Parameter-Efficient Vocabulary Expansion via Cross-Lingual Embedding Alignment

Andrew Ivan Soegeng[1,2], **Muhammad Reza Qorib**[3], **Weishi Wang**[2], **Daniel Dahlmeier**[2], **Hwee Tou Ng**[1]
[1]National University of Singapore, [2]SAP, [3]Carnegie Mellon University
{andrew.soegeng,weishi.wang,d.dahlmeier}@sap.com
mrqorib@cmu.edu
dcsnght@nus.edu.sg

Abstract

Continual pretraining with vocabulary expansion is often used to make large language models (LLMs) more effective and efficient at handling new languages. However, when LLMs are extended to many languages simultaneously, the number of model parameters increases significantly. In addition, the newly added tokens do not receive the same number of training steps as the tokens in the original vocabulary, which sometimes leads to vocabulary expansion yielding only limited performance improvements. We propose leveraging the semantic equivalence between words in the model's primary language and those in the extended vocabulary to transfer learned representations. In this paper, we introduce MATE, a parameter-efficient vocabulary expansion method that enables effective cross-lingual transfer with 8× fewer additional parameters per token. MATE reduces the average tokenization compression ratio and consistently outperforms baselines on various multilingual tasks, especially for low-resource languages.

1 Introduction

Large Language Models (LLMs) are predominantly pre-trained on English and a few high-resource languages (Brown et al., 2020; Touvron et al., 2023; OLMo et al., 2025). Adapting them to unseen, low-resource languages is often bottlenecked by static subword tokenizers (e.g., BPE (Sennrich et al., 2016)). When applied to new scripts, these tokenizers tend to fragment words inefficiently into byte-level sequences, resulting in high token fertility, increased latency, and diluted semantics (Rust et al., 2021). For example, OLMo 2 (OLMo et al., 2025) tokenizes a Burmese text into a sequence 12 times longer than its English equivalent.

Prevailing solutions rely on vocabulary adaptation, such as replacing or expanding the tokenizer (Li et al., 2025; Fujii et al., 2024; Cui et al., 2023). However, these methods have several drawbacks.

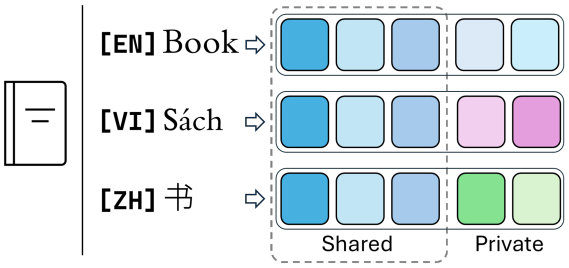

Figure 1: Illustration of embedding vectors for words with the same meaning across different languages using MATE. With English as the anchor language, some embedding dimensions in the other languages are weight-tied to improve cross-lingual transfer, while learnable private parameters capture language-specific nuances.

First, they increase the number of model parameters. More importantly, standard vocabulary expansion—which initializes new target-language tokens as independent parameters—often fails to improve downstream performance (Fujii et al., 2024). This may occur because the new tokens are undertrained, as they receive far fewer training steps than the original vocabulary. In addition, such practices can also lead to catastrophic forgetting (Csaki et al., 2024).

To address these issues, we propose MATE (Multilingual Alignment for Token Embeddings), a parameter-efficient vocabulary expansion method grounded in cross-lingual embedding alignment. Rather than initializing the embedding of a target-language token t independently, we identify its semantically equivalent token e from the model's primary language (the anchor language) that exist in the original vocabulary and align the embeddings of e and t by partially tying their parameters. In this way, the new token inherits rich pre-trained features while using fewer parameters, as illustrated in Figure 1.

Our contributions can be summarized as follows:

- We introduce a parameter-efficient vocabulary expansion method that structurally enforces

semantic alignment, improving cross-lingual transfer while reducing parameter count.

- We demonstrate that our approach outperforms baseline fine-tuning and common vocabulary expansion methods in both tokenization efficiency and downstream task performance, including machine translation, question answering, reading comprehension, and multilingual reasoning.
- We show that parameter sharing not only makes the model more efficient, but also helps it to achieve better downstream task performance despite having fewer parameters.
- We show that our method is robust to polysemy and scales well to larger models.

2 Related Work

2.1 Vocabulary Expansion

Vocabulary adaptation is often employed to adapt pre-trained LLMs to new languages, especially to improve tokenization efficiency and potentially improve the model's performance on the new languages. Existing approaches generally fall into two categories: vocabulary expansion, which merges new tokens from a target-language tokenizer into the base vocabulary (Cui et al., 2023; Fujii et al., 2024; Nguyen et al., 2024b; Bari et al., 2025; Yamaguchi et al., 2024b, 2025) and vocabulary replacement, which substitutes the base tokenizer entirely (Minixhofer et al., 2022; Dobler and de Melo, 2023; Downey et al., 2023; Li et al., 2025; Sakajo et al., 2025; Yamaguchi et al., 2024a; Minixhofer et al., 2024). Our work focuses on vocabulary expansion as we want to expand the base model's capabilities to new languages while retaining the capability in its original languages.

A critical challenge in this process is initializing the embeddings of the new tokens, as random initialization yields suboptimal performance (Minixhofer et al., 2022; Dobler and de Melo, 2023). Prior work addresses this through compositional, distributional, or semantic-transfer strategies: averaging original subword embeddings (Fujii et al., 2024), initializing new embeddings from script-wise statistics of the pretrained embedding space (Downey et al., 2023), interpolating source-token embeddings using nearest neighbors in external static embedding spaces (Minixhofer et al., 2022), initializing factorized PLM embedding coordinates as similarity-weighted combinations of source subword coordinates (Liu et al., 2024), training auxiliary predictors (Dobler and de Melo, 2023), using hypernetwork predictors (Özeren et al., 2025), transferring from target-language models (Lee et al., 2025), copying the closest source-token embedding under GloVe (Pennington et al., 2014) similarity (Li et al., 2025), or using bilingual dictionaries to map new target-language subwords to source-language subwords (Sakajo et al., 2025). Unlike approaches that only set the initial embedding values or use low-rank factorization without explicit lexical parameter sharing, MATE initializes part of each new token embedding using the exact pre-trained embedding of its English anchor and maintains *persistent weight tying* throughout training. This anchors target-language tokens to established English representations, enabling immediate cross-lingual alignment while being 8× more parameter efficient than standard vocabulary expansion.

2.2 Cross-Lingual Input and Output Embedding Alignment

Prior work has explored aligning the input embeddings of semantically equivalent words. In the context of transformers, Kautsar and Koto (2025) fully aligned the embeddings of 13 low-resource languages to English within mBERT, demonstrating performance improvements in classification tasks. More closely related to our work, Liu et al. (2019) introduced a partial alignment strategy for vanilla transformers in neural machine translation. Their method aligned word embeddings with similar lexical meaning, same word form, and randomly chosen unrelated words. Since Liu et al. (2019) worked on an encoder-decoder architecture, their parameter sharing was applied on two disjoint vocabularies (i.e., encoder for source language and decoder for target language). Unlike prior work, we apply partial alignment to expand the vocabulary of decoder-only generative LLMs, which use a unified vocabulary for all languages, for input and output.

3 Methodology

In this section, we first describe our vocabulary expansion strategy for broad multilingual support in Section 3.1, followed by our method for representing the added tokens via parameter sharing with their semantically equivalent tokens in the anchor language in Section 3.2. Since most LLMs are primarily built for English, we explain our method using English as the anchor language.

3.1 Pairwise Vocabulary Expansion

Our approach utilizes bilingual dictionaries mapping words from each target language to English. However, existing resources such as MUSE (Conneau et al., 2018) are insufficient: the MUSE bilingual dictionaries contain only approximately 21,600 word pairs, of which only 3,470 English words overlap with the base OLMo 2 tokenizer vocabulary. Furthermore, MUSE does not cover all target languages.

To address these limitations, we construct our own translation dictionaries by running a word aligner on parallel corpora for each language pair. Let $\mathcal{L} = \{\ell_1, \ldots, \ell_n\}$ denote the set of target languages. Our alignment pipeline consists of two phases: *Word Mapping* and *Token-ID Mapping*.

Word Mapping Since not all languages are space-separated, we first tokenize the parallel corpora $\mathcal{D}_\ell$ using language-specific word tokenizers to obtain word-level units (Table 8 in Appendix). For each language $\ell \in \mathcal{L}$, we extract bidirectional alignments using `fast_align` (Dyer et al., 2013). These alignments are aggregated into one-to-many mappings, where $c_{t,e}$ denotes the frequency with which target-language word t is aligned to English word e.

To ensure semantic quality, we apply a rigorous filtering pipeline. Let $\mathcal{W}$ denote the combined English lexicon from the NLTK Corpus word list (Bird and Loper, 2004) and WordNet (Miller, 1994), and $\mathcal{S}$ denote English stopwords from SpaCy (Honnibal et al., 2020). We discard an alignment pair (t, e) if it meets any of the following criteria:

- **English Lexicon Filtering:** The English word e does not exist in the English lexicon $\mathcal{W}$, or the target-language word t appears in $\mathcal{W}$ (to prevent code-switched English words from being recognized as part of the target language).
- **Stopword Filtering:** e is a stopword ($e \in \mathcal{S}$).
- **Script Validity:** The target-language word t contains characters outside the standard Unicode script range for language ℓ.
- **Low Frequency:** The alignment frequency $c_{t,e}$ is less than 10 (to reduce noise from alignment errors or rare contexts).

