**General Paper** Peer-Reviewed

# MITRE: Efficient Pre-trained Models for Multilingual Neural Machine Translation with Registering

Zhi Qu[a], Yiran Wang[b], Jiannan Mao[b],
Jin Tei[b], Hideki Tanaka[b], Masao Utiyama[b] and Taro Watanabe[a]

Multilingual neural machine translation (MNMT) aims to support arbitrary translations across multiple languages. MNMT has recently seen dramatic improvements with large language models (LLMs), yet LLMs often require substantial computational resources and may be insufficient for MNMT in terms of quality. In this work, we present MITRE (**m**ulti**i**lingual **t**ranslation with **re**gisters), a series of pre-trained MNMT-specific models trained on 9.3 billion sentence pairs across 24 languages collected from public corpora to compete with commercial LLMs. Built on the decoder-only architecture, MITRE integrates a novel mechanism called registering, which inserts a sequence of artificial tokens, namely registers, between source and target tokens and modifies the attention mask such that generation pays attention exclusively to the activated registers. Through experiments on EC-40, a large-scale training set that enables fair methodological comparison, we first demonstrate that registering advances the state-of-the-art in MNMT-specific methods. Second, we show that one of our models, MITRE-913M, outperforms NLLB-3.3B in most cases, and achieves performance close to GPT-4o mini with less than 1 billion parameters measured by spBLEU, chrF++, and COMET. Third, fine-tuning experiments in various scenarios show that our models have strong fine-tuning adaptability compared to NLLB series. Finally, based on analysis, we show that registers reflect the semantics of corresponding source tokens in the target language space.

**Key Words**: *Machine Translation, Multilingualism, Zero-shot Translation*

## 1 Introduction

Multilingual neural machine translation (MNMT) aims for arbitrary translation between multiple languages. In the early development of MNMT, a method proposed by Johnson et al. (2017)

[a] Nara Institute of Science and Technology
[b] National Institute of Information and Communications Technology
This paper is an extended version of our preliminary paper (Qu et al. 2025) presented in the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL 2025 main) as an oral presentation. Compared to the original paper, the main topic has been shifted from the proposed methodology to our pre-trained models with additional experimental results and analyses in Sections 4, 5, and 6.2.
Our codes and models are available at `https://github.com/zhiqu22/mitre`.

enabled the training of a single MNMT-specific model using parallel data only, in which a language tag is appended as a translation instruction to the input sequence. Such models (Zhang et al. 2020; Fan et al. 2021; NLLB Team 2022) demonstrate the ability to translate directions that are not seen during training, known as zero-shot translation, which is helpful in addressing data scarcity in certain translation directions and facilitates the scaling of MNMT systems to hundreds of languages. However, this paradigm has recently faced a new challenge: large language models (LLMs) have demonstrated impressive translation performance, often outperforming state-of-the-art MNMT-specific models in several translation directions (Zhu et al. 2024), although LLMs require substantial computational resources. Meanwhile, MNMT-specific models suffer from a trade-off among three competing goals: computational costs, broad language coverage, and high translation quality. A key challenge is the off-target problem (Chen et al. 2023; Tan and Monz 2023), where generated translations fail to match the intended target language. This issue becomes increasingly severe as the number of supported languages grows under a limited model scale, posing a substantial challenge to translation quality.

We present MITRE (**m**ulti**i**lingual **t**ranslation with **re**gisters) to reconcile computational costs, broad language coverage, and high translation quality. MITRE is a series of pre-trained MNMT-specific models trained on 9.3 billion sentence pairs across 24 languages collected from public corpora. MITRE adopts a decoder-only (Dec-only) architecture and incorporates a novel method, i.e., *registering*, which effectively mitigates the off-target problem without introducing additional parameters. Specifically, we implement registering by inserting a sequence of artificial tokens, called *registers*, between the source and target tokens, inspired by Rios et al. (2020) and Chen et al. (2023), which demonstrate that constraining generation within the target language space is crucial for reducing the off-target problem. These registers are aligned one-to-one with the source tokens, carry no actual semantics, and serve only to encode the identity of the target language. We then modify the attention mask such that the generation of target tokens depends solely on registers, rather than directly on source tokens as in standard encoder–decoder or decoder-only architectures (Figure 1). Based on this design, each register is expected to capture the semantics of its aligned source token and represent it within the target language space, thereby imposing a strong constraint that anchors generation exclusively in the target language. By incorporating registering, we pre-train two MITRE models, i.e., MITRE-466M and MITRE-913M, comprising 466M and 913M parameters, respectively.

