{sup})} \leftarrow \mathbf{0}_{|\mathcal{V}_{\text{assist}}| \times |\mathcal{V}_{\text{main}}|}$   
**274   for**  $i = 1, 2, \dots, |\mathcal{V}_{\text{assist}}|$  **do**  
**275**  $v_i^{(\text{assist})} \leftarrow \text{Tok}_{\text{assist}}.\text{decode}(i)$   
**276**  $[t_1, \dots, t_m] \leftarrow \text{Tok}_{\text{main}}.\text{encode}(v_i^{(\text{assist})})$   
**277**  $\mathbf{M}_{i,t_1}^{(\text{pre})} \leftarrow 1$   
**278** **for**  $j = 1, 2, \dots, |\mathcal{V}_{\text{main}}|$  **do**  
**279**  $v_j^{(\text{main})} \leftarrow \text{Tok}_{\text{main}}.\text{decode}(j)$   
**280**  $[t'_1, \dots, t'_n] \leftarrow \text{Tok}_{\text{assist}}.\text{encode}(v_j^{(\text{main})})$   
**281**  $\mathbf{M}_{t'_1,j}^{(\text{sup})} \leftarrow 1$   
**282**  $\mathbf{M}^{(\text{assist} \rightarrow \text{main})} \leftarrow \mathbf{M}^{(\text{pre})} + \alpha \cdot \mathbf{M}^{(\text{sup})}$   
**283** **Return:**  $\mathbf{M}^{(\text{assist} \rightarrow \text{main})}$ 


---

**284**  
**285** where  $N = |\mathbb{1}(v_i^{(\text{main})} \in \text{Sup}(v_j^{(\text{assist})}))|$  is the number of target tokens that are the prefix of  $v_i^{(\text{main})}$ . The  
**286** above equation provides insights on the translation process: with probability  $\text{Pr}(\text{Pre}) = \frac{1}{1+\alpha N}$ , the  
**287** probability mass on assist token is transited to the mapped prefix main token, and with probability  
**288**  $\text{Pr}(\text{Sup}) = \frac{\alpha N}{1+\alpha N}$ , the probability mass on assist token is uniformly transited to the mapped  
**289** superstring main tokens.

**290**  
**291** 3.4 UNCERTAINTY-AWARE TOKEN-LEVEL ENSEMBLING  
**292**

**293** After establishing a shared alignment space, we ensemble token-level predictions from multiple  
**294** LLMs with an *uncertainty-aware fusion strategy*.

**295** Different models exhibit varying levels of confidence depending on the input. Treating all models  
**296** equally may amplify noise or disproportionately emphasize low-confidence predictions. To address  
**297** this, we use entropy-based weighting, following observations from UniTE (Yao et al., 2025) that  
**298** low-confidence tokens tend to introduce noise. To improve robustness, we only compute entropy over  
**299** the top- $k$  tokens in each model’s predicted distribution (Ma et al., 2025), filtering out uninformative  
**300** tails.

**301** Specifically, define  $\text{Top}_k(\mathbf{p}^{(\text{assist})})$  as the top- $k$  highest scoring assist tokens. The uncertainty-based  
**302** weight  $w_{\text{ent}}^{(\text{assist})}$  is computed as the inverse entropy of the Top- $k$  tokens, that is

$$\text{304} \quad w_{\text{ent}}^{(\text{assist})} = \frac{1}{H(\mathbf{p}^{(\text{assist})})}, \quad \text{where } H(\mathbf{p}^{(\text{assist})}) = - \sum_{v \in \text{Top}_k(\mathbf{p}^{(\text{assist})})} \frac{\mathbf{p}^{(\text{assist})}(v)}{Z} \log \frac{\mathbf{p}^{(\text{assist})}(v)}{Z},
 \text{305}$$

**306** where  $Z = \sum_{v \in \text{Top}_k(\mathbf{p}^{(\text{assist})})} \mathbf{p}^{(\text{assist})}(v)$ . A similar procedure can be applied to the main model,  
**307** denoted  $w_{\text{ent}}^{(\text{main})}$ . We then normalize all weights across the main model and all assist models so that  
**308** they sum to one. The final ensembled distribution over the main model’s vocabulary is:

$$\text{310} \quad \tilde{\mathbf{p}} = w^{(\text{main})} \cdot \mathbf{p}^{(\text{main})} + \sum_{\kappa=1}^K w^{(\text{assist}\#\kappa)} \cdot \mathbf{T}^{(\text{assist}\#\kappa)^{\top}} \mathbf{p}^{(\text{assist}\#\kappa)},
 \text{312}$$

**313** where  $[w^{(\text{main})}, w^{(\text{assist}\#1)}, \dots, w^{(\text{assist}\#K)}] \in \Delta^{K+1}$  are the normalized uncertainty-based weights that  
**314** control the contribution of the main model and assist models. This formulation maintains alignment  
**315** with the main model’s prediction while allowing assist models to refine uncertain or ambiguous cases  
**316** when they are confident.

**317** All token-translation matrices are precomputed and cached as highly sparse matrices. Each column  
**318** (corresponding to a single assist-model token) has only one or a small number of non-zero entries  
**319** that point to its top-ranked aligned tokens in the main model’s vocabulary. This sparsity enables very  
**320** fast lookup and efficient memory usage.

**321**  
**322** 4 EXPERIMENTS

**323** In this section, we design experiments to answer the following research questions:

324  
 325 Table 2: Evaluation results across datasets for base models and LLM ensemble methods in two-model  
 326 setting. Ensembling uses **LLaMA-3.1-8B-Instruct** (light blue) as the main model and **InternLM-2.5-7B-Chat** (light pink) as the assist model. Best results are in **bold**, and second-best are underlined.  
 327

Model	NQ	TriviaQA	ARC-c	MMLU	GSM8K	PIQA	Avg
<i>Base Models</i>							
Gemma-2-9B-IT	13.49	52.53	49.57	41.56	78.23	61.96	49.22
GLM-4-9B-Chat	24.27	58.94	88.06	64.68	76.41	82.58	65.82
InternLM-2.5-7B-Chat	26.62	63.23	<u>85.13</u>	<u>70.57</u>	83.93	<u>87.47</u>	69.49
LLaMA-3.1-8B-Instruct	27.51	<u>65.57</u>	78.72	67.38	82.87	<u>82.86</u>	67.82
Mistral-7B-Instruct	25.37	66.61	73.42	58.76	61.74	69.55	59.91
Qwen2.5-7B-Instruct	15.04	53.23	87.35	70.46	78.67	85.25	65.67
Yi-1.5-9B-Chat	14.54	51.74	80.64	66.27	64.69	80.64	59.75
<i>LLM Ensemble Methods</i>							
LLM-Blender	25.10	61.12	76.01	65.89	80.15	80.43	64.78
GAC	22.86	65.45	70.83	68.92	75.62	78.90	63.10
CES	<u>30.54</u>	66.20	81.80	68.10	81.10	<u>87.72</u>	69.24
DeepEn	28.63	66.32	75.49	68.23	81.33	79.24	66.21
Unite	29.11	<u>67.45</u>	<u>83.23</u>	<u>69.56</u>	<u>84.43</u>	86.63	<u>70.07</u>
<b>ToT (Ours)</b>	<b>32.30</b>	<b>72.58</b>	<b>85.74</b>	<b>71.89</b>	<b>87.41</b>	<b>88.69</b>	<b>73.77</b>
<i>Improve over Base LLM</i>	(+4.79)	(+7.01)	(+7.02)	(+4.51)	(+4.54)	(+5.83)	(+5.95)
Oracle (Roofline)	39.11	78.53	89.06	80.48	91.20	94.90	78.88