After filtering, we retain the top $K = 50{,}000$ most frequent non-English words per language. To prevent ambiguity from cross-lingual homographs (identical words with different meanings across languages), we enforce language exclusivity: any non-English word t that appears in multiple languages is excluded. We denote the resulting frequency-weighted word alignment set for language ℓ as $\mathcal{B}_\ell = \{(t, e, c_{t,e})\}$, where a target-language word t may be associated with multiple English candidates e. The collection across all target languages is $\mathcal{B} = \{\mathcal{B}_\ell\}_{\ell \in \mathcal{L}}$.

Token-ID Mapping The Word Mapping stage therefore produces a frequency-weighted multimap rather than a function. We convert this multimap into the token-level alignment dictionary used by MATE in three steps. First, we keep only triples $(t, e, c_{t,e})$ where the non-English word t is absent from the base vocabulary V_{base} and the English word e is already present in V_{base}. This preserves non-English tokens that already exist in the original tokenizer, while ensuring that every retained English anchor has an existing pretrained embedding. Second, if a retained non-English word t is aligned to multiple English words, we choose the most frequent alignment as its unique anchor, $e^* = \arg\max_e c_{t,e}$. This converts the filtered one-to-many word mappings into one-to-one token alignments. Finally, we add each retained non-English word t to the vocabulary with a new token ID and record the alignment $\mathcal{A} : t \to e^*$. Since all added items come directly from filtered word-level mappings, the added tokens are full words rather than subword fragments. Table 9 (Appendix) lists the number of tokens added per language.

3.2 Embedding Weight Tying

To promote a shared semantic space and cross-lingual transfer while minimizing the parameter increase caused by vocabulary expansion, we apply partial weight tying between the input and output embeddings of a new token $\boldsymbol{v_t}$ and its English anchor $\boldsymbol{v_e}$. For each pair $(\boldsymbol{v_t}, \boldsymbol{v_e})$, we tie the first k dimensions of their embeddings. These k dimensions form the *shared embedding*, which remains identical for both tokens throughout training, while the remaining $d - k$ dimensions (where d is the embedding dimension) form the *private embedding*, capturing language-specific nuances.

We can view the embedding vector $\boldsymbol{v_t} = \{v_{t(1)}, v_{t(2)}, \ldots, v_{t(d)}\} \in \mathbb{R}^d$ as a concatenation of the first k parameters of the English anchor embedding vector and a private embedding vector $\boldsymbol{p_t} = \{p_{t(1)}, p_{t(2)}, \ldots, p_{t(d-k)}\} \in \mathbb{R}^{d-k}$. Each pa-

rameter $v_{t(i)}$ is defined as follows:

$$v_{t(i)} = \begin{cases} v_{e(i)} & \text{for } 1 \leq i \leq k \\ p_{t(i-k)} & \text{for } i > k \end{cases} \quad (1)$$

The shared embedding does not require weight initialization, as its parameters are tied to the English embedding. For the private embedding, each parameter $p_{t(i)}$ is initialized using the corresponding dimension of the English anchor token embedding $v_{e(k+i)}$, but it learns its own weights independently during training.

3.3 Inference

At inference time, MATE applies a lossless two-phase tokenization scheme that remains compatible with Byte Pair Encoding (BPE) (Sennrich et al., 2016). First, it scans the input text, greedily extracts added tokens and then tokenizes them. Second, it applies the original BPE merge rules to the remaining text spans and interleaves the resulting base-token IDs with the aligned-token IDs. The full procedure and pseudocode are provided in Appendix B.

4 Experiments

To validate the effectiveness of our proposed embedding alignment strategy, we conduct comprehensive experiments using the OLMo 2 1B model (OLMo et al., 2025), expanding its vocabulary to cover ten Southeast Asian languages: Indonesian (`id`), Khmer (`km`), Lao (`lo`), Malay (`ms`), Burmese (`my`), Tamil (`ta`), Thai (`th`), Tagalog (`tl`), Vietnamese (`vi`), and Chinese (`zh`).

4.1 Experimental Setup

Since parallel data have been shown to be more effective than monolingual corpora for improving LLMs' multilingual capabilities (Qorib et al., 2025), we follow the training setup of Nguyen et al. (2026) and adapt the LLM by performing continual pretraining (CPT) exclusively on parallel data, ensuring that our results are comparable to theirs. We use sentence pairs of English and each of the ten Southeast Asian languages from the NLLB parallel corpus (Costa-jussà et al., 2024) as the training data. To prevent catastrophic forgetting, we allocate 25% of the training tokens to replay data, using the stage 2 training data of OLMo 2. The total CPT dataset amounts to 10B tokens (measured using the OLMo 2 tokenizer). To ensure a fair comparison, all models—including the baselines and our proposed method—are trained with identical training data content and ordering. The number of sentences for each language in the training data is provided in Table 10 (Appendix).

Token Mapping Generation Dataset We use `fast_align` to generate token mappings, performing the alignment independently for each of the ten target languages. To ensure a sufficient number of mappings, we select source corpora based on resource availability. For Indonesian, Malay, Burmese, Tamil, Thai, Tagalog, Vietnamese, and Chinese, we use the NLLB parallel corpus (Costa-jussà et al., 2024). For Khmer and Lao, we use the FineTranslations parallel corpus (Penedo et al., 2026) to obtain more robust alignments.

Embedding Alignment Configuration For our proposed method, we introduce a hyperparameter that controls the proportion of embedding parameters that are tied. For each aligned token pair, the first 87.5% of the input and output embedding parameters are shared between the new non-English token and its English anchor, while the remaining 12.5% are left as learnable private parameters.

4.2 Baselines

We benchmark our proposed alignment strategy against six distinct baselines.

1. **Baseline:** We fine-tune the original model directly on the target language data without any architectural modification or vocabulary expansion. This baseline utilizes the original English-centric tokenizer, forcing non-English text to be fragmented into shorter subword tokens or raw bytes.

2. **Vocab Expansion + Mean Init:** To ensure a strictly fair comparison with the additional capacity introduced by MATE, we expand the base tokenizer until the total parameter count exactly matches that of our proposed method. For each newly added token, the input embedding and output weight are initialized as the mean of the component tokens obtained when the new token is tokenized by the original tokenizer. The detailed vocabulary expansion and merge-selection procedure is described in Appendix G.1.

3. **Vocab Expansion + FOCUS:** This configuration uses the same parameter-matched vocab-

ulary expansion protocol as *Vocab Expansion + Mean Init*, but replaces mean initialization with the method proposed by FOCUS (Dobler and de Melo, 2023). The detailed implementation is described in Appendix G.2.

4. **Vocab Expansion + ZeTT:** This configuration also uses the same expanded vocabulary, but initializes the newly added token embedding parameters with the prediction of a hypernetwork proposed by Zero-Shot Tokenizer Transfer (ZeTT) (Minixhofer et al., 2024). The detailed implementation is described in Appendix G.3.

5. **Vocab Expansion + Sakajo et al.:** Unlike the other vocabulary expansion baselines, which reuse the same parameter-matched expanded tokenizer as *Vocab Expansion + Mean Init*, this baseline uses a different expanded vocabulary. Following the dictionary-based tokenizer transfer strategy of Sakajo et al. (2025), we train a byte-level BPE tokenizer on bilingual dictionary entries derived from the word-to-word mapping $\mathcal{B}$ produced by our Word Mapping stage. We then merge this dictionary-trained tokenizer into the original OLMo tokenizer using the same ID-preserving merge procedure as *Vocab Expansion + Mean Init*. The embeddings of newly added tokens are initialized using the dictionary-based subword mapping strategy proposed by Sakajo et al. (2025). The detailed implementation is described in Appendix G.4.

6. **Vocab Expansion + TokAlign:** This configuration follows the same parameter-matched vocabulary expansion protocol as the *Vocab Expansion + Mean Init* baseline above. However, instead of mean initialization, the input and output embeddings for the newly added tokens are initialized using TokAlign (Li et al., 2025). The detailed weight initialization procedure is described in Appendix G.5.

Training Hyperparameters All models fine-tuned in this paper use the same training hyperparameters. The full details, including the learning rate, batch size, and optimizer settings, are documented in Appendix H.

4.3 Evaluation

We evaluate cross-lingual generation and understanding on four benchmarks: FLORES-200 devtest translation (Costa-jussà et al., 2024) in both English-to-Target (E→X) and Target-to-English (X→E) directions, XQuAD question answering (Artetxe et al., 2020), Belebele reading comprehension (Bandarkar et al., 2024), and Multilingual MMLU general knowledge (Lai et al., 2023). Translation uses 5-shot prompting and is reported with BLEU and chrF++; XQuAD is reported with exact match (EM).

We measure tokenization efficiency using Compression Ratio (CR) on the FLORES-200 test set. For language ℓ, let T_ℓ be the number of tokens produced by the evaluated method and T_e be the number of English tokens. We fix T_e by always tokenizing the English reference with the original OLMo 2 tokenizer, so differences in CR reflect target-language tokenization efficiency:

$$CR_\ell = \frac{T_\ell}{T_e} \tag{2}$$

Lower CR is better, with values closer to 1.0 indicating tokenization as compact as English. We summarize efficiency using average CR ($\overline{CR}$) and disparity across languages using the Gini coefficient:

$$G = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N}|CR_i - CR_j|}{2N^2\overline{CR}} \tag{3}$$

For tokenization speed, we use a separate FineWeb2 subset (Penedo et al., 2025) containing exactly 0.23 GB of text for English and each of the ten target languages (`id`, `km`, `lo`, `ms`, `my`, `ta`, `th`, `tl`, `vi`, and `zh`), then concatenate all language-specific subsets into a single corpus. We report throughput, total produced tokens, and total time on an Intel(R) Xeon(R) Platinum 8559C CPU with 192 CPU threads. For downstream tasks, we assess statistical significance using paired bootstrap resampling (Koehn, 2004) with 1,000 samples.