We conduct experiments by following prior studies (Tan and Monz 2023; Bu et al. 2024; Wu et al. 2024; Galiano-Jiménez et al. 2025; Zou et al. 2025) to evaluate results using four automatic metrics: spBLEU (NLLB Team 2022) and chrF++ (Popović 2015, 2017) for surface similar-

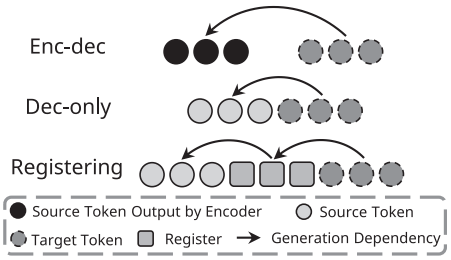



**Figure 1** Overview of our proposed method. This visualizes the generation dependencies across different architectures, where each *Token* refers to the representation corresponding to the token.

ity, COMET (Rei et al. 2020) for semantic similarity, and off-target ratio (Zhang et al. 2020) for target language faithfulness. First, we compare registering to other related MNMT-specific methodologies using EC-40 (Tan and Monz 2023), a large-scale English-centric training set, i.e., translating into/from English, designed to assess zero-shot translation capabilities. Experimental results show that, compared to related MNMT-specific methodologies, our method improves spBLEU, chrF++, and COMET scores by up to 4.06, 9.34, and 7.47 points, respectively, on average across 1,640 translation directions while using fewer parameters, and reduces the off-target ratio from 26.69% to 3.65%. Second, we compare our pre-trained MITRE models against a wide range of competitive systems, including M2M series (Fan et al. 2021), NLLB series (NLLB Team 2022), and GPT series (Brown et al. 2020; OpenAI 2024) across 24 languages. Results indicate that MITRE-913M achieves significantly higher spBLEU and chrF++ scores than NLLB-3.3B, the SOTA open-sourced model specific to MNMT, and attains comparable COMET scores. Compared with GPT-4o mini (OpenAI 2024), MITRE-913M yields lower COMET scores but remains competitive in spBLEU and performs significantly better in chrF++. Third, we fine-tune the pre-trained MITRE models using both full-parameter updates and parameter-efficient tuning via LoRA (Hu et al. 2022) in three distinct scenarios, demonstrating the superior fine-tuning adaptability of MITRE compared to NLLB series. Finally, by analyzing the attention mechanisms and the representation distribution in translation instances at the token level, we confirm that the register mirrors the corresponding source token in the target language space.

## 2 Related Work

Johnson et al. (2017) proposed adding a language tag as a translation instruction at the beginning of source tokens, marking the beginning of MNMT-specific models. Subsequently, it became

the primary development objective for multilingual models that simultaneously achieve high translation quality, broad language coverage, and computational efficiency. Early work by Zhang et al. (2020) supported around 100 languages but suffered from a severe off-target problem due to English-centric training data, despite using back translation (Edunov et al. 2018) as data augmentation. Then, based on the goal of maximizing the cross-lingual generalization capability, i.e., the ability to make low-resource languages benefit from high-resource languages, Fan et al. (2021) trained the M2M series using many non-English translation directions, explicitly moving away from the English-centric design. More recently, NLLB Team (2022) introduced the NLLB series, the current state-of-the-art MNMT-specific models supporting more than 200 languages, which benefited from significantly expanded multilingual training data, extensive data augmentation, e.g., back translation and distillation (Hinton et al. 2015), and increased model capacity.