344  
 345 • **RQ1:** Does our ensemble framework consistently improve the performance of base models across  
 346 diverse tasks?  
 347 • **RQ2:** How do different components, such as uncertainty modeling and prefix/superstring-based  
 348 token mappings, and alignment strategies contribute to ensemble effectiveness?  
 349 • **RQ3:** How efficient is our method in terms of preprocessing overhead, inference latency, and  
 350 GPU memory compared to existing ensemble baselines?  
 351 • **RQ4:** How does performance scale with the number and capacity of involved models?

#### 353 4.1 EXPERIMENTAL SETUP

355 **Models.** We conduct all experiments using a diverse set of open-source chat and  
 356 instruct-tuned models, including LLaMA-3.1-8B-Instruct (Grattafiori et al., 2024),  
 357 InternLM-2.5-7B/20B-Chat (Team, 2023), GLM-4-9B-Chat (GLM et al., 2024),  
 358 Mistral-7B-Instruct-v0.3 (Jiang et al., 2023a), Qwen2.5-7B-Instruct (Yang et al.,  
 359 2024), Yi-1.5-9B-Chat (Young et al., 2024), and Gemma-2-9B-IT (Team et al., 2024). These  
 360 models reflect a range of widely adopted architectures spanning practical model sizes and are among  
 361 the most recent publicly available versions at the time of the experiments.

362 **Baselines.** We compare against four representative test-time ensembling methods. **LLM-Blender** (Jiang et al., 2023b) uses a reward model (PairRanker) and a fusion model (GenFuser) to rerank and merge responses. Due to over-generation issues with GenFuser, we use only the reward-based selection. **GAC** (Yu et al., 2024) maps token probabilities into a unified space using a learned matrix and performs token-level aggregation. **CES** (Phan et al., 2025) ensembles probabilities at the sub-token (byte) level for fill-in-the-middle tasks. **DeepEn** (Huang et al., 2024) projects model outputs into a shared latent space based on overlapping vocabularies, using relative representation theory. **UniTe** (Yao et al., 2025) uses a top- $k$  union strategy to align and filter tokens.

371 **Benchmarks.** We evaluate nine benchmarks across four task categories. **(1) Comprehensive**  
 372 **exams:** MMLU (5-shot) (Hendrycks et al., 2020) and ARC-C (0-shot) (Clark et al., 2018), assessing  
 373 subject knowledge and natural science reasoning. **(2) Reasoning:** GSM8K (Cobbe et al., 2021)  
 374 (4-shot with **chain-of-thought prompting** (Wei et al., 2022)), PIQA (0-shot) (Bisk et al., 2020), and  
 375 HumanEval (0-shot) (Chen et al., 2021), covering arithmetic, commonsense, and program synthesis.  
 376 **(3) Knowledge-intensive QA:** TriviaQA (5-shot) (Joshi et al., 2017) and NaturalQuestions  
 377 (NQ) (5-shot) (Kwiatkowski et al., 2019). **(4) Summarization:** SAMSum (0-shot) (Gliwa et al.,  
 378 2019), a dialogue summarization dataset. Dataset details are in Appendix A.

378 **Experimental Setup.** We evaluate both two-model and multi-model ensembling across the afore-  
 379 mentioned benchmarks. For the two-model setting, we select `LLaMA-3.1-8B-Instruct` and  
 380 `InternLM-2.5-7B-Chat` as the main pair, chosen for their strong and comparable performance.  
 381 For multi-model settings, we include `Qwen2.5-7B-Instruct` and additional models to assess  
 382 scalability. We also explore diverse model pairings to test generality. Unless otherwise specified, we  
 383 fix the hyperparameters to  $\alpha = 0.5$  and set the top- $k$  tokens for uncertainty estimation to  $k = 50$ . All  
 384 experiments are conducted on NVIDIA A100 80GB GPUs.

## 385 386 4.2 MAIN RESULTS (RQ1)

389 As shown in Table 2, our method (ToT) achieves the best performance across all benchmarks, with  
 390 an average improvement of +5.95 points over the main model (`LLaMA-3.1-8B-Instruct`). It  
 391 delivers consistent gains across tasks (e.g., +4.79 on NQ, +7.01 on TriviaQA, +7.02 on ARC-C),  
 392 and remains robust even when the main model underperforms, for example, surpassing both base  
 393 models on ARC-C (85.74 vs. 78.72 and 85.13).

394 We also report a *rooftline accuracy*, the percentage of examples answered correctly by at least one  
 395 base model, as an empirical upper bound for training-free ensembles. ToT closely tracks this upper  
 396 bound across tasks, especially on reasoning-oriented benchmarks (GSM8K and ARC-c), indicating  
 397 that it effectively aggregates complementary reasoning signals at inference. The remaining headroom  
 398 is concentrated in knowledge-intensive settings, suggesting that integrating external knowledge or  
 399 task-specific priors could further narrow the gap. Compared to prior methods, `LLM-Blender`  
 400 relies on coarse-grained output ranking and shows limited improvement. `Unite` uses top- $k$  token  
 401 selection but lacks generation-awareness. Other methods like `GAC` and `DeepEn` suffer from poor  
 402 token alignment across vocabularies. These results highlight the value of tokenizer-aware alignment  
 403 and uncertainty modeling in test-time LLM ensembling.

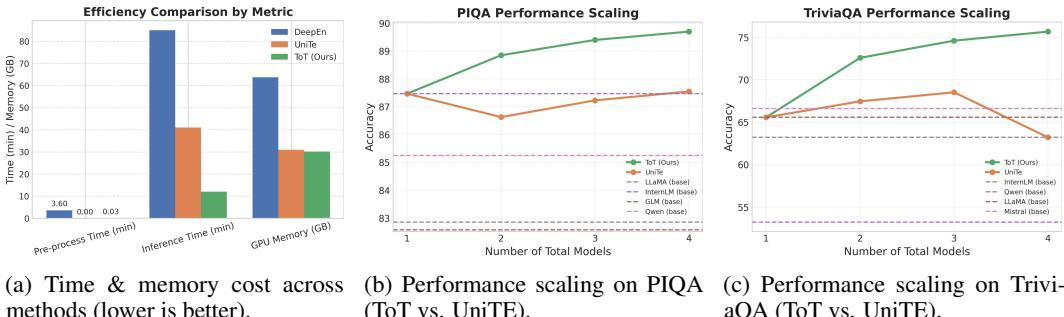
## 404 405 4.3 ABLATION STUDY (RQ2))