We report VOCAB EXPANSION + SAKAJO ET AL. separately for tokenization because it uses a dictionary-trained expanded vocabulary; the other vocabulary-expansion variants share the same tokenizer as VOCAB EXPANSION.

5 Results

5.1 Downstream Task Performance

As shown in Table 1, MATE achieves the highest task-average score across all downstream tasks, with statistically significant improvements over all baselines. It improves over the baseline on both

Method	E→X Translation		X→E Translation		XQuAD	Belebele	M_MMLU
	BLEU	chrF++	BLEU	chrF++	EM (%)	Acc. (%)	Acc. (%)
Baseline	24.12	39.22	26.35	46.19	4.35	24.47	28.42
VE + Mean Init	25.45	43.52*	27.59	50.69*	17.35*	25.89*	27.71
VE + FOCUS	21.42	36.33	26.11	47.33	13.91	24.66	28.53
VE + ZeTT	22.96	39.52	23.63	45.07	13.87	25.82*	27.79
VE + Sakajo et al.	24.63*	42.15*	27.58*	50.14*	17.84*	27.24*	27.56
VE + TokAlign	23.54	40.40*	26.02	47.92*	14.56	26.56	28.63
MATE	**27.62***	**46.13***	**31.14***	**54.94***	**29.18***	**27.73***	**29.11***

Table 1: Task-average downstream performance. Translation uses BLEU and chrF++; other metrics are shown in the headers. Bold/underline mark the best/second-best scores. * indicates statistically significant improvement over Baseline ($p < 0.05$). Language-specific results are in Appendix F.

Tokenizer	Avg CR↓	Gini↓
Baseline	4.77	0.42
VE	2.02	0.24
VE + Sakajo et al.	2.47	0.31
MATE	**1.55**	**0.13**

Table 2: Tokenization efficiency by average Compression Ratio (CR) and Gini coefficient across 11 languages. Lower is better; bold/underline mark the best/second-best scores. Language-specific results are in Appendix E.

Tokenizer	Tokens/s↑	Tokens↓	Time (s)↓
Baseline	**658.64K**	1.11B	1680.85
VE	369.06K	575.59M	1559.62
VE + Sakajo et al.	401.20K	646.17M	1610.59
MATE	398.43K	**523.53M**	**1313.98**

Table 3: Tokenization speed benchmark on the concatenated FineWeb2 subset. Tokens/s reports aggregate throughput across all evaluated languages.

generation and understanding tasks, indicating that the gains generalize across multilingual evaluations. In translation, the largest improvements occur in Khmer, Lao, and Burmese, the three languages with the least training data (Tables 10 and 12), suggesting that MATE mitigates data scarcity through transfer from semantically equivalent English tokens. Per-language scores are provided in Appendix F. A broader resource-level analysis in Appendix I shows that MATE's gains are largest in lower-resource settings, with an exponential decay trend explaining most of the variance in translation improvements.

5.2 Tokenization Efficiency and Equity

In addition to achieving the best downstream task performance, MATE also improves over all baselines in tokenization efficiency and equity, yielding the best average efficiency (Avg CR = 1.55) and the most equitable performance distribution across the ten languages ($G = 0.13$) (Table 2). Per-language scores are provided in Appendix E.

Table 3 shows that raw throughput alone does not determine total tokenization time. The BASELINE tokenizer has the highest tokens/s because its smaller vocabulary requires fewer BPE merge operations, but it produces the largest token count. MATE has the lowest total time because it produces substantially fewer tokens while retaining throughput close to the vocabulary-expansion tokenizers, and it can be implemented with the Hugging Face Tokenizers library (Moi and Patry, 2023) without custom tokenization code.

6 Ablation Study

We isolate the contribution of each MATE component in Table 4: PAIRWISE VOCABULARY EXPANSION (PVE), EMBEDDING WEIGHT TYING (EWT), and an alternative to EWT that initializes aligned tokens from their English anchors without tying the parameters, which we call SEMANTIC EMBEDDING INITIALIZATION (SEI). The configurations are:

1. **Baseline:** The base OLMo 2 1B model without modification.
2. **+PVE:** Expand the vocabulary using all translation-mapped tokens from the alignment set $\mathcal{A}$. New token embeddings are initialized randomly.

Config	Params	E→X BLEU	X→E BLEU	XQuAD EM (%)	Belebele Acc. (%)	M_MMLU Acc. (%)	Avg
Baseline	1.48B	24.12	26.35	4.35	24.47	28.42	21.54
+PVE	2.81B	26.74	27.05	13.43	26.08	27.84	24.23
+PVE+SEI	2.81B	**28.07**	30.03	27.06	27.39	28.39	28.19
MATE	1.65B	27.62	**31.14**	**29.18**	**27.73**	**29.11**	**28.96**

Table 4: Ablation performance including parameter counts and macro-averages. M_MMLU denotes Multilingual MMLU (Lai et al., 2023). Bold indicates the best score, and underline indicates the second best.

Ratio	Params	E→X BLEU	X→E BLEU	XQuAD EM (%)	Belebele Acc. (%)	M_MMLU Acc. (%)	Avg
0%	2.81B	28.07	30.03	27.06	27.39	28.39	28.19
12.5%	2.64B	**28.40**	**32.41**	23.07	**27.98**	28.65	28.10
43.75%	2.23B	28.11	31.74	26.89	25.80	28.37	28.18
87.5%	1.65B	27.62	31.14	**29.18**	27.73	**29.11**	**28.96**
93.75%	1.57B	27.00	30.01	23.93	25.88	28.90	27.14

Table 5: Sensitivity to the embedding alignment ratio. Scores are macro-averages across the evaluated languages for each task. Bold indicates the best score, and underline indicates the second best.

3. **+PVE+SEI:** The embedding of each new token v_t is initialized using the pre-trained embedding of its English anchor token v_e.
4. **+PVE+EWT (MATE):** The input and output embeddings of the new tokens are partially weight-tied to the English anchor tokens via a shared embedding.

The ablation shows that each component contributes to performance. +PVE+SEI improves over +PVE, confirming the value of English-anchor initialization, while MATE achieves the highest average score despite using far fewer parameters than +PVE+SEI and +PVE. Detailed results are in Appendix K.

7 Analysis

7.1 Embedding Alignment Ratio Sensitivity

The embedding alignment ratio is the fraction of each new token embedding shared with its English anchor. Higher ratios reduce parameters but impose stronger cross-lingual sharing. We vary this ratio while fixing the vocabulary, training data, and hyperparameters, comparing 0%, 12.5%, 43.75%, 87.5%, and 93.75%; 87.5% is the default MATE configuration. Results are shown in Table 5.

The optimal ratio varies by task: 12.5% performs best on translation and Belebele, whereas 87.5% performs best on XQuAD and M_MMLU and achieves the highest average. Since 12.5% retains a substantially larger model and 93.75% overconstrains the embeddings, we use 87.5% as the best tradeoff between accuracy and parameter count. Complete task-level results are in Appendix J.

7.2 Robustness to Polysemy

One potential concern with explicit cross-lingual embedding alignment is how it handles polysemous words. Since MATE structurally ties the embedding of a target-language token to a single English anchor token, there is a risk that the target token may become semantically restricted to only the meaning of its anchor token, thereby losing its ability to represent other meanings. For example, the word "加油" in Chinese can mean "refuel" or "cheer" in English. If we tie the embedding of "加油" to "cheer", it may disrupt the model's performance on sentences where "加油" means "refuel".

To investigate this issue, we evaluate the model's ability to accurately disambiguate polysemous Chinese words across different contexts using the Chinese subset of the OntoNotes dataset (Pradhan et al., 2013; Hadiwinoto et al., 2019). The original OntoNotes word sense disambiguation data consists of triplets of test sentences, ambiguous Chinese words, and sense IDs, with each sense defined in English. We first paraphrase each English sense definition using a single English word using

Model (7B)	E→X BLEU	X→E BLEU	XQuAD EM (%)	Bel. Acc. (%)	M_MMLU Acc. (%)	Avg
Baseline	29.30	36.34	28.72	42.77	**40.39**	35.50
VE + Sakajo et al.	29.21	35.18	23.05	39.22	36.75	32.68
MATE	**30.56**	**36.58**	**36.26***	**45.98***	40.22	**37.92**

Table 7: Performance comparison on the OLMo 2 7B model to demonstrate scalability. Bel. denotes Belebele. M_MMLU denotes Multilingual MMLU. Bold marks the highest score per column. * indicates statistically significant improvement over Baseline ($p < 0.05$).

GPT-4o (Hurst et al., 2024) to obtain a candidate translation. We then follow the prompting method of Kang et al. (2024). Specifically, given an ambiguous Chinese word in a sentence, the model is prompted with: "在"{Chinese sentence}"这句话中, "{Chinese word}"这个词翻译成英文为" (In English: "In the sentence '{Chinese sentence}', the word '{Chinese word}' is translated into English as"). We then compute the log probabilities of all candidate English translations and consider the prediction correct if the most probable translation appears in the ground-truth translation set. The evaluation dataset contains 8,355 instances covering 324 distinct ambiguous Chinese words.