The training of the pre-trained MNMT models discussed above relies solely on the vanilla Transformer architecture (Vaswani et al. 2017), without incorporating any specialized methods, and instead leverages data augmentation. This is largely because previous methods proposed to improve MNMT often incur high computational complexity, making it infeasible to enhance cross-lingual generalization capacity under reasonable computational budgets. Specifically, early studies attempted to isolate generation across languages through language-specific dictionaries or language-specific components (Rios et al. 2020; Qu and Watanabe 2022) to reduce the off-target problem, but these *explicit methods* that modified the modeling process introduced significant computational costs and hindered cross-lingual knowledge sharing (Zhang et al. 2021). As a compromise, Chen et al. (2023) proposed augmenting shared vocabularies with language-specific subsets. In parallel, some studies have introduced *implicit methods* to improve zero-shot translation by optimizing internal representations, for example, aligning semantic spaces across languages (Pan et al. 2021; Bu et al. 2024) or enhancing translation instructions toward the target language (Stap et al. 2023; Qu et al. 2025; Sun et al. 2024), although these methods typically incur additional training costs. In addition, to evaluate the generalization ability brought by an MNMT methodology, the improvement of zero-shot translation capacity in an English-centric training set has become the main indicator (Aharoni et al. 2019; Arivazhagan et al. 2019a; Gu et al. 2019; Zhang et al. 2021; Stap et al. 2023; Tan and Monz 2023).

Recently, commercial LLMs, such as the GPT series (Brown et al. 2020; OpenAI 2024), have demonstrated impressive multilingual translation capabilities (Zhu et al. 2024), but such closed-source models are limited by high computational costs and their non-extensibility, e.g., extending to new languages or fine-grained fine-tuning. Consequently, fine-tuning open-source LLMs into MNMT-specific models has emerged as an attractive research direction, though current progress

is limited. Yang et al. (2023) attempted to fine-tune an LLM to an MNMT-specific model that supports more than 100 languages, but its performance still lagged behind commercial LLMs and state-of-the-art MNMT-specific models. Subsequent works (Xu et al. 2024; Alves et al. 2024) achieved higher translation quality but at the expense of language coverage, supporting only 5 and 10 languages, respectively. Although Xu et al. (2025) further expanded coverage to 50 languages, their approach did not support arbitrary translation among these languages.

## 3 Multilingual Translation With Registers

We propose *registering* that combines the advantages of the explicit and implicit strategies, i.e., altering model structure and optimizing language representation, respectively, by separating generation-related representations per language without introducing additional parameters, thus incurring only modest computational overhead.

### 3.1 Multilingual Neural Machine Translation

Formally, given a multilingual corpus $\mathbb{C}$ spanning multiple translation directions, each instance in $\mathbb{C}$ is defined as $(\boldsymbol{x}^u, \boldsymbol{y}^v)$, comprising a sequence of source tokens $\boldsymbol{x}^u = x_1^u, \ldots, x_I^u$ in language $u$ and a sequence of target tokens $\boldsymbol{y}^v = y_1^v, \ldots, y_J^v$ in language $v$. In addition, we follow Sutskever et al. (2014) to shift the tag indicating the end of a sequence, i.e., [eos], to the beginning of the target sequence as a trigger of generation instead of introducing another tag, such as [bos] indicating the beginning of a sequence. We introduce a set of language tags $\mathbb{L} = \{l_1, \ldots, l_K\}$, where language tags are artificial tokens individually corresponding to $K$ languages in $\mathbb{C}$. Following Wu et al. (2021), we add a tag indicating the language $v$ at the beginning of $\boldsymbol{x}^u$ as the translation instruction for multilingual neural machine translation (MNMT), denoted by $l^v$. Consequently, the input fed into the model specific to MNMT becomes $\boldsymbol{x'} = l^v, x_1^u, \ldots, x_I^u$, and we define a transformed corpus $\mathbb{C}' = \{(\boldsymbol{x'}, \boldsymbol{y}^v) \mid (\boldsymbol{x}^u, \boldsymbol{y}^v) \in \mathbb{C}\}$. We train the model over all instances of $\mathbb{C}'$ by optimizing the following cross entropy loss:

$$\mathcal{L}_{\text{ce}} = -\sum_{\boldsymbol{x'}, \boldsymbol{y}^v \in \mathbb{C}'} \sum_{j=1}^{J} \log\ p(y_j^v \mid \boldsymbol{x'}, \boldsymbol{y}_{<j}^v), \tag{1}$$

where $p(y_j^v \mid \boldsymbol{x'}, \boldsymbol{y}_{<j}^v)$ represents the probability distribution for generating $y_j^v$ by the MNMT-specific model. The current state-of-the-art models (Fan et al. 2021; NLLB Team 2022) utilize the encoder-decoder architecture (Enc-dec), where the generation of $y_j$ can be expressed as:

$$y_j^v = \text{decoder}(\text{encoder}(\boldsymbol{x'}), \boldsymbol{y}_{<j}^v). \tag{2}$$