407 We perform ablation experiments on the NQ benchmark to  
 408 assess the contribution of each module and the effectiveness  
 409 of alternative alignment strategies (Table 3). Our two-model  
 410 ToT ensemble improves the main model by +4.79 points  
 411 (32.30), and adding a third model (Mistral-7B-Instruct) fur-  
 412 ther boosts performance to 32.94, demonstrating scalability.  
 413 Among core components, removing uncertainty weighting  
 414 yields a minor drop (32.11), indicating stability benefits.  
 415 In contrast, removing the prefix-based alignment matrix  
 $M^{pre}$  significantly reduces performance (30.75), confirming  
 416 the importance of generation-aware alignment. Omitting  
 417 the reverse mapping  $M^{sup}$  (31.86) or the clipping factor  $\beta$   
 418 (32.12) also leads to degradation, highlighting the utility of  
 419 bidirectional alignment and confidence balancing. We also  
 420 compare ToT with alternative alignment strategies, including  
 421 semantic similarity (27.63), lexical matching (28.55), and  
 422 edit-distance-based alignment (Wan et al., 2024a) (29.12).  
 423 All underperform our method, underscoring the benefit of  
 424 leveraging tokenizer priors over surface heuristics.

425 Hyperparameter studies for  $\alpha$  -  $k$  (Appendix B.1) show stable performance across wide ranges, with  
 426 fixed defaults near-optimal and consistently outperforming baselines. Results across diverse model  
 427 combinations (Appendix B.2) demonstrate gains that persist when swapping the *main* and *assist*  
 428 roles; the improvement magnitude correlates with the assist model’s strength. Token-alignment  
 429 visualizations (Appendix B.3) reveal sparse, semantically grounded mappings and cases where the  
 430 fused distribution corrects errors even when both base models’ predictions are wrong. Appendix B.4  
 431 presents empirical studies on tokenizer alignment properties, scalability across models and languages,  
 and the effect of alternative uncertainty measures that further confirm our approach’s effectiveness.

Table 3: Ablation study on NQ. We compare ToT with different variants and token mapping strategies.

Setting	NQ Acc.
<i>Baselines</i>	
InternLM-2.5-7B-Chat	26.62
LLaMA-3.1-8B-Instruct	27.51
ToT (2 models)	<b>32.30</b>
+Mistral-7B (3 models)	<b>32.94</b>
<i>Ablations</i>	
w/o Uncertainty	32.11
w/o $M^{pre}$	30.75
w/o $M^{sup}$	31.86
w/o $\beta$	32.12
<i>Alternative Mapping Strategies</i>	
with Semantic	27.63
with Lexical	28.55
with MinED (Wan et al., 2024a)	29.12



(a) Time & memory cost across methods (lower is better). (b) Performance scaling on PIQA (ToT vs. UniTE). (c) Performance scaling on TriviaQA (ToT vs. UniTE).

Figure 2: Efficiency and scaling overview. **Left:** wall-clock time and peak memory across ensembling methods. **Middle/Right:** scaling behavior of ToT vs. UniTE across base models.

#### 4.4 EFFICIENCY ANALYSIS (RQ3)

Figure 2a compares the time and memory costs of different ensembling methods. ToT achieves clear efficiency advantages in both preprocessing and inference. Unlike DeepEn, which requires costly offline alignment, ToT relies only on lightweight decode-encode operations to construct a sparse mapping (0.03 min vs. 3.6 min). At inference, ToT performs a single sparse matrix multiplication without altering model architecture or decoding logic, yielding substantially lower time (12 min vs. 85 min) and memory use (30.2 GB vs. 63.7 GB). Compared to UniTE, ToT is also more efficient due to its sparse and fixed-tokenizer design. Overall, ToT delivers strong accuracy (Table 2) while maintaining practical speed and memory scalability.

#### 4.5 SCALABILITY ACROSS MODEL SIZES AND QUANTITIES (RQ4)

To evaluate scalability, we consider two settings: (1) ensembling models of different scales, and (2) increasing the number of ensembled models.

As shown in Table 4, ToT achieves substantial gains when applied to large-scale models. By ensembling InternLM2.5-20B-Chat with LLaMA-3.1-8B-Instruct, ToT consistently outperforms both base models across all four tasks (e.g., +10.51 on HumanEval, +7.31 on NQ). This highlights the flexibility of our approach in handling model pairs with heterogeneous capacities and confirms its effectiveness in large-model regimes.

Furthermore, Figure 2b-2c illustrates performance scaling trends as more models are added to the ensemble. For both TriviaQA and PIQA, ToT demonstrates a clear upward trajectory, achieving steady improvements from 2 to 4 models. In contrast, UniTE's performance plateaus or even degrades when additional models are introduced, likely due to its less robust token selection strategy and lack of generation preference modeling. This further confirms ToT's robustness and generality when scaling across both model diversity and ensemble size.

## 5 CONCLUSION

We propose a lightweight and effective framework ToT for token-level ensembling of LLMs, addressing the core challenge of tokenizer mismatch. Our method aligns heterogeneous token spaces via a simple *token translation* based on tokenizer priors, enhanced by bidirectional alignment and uncertainty-aware fusion. Extensive experiments across tasks, model sizes, and ensemble settings show that ToT consistently outperforms existing baselines while maintaining high efficiency. Looking ahead, a key challenge is selecting complementary model combinations. As the LLM landscape grows, developing adaptive strategies for model pairing and ensemble configuration will be essential for maximizing performance and diversity.

486     **Ethical Statement.** This work focuses on algorithmic methods for test-time ensembling of publicly  
 487     available large language models. No human subjects or sensitive personal data are involved. All  
 488     datasets used in our experiments (e.g., MMLU, ARC-C, GSM8K, TriviaQA, NaturalQuestions,  
 489     SAMSum, HumanEval, PIQA) are standard public benchmarks widely adopted in prior research.  
 490     We have taken care to follow the ICLR Code of Ethics by ensuring appropriate dataset usage,  
 491     acknowledging limitations, and avoiding any harmful applications. The proposed method is purely  
 492     methodological and does not pose direct risks related to privacy, security, or fairness.

493     **Reproducibility Statement.** We provide detailed descriptions of our methodology in Section 3,  
 494     including the construction of token translation matrices, bidirectional alignment, and uncertainty-  
 495     aware weighting. Experimental settings, datasets, and evaluation protocols are specified in Section  
 496     4 and Appendix A. Ablation studies and hyperparameter analyses are presented in Section 4.3  
 497     and Appendix B to demonstrate robustness. All token translation matrices are precomputable and  
 498     lightweight. To further support reproducibility, we've incorporated code, precomputed mappings,  
 499     and experiment scripts in the supplementary material.

500  
 501     **REFERENCES**

503     Aakriti Agrawal, Mucong Ding, Zora Che, Chenghao Deng, Anirudh Satheesh, John Langford, and  
 504     Furong Huang. Ensemw2s: Can an ensemble of llms be leveraged to obtain a stronger llm? *arXiv*  
 505     preprint *arXiv:2410.04571*, 2024.