On this task, MATE achieves an accuracy of 46.04%, outperforming the Baseline model's 43.84% (Table 6).

Model	Accuracy (%)
Baseline	43.84
MATE	**46.04**

Table 6: Accuracy comparison on the Chinese subset of the OntoNotes dataset.

This improvement over the baseline indicates that MATE is as robust to polysemy as the baseline, even when a part of their embedding parameters is tied to the embedding of a single English anchor token. Its robustness is not surprising. In a standard decoder-only LLM without embedding alignment, each polysemous word is represented by the same input embedding regardless of its meaning.

7.3 Scalability to Larger Models

To verify scalability, we apply MATE to OLMo 2 7B (OLMo et al., 2025) and compare it with a vanilla 7B baseline and a dictionary-based vocabulary-expansion baseline (VE + Sakajo et al.) trained on the same data as our 1B experiments. As shown in Table 7, MATE achieves the best average score across five tasks, outperforming both the vanilla baseline (37.92 vs. 35.50) and VE + Sakajo et al. (37.92 vs. 32.68). The gains are smaller than in the 1B setting because the 7B baseline is substantially stronger; per-language results are in Appendix L.

8 Conclusion

In this paper, we introduced MATE, a parameter-efficient vocabulary expansion scheme that partially weight-ties non-English tokens to their semantically equivalent English tokens. This approach improves cross-lingual transfer from English while preserving the model's capacity to learn the unique, language-specific characteristics of each token. MATE significantly improves parameter efficiency, tokenization efficiency, and downstream task performance across ten new languages.

Our comprehensive evaluation demonstrates that this method yields exceptionally strong gains for lower resource languages, actively reducing the performance disparity between high- and low-resource settings. Our analysis shows that parameter sharing across tokens with semantically similar meanings helps the model achieve better downstream task performance despite having fewer parameters. We also show that our method is robust to polysemy and achieves the best average performance in the 7B setting, establishing MATE as a highly effective approach for vocabulary expansion.

9 Limitations

While MATE demonstrates significant improvements in cross-lingual transfer and parameter efficiency, we acknowledge several limitations in our current work. First, we limit our investigation to extending OLMo 2 to ten Southeast Asian languages—comprising a diverse mix of high- and low-resource languages with both Latin and non-Latin scripts. While we believe our method is applicable to other languages, we leave that exploration

to future work.

Second, although our scalability experiment shows that MATE remains effective on OLMo 2 7B, we have only evaluated models up to the 7B parameter scale. Larger models may exhibit different scaling behavior, training dynamics, or sensitivity to vocabulary expansion, and we leave evaluation beyond 7B to future work.

Third, the models evaluated in this paper are base foundation models that have not yet been instruction-tuned, and they have not undergone rigorous safety tuning, red-teaming, or value alignment. Consequently, they may reflect biases present in the training corpora or generate unsafe content. Therefore, they are intended primarily for research purposes and should be carefully evaluated before being deployed in user-facing applications.

References

Vu Anh. 2018. Underthesea: Vietnamese NLP toolkit.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of ACL*.

Thura Aung and Ye Kyaw Thu. 2025. myTokenize: comprehensive tokenization library for Myanmar language.

Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Albert Q. Jiang, Jia Deng, Stella Biderman, and Sean Welleck. 2024. Llemma: an open language model for mathematics. In *Proceedings of ICLR*.

Lucas Bandarkar, Davis Liang, Benjamin Muller, Mikel Artetxe, Satya Narayan Shukla, Donald Husa, Naman Goyal, Abhinandan Krishnan, Luke Zettlemoyer, and Madian Khabsa. 2024. The Belebele benchmark: a parallel reading comprehension dataset in 122 language variants. In *Proceedings of ACL*.

M Saiful Bari, Yazeed Alnumay, Norah A Alzahrani, Nouf M Alotaibi, Hisham Abdullah Alyahya, Sultan AlRashed, Faisal Abdulrahman Mirza, Shaykhah Z Alsubaie, Hassan A Alahmed, Ghadah Alabduljabbar, and 1 others. 2025. ALLaM: large language models for Arabic and English. In *Proceedings of ICLR*.

Steven Bird and Edward Loper. 2004. NLTK: the natural language toolkit. In *Proceedings of ACL*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, and others. 2020. Language models are few-shot learners. In *Proceedings of NeurIPS*.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *Proceedings of ICLR*.

Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, and 20 others. 2024. Scaling neural machine translation to 200 languages. *Nature*.

Zoltan Csaki, Pian Pawakapan, Urmish Thakker, and Qiantong Xu. 2024. Efficiently adapting pretrained language models to new languages. *arXiv preprint arXiv:2311.05741*.

Yiming Cui, Ziqing Yang, and Xin Yao. 2023. Efficient and effective text encoding for Chinese LLaMA and Alpaca. *arXiv preprint arXiv:2304.08177*.

Konstantin Dobler and Gerard de Melo. 2023. FOCUS: effective embedding initialization for monolingual specialization of multilingual models. In *Proceedings of EMNLP*.

C. M. Downey, Terra Blevins, Nora Goldfine, and Shane Steinert-Threlkeld. 2023. Embedding structure matters: comparing methods to adapt multilingual vocabularies to new languages. In *Proceedings of MRL*.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of NAACL*.

Kazuki Fujii, Taishi Nakamura, Mengsay Loem, Hiroki Iida, Masanari Ohi, Kakeru Hattori, Hirai Shota, Sakae Mizuki, Rio Yokota, and Naoaki Okazaki. 2024. Continual pre-training for cross-lingual LLM adaptation: enhancing Japanese language capabilities. In *Proceedings of COLM*.

Christian Hadiwinoto, Hwee Tou Ng, and Wee Chung Gan. 2019. Improved word sense disambiguation using pre-trained contextualized word representations. In *Proceedings of EMNLP-IJCNLP*.

Viet Hoang. 2023. khmer-nltk: Khmer language processing toolkit.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: industrial-strength natural language processing in Python.

Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, and members of OpenAI. 2024. GPT-4o system card. *arXiv preprint arXiv:2410.21276*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of EACL*.

Haoqiang Kang, Terra Blevins, and Luke Zettlemoyer. 2024. Translate to disambiguate: zero-shot multilingual word sense disambiguation with pretrained language models. In *Proceedings of EACL*.

Muhammad Dehan Al Kautsar and Fajri Koto. 2025. Parallel tokenizers: rethinking vocabulary design for cross-lingual transfer. *arXiv preprint arXiv:2510.06128*.

Denis Kocetkov, Raymond Li, Loubna Ben Allal, Jia Li, Chenghao Mou, Carlos Muñoz Ferrandis, Yacine Jernite, Margaret Mitchell, Sean Hughes, Thomas Wolf, Dzmitry Bahdanau, Leandro von Werra, and Harm de Vries. 2023. The Stack: 3 TB of permissively licensed source code. *Transactions on Machine Learning Research*.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP*.

Anoop Kunchukuttan. 2020. The IndicNLP library.

Viet Lai, Chien Nguyen, Nghia Ngo, Thuat Nguyen, Franck Dernoncourt, Ryan Rossi, and Thien Nguyen. 2023. Okapi: instruction-tuned large language models in multiple languages with reinforcement learning from human feedback. In *Proceedings of EMNLP*.

Seungyoon Lee, Seongtae Hong, Hyeonseok Moon, and Heuiseok Lim. 2025. Semantic aware linear transfer by recycling pre-trained language models for cross-lingual transfer. In *Findings of ACL*.

Chong Li, Jiajun Zhang, and Chengqing Zong. 2025. TokAlign: efficient vocabulary adaptation via token alignment. In *Proceedings of ACL*.

Xuebo Liu, Derek F. Wong, Yang Liu, Lidia S. Chao, Tong Xiao, and Jingbo Zhu. 2019. Shared-private bilingual word embeddings for neural machine translation. In *Proceedings of ACL*.

Yihong Liu, Peiqin Lin, Mingyang Wang, and Hinrich Schuetze. 2024. OFA: a framework of initializing unseen subword embeddings for efficient large-scale multilingual continued pretraining. In *Findings of NAACL*.

Andre Martins and Ramon Astudillo. 2016. From softmax to sparsemax: a sparse model of attention and multi-label classification. In *Proceedings of ICML*.

George A. Miller. 1994. WordNet: a lexical database for English. In *Proceedings of HLT*.

Benjamin Minixhofer, Fabian Paischer, and Navid Rekabsaz. 2022. WECHSEL: effective initialization of subword embeddings for cross-lingual transfer of monolingual language models. In *Proceedings of NAACL*.

Benjamin Minixhofer, Edoardo M. Ponti, and Ivan Vulić. 2024. Zero-shot tokenizer transfer. In *Proceedings of NeurIPS*.

Anthony Moi and Nicolas Patry. 2023. Hugging Face tokenizers: fast state-of-the-art tokenizers optimized for research and production.

Tan Sang Nguyen, Muhammad Reza Qorib, and Hwee Tou Ng. 2026. OpenSeal: good, fast, and cheap construction of an open-source Southeast Asian LLM via parallel data. *arXiv preprint arXiv:2602.02266*.

Thuat Nguyen, Chien Van Nguyen, Viet Dac Lai, Hieu Man, Nghia Trung Ngo, Franck Dernoncourt, Ryan A. Rossi, and Thien Huu Nguyen. 2024a. CulturaX: a cleaned, enormous, and multilingual dataset for large language models in 167 languages. In *Proceedings of LREC-COLING*.