Alternatively, Gao et al. (2022) and Zhang et al. (2022) show that MNMT-specific models can be implemented with a decoder-only architecture (Dec-only). We follow Gao et al. (2022) to train a Dec-only MNMT-specific model with Equation (1) rather than the language modeling loss (Radford et al. 2018). Thus, the generation is described as:

$$y_j^v = \text{decoder}(\boldsymbol{x}', \boldsymbol{y}_{<j}^v). \tag{3}$$

## 3.2 Registering

Gu et al. (2019) and Qu et al. (2024) suggest that the language tag $l^v$ is a key factor influencing the off-target problem. Specifically, although $l^v$ is a translation instruction distinctly representing the target language, translation generation may not strictly depend on $l^v$ due to the spread of attention over source tokens other than $l^v$ in the attention mechanism. In this work, we propose *registering* to address this problem. Specifically, we create registers by duplicating the language tag of the target language, where the length of registers equals the length of source tokens. By modifying the attention mechanism, the semantics of all source tokens are implicitly projected into the corresponding target language space through explicit registers. As a result, the risk of off-target problems is minimized because the generation of target tokens depends solely on these registers, namely, the generation is constrained within the target language space.

Although previous MNMT models are primarily implemented with the Enc-dec architecture (Fan et al. 2021; NLLB Team 2022; Cao et al. 2024; Bu et al. 2024), we begin with the Dec-only architecture, as Dec-only is more parameter-efficient than Enc-dec and potentially benefits the knowledge transfer (Wang et al. 2022). Formally, given a Dec-only model and an instance $(\boldsymbol{x}', \boldsymbol{y}^v)$, we begin by creating a sequence of artificial tokens, i.e., *registers*. Each register $r$ is a copy of $l^v$ with one-to-one correspondence to each source token, and the sequence of registers is denoted by $\boldsymbol{r}^v = r_1^v, \ldots, r_{I+1}^v$, where $I+1$ refers to the length of $\boldsymbol{x}'$. We then insert registers, i.e., $\boldsymbol{r}^v$, into the sequence between $\boldsymbol{x}'$ and $\boldsymbol{y}^v$, thus reformulating the generation process of Equation (3) as:

$$y_j^v = \text{decoder}(\boldsymbol{x}', \boldsymbol{r}^v, \boldsymbol{y}_{<j}^v). \tag{4}$$

The key step of registering is to modify the Transformer attention mask (Vaswani et al. 2017) to remove the direct dependence of target tokens on source tokens and preserve the reliance on registers. As shown in Figure 2, our attention mask is based on the prefix Dec-only scheme (Dong et al. 2019), where the attention scores for the source tokens are computed with respect to other source tokens bidirectionally, while the target tokens rely only on attention for the

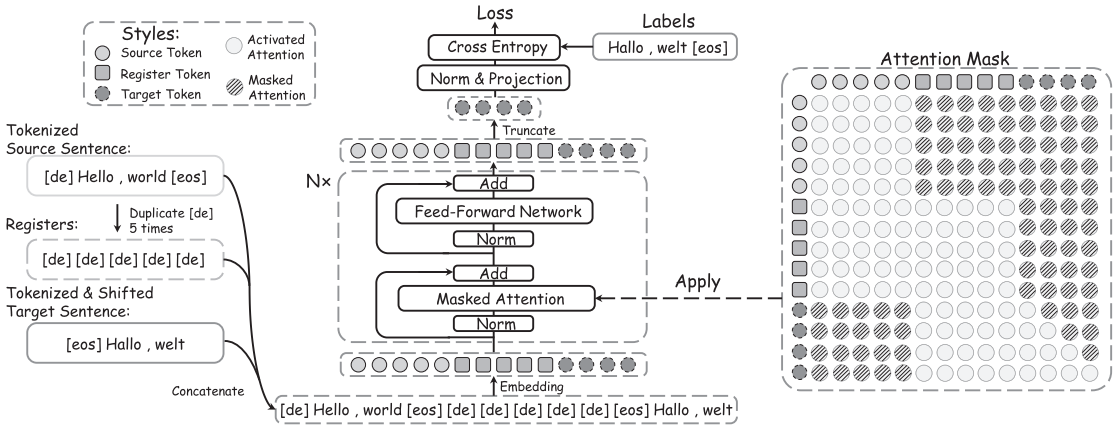



**Figure 2** Illustration of registering, showing an example of translation from English to German. The model stacks $N$ layers with each layer following the Transformer decoder layer structure with pre-normalization. Each circle represents a token and its representation in the generation.