507     Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical  
 508     commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*,  
 509     volume 34, pp. 7432–7439, 2020.

510     Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared  
 511     Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large  
 512     language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

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

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

521     Tao Feng, Yanzhen Shen, and Jiaxuan You. Graphrouter: A graph-based router for llm selections.  
 522     *arXiv preprint arXiv:2410.03834*, 2024.

524     Bogdan Gliwa, Iwona Mochol, Maciej Bieseck, and Aleksander Wawer. Samsum corpus: A human-  
 525     annotated dialogue dataset for abstractive summarization. *arXiv preprint arXiv:1911.12237*,  
 526     2019.

527     Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas,  
 528     Guanyu Feng, Hanlin Zhao, et al. Chatglm: A family of large language models from glm-130b to  
 529     glm-4 all tools. *arXiv preprint arXiv:2406.12793*, 2024.

531     Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad  
 532     Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of  
 533     models. *arXiv preprint arXiv:2407.21783*, 2024.

534     Kevin Gu, Eva Tuecke, Dmitriy Katz, Raya Horesh, David Alvarez-Melis, and Mikhail Yurochkin.  
 535     Chared: Character-wise ensemble decoding for large language models. *arXiv preprint*  
 536     *arXiv:2407.11009*, 2024.

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

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

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

546 Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec  
 547 Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint*  
 548 *arXiv:2412.16720*, 2024.

549 Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot,  
 550 Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier,  
 551 Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas  
 552 Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *CoRR*, abs/2310.06825, 2023a.  
 553 doi: 10.48550/ARXIV.2310.06825. URL <https://doi.org/10.48550/arXiv.2310.06825>.

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

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

559 Ehsan Kamalloo, Nouha Dziri, Charles Clarke, and Davood Rafiei. Evaluating open-domain question  
 560 answering in the era of large language models. In *Proceedings of the 61st Annual Meeting of the*  
 561 *Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5591–5606, 2023.

562 Taku Kudo. Subword regularization: Improving neural network translation models with multiple  
 563 subword candidates. *arXiv preprint arXiv:1804.10959*, 2018.

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

568 Keming Lu, Hongyi Yuan, Runji Lin, Junyang Lin, Zheng Yuan, Chang Zhou, and Jingren Zhou.  
 569 Routing to the expert: Efficient reward-guided ensemble of large language models. *arXiv preprint*  
 570 *arXiv:2311.08692*, 2023.

571 Huan Ma, Jingdong Chen, Guangyu Wang, and Changqing Zhang. Estimating llm uncertainty with  
 572 logits. *arXiv preprint arXiv:2502.00290*, 2025.

573 OpenAI. Gpt-4 technical report. 2023. URL <https://api.semanticscholar.org/CorpusID:257532815>.

574 Buu Phan, Brandon Amos, Itai Gat, Marton Havasi, Matthew J Muckley, and Karen Ullrich. Exact  
 575 byte-level probabilities from tokenized language models for fin-tasks and model ensembles. In  
 576 *The Thirteenth International Conference on Learning Representations*, 2025.

577 Yangjun Ruan, Saurabh Singh, Warren Morningstar, Alexander A Alemi, Sergey Ioffe, Ian Fischer,  
 578 and Joshua V Dillon. Weighted ensemble self-supervised learning. *arXiv preprint*  
 579 *arXiv:2211.09981*, 2022.

580 Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with  
 581 subword units. *arXiv preprint arXiv:1508.07909*, 2015.

582 Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya  
 583 Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al.  
 584 Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*,  
 585 2024.

594 InternLM Team. Internlm: A multilingual language model with progressively enhanced capabilities,  
 595 2023.

596

597 Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée  
 598 Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez,  
 599 Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation  
 600 language models. *ArXiv*, abs/2302.13971, 2023. URL <https://api.semanticscholar.org/CorpusID:257219404>.

601

602 Fanqi Wan, Xinting Huang, Deng Cai, Xiaojun Quan, Wei Bi, and Shuming Shi. Knowledge fusion  
 603 of large language models. In *ICLR*, 2024a.

604

605 Fanqi Wan, Longguang Zhong, Ziyi Yang, Ruijun Chen, and Xiaojun Quan. Fusechat: Knowledge  
 606 fusion of chat models. *arXiv preprint arXiv:2408.07990*, 2024b.

607

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

611

612 Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. C-pack: Packaged resources to  
 613 advance general chinese embedding, 2023.

614

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

617

618 Yangyifan Xu, Jinliang Lu, and Jiajun Zhang. Bridging the gap between different vocabularies for  
 619 llm ensemble. In *Proceedings of the 2024 Conference of the North American Chapter of the  
 620 Association for Computational Linguistics: Human Language Technologies (Volume 1: Long  
 621 Papers)*, pp. 7133–7145, 2024b.

622

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

626

627 An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li,  
 628 Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint  
 629 arXiv:2412.15115*, 2024.

630

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

634

635 Alex Young, Bei Chen, Chao Li, Chengan Huang, Ge Zhang, Guanwei Zhang, Guoyin Wang, Heng  
 636 Li, Jiangcheng Zhu, Jianqun Chen, et al. Yi: Open foundation models by 01. ai. *arXiv preprint  
 637 arXiv:2403.04652*, 2024.

638

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

642

643 Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B  
 644 Hashimoto. Benchmarking large language models for news summarization. *Transactions of the  
 645 Association for Computational Linguistics*, 12:39–57, 2024.

646

647

648  
649

## Appendix

650  
651  
652

### CONTENTS

653  
654  
655  
656  
657  
658  
659  
660  
661  
662

<b>A Dataset Statistics</b>	<b>14</b>
-----------------------------	-----------

<b>B Further Ablation Study and Visualization</b>	<b>14</b>
B.1 Hyper-parameter Study	14
B.2 Ablation Study on Various Model Ensembling	14
B.3 Demonstration of Alignment and Ensembling	16
B.4 Empirical Analysis of Tokenizers and Performance	19

663  
664  
665  
666  
667  
668  
669  
670  
671

<b>C Formal Analysis of Token Translation</b>	<b>20</b>
C.1 Problem Setup	20
C.2 Tokenizer Assumptions	21
C.3 Causal Alignment	21
C.4 ToT Recovers the Exact Alignment	21
C.5 Causal Validity	22

672  
673

<b>D Use of Large Language Models</b>	<b>22</b>
---------------------------------------	-----------