Xuan-Phi Nguyen, Wenxuan Zhang, Xin Li, Mahani Aljunied, Zhiqiang Hu, Chenhui Shen, Yew Ken Chia, Xingxuan Li, Jianyu Wang, Qingyu Tan, Liying Cheng, Guanzheng Chen, Yue Deng, Sen Yang, Chaoqun Liu, Hang Zhang, and Lidong Bing. 2024b. SeaLLMs – large language models for Southeast Asia. In *Proceedings of ACL*.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, and 24 others. 2025. 2 OLMo 2 furious. *arXiv preprint arXiv:2501.00656*.

Guilherme Luccas Özeren, Yihong Liu, and Hinrich Schuetze. 2025. HYPEROFA: expanding LLM vocabulary to new languages via hypernetwork-based embedding initialization. In *Proceedings of ACL SRW*.

Guilherme Penedo, Hynek Kydlíček, Vinko Sabolčec, Bettina Messmer, Negar Foroutan, Amir Hossein Kargaran, Colin Raffel, Martin Jaggi, Leandro von Werra, and Thomas Wolf. 2025. FineWeb2: one pipeline to scale them all – adapting pre-training data processing to every language. In *Proceedings of COLM*.

Guilherme Penedo, Hynek Kydlíček, Amir Hossein Kargaran, and Leandro von Werra. 2026. FineTranslations.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: global vectors for word representation. In *Proceedings of EMNLP*.

Wannaphong Phatthiyaphaibun. 2022. LaoNLP: Lao language natural language processing.

Wannaphong Phatthiyaphaibun, Korakot Chaovavanich, Charin Polpanumas, Arthit Suriyawongkul, Lalita Lowphansirikul, Pattarawat Chormai, and Peerat Limkonchotiwat. 2023. PyThaiNLP: Thai natural language processing in Python. In *Proceedings of NLP-OSS*.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using OntoNotes. In *Proceedings of CoNLL*.

Muhammad Reza Qorib, Junyi Li, and Hwee Tou Ng. 2025. Just go parallel: improving the multilingual capabilities of large language models. In *Proceedings of ACL*.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2021. How good is your tokenizer? on the monolingual performance of multilingual language models. In *Proceedings of ACL*.

Haruki Sakajo, Yusuke Ide, Justin Vasselli, Yusuke Sakai, Yingtao Tian, Hidetaka Kamigaito, and Taro Watanabe. 2025. Dictionaries to the rescue: cross-lingual vocabulary transfer for low-resource languages using bilingual dictionaries. In *Findings of ACL*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of ACL*.

Junyi Sun. 2012. Jieba: Chinese text segmentation.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, and others. 2023. LLaMA: open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Atsuki Yamaguchi, Terufumi Morishita, Aline Villavicencio, and Nikolaos Aletras. 2025. Adapting chat language models using only target unlabeled language data. *Transactions on Machine Learning Research*.

Atsuki Yamaguchi, Aline Villavicencio, and Nikolaos Aletras. 2024a. An empirical study on cross-lingual vocabulary adaptation for efficient language model inference. In *Findings of EMNLP*.

Atsuki Yamaguchi, Aline Villavicencio, and Nikolaos Aletras. 2024b. How can we effectively expand the vocabulary of LLMs with 0.01 GB of target language text? *arXiv preprint arXiv:2406.11477*.

A Language-Specific Word Tokenizers

We list the language-specific word tokenizers in Table 8.

Code	Tokenizer
en	`spacy` (Honnibal et al., 2020)
id	`spacy` (Honnibal et al., 2020)
km	`khmer-nltk` (Hoang, 2023)
lo	`laonlp` (Phatthiyaphaibun, 2022)
ms	`spacy` (Honnibal et al., 2020)
my	`myTokenize` (Aung and Thu, 2025)
ta	`indicnlp` (Kunchukuttan, 2020)
th	`pythainlp` (Phatthiyaphaibun et al., 2023)
tl	`spacy` (Honnibal et al., 2020)
vi	`underthesea` (Anh, 2018)
zh	`jieba` (Sun, 2012)

Table 8: Language-specific word tokenizers used to preprocess parallel corpus $\mathcal{D}_\ell$.

B MATE Tokenization Inference Scheme

Standard BPE tokenization applies a fixed set of learned merge rules to segment text into subword tokens. When new tokens are added to the vocabulary without corresponding merge rules, the BPE algorithm cannot produce them during tokenization. To address this, we introduce a two-phase tokenization scheme that first identifies and extracts the added aligned tokens from the input text, then applies standard BPE tokenization to the remaining text.

Let V_{base} denote the base tokenizer's vocabulary and R_{merge} its existing BPE merge rules. We define V_{added} as the set of added tokens, and $V_{new} = V_{base} \cup V_{added}$ as the augmented vocabulary.

In Phase 1 (priority tokenization), the algorithm scans the input string left to right and attempts to match substrings against V_{added} using a greedy longest-match-first strategy that selects the longest candidate at each position to minimize fragmentation. The matching criterion is language-dependent: for space separated languages (e.g., Indonesian, Malay, Tagalog, Vietnamese), a substring is matched to $t \in V_{added}$ only if it constitutes a full word delimited by whitespace on both sides, preventing partial matches within longer words; for non-space separated languages (e.g., Chinese, Thai, Khmer, Lao, Burmese, Tamil), whitespace is not available, so we permit flexible substring matching without requiring surrounding whitespace. Matched tokens are stored as aligned token IDs, while unmatched characters are accumulated into free text segments. In Phase 2 (BPE tokenization), each free text segment is tokenized using the original BPE rules R_{merge} over V_{base}, and the resulting subword tokens are interleaved with the aligned token IDs to produce the final token sequence. This two-phase design ensures that the added aligned tokens are always tokenized as single tokens, preserving the one-to-one correspondence between target-language tokens and their English anchors that underlies MATE's embedding alignment. The algorithm is provided in Algorithm 1.

C Number of Added Tokens

Table 9 details the number of unique tokens added to the vocabulary for each language using the process described in Section 3.1.

Language	Added Tokens
Indonesian (id)	15,562
Khmer (km)	8,390
Lao (lo)	32,109
Malay (ms)	16,205
Burmese (my)	44,760
Tamil (ta)	49,843
Thai (th)	44,559
Tagalog (tl)	45,251
Vietnamese (vi)	18,654
Chinese (zh)	48,773
Total Added	**324,106**

Table 9: Number of new tokens added to the vocabulary for each target language after filtering.

D Training Data Statistics

Language	Training Sentences
English (en)	110,138,188
Indonesian (id)	13,429,498
Khmer (km)	5,859,442
Lao (lo)	4,166,616
Malay (ms)	19,462,686
Burmese (my)	5,865,195
Tamil (ta)	7,612,214
Thai (th)	10,081,723
Tagalog (tl)	19,474,026
Vietnamese (vi)	10,913,301
Chinese (zh)	13,273,487

Table 10: Number of sentences per language in the training data.

E Detailed Tokenization Efficiency Results

F Detailed Downstream Evaluation Results

In this section, we provide the language-specific results summarized in Table 1. The average column reports the macro-average across the languages evaluated for each task.

G Baseline Vocabulary Expansion Details

G.1 Vocab Expansion + Mean Init

The VOCAB EXPANSION + MEAN INIT baseline extends the original OLMo 2 tokenizer by adding new Byte Pair Encoding (BPE) merges trained specifically on target-language data. To ensure a strictly fair comparison, the number of added tokens (X) is tightly constrained so that the total parameter count strictly matches that of our proposed MATE method.

Because each new token introduces a new dimension vector in both the input token embedding matrix and the output language-modeling head matrix, adding a single token increases the total parameter count by $2 \times d$, where d is the model's hidden embedding dimension. Therefore, we calculate the exact number of required new tokens as $X = \lfloor (P_{target} - P_{base})/(2 \times d) \rfloor$.

First, we train a temporary reference tokenizer on a multilingual corpus constructed from the non-English segments of the parallel dataset, covering

Algorithm 1 MATE Tokenization Inference Scheme

Require: Input string **s**, Script category $C \in \{\text{Latin}, \text{Non-Latin}\}$
Require: V_{added} (Aligned tokens), V_{base} (Base vocab), R_{merge} (BPE rules)
$Segments \leftarrow []$
$u \leftarrow$ "" ▷ Buffer for free text
$i \leftarrow 0$
▷ **Phase 1: Priority Tokenization**
while $i <$ length(**s**) **do**
  $M \leftarrow \{t \in V_{added} \mid \mathbf{s}[i : i + \text{length}(t)] == t\}$
  if $C ==$ Latin **then**
    $M \leftarrow \{t \in M \mid \text{IsWordBoundary}(t, \mathbf{s}, i)\}$
  end if
  if $M \neq \emptyset$ **then**
    $k \leftarrow \arg\max_{t \in M} \text{length}(t)$ ▷ Greedy longest-match
    if $u \neq$ "" **then**
      $Segments$.Append(u) ▷ Store free text segment
      $u \leftarrow$ "" ▷ Clear buffer
    end if
    $Segments$.Append(ID(k)) ▷ Store locked aligned ID
    $i \leftarrow i + \text{length}(k)$
  else
    $u \leftarrow u + \mathbf{s}[i]$ ▷ Accumulate free text character
    $i \leftarrow i + 1$
  end if
end while
if $u \neq$ "" **then**
  $Segments$.Append(u)
end if
▷ **Phase 2: BPE Tokenization & Sequence Reconstruction**
$Output \leftarrow []$
for $seg \in Segments$ **do**
  if IsString(seg) **then**
    $Output$.Extend(BPE($seg \mid V_{base}, R_{merge}$))
  else
    $Output$.Append(seg) ▷ Append locked ID
  end if
end for
return $Output$