previous ones. We then adjust the mask to control token-level representation according to the following rules: (1) $\boldsymbol{r}^v$ pays attention to $\boldsymbol{x}'$; (2) $\boldsymbol{r}^v$ computes attention bidirectionally for $\boldsymbol{r}^v$; (3) $y_j^v$ pays attention to $\boldsymbol{r}^v$ and $\boldsymbol{y}_{<j}^v$. With this design, the generation of $\boldsymbol{y}^v$ can solely rely on the representation of $\boldsymbol{r}^v$, where the representation of $r_i^v$ not only acts as a representational container of the target language but also carries the semantics of $x_i^u$. In addition, although $\boldsymbol{r}^v$ serves as transferring semantics of $x_i^u$ into the target language representational space, we let the source tokens $\boldsymbol{x}'$ follow Equation (3), because $\boldsymbol{r}^v$ and $\boldsymbol{y}_{<j}^v$ are masked from the attention mechanism of $\boldsymbol{x}'$. As a result, the semantics in the source language are explicitly transferred into the target language space, and the generation is purely dependent on the register, thereby addressing the off-target problem.

## 4 Experiment: MNMT-specific Methodologies

To validate the advantages of our proposed method over existing MNMT-specific methodologies under controlled experimental conditions, we train a set of models that incorporate different but related methods on the same dataset. This experimental setup enables a fair comparison across architectural choices, including Enc-dec and Dec-only models, and allows us to assess the scalability of our method via experiments at two different model scales.

## 4.1 Dataset

We conduct experiments on a large-scale zero-shot translation benchmark proposed by Tan and Monz (2023) to evaluate the effectiveness and scalability of *Registering*. We use EC-40 as the training data and Flores[1] as the test data.[2]

EC-40 is an English-centric parallel corpus in which each non-English language is paired exclusively with English. It contains approximately 120 million sentence pairs spanning 41 languages from five language families: Germanic, Romance, Slavic, Indo-Aryan, and Afro-Asiatic.[3] For each family (excluding English), eight languages are evenly distributed across four resource tiers, i.e., High, Medium, Low, and Extra Low, corresponding to 5M, 1M, 100K, and 50K sentence pairs, respectively.

Flores is a high-quality multilingual evaluation benchmark covering over 200 languages. For each language, Flores provides two splits, *dev* and *devtest*, consisting of 997 and 1,012 sentences, respectively. All sentences are semantically parallel across languages, enabling flexible construction of evaluation directions.

In our experimental setting, models trained on EC-40 support 80 supervised translation directions and 1,560 zero-shot translation directions, totaling 1,640 translation directions. We extract and recombine the corresponding language pairs from Flores for validation and testing. Following the standard process suggested from NLLB Team (2022), we use the *dev* split for validation and the *devtest* split for testing, where validation is performed only on supervised directions, while testing covers all possible directions.

## 4.2 Automatic Evaluation Metric

We evaluate translation performance using four automatic metrics, which are categorized into three complementary aspects of translation quality: (1) *Surface similarity*, measured by spBLEU (Papineni et al. 2002; Goyal et al. 2021) and chrF++ (Popović 2015, 2017); (2) *Semantic similarity*, assessed by COMET (Rei et al. 2020, 2022); and (3) *Target-language faithfulness*, quantified by the off-target ratio (Zhang et al. 2020). Since no single automatic metric can fully substitute human evaluation (Proietti et al. 2025), a comprehensive evaluation by spBLEU, chrF++, and COMET together has become the standard recipe in recent MNMT studies (Tan and Monz 2023; Bu et al. 2024; Wu et al. 2024; Galiano-Jiménez et al. 2025; Zou et al. 2025), as these metrics capture complementary aspects of translation quality.

[1] `https://github.com/openlanguagedata/flores`

[2] As noted by Tan and Monz (2023), there is no overlap between EC-40 and Flores; we additionally verified this property prior to our experiments.

[3] Details of EC-40 are provided in Appendix F.