674

<b>E Limitations</b>	<b>22</b>
----------------------	-----------

676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701

702 **A DATASET STATISTICS**  
703

704 We evaluate nine benchmark datasets across four representative task categories to assess language  
 705 model performance under diverse reasoning and comprehension requirements. **(1) Comprehensive**  
 706 **Exams:** MMLU (5-shot, 5,000 test examples) (Hendrycks et al., 2020) is a multiple-choice benchmark  
 707 that spans 57 diverse subjects across STEM, humanities, social sciences, and more. Each question  
 708 contains four answer choices, and models are evaluated using in-context few-shot prompting and  
 709 accuracy as the metric. ARC-Challenge (0-shot, 1,172 examples) (Clark et al., 2018) is a science  
 710 reasoning benchmark targeting difficult grade-school level questions with multiple-choice formats.  
 711 Models must directly select the correct answer without in-context examples. **(2) Reasoning:** GSM8K  
 712 (4-shot with chain-of-thought prompting (Wei et al., 2022), 1,319 examples) (Cobbe et al., 2021) con-  
 713 sists of grade-school math word problems. The model is prompted with four examples that explicitly  
 714 demonstrate step-by-step intermediate reasoning, and the final prediction is considered correct only  
 715 if the numeric answer matches the gold label exactly. PiQA (0-shot, 1,838 examples) (Bisk et al.,  
 716 2020) evaluates physical commonsense reasoning: models are given a naturalistic question involving  
 717 physical interactions and must choose between two plausible outcomes. HumanEval (0-shot, 164  
 718 examples) (Chen et al., 2021) is a code generation benchmark where each input specifies a function  
 719 signature and docstring in Python, and the model is expected to synthesize correct code. Performance  
 720 is evaluated using `pass@1`, indicating the percentage of problems solved correctly in a single attempt.  
 721 **(3) Knowledge-Intensive QA:** TriviaQA (5-shot, 11,313 examples) (Joshi et al., 2017) contains  
 722 open-domain factoid questions curated from trivia websites, with Wikipedia-based ground truth  
 723 answers. Each input includes five QA exemplars followed by a new question; accuracy is measured  
 724 by exact match. NaturalQuestions (NQ) (5-shot, 3,610 examples) (Kwiatkowski et al., 2019)  
 725 consists of real anonymized Google search queries, paired with short answers derived from Wikipedia  
 726 articles. We follow prior work in using short-form extractive answer strings and compute exact match  
 727 as the evaluation metric. **(4) Summarization:** SAMSum (0-shot, 818 examples) (Gliwa et al., 2019)  
 728 is a single-document summarization task where the input is a multi-turn dialogue between fictitious  
 729 participants, and the model must produce a concise and coherent summary. No demonstrations are  
 730 provided, and performance is measured using ROUGE-L.

731 Following prior work (Yao et al., 2025; Huang et al., 2024), we construct the prompt by providing  
 732 the question followed by the answer format, optionally including in-context examples and chain-of-  
 733 thought prompting using the phrase “let’s think step-by-step” (Wei et al., 2022).

734 **B FURTHER ABLATION STUDY AND VISUALIZATION**  
735

736 In this section, we first perform a comprehensive hyperparameter study to demonstrate the robustness  
 737 of our method under various settings. We then conduct an ablation study using different pairs of  
 738 large language models (LLMs) in the ensemble to further verify its robustness across diverse model  
 739 combinations. Finally, we present representative output examples and token alignment visualizations  
 740 to provide qualitative insights into the behavior and effectiveness of our approach.

741 **B.1 HYPER-PARAMETER STUDY**  
742

743 Figures 3, 4, and 5 demonstrate robustness across the key hyperparameters  $\alpha$ ,  $\beta$ , and  $k$ . Here,  
 744  $\alpha$  controls the blend between prefix and superstring mappings when aligning token sequences;  $\beta$   
 745 controls the *strength of the main model* by enforcing a minimum contribution—implemented as  
 746 clipping the normalized weight so that  $w^{(\text{main})} \leftarrow \max(\beta, w^{(\text{main})})$  and renormalizing the remaining  
 747 mass across assist models; and  $k$  sets how many (top) tokens are used to compute model uncertainty.  
 748 Despite their different roles, all three are easy to configure and yield stable performance over wide  
 749 ranges. Across settings, our method consistently outperforms the base models, making it practical  
 750 without tuning; thus we fix these values uniformly in all experiments.

752 **B.2 ABLATION STUDY ON VARIOUS MODEL ENSEMBLING**  
753

754 Figure 6 provides a comprehensive view of the performance gains achieved when ensembling different  
 755 main and assist model pairs. Two key observations emerge.



Figure 3: Performance with respect to  $\alpha$ , representing the ratio between prefix and superstring mappings on the NQ dataset with InternLM and LLaMA models.

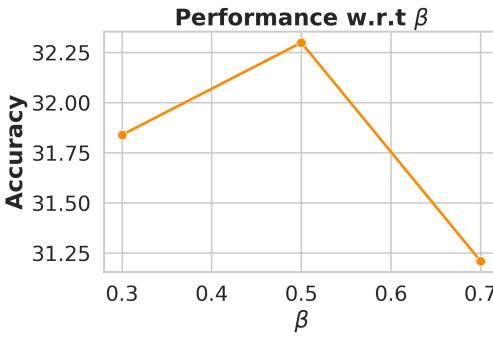


Figure 4: Performance with respect to  $\beta$ , denoting the minimum contribution of the main model to the final prediction, evaluated on the NQ dataset with InternLM and LLaMA.

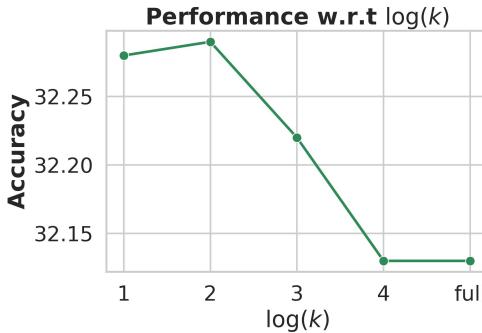


Figure 5: Performance with respect to  $\log k$ , where  $k$  is the number of tokens used to compute uncertainty, evaluated on the NQ dataset using InternLM and LLaMA.

800  
801  
802  
803  
804  
805  
806  
807  
808  
809

First, our method is highly robust, as it consistently delivers substantial performance improvements across all combinations of main and assist models. No matter which model is used as the main model—GLM, InternLM, LLaMA, or Mistral—the introduction of an assist model yields a notable gain over the base performance. This demonstrates that our ensembling strategy generalizes well and is not overly sensitive to the choice of models involved.

Second, the magnitude of the gain tends to correlate with the strength of the assist model. Specifically, using stronger models like InternLM and LLaMA as assist models leads to the largest improvements,

810 with performance gains exceeding 5 points in some cases. For example, pairing InternLM as the assist  
 811 model with GLM as the main model yields the highest observed gain. This suggests that stronger  
 812 assist models provide more informative or complementary signals during ensembling, thus boosting  
 813 the overall prediction quality.

814 Together, these results validate the versatility and effectiveness of our method, showing it works well  
 815 in diverse model settings and particularly excels when high-capacity assist models are available.  
 816

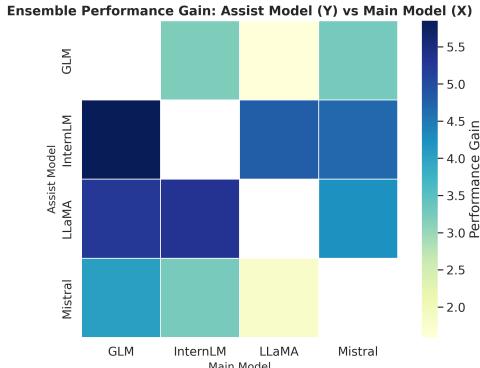


Figure 6: Performance gain with different main and assist models.