Method	en	id	km	lo	ms	my	ta	th	tl	vi	zh	Avg↓	Gini↓
Baseline	1.00	1.55	8.75	9.54	1.62	11.65	7.64	4.38	2.06	2.44	1.87	4.77	0.42
Vocab Expansion	1.00	**1.26**	3.32	2.78	**1.29**	3.23	2.74	2.33	1.65	**1.58**	1.01	2.02	0.24
VE + Sakajo et al.	1.00	1.31	4.92	2.88	1.35	4.12	4.54	2.88	**0.93**	1.86	1.40	2.47	0.31
MATE	1.00	1.53	**1.88**	**1.26**	1.61	**1.92**	**1.79**	**1.29**	1.68	2.09	**0.97**	**1.55**	**0.13**

Table 11: Language-specific tokenization efficiency results comparing Compression Ratio (CR) across 11 languages. Gini measures inequality of compression ratios across languages. (↓) indicates lower values are better. The best value is shown in bold, and the second-best value is underlined. Vocab Expansion + FOCUS, Vocab Expansion + ZeTT, and Vocab Expansion + TokAlign are omitted as they use the same tokenizer as Vocab Expansion.

all 10 target languages in our study. The original tokenizer is then expanded by merging in rules from this temporary tokenizer while preserving every original token ID and original merge rule. We scan the temporary tokenizer's BPE merge list in order, keep only merges that are absent from the base tokenizer, and append those novel merges to the base merge list until the target token count X is reached. This procedure adds only non-overlapping BPE merges from the target-language tokenizer, leaving the original tokenizer behavior unchanged for existing tokens while enabling more compact tokenization of the target languages. This process is formally detailed in Algorithm 2. For each added token t, we tokenize its surface form with the original tokenizer into component tokens $(s_1, \ldots, s_m)$ and initialize both the input embedding and output weight of t as the mean of the corresponding component vectors.

	id	km	lo	ms	my	ta	th	tl	vi	zh	Avg
English → Target											
Baseline	45.69	3.19	6.65	40.99	2.33	8.81	21.22	34.31	39.50	38.51	24.12
VE + Mean Init	46.35	9.61	13.08	40.72	4.78	9.82	18.56	34.06	40.21	37.27	25.45
VE + FOCUS	44.72	0.79	1.69	39.22	0.04	5.67	16.67	30.90	39.02	35.44	21.42
VE + ZeTT	46.25	3.65	6.07	39.59	0.52	6.39	17.31	34.20	39.24	36.39	22.96
VE + Sakajo et al.	**47.23**	5.05	9.65	41.02	3.43	8.41	18.98	34.36	39.76	38.43	24.63
VE + TokAlign	46.39	0.69	10.60	40.75	0.79	7.48	17.86	33.98	**40.27**	36.54	23.54
MATE	46.54	**14.63**	**18.02**	**41.51**	**10.11**	**10.03**	**22.71**	**34.67**	39.10	**38.84**	**27.62**
Target → English											
Baseline	42.98	23.10	6.13	42.61	5.85	11.88	27.35	43.86	34.71	25.06	26.35
VE + Mean Init	38.19	22.17	23.30	38.81	8.91	19.47	**28.79**	36.84	33.96	25.45	27.59
VE + FOCUS	**43.68**	0.39	16.00	**44.09**	7.76	18.26	26.31	43.98	34.82	25.77	26.11
VE + ZeTT	43.54	10.38	2.19	43.23	0.91	7.14	26.95	43.40	32.98	25.56	23.63
VE + Sakajo et al.	43.43	16.77	13.25	43.83	6.74	17.91	27.66	44.08	**35.70**	**26.38**	27.58
VE + TokAlign	37.73	16.24	14.65	40.02	3.51	18.21	27.04	43.60	35.38	23.78	26.02
MATE	42.85	**27.22**	**30.10**	42.93	**12.70**	**22.24**	28.56	**44.42**	34.04	26.32	**31.14**

Table 12: Language-specific translation BLEU scores on FLORES-200. **Bold** indicates the best score per column, and underline indicates the second best.

	id	km	lo	ms	my	ta	th	tl	vi	zh	Avg
English → Target											
Baseline	66.74	17.99	20.24	64.09	12.61	35.59	36.98	58.02	55.51	24.45	39.22
VE + Mean Init	68.13	28.51	31.65	65.13	19.43	**41.46**	36.81	59.26	58.06	**26.78**	43.52
VE + FOCUS	67.04	10.76	10.23	63.60	0.89	35.69	37.71	55.45	57.19	24.76	36.33
VE + ZeTT	67.91	16.58	24.85	64.19	5.60	35.74	38.41	59.13	57.03	25.76	39.52
VE + Sakajo et al.	**68.60**	18.60	30.79	65.24	17.06	37.74	**40.27**	59.38	57.62	26.22	42.15
VE + TokAlign	67.97	22.74	19.89	**65.58**	15.56	32.37	36.66	59.26	**58.15**	25.81	40.40
MATE	68.19	**38.69**	**37.38**	65.42	**26.58**	41.42	40.18	**59.83**	57.09	26.51	**46.13**
Target → English											
Baseline	63.23	42.99	23.90	62.64	15.51	42.62	51.11	62.85	56.91	40.18	46.19
VE + Mean Init	59.82	47.57	47.39	60.06	30.88	40.79	**54.18**	57.77	57.07	51.36	50.69
VE + FOCUS	**66.02**	3.47	38.91	**65.80**	29.14	42.50	52.72	65.09	58.38	51.30	47.33
VE + ZeTT	65.60	35.42	18.38	65.06	13.50	27.13	52.65	64.69	57.49	50.77	45.07
VE + Sakajo et al.	65.52	40.18	29.30	65.43	28.77	42.53	53.34	65.00	**58.69**	**52.59**	50.14
VE + TokAlign	65.91	39.74	26.76	65.48	25.79	29.89	52.71	**65.42**	56.51	50.95	47.92
MATE	65.05	**52.13**	**54.08**	64.86	**37.77**	**46.56**	53.68	65.09	57.73	52.48	**54.94**

Table 13: Language-specific translation chrF++ scores on FLORES-200. **Bold** indicates the best score per column, and underline indicates the second best.

G.2 Vocab Expansion + FOCUS

The VOCAB EXPANSION + FOCUS baseline uses the same expanded tokenizer as VOCAB EXPANSION + MEAN INIT. We first expand both the input embedding matrix and output language-modeling head to the new vocabulary size. Each new row is initialized with the mean of the original-token components produced by the base tokenizer, following the procedure in Appendix G.1. FOCUS initialization is then applied only to new tokens for which the target-language corpus provides sufficient evidence.

For each target language, we tokenize its text

Method	en	th	vi	zh	Avg
Baseline	14.20	0.08	1.34	1.76	4.35
VE + Mean Init	47.56	4.20	7.14	10.50	17.35
VE + FOCUS	44.96	0.25	6.81	3.61	13.91
VE + ZeTT	48.49	2.35	2.02	2.61	13.87
VE + Sakajo et al.	51.34	3.70	12.02	4.29	17.84
VE + TokAlign	47.90	2.69	4.54	3.11	14.56
MATE	**52.10**	**27.98**	**15.88**	**20.76**	**29.18**

Table 14: Language-specific XQuAD exact match scores (%). **Bold** indicates the best score per column, and underline indicates the second best.

Method	en	id	km	lo	ms	my	ta	th	tl	vi	zh	Avg
Baseline	28.67	23.78	21.78	21.33	25.56	23.11	24.67	25.89	24.89	22.44	27.00	24.47
VE + Mean Init	27.56	25.00	23.56	27.00	26.11	24.89	25.33	26.22	26.22	24.67	28.22	25.89
VE + FOCUS	30.11	25.78	23.11	21.33	24.78	23.00	22.33	25.89	25.33	22.44	27.11	24.66
VE + ZeTT	30.44	26.00	26.33	24.89	24.11	26.22	24.67	25.22	**27.67**	23.89	24.56	25.82
VE + Sakajo et al.	29.33	**27.78**	27.11	**28.11**	25.78	24.78	**27.22**	**27.44**	26.22	**27.22**	28.67	27.24
VE + TokAlign	**33.22**	25.78	25.33	25.00	25.56	25.67	24.56	27.33	26.33	24.89	28.44	26.56
MATE	31.44	27.22	**27.44**	27.22	**26.33**	**27.00**	27.00	**27.44**	27.33	26.33	**30.33**	**27.73**

Table 15: Language-specific Belebele reading comprehension accuracy (%). **Bold** indicates the best score per column, and underline indicates the second best.

Method	en	id	ta	vi	zh	Avg
Baseline	32.00	29.36	25.48	28.29	26.97	28.42
VE + Mean Init	32.17	27.36	25.00	27.77	26.23	27.71
VE + FOCUS	31.60	28.87	**27.20**	28.43	26.56	28.53
VE + ZeTT	32.98	27.70	25.02	27.50	25.75	27.79
VE + Sakajo et al.	32.61	27.51	25.21	26.57	25.92	27.56
VE + TokAlign	32.58	29.02	27.11	27.73	26.70	28.63
MATE	**33.36**	**29.45**	25.53	**29.06**	**28.17**	**29.11**

Table 16: Language-specific Multilingual MMLU accuracy (%). **Bold** indicates the best score per column, and underline indicates the second best.

with the expanded tokenizer and train a fastText model (Joulin et al., 2017) on the resulting token sequence. We identify the overlapping set of original vocabulary tokens that appear in the target-language corpus and have valid fastText vectors. For each new token that appears at least 10 times and has not already been initialized by another language, we compute cosine similarities between its fastText vector and the fastText vectors of the overlapping original tokens. These similarities are converted into sparse convex weights using sparsemax (Martins and Astudillo, 2016). The new token row is then initialized as the sparsemax-weighted combination of the corresponding original-token embedding rows, and the same initialized vector is written to both the input embedding matrix and the output language-modeling head. New tokens that are not categorized as any language are initialized using the mean-initialization procedure similar to *Vocab Expansion + Mean Init*.