**spBLEU** spBLEU (Goyal et al. 2021) is a variant of BLEU (Papineni et al. 2002; Post 2018). It unifies tokenization across languages through a shared open-source tokenizer.[4] We adopt spBLEU following the official setup used in Flores (Goyal et al. 2021) because it alleviates inconsistencies in BLEU scores across different writing systems. By standardizing tokenization, spBLEU ensures fairer comparisons across diverse scripts.

**chrF++** chrF++ (Popović 2015, 2017) measures character-level n-gram overlap between hypotheses and references, balancing precision and recall. Following NLLB Team (2022), we report chrF++ together with spBLEU since it complements lexical-level metrics by being less sensitive to tokenization artifacts and better capturing orthographic similarity across writing systems in MNMT (Mullov et al. 2024).

**COMET** COMET (Rei et al. 2020) evaluates translation quality at a representation level by comparing the generated translation, reference, and source sentence. All COMET scores are computed using the official *Unbabel/wmt22-comet-da* model (Rei et al. 2022). We include COMET because it captures semantic similarity beyond surface information and correlates more strongly with human judgments. However, due to uneven training data across languages, COMET may exhibit inconsistent reliability for low-resource languages. Additionally, given that not all languages are supported by COMET, we report COMET scores only for the supported languages listed in Appendix A.

**Off-target Ratio** The off-target ratio (Zhang et al. 2020) quantifies target-language faithfulness by measuring the proportion of translations whose predicted language differs from the intended target. We include this metric to evaluate the effectiveness of our method in zero-shot translations. The ratio is computed using an open-source language detection tool,[5] which identifies generated language via language-specific tokens. As the detector is not fully reliable and only supports a subset of languages, we treat it as a secondary metric and report this metric for the supported ones listed in Appendix A.

**Statistical Significance Testing** To verify the robustness of evaluation results on spBLEU, chrF++, and COMET, we perform statistical significance testing (Koehn 2004). Specifically, we apply paired bootstrap resampling with 1,000 iterations and a resampling ratio of 0.5, and consider differences significant when $p < 0.05$.

[4] `https://tinyurl.com/flores200sacrebleuspm`
[5] `https://github.com/LlmKira/fast-langdetect`

### 4.3 Configuration and Baseline

To enable a thorough comparison with the related and state-of-the-art methodologies, we implement two architectures in comparison: Enc-dec, which is employed in most current MNMT-specific models, and Dec-only, which is our modeling backbone. All models follow the Transformer architecture (Vaswani et al. 2017) in a pre-norm setting, with an embedding size of 1,024, an FFN inner dimension of 4,096, and 16 attention heads, following the official configuration of EC-40 (Tan and Monz 2023). We construct two model configurations with different depths to evaluate the scalability of our proposed method. In the shallow configuration, both architectures comprise a total of 12 layers. The Enc-dec model uses 6 encoder and 6 decoder layers and has 242 million parameters, while the Dec-only model consists of 12 stacked decoder layers with 217 million parameters. In the deep configuration, we double the total depth to 24 layers. The Enc-dec model employs 12 encoder and 12 decoder layers and has 418 million parameters, whereas the Dec-only model uses 24 decoder layers with 368 million parameters.

We employ Fairseq (Ott et al. 2019), an open-source toolkit, to implement the vanilla Enc-dec and Dec-only models as baselines. Since our training set is EC-40, we faithfully follow the hyperparameters, subword vocabulary,[6] and training data provided by the EC-40 repository.[7] Training is conducted on eight Tesla V100 GPUs with the *memory-efficient-fp16* option enabled in Fairseq, a maximum input of 2,048 source tokens per GPU, and a gradient accumulation of 16 steps. Both input and output token lengths are limited to 256, and we share the embedding layer between the encoder and decoder. The hyperparameters from EC-40 are summarized as follows: a random seed of 1,234, a learning rate of 0.0005 with the inverse square root schedule and a warmup of 4,000 steps, the Adam optimizer (Kingma and Ba 2017), dropout of 0.1, attention dropout of 0.1, a label smoothing rate of 0.1, no weight decay, and the temperature sampling with $T = 5$ (Arivazhagan et al. 2019b). We then train for 200,000 steps, averaging the last five checkpoints and saving by epoch. For validation during training, we include only supervised translation directions, i.e., the 80 English-involved directions used in training, as validating zero-shot directions would be considered cheating when measuring the cross-lingual generalization ability. Finally, we employ beam search with a beam size of 5 in inference.