### B.3 DEMONSTRATION OF ALIGNMENT AND ENSEMBLING

835 Table 5 compares three token mapping methods: string similarity, embedding similarity, and our  
 836 proposed Token Translation. String similarity is based on edit distance at the string level, while  
 837 embedding similarity measures cosine similarity between token embeddings. In contrast, Token  
 838 Translation (using prefix-based mapping) leverages both the model’s generation preferences and  
 839 tokenizer priors to identify mappings that align more closely with actual generation behavior. As  
 840 evidenced by the performance results in Table 3, both string- and embedding-based methods often pro-  
 841 duce suboptimal or irrelevant mappings—for example, mapping “Papa” to “Luna” or “Administrator”  
 842 to a Chinese token—because they overlook how tokens function during decoding. In contrast, Token  
 843 Translation consistently produces compact, semantically coherent segments such as “Admin” and  
 844 “P” that better match the tokenizer’s segmentation and generation dynamics. This demonstrates the  
 845 effectiveness of our approach in generating mappings that are not only linguistically and structurally  
 846 aligned, but also more suitable for downstream ensemble generation.

846 Table 5: Comparison of Token Mapping Methods

847

Method	Source Token	Mapped Token
<b>String sim</b>	Administrator	Administr
<b>Embedding sim</b>	Administrator	Administrator in Chinese
<b>Token Translation</b>	Administrator	Admin
<b>String sim</b>	_CAT	_CA
<b>Embedding sim</b>	_CAT	_VAR
<b>Token Translation</b>	_CAT	_C
<b>String sim</b>	Papa	Pap
<b>Embedding sim</b>	Papa	Luna
<b>Token Translation</b>	Papa	P

859 Next, we present output examples of our ensemble method in Tables 6 and 7, which not only correct  
 860 errors made by the base models but also outperform baseline methods on the GSM8K dataset.  
 861 Correct answers are highlighted in green, while incorrect ones are marked in red. These cases clearly  
 862 demonstrate the effectiveness of our method in enhancing reasoning accuracy and robustness by  
 863 leveraging complementary strengths of different models.

864  
865  
866  
867  
868  
869  
870  
871  
872  
873

874  
875

Table 6: Case study of ToT and other ensembling methods on GSM8K.

876	<b>Question</b>	Blake and Kelly are having a contest to see who can run the most in 15 minutes. They decide to do it on a football field that is 100 yards long. Blake runs back and forth 15 times. Kelly runs back and forth once, and then decides that she doesn't want to run next to Blake, so she starts to run to the 40-yard line and back. She does this 34 times. How much farther does the winner run than the loser?
877	<b>LLaMA</b>	Blake runs 15 times back and forth on a 100-yard field, so he runs $15*2*100=3000$ yards. Kelly runs 34 times to the 40-yard line and back, so she runs $34*2*40=2720$ yards. Blake runs 3000 yards and Kelly runs 2720 yards, so Blake runs $3000-2720=280$ yards more than Kelly. The answer is <b>280</b>
878	<b>InternLM</b>	First, let's calculate how far Blake runs. A football field is 100 yards long, and he runs back and forth 15 times. So, Blake runs $100 \text{ yards} * 15 = 1500$ yards. Next, let's calculate how far Kelly runs. She runs back and forth once, which is 100 yards $* 2 = 200$ yards. Then, she starts running to the 40-yard line and back, which is 40 yards $* 2 = 80$ yards. She does this 34 times, so she runs an additional $80 \text{ yards} * 34 = 2720$ yards. In total, Kelly runs $200 \text{ yards} + 2720 \text{ yards} = 2920$ yards. To find out how much farther the winner runs than the loser, we subtract the distance that Blake ran from the distance that Kelly ran: $2920 \text{ yards} - 1500 \text{ yards} = 1420$ yards. The answer is <b>1420</b>
879	<b>UniTE</b>	Blake and Kelly are having a running contest on a 100-yard football field. Blake runs back and forth 15 times, and since each round trip is 100 yards, he runs a total of $15 \times 100 = 1500$ yards. Kelly first runs one full trip across the field and back, which is $2 \times 100 = 200$ yards. After that, she runs to the 40-yard line and back 34 times. Since the 40-yard line is not the full length, we compute her additional distance as $40 \times 34 = 1360$ yards. Adding this to her initial 200 yards, Kelly runs a total of 1560 yards. Comparing the two, Kelly runs $1560 - 1500 = 60$ yards more than Blake. The answer is <b>60</b>
880	<b>ToT (Ours)</b>	Blake runs back and forth 15 times on a 100-yard field, so he runs $15*2*100=3000$ yards. Kelly runs back and forth once, which is $2*100=200$ yards. Then she runs to the 40-yard line and back, which is $2*40=80$ yards. She does this 34 times, so she runs $34*80=2720$ yards. The total distance that Kelly runs is $200+2720=2920$ yards. The difference between the distances that Blake and Kelly run is $3000-2920=80$ yards. The answer is <b>80</b>

909  
910  
911  
912  
913  
914  
915  
916  
917

918  
919  
920  
921  
922  
923  
924  
925  
926  
927

Table 7: Case study of ToT and other ensembling methods on GSM8K.