G.3 Vocab Expansion + ZeTT

The VOCAB EXPANSION + ZETT baseline also uses the same expanded tokenizer as VOCAB EXPANSION + MEAN INIT, but replaces the final initialization step with predictions from a ZeTT hypernetwork (Minixhofer et al., 2024). The ZeTT hypernetwork is trained to transfer token embeddings

Algorithm 2 VOCAB EXPANSION Tokenizer Expansion Procedure.

Require: Base Model Params P_{base}, Target Model Params P_{target}, Embedding Dim d
Require: Base Tokenizer $\mathcal{T}_{base}$, Target Corpus $\mathcal{D}_{target}$
Ensure: Expanded Tokenizer $\mathcal{T}_{expanded}$

$V_{base} \leftarrow \text{VOCABSIZE}(\mathcal{T}_{base})$
$X \leftarrow \lfloor \frac{P_{target} - P_{base}}{2 \times d} \rfloor$ ▷ Step 1: Calculate target token addition
▷ Step 2: Train temporary BPE tokenizer
$\mathcal{T}_{temp} \leftarrow \text{TRAINBPE}(\mathcal{D}_{target}, \text{vocab\_size} = V_{base} + X)$
$S_{new} \leftarrow \emptyset$
$\mathcal{M}_{temp} \leftarrow \text{GETMERGESSORTED}(\mathcal{T}_{temp})$
▷ Step 3: Filter and append novel merges
for m **in** $\mathcal{M}_{temp}$ **do**
  if $m \notin \mathcal{T}_{base}$ **and** $|S_{new}| < X$ **then**
    $S_{new} \leftarrow S_{new} \cup \{m\}$
  end if
end for
$\mathcal{T}_{expanded} \leftarrow \text{ADDMERGES}(\mathcal{T}_{base}, S_{new})$
return $\mathcal{T}_{expanded}$

across tokenizers: given a token's byte-level surface form, language identifier, and the source model's existing embedding table, it learns to predict the corresponding input and output embedding rows for a target tokenizer.

At inference time, we expand the OLMo checkpoint to the target embedding size, load the trained ZeTT hypernetwork, and concatenate the original input and output embedding tables to form the source embedding representation used by the hypernetwork. Newly added token IDs are grouped by language using the token-language assignment file generated by VOCAB EXPANSION + FOCUS. For each language group, the new token strings are encoded with the byte-level hypernetwork tokenizer into fixed-length surface-form matrices. The hypernetwork then predicts input and output embedding rows in batches, conditioned on the language ID and source embedding representation, and assigns the predicted rows to the corresponding new token IDs. New tokens that are not categorized as any language are initialized using the mean-initialization procedure similar to *Vocab Expansion + Mean Init*.

G.4 Vocab Expansion + Sakajo et al.

The VOCAB EXPANSION + SAKAJO ET AL. baseline follows the dictionary-based tokenizer transfer procedure of Sakajo et al. (2025). We use the word-to-word mapping $\mathcal{B}$ produced by the Word Mapping stage as bilingual dictionary entries, where each target-language word is paired with its English definitions. A BPE tokenizer is trained from the dictionary entries using a byte-level pre-tokenizer. We then merge this tokenizer into the original OLMo tokenizer using the same ID-preserving merge procedure as VOCAB EXPANSION + MEAN INIT in Appendix G.1, yielding a merged tokenizer whose original-token behavior is unchanged.

To initialize the expanded model, we run the dictionary-transfer alignment procedure between the source tokenizer and the target tokenizer. The alignment procedure constructs token-level mappings from each target-tokenizer entry to one or more source-tokenizer entries, with probabilities estimated from the bilingual dictionary alignment. We then reindex this mapping to the merged tokenizer: original source tokens keep identity mappings, while newly appended target tokens use their aligned source-token distributions. After expanding the input embedding matrix and output language-modeling head to the padded embedding size, each new token is initialized from its aligned source tokens. If a new token maps to a single source token, its input and output rows are copied from that source token. If it maps to multiple source tokens, its rows are initialized as the probability-weighted average of the aligned source-token rows. Tokens without a valid dictionary alignment are initialized with mean initialization.

G.5 Vocab Expansion + TokAlign

The VOCAB EXPANSION + TOKALIGN baseline utilizes the same vocabulary expansion procedure (Algorithm 2) to determine the new token set. However, it replaces mean initialization with a weight initialization strategy proposed by TokAlign (Li et al., 2025). We reproduce their weight initialization pipeline using the following steps:

1. **Dual-Vocabulary Tokenization:** We tokenize a large reference corpus twice—comprising a mixture of CulturaX (Nguyen et al., 2024a) (40%), The Stack (Kocetkov et al., 2023) (30%), and Proof-Pile-2 (Azerbayev et al., 2024) (30%)—following their paper. The first pass uses the original source tokenizer, and the second uses the expanded target tokenizer. This generates parallel token sequences (S_{src} and S_{tgt}) of the same content.

2. **Global Co-occurrence Learning:** We train static GloVe embeddings on both sequence sets. This step captures global token-token co-occurrence statistics to establish a semantic geometry for both vocabularies.

3. **Semantic Mapping:** For every newly added token t_{new}, we identify its closest token t_{old} in the base vocabulary using cosine similarity:

$$\text{Anchor}(t_{\text{new}}) = \arg\max_{t_{\text{old}} \in V_{\text{src}}} \cos(\boldsymbol{v}_{t_{\text{new}}}, \boldsymbol{v}_{t_{\text{old}}})$$

This creates a list of mappings from each new token to an existing anchor token.

4. **Weight Transfer:** Finally, the input embeddings and output weights of t_{new} are explicitly copied from its identified anchor token. This provides a semantically relevant "warm start" for the newly expanded parameters prior to fine-tuning.

H Training Hyperparameters

Hyperparameter	
Global batch size	512
Micro batch size (per device)	4
Learning rate	2.0×10^{-4}
Weight decay	0.1
Optimizer	AdamW
(β_1, β_2)	(0.9, 0.95)
Gradient clip	1.0
Max sequence length	4096

Table 17: Training hyperparameters for the 1B model.

I Performance Gains across Resource Levels

To analyze when MATE is most beneficial, we plot the performance gain from extending the vocabulary of each language using MATE (i.e., $score(\text{MATE}) - score(\text{Baseline})$) against the number of training sentences for that language (Figure 2), and fit an exponential decay model $y = ae^{-bx}$ by minimizing the sum of squared residuals. The exponential decay model explains most of the variance in translation improvements, with $R^2 = 0.84$ for E→X and $R^2 = 0.83$ for X→E. For Belebele, the fit is moderate ($R^2 = 0.53$), indicating that the relationship between data size and

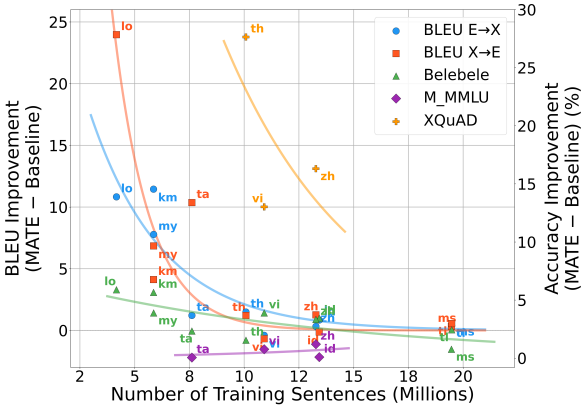

Figure 2: MATE's improvement over the baseline as a function of the number of training sentences per language. The lines show the line of best fit for each downstream task.

improvement is less pronounced for reading comprehension. M_MMLU does not exhibit a clear decay trend; however, the sample size is much smaller (4 languages) than for the other tasks, making it difficult to establish a reliable relationship.

Overall, these results show that while MATE frequently improves performance on high-resource languages (e.g., a 3.3% accuracy gain on Chinese reading comprehension and a 1.26 BLEU improvement on Chinese→English translation), its impact becomes dramatically larger for low-resource languages. For example, MATE improves Lao-to-English translation by 23.97 BLEU over the baseline. These findings demonstrate that MATE is particularly effective at unlocking the latent multilingual capabilities of LLMs in low-resource settings, where traditional approaches struggle the most.

J Detailed Evaluation Results for Embedding Alignment Ratio Sensitivity

In this section, we provide the full task-level results for the embedding alignment ratio sensitivity study summarized in Table 5. Table 18 reports translation BLEU scores, Table 19 reports XQuAD exact match scores, Table 20 reports Belebele accuracy, and Table 21 reports Multilingual MMLU accuracy.