In addition to the vanilla Transformer models, we reproduce four related methods mentioned in Section 2 as baselines, consisting of three methods designed for Enc-dec and one designed for Dec-only: (1) LAVS (Chen et al. 2023) explicitly edits the vocabulary by adding language-

[6] The EC-40 subword vocabulary is implemented with SentencePiece (Kudo and Richardson 2018) using a unigram language model (Kudo 2018).

[7] `https://github.com/Smu-Tan/ZS-NMT-Variations`

specific sub-vocabularies for each language. We follow their original Enc-dec setup and reuse the official implementation,[8] which introduces approximately 12k language-specific tokens (around 12.8M additional parameters). (2) CL (Pan et al. 2021) implicitly aligns sentence-level semantic representations across languages using encoder outputs and a contrastive learning objective. We reuse their implementation[9] and set the temperature to 0.1, as recommended in their work. (3) LCS (Sun et al. 2024) strengthens translation instructions for Enc-dec by implicitly biasing token representations in the encoder with a target language embedding. Following their translation strategy from Fan et al. (2021), we prepend language tags to both source and target sequences and apply the biasing at the 5th encoder layer in 12-layer models (with 6 encoder layers) or the 8th layer in 24-layer encoder models. We re-implement this method as no public code is available. (4) TDO (Qu et al. 2024) enhances Dec-only models by explicitly dividing the process into two phases: the first phase encodes source tokens enriched with target language features, and the second phase concatenates these with target tokens for input to the model. We adopt their official implementation[10] and follow their settings after ablation, using 3 layers for the first stage in 12-layer models and 6 layers in 24-layer models to strengthen zero-shot translation performance.

## 4.4 Result

Although languages in EC-40 can be categorized along two dimensions, i.e., resource size and language family, we follow Tan and Monz (2023) to present the results in Table 1 grouped by resource size. Experimental results in Table 1 demonstrate that our method consistently outperforms all baselines across various configurations and metrics, with statistically significant improvements over the vanilla Dec-only model in all directions. The most notable improvement is in the off-target ratio, which is reduced from 48.40% to 4.65% in 12-layer models and from 26.69% to 3.65% in 24-layer models. Even if the tool used for measuring the off-target ratio is not entirely reliable, the results demonstrate that registering effectively resolves the off-target issue. Additionally, given that LAVS is driven by additional language-specific parameters, the substantial gains from LAVS demonstrate that simply adding language-specific parameters is not a cost-efficient solution. We also observe that the gains from related methods tend to diminish as the model scale increases. In the 12-layer models, the highest gains among the four related methods over vanilla models in zero-shot translation are 3.88, 6.70, and 4.46 points for spBLEU,

[8] https://github.com/PKUnlp-icler/Off-Target-MNMT
[9] https://github.com/PANXiao1994/mRASP2
[10] https://github.com/zhiqu22/PhasedDecoder