928	<b>Question</b>	John buys milk for 2 dollars, eggs for 3 dollars, light bulbs for 3 dollars, cups for 3 dollars, and roach traps for 4 dollars. If there is a 10% tax on all nonfood items how much did everything cost?
929	<b>LLaMA</b>	<p>John buys milk for 2 dollars, eggs for 3 dollars, and roach traps for 4 dollars, so the total cost of the food items is <math>2+3+4=9</math> dollars.</p> <p>The nonfood items are the light bulbs, cups, and roach traps. The total cost of the nonfood items is <math>3+3+4=10</math> dollars.</p> <p>There is a 10% tax on the nonfood items, so the tax is 10% of 10 dollars, which is <math>0.1*10=1</math> dollar.</p> <p>The total cost of the nonfood items with tax is <math>10+1=11</math> dollars.</p> <p>The total cost of everything is the cost of the food items plus the cost of the nonfood items with tax, so the total cost is <math>9+11=</math> <span style="border: 1px solid red; padding: 2px;">20</span></p>
930	<b>InternLM</b>	<p>First, let's calculate the cost of the non-food items before tax. The non-food items are the light bulbs, cups, and roach traps.</p> <p>The cost of the light bulbs is \$3, the cups are also \$3, and the roach traps are \$4.</p> <p>So the total cost of the non-food items is <math>\\$3 + \\$3 + \\$4 = \\$10</math>.</p> <p>The answer is <span style="border: 1px solid red; padding: 2px;">10</span></p>
931	<b>UniTE</b>	<p>John buys milk for 2 dollars, eggs for 3 dollars, and roach traps for 4 dollars, so the total for food items is <math>2 + 3 + 4 = 9</math> dollars.</p> <p>The nonfood items include light bulbs, cups, and roach traps, totaling <math>3 + 3 + 4 = 10</math> dollars.</p> <p>A 10% tax applies to nonfood items, so the tax is 10% of 10 dollars, which is <math>0.1 \times 10 = 1</math> dollar.</p> <p>Adding the tax, the nonfood total becomes <math>10 + 1 = 11</math> dollars.</p> <p>Combining the food and taxed nonfood totals, the final amount is <math>9 + 11 =</math> <span style="border: 1px solid red; padding: 2px;">20</span></p>
932	<b>ToT (Ours)</b>	<p>First, let's calculate the total cost of the non-food items. The non-food items are the light bulbs, cups, and roach traps.</p> <p>The cost of the light bulbs is \$3, the cups are \$3, and the roach traps are \$4.</p> <p>So the total cost of the non-food items is <math>\\$3 + \\$3 + \\$4 = \\$10</math>.</p> <p>Next, we need to calculate the tax on the non-food items. The tax rate is 10%, so we need to find 10% of \$10.</p> <p>To do this, we multiply \$10 by 0.10 (which is the decimal equivalent of 10%).</p> <p>So the tax on the non-food items is <math>\\$10 * 0.10 = \\$1</math>.</p> <p>Now, we can add the tax to the cost of the non-food items to find the total cost of the non-food items.</p> <p>The total cost of the non-food items is <math>\\$10 + \\$1 = \\$11</math>.</p> <p>Finally, we need to add the cost of the milk and eggs to the total cost of the non-food items to find the total cost of everything.</p> <p>The cost of the milk is \$2 and the cost of the eggs is \$3.</p> <p>So the total cost of everything is <math>\\$2 + \\$3 + \\$11 =</math> <span style="border: 1px solid green; padding: 2px;">16</span></p>
933		
934		
935		
936		
937		
938		
939		
940		
941		
942		
943		
944		
945		
946		
947		
948		
949		
950		
951		
952		
953		
954		
955		
956		
957		
958		
959		
960		
961		
962		
963		
964		
965		
966		
967		
968		
969		
970		
971		

972 B.4 EMPIRICAL ANALYSIS OF TOKENIZERS AND PERFORMANCE  
973

974 To complement the formal analysis in Appendix C, we present an empirical study of tokenizer  
975 behavior and its effect on cross-model alignment. This analysis examines how heterogeneous  
976 vocabularies interact in practice and how these structural properties influence ensemble performance.  
977 We begin by measuring token alignment coverage across representative LLM families.

978  
979 B.4.1 TOKEN ALIGNMENT COVERAGE ACROSS MODELS  
980

981 We begin by quantifying token-level alignment coverage across five representative LLM families.  
982 Table 8 reports the percentage of tokens in one vocabulary that find direct alignment matches in  
983 another. Coverage varies substantially across model pairs (e.g., 85.43% for LLaMA3→Qwen2.5  
984 vs. 36.30% for GLM4→Mistral), revealing the strong tokenizer-induced structural differences that  
985 motivate a deterministic translation mechanism such as ToT.

986 Table 8: Cross-token alignment coverage among representative LLM tokenizers.  
987

Model Pair	internlm2 (92k)	Qwen2.5 (152k)	LLaMA3 (128k)	GLM4 (151k)	Mistral (32k)
internlm2 (92k)	–	61.51%	62.70%	54.75%	79.65%
Qwen2.5 (152k)	61.51%	–	85.43%	33.05%	75.57%
LLaMA3 (128k)	62.70%	85.43%	–	39.15%	75.79%
GLM4 (151k)	54.75%	33.05%	39.15%	–	36.30%
Mistral (32k)	79.65%	75.57%	75.79%	36.30%	–

994  
995 B.4.2 FORWARD/BACKWARD MAPPING PRECISION  
996

997 Next, we examine how many non-shared tokens can be mapped via ToT. Table 9 reports results when  
998 mapping several assist models into the internlm2 vocabulary. Because vocabularies differ greatly  
999 in size and segmentation granularity, forward (assist→main) and backward (main→assist) mapping  
1000 coverage can be highly asymmetric—for example, Qwen2.5→internlm2 achieves 91.35% forward  
1001 coverage but only 23.09% backward coverage. Smaller vocabularies (e.g., Mistral’s 32k) also show  
1002 the opposite pattern. These results illustrate why alignment would be better to consider tokenizer  
1003 directionality rather than assume symmetric similarity.

1004 Table 9: Forward (assist→main) and backward (main→assist) mapping coverage into internlm2.  
1005

Model → internlm2	Direction	Mapped	Non-Shared	Coverage
Qwen2.5 (152k)	Forward	32,540	35,616	0.9135
	Backward	21,969	95,136	0.2309
LLaMA3 (128k)	Forward	17,004	34,521	0.4927
	Backward	6,283	70,233	0.0894
GLM4 (151k)	Forward	37,705	41,876	0.9006
	Backward	32,371	100,884	0.3210
Mistral (32k)	Forward	1,061	66,443	0.0159
	Backward	5,719	6,667	0.8576

1014 B.4.3 SEMANTIC COHERENCE ANALYSIS  
1015

1016 To evaluate whether ToT’s alignments are semantically meaningful, Table 10 reports the average  
1017 BGE embedding (Xiao et al., 2023) similarity between paired tokens. Scores are consistently  
1018 high (0.70–0.97), confirming semantic coherence while remaining below the theoretical maximum,  
1019 indicating that alignment is still governed by tokenizer-induced structural decomposition rather than  
1020 semantic nearest neighbors.

1021  
1022 B.4.4 SCALING TO LARGER BACKBONES  
1023

1024 To assess scalability, we evaluate ToT on Mixtral-8×7B and Qwen2.5-32B-Instruct. Table 11 shows  
1025 that ToT consistently improves over both individual models and the UniTe baseline, demonstrating  
that the alignment benefits persist even for strong dense and MoE backbones.

Table 10: Semantic similarity of aligned tokens using BGE embeddings.

Model → internlm2	Direction	Avg Sim	Max Avg Sim
Qwen2.5 (152k)	Forward	0.7778	0.8347
	Backward	0.9407	0.9904
LLaMA3 (128k)	Forward	0.6999	0.7969
	Backward	0.8863	0.9880
GLM4 (151k)	Forward	0.7818	0.8398
	Backward	0.9669	0.9959
Mistral (32k)	Forward	0.7142	0.7679
	Backward	0.8411	0.9508

Table 11: Scalability evaluation on larger backbones.

Model	SAMSum	NQ	Avg
Mixtral-8×7B	32.8	30.9	31.9
Qwen2.5-32B-Instruct	34.1	33.7	33.9
UniTe (baseline)	34.0	34.5	34.3
<b>ToT (Ours)</b>	<b>35.3</b>	<b>36.9</b>	<b>36.1</b>

#### B.4.5 MULTILINGUAL EVALUATION: FLORES En→De AND En→Zh

We further test robustness on FLORES En→De and En→Zh translation. Table 12 shows that ToT improves over both single models and UniTe, particularly in En→Zh where tokenization differences are largest.