K Detailed Evaluation Results for Ablation Study

In this section, we provide the full, language-specific breakdown of the ablation study summarized in the main text. Table 22 details the translation performance (BLEU) for both E→X and X→E directions across all evaluated languages. Ta-

	id	km	lo	ms	my	ta	th	tl	vi	zh	Avg
English → Target											
0%	46.23	15.36	17.14	41.22	11.46	11.32	23.74	34.66	39.34	40.22	28.07
12.5%	47.35	16.40	18.20	41.69	11.24	11.35	23.20	34.96	39.43	40.21	28.40
43.75%	46.47	15.84	16.49	41.82	11.53	11.16	23.64	34.72	39.56	39.88	28.11
87.5%	46.54	14.63	18.02	41.51	10.11	10.03	22.71	34.67	39.10	38.84	27.62
93.75%	46.74	13.50	13.75	41.75	10.04	9.41	22.64	34.66	39.28	38.20	27.00
Target → English											
0%	44.57	26.58	31.09	44.59	14.19	24.03	7.55	45.24	35.83	26.65	30.03
12.5%	44.48	28.16	30.88	43.86	16.36	23.30	29.72	45.12	35.65	26.61	32.41
43.75%	44.04	26.75	30.97	44.23	14.49	23.13	28.78	45.40	33.37	26.26	31.74
87.5%	42.85	27.22	30.10	42.93	12.70	22.24	28.56	44.42	34.04	26.32	31.14
93.75%	43.45	25.84	29.95	42.86	14.94	21.68	28.31	35.60	31.80	25.64	30.01

Table 18: Language-specific translation BLEU scores for embedding alignment ratio sensitivity.

Ratio	en	th	vi	zh	Avg
0%	52.02	20.42	15.13	20.67	27.06
12.5%	51.43	14.37	11.68	14.79	23.07
43.75%	50.76	22.02	13.03	21.76	26.89
87.5%	52.10	27.98	15.88	20.76	29.18
93.75%	47.82	20.76	9.08	18.07	23.93

Table 19: Language-specific XQuAD exact match scores (%) for embedding alignment ratio sensitivity.

Ratio	en	id	km	lo	ms	my	ta	th	tl	vi	zh	Avg
0%	31.67	25.33	27.33	24.44	26.44	24.67	28.67	29.11	28.44	25.89	29.33	27.39
12.5%	31.33	26.78	26.44	26.11	25.33	27.44	28.44	29.56	26.67	29.00	30.67	27.98
43.75%	31.00	23.67	25.11	23.00	24.33	22.89	26.33	27.44	25.33	26.00	28.67	25.80
87.5%	31.44	27.22	27.44	27.22	26.33	27.00	27.00	27.44	27.33	26.33	30.33	27.73
93.75%	28.00	24.67	25.44	23.22	26.33	25.56	25.78	26.44	26.11	24.67	28.44	25.88

Table 20: Language-specific Belebele accuracy (%) for embedding alignment ratio sensitivity.

Ratio	en	id	ta	vi	zh	Avg
0%	33.56	28.30	24.98	27.88	27.24	28.39
12.5%	33.10	28.45	25.98	27.80	27.90	28.65
43.75%	33.10	28.75	24.53	28.14	27.35	28.37
87.5%	33.36	29.45	25.53	29.06	28.17	29.11
93.75%	33.07	29.52	26.34	28.21	27.37	28.90

Table 21: Language-specific Multilingual MMLU accuracy (%) for embedding alignment ratio sensitivity.

ble 23 reports XQuAD exact match scores, Table 24 presents the language-specific accuracy scores on the Belebele reading comprehension benchmark, and Table 25 provides the breakdown for the Multilingual MMLU benchmark.

L Detailed Evaluation Results for Scalability to Larger Models

In this section, we provide the full, language-specific breakdown comparing the baseline OLMo 2 7B model, VE + SAKAJO ET AL., and our pro-

	id	km	lo	ms	my	ta	th	tl	vi	zh	Avg
English → Target											
Baseline	45.69	3.19	6.65	40.99	2.33	8.81	21.22	34.31	39.50	38.51	24.12
+PVE	46.38	9.62	14.84	41.16	10.10	9.34	22.91	34.29	**39.61**	39.19	26.74
+PVE+SEI	46.23	**15.36**	17.14	41.22	**11.46**	**11.32**	**23.74**	34.66	39.34	**40.22**	**28.07**
MATE	**46.54**	14.63	**18.02**	**41.51**	10.11	10.03	22.71	**34.67**	39.10	38.84	27.62
Target → English											
Baseline	42.98	23.10	6.13	42.61	5.85	11.88	27.35	43.86	34.71	25.06	26.35
+PVE	43.72	26.38	12.21	43.64	13.13	22.35	19.22	44.39	19.29	26.13	27.05
+PVE+SEI	**44.57**	26.58	**31.09**	**44.59**	**14.19**	**24.03**	7.55	**45.24**	**35.83**	**26.65**	30.03
MATE	42.85	**27.22**	30.10	42.93	12.70	22.24	**28.56**	44.42	34.04	26.32	**31.14**

Table 22: Language-specific translation BLEU scores (FLORES-200) for the ablation study. **Bold** indicates the best score per column, underline indicates the second best.

	en	th	vi	zh	Avg
Baseline	14.20	0.08	1.34	1.76	4.35
+PVE	42.94	1.09	5.80	3.87	13.43
+PVE+SEI	52.02	20.42	15.13	20.67	27.06
MATE	**52.10**	**27.98**	**15.88**	**20.76**	**29.18**

Table 23: Language-specific XQuAD exact match scores (%) for the ablation study. **Bold** indicates the best score per column, underline indicates the second best.

	en	id	km	lo	ms	my	ta	th	tl	vi	zh	Avg
Baseline	28.67	23.78	21.78	21.33	25.56	23.11	24.67	25.89	24.89	22.44	27.00	24.47
+PVE	29.67	24.33	23.56	24.78	25.89	23.78	26.56	25.78	25.22	**29.56**	27.78	26.08
+PVE+SEI	**31.67**	25.33	27.33	24.44	**26.44**	24.67	**28.67**	**29.11**	**28.44**	25.89	29.33	27.39
MATE	31.44	**27.22**	**27.44**	**27.22**	26.33	**27.00**	27.00	27.44	27.33	26.33	**30.33**	**27.73**

Table 24: Language-specific Belebele reading comprehension accuracy (%) for the ablation study. **Bold** indicates the best score per column, underline indicates the second best.

	en	id	ta	vi	zh	Avg
Baseline	32.00	29.36	25.48	28.29	26.97	28.42
+PVE	32.18	28.28	24.79	26.99	26.94	27.84
+PVE+SEI	**33.56**	28.30	24.98	27.88	27.24	28.39
MATE	33.36	**29.45**	**25.53**	**29.06**	**28.17**	**29.11**

Table 25: Language-specific M_MMLU accuracy (%) for the ablation study. **Bold** indicates the best score per column, underline indicates the second best.

posed method (MATE) at the 7B scale. Table 26 details the translation performance (BLEU) for both English → Target and Target → English directions. Table 27 reports exact match on XQuAD, Table 28 presents the language-specific accuracy scores on Belebele, and Table 29 provides the breakdown for Multilingual MMLU.

Model	id	km	lo	ms	my	ta	th	tl	vi	zh	Avg
English → Target											
Baseline	47.84	17.06	16.67	42.95	7.58	13.80	26.31	36.63	42.19	42.01	29.30
VE + Sakajo et al.	**49.28**	16.12	17.06	**43.41**	12.02	**14.43**	25.34	N/A	**42.79**	**42.47**	29.21
MATE	48.54	**17.85**	**22.01**	43.14	**13.77**	12.05	**27.25**	**36.75**	42.47	41.75	**30.56**
Target → English											
Baseline	**48.32**	31.93	32.93	48.20	18.95	**29.54**	33.23	50.17	**39.56**	**30.58**	36.34
VE + Sakajo et al.	47.99	29.07	30.64	48.44	17.99	28.12	32.86	**50.48**	36.58	29.62	35.18
MATE	48.04	**32.32**	**35.29**	**48.50**	**20.83**	27.54	**34.11**	50.02	39.33	29.79	**36.58**

Table 26: Translation performance (BLEU) on FLORES-200 for OLMo 2 7B. **Bold** indicates the best score per column.

Model	en	th	vi	zh	Avg
Baseline	**72.18**	4.71	**24.45**	13.53	28.72
VE + Sakajo et al.	60.92	5.63	16.47	9.16	23.05
MATE	63.36	**37.06**	18.57	**26.05**	**36.26**

Table 27: Exact match (%) on XQuAD for OLMo 2 7B. **Bold** indicates the best score per column.

Model	en	id	km	lo	ms	my	ta	th	tl	vi	zh	Avg
Baseline	66.56	**50.67**	27.67	28.44	**52.33**	25.56	32.67	37.00	47.00	**46.11**	**56.44**	42.77
VE + Sakajo et al.	61.78	45.44	27.56	28.44	45.11	25.11	31.33	34.67	40.78	38.78	52.44	39.22
MATE	**70.33**	50.56	**35.22**	**33.11**	50.78	**31.22**	**38.44**	**45.56**	**49.22**	45.78	55.56	**45.98**

Table 28: Commonsense reasoning accuracy (%) on Belebele for OLMo 2 7B. **Bold** indicates the best score per column.

Model	en	id	ta	vi	zh	Avg
Baseline	**56.30**	**41.71**	29.17	36.89	37.87	**40.39**
VE + Sakajo et al.	50.36	39.08	27.93	31.57	34.80	36.75
MATE	53.28	41.20	**30.37**	**37.00**	**39.23**	40.22

Table 29: General reasoning accuracy (%) on Multilingual MMLU for OLMo 2 7B. **Bold** indicates the best score per column.