Table 12: Multilingual evaluation on FLORES subsets.

Dataset	Model	BLEU (↑)
En→De	LLaMA	29.87
	InternLM	26.74
	UniTe	33.17
	<b>Ours</b>	<b>36.13</b>
En→Chinese	LLaMA	27.45
	InternLM	32.10
	UniTe	34.82
	<b>Ours</b>	<b>37.56</b>

#### B.4.6 UNCERTAINTY MEASURE COMPARISON

Finally, we compare several uncertainty metrics within the same routing framework. As shown in Table 13, top- $k$  entropy achieves the strongest average performance. This aligns with the observation that long-tail logits introduce noise, while the top- $k$  region contains the meaningful decision mass for autoregressive decoding.

## C FORMAL ANALYSIS OF TOKEN TRANSLATION

In this section, we provide a rigorous justification for our Token Translation (ToT) framework. We show that ToT is not a heuristic approximation, but a deterministic implementation of *Exact Causal Alignment* between heterogeneous token spaces. Under standard tokenizer assumptions, ToT recovers the ground-truth conditional probability required for cross-model alignment.

### C.1 PROBLEM SETUP

Let  $\mathcal{V}_{\text{assist}}$  and  $\mathcal{V}_{\text{main}}$  denote the vocabularies of the assist and main models, respectively. We posit a shared, continuous text space  $\mathcal{S}$  representing all possible raw text strings. The alignment objective is

Table 13: Comparison of uncertainty measures for routing.

Uncertainty Measure	Description	Avg Score ( $\uparrow$ )
Entropy	Full-distribution entropy	32.13
Variance	Variance of the distribution	31.10
Calibration	$1 - \max(p)$ confidence proxy	31.56
Disagreement	$\text{KL}(\text{main} \parallel \text{assist})$	31.28
<b>Top-<math>k</math> Entropy (Ours)</b>	Entropy over normalized top- $k$ mass	<b>32.30</b>

1089  
1090 to compute

$$P(v^{(\text{main})} \mid v^{(\text{assist})}), \quad v^{(\text{assist})} \in \mathcal{V}_{\text{assist}}, \quad v^{(\text{main})} \in \mathcal{V}_{\text{main}}.$$

1091  
1092  
1093 The true conditional probability is

$$P(v^{(\text{main})} \mid v^{(\text{assist})}) = \sum_{s \in \mathcal{S}} \underbrace{P(v^{(\text{main})} \mid s)}_{\text{Main Prior}} \cdot \underbrace{P(s \mid v^{(\text{assist})})}_{\text{Assist Decoding}}. \quad (1)$$

## 1100 C.2 TOKENIZER ASSUMPTIONS

1101 Modern BPE and Unigram tokenizers satisfy the following assumptions.

1102  
1103 **Assumption 1 (Deterministic Decoding).** Each assist token  $v^{(\text{assist})}$  deterministically decodes to a  
1104 unique raw string:

$$P(s \mid v^{(\text{assist})}) = \mathbb{1}[s = \text{Tok}_{\text{assist}}.\text{decode}(v^{(\text{assist})})].$$

1105  
1106  
1107 **Assumption 2 (Canonical Encoding Prior).** For any string  $s$ , the main tokenizer produces a  
1108 unique canonical sequence  $\text{Tok}_{\text{main}}.\text{encode}(s) = [t_1, t_2, \dots]$ . Due to autoregressive causality,

$$P(v^{(\text{main})} \mid s) = \mathbb{1}[v^{(\text{main})} = \text{First}(\text{Tok}_{\text{main}}.\text{encode}(s))].$$

## 1114 C.3 CAUSAL ALIGNMENT

1115 Under Assumptions 1–2, the intractable sum in Eq. 1 collapses to

$$P_{\text{true}}(v^{(\text{main})} \mid v^{(\text{assist})}) = \mathbb{1}[v^{(\text{main})} = \text{First}(\text{Tok}_{\text{main}}.\text{encode}(\text{Tok}_{\text{assist}}.\text{decode}(v^{(\text{assist})})))].$$

1116 This is the exact causal mapping from assist-token space to main-token space.

## 1122 C.4 TOT RECOVERS THE EXACT ALIGNMENT

1123 ToT constructs a forward translation matrix  $M^{(\text{pre})}$ :

$$\hat{P}_{\text{ToT}}(v^{(\text{main})} \mid v^{(\text{assist})}) \triangleq M_{v^{(\text{assist})}, v^{(\text{main})}}^{(\text{pre})},$$

1127 where

$$M_{i,j}^{(\text{pre})} = \mathbb{1}[j = \text{First}(\text{Tok}_{\text{main}}.\text{encode}(\text{Tok}_{\text{assist}}.\text{decode}(i)))].$$

1130 Thus,

$$\mathbb{E}[\hat{P}_{\text{ToT}}] = P_{\text{true}},$$

1131 showing that ToT exactly recovers the deterministic causal alignment induced by the tokenizers.

1134 C.5 CAUSAL VALIDITY  
11351136 Consider an assist token that expands to a multi-token main sequence  $[t_1, t_2, \dots, t_n]$ . Define:1137  
1138 •  $E_{\text{seq}}$ : the event that the main model generates the full sequence;  
1139 •  $E_{t_1}$ : the event that it generates  $t_1$ .

1140 Since

1141 
$$E_{\text{seq}} \subseteq E_{t_1} \implies P(E_{\text{seq}}) \leq P(E_{t_1}),$$

1142 the first token  $t_1$  is a necessary causal precursor for generating the entire sequence. If the assist model  
1143 assigns probability  $p$  to  $v^{(\text{assist})}$ , then the main model must assign at least  $p$  to  $t_1$ . Therefore,

1144 
$$\hat{P}(t_1) \leftarrow p$$

1145 preserves probability mass and yields a deterministic causal mapping for next-token prediction.  
11461147 As shown in Table 3, this prefix-only mapping already surpasses strong baselines. Moreover, the  
1148 backward translation matrix  $M^{(\text{sup})}$  restores the residual probability mass spread across longer  
1149 sequences, providing improved robustness through bidirectional alignment.  
11501151 D USE OF LARGE LANGUAGE MODELS  
11521153 During the preparation of this paper, we made limited use of large language models (LLMs), specifically  
1154 ChatGPT, as an auxiliary writing tool. The LLM was used exclusively for stylistic refinement,  
1155 including improvements to fluency, grammar, and readability of text initially drafted by the authors.  
1156 All scientific content, including problem formulation, methodology, experiments, analyses, and  
1157 overall narrative, was entirely conceived and validated by the authors. Thus, the role of LLMs was  
1158 restricted to text polishing and does not constitute authorship.  
11591160 E LIMITATIONS  
11611162 Our approach assumes access to model tokenizers and output probabilities, which may not be available  
1163 for all proprietary APIs. Additionally, selecting optimal model combinations for ensembling remains  
1164 an open challenge.  
11651166  
1167  
1168  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187