| #L | Method | spBLEU ↑ | | | | | | chrF++ ↑ | | | | | | COMET ↑ | | | | | | off. ↓ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | High | Med | Low | E.L. | zero. | avg. | High | Med | Low | E.L. | zero. | avg. | High | Med | Low | E.L. | zero. | avg. | |
| 12 | Enc-dec | 11.05 | 9.80 | 3.95 | 2.59 | 5.86 | 6.99 | 25.61 | 24.61 | 14.25 | 13.34 | 18.16 | 19.68 | 49.61 | 44.93 | 28.15 | 31.17 | 52.29 | 53.72 | 48.40 |
| | +CL | 14.19 | 14.18 | 7.55 | 6.04 | 9.74 | 10.68 | 30.87 | 31.27 | 21.66 | 19.33 | 24.86 | 26.04 | 52.42 | 47.60 | 31.42 | 34.40 | 56.75 | 57.93 | 19.08 |
| | +LCS | 13.67 | 13.17 | 6.07 | 4.10 | 8.35 | 9.37 | 31.05 | 30.90 | 19.84 | 17.45 | 23.70 | 24.95 | 52.22 | 47.73 | 30.19 | 32.50 | 55.44 | 56.71 | 22.43 |
| | +LAVS | 12.22 | 11.19 | 3.98 | 2.99 | 6.40 | 7.54 | 26.66 | 25.77 | 15.54 | 14.64 | 19.38 | 20.88 | 51.57 | 46.13 | 28.71 | 32.16 | 54.64 | 56.08 | 42.98 |
| | Dec-only | 11.51 | 11.03 | 6.06 | 4.20 | 7.30 | 8.34 | 27.10 | 27.16 | 18.01 | 16.17 | 21.00 | 22.34 | 50.88 | 45.88 | 30.13 | 32.64 | 54.41 | 55.71 | 22.21 |
| | +TDO | 13.50 | 12.75 | 6.88 | 4.60 | 8.62 | 9.61 | 29.87 | 29.61 | 19.75 | 18.00 | 23.32 | 24.57 | 52.48 | 47.40 | 30.76 | 33.59 | 56.16 | 57.36 | 27.18 |
| | +Ours | **15.46** | **14.62** | **8.99** | **7.94** | **11.05** | **11.92** | **33.55** | **32.47** | **24.50** | **24.68** | **28.00** | **29.02** | **54.87** | **49.59** | **33.83** | **37.11** | **60.23** | **61.19** | **4.65** |
| 24 | Enc-dec | 15.02 | 14.50 | 5.04 | 3.47 | 8.64 | 9.69 | 31.84 | 31.48 | 16.02 | 15.33 | 22.60 | 23.95 | 54.76 | 49.32 | 29.68 | 32.49 | 56.84 | 58.08 | 26.69 |
| | +CL | 15.97 | 16.15 | 8.16 | 5.96 | 10.79 | 11.76 | 32.52 | 32.84 | 22.05 | 18.30 | 25.51 | 26.73 | 55.20 | 49.96 | 32.53 | 35.06 | 59.26 | 60.38 | 19.99 |
| | +LCS | 16.19 | 15.31 | 5.39 | 3.70 | 9.25 | 10.28 | 33.81 | 32.99 | 18.10 | 16.87 | 24.26 | 25.57 | 55.63 | 50.74 | 30.80 | 33.94 | 57.61 | 58.81 | 23.47 |
| | +LAVS | 16.39 | 16.31 | 6.31 | 3.57 | 10.03 | 11.01 | 34.49 | 33.90 | 18.62 | 15.77 | 25.02 | 26.18 | 56.30 | 50.73 | 31.60 | 32.46 | 57.90 | 58.95 | 21.73 |
| | Dec-only | 15.07 | 15.02 | 7.40 | 5.11 | 9.84 | 10.82 | 31.96 | 32.20 | 20.13 | 18.84 | 24.82 | 26.04 | 55.29 | 50.07 | 31.85 | 34.44 | 58.84 | 59.95 | 19.01 |
| | +TDO | 15.79 | 15.56 | 7.96 | 5.40 | 10.40 | 11.37 | 32.22 | 32.24 | 20.85 | 19.32 | 25.23 | 26.44 | 55.44 | 50.48 | 32.57 | 35.72 | 59.77 | 60.84 | 23.14 |
| | +Ours | **16.98** | **16.57** | **9.88** | **8.37** | **12.26** | **13.12** | **35.00** | **34.56** | **26.01** | **25.61** | **29.53** | **30.51** | **56.92** | **51.82** | **35.37** | **38.04** | **62.66** | **63.54** | **3.65** |

**Table 1** Averaged spBLEU, chrF++, and COMET scores of results on EC-40. The last column (off.) lists the off-target ratio averaged from all zero-shot directions. We report average scores by grouping languages that have the same resource tier with each column averaging over the directions translating from languages outside of this group to the languages inside of this group. #L, E.L., zero, and avg. abbreviate the number of layers, tier of extra low, the average of zero-shot translations, and the average of all translations, respectively. The best score in each column of a block is in bold.

chrF++, and COMET, respectively. In the 24-layer models, the improvements decrease to 2.15, 2.91, and 2.42 points. By contrast, our method achieves more consistent improvements, where 5.19, 7.00, and 5.78 points in the 12-layer models and 3.62, 4.71, and 3.82 points in the 24-layer models, respectively. These comparisons collectively demonstrate that registering exhibits superior scalability across model depths.

## 5 Experiment: Pre-trained Models

The methodology-level experiments presented in Section 4 demonstrate the effectiveness and scalability of our method, thereby laying the foundation for the more practical pre-training with the larger training data. In this section, we pre-train large-scale MNMT models so that they perform comparable against the state-of-the-art open-source MNMT-specific models and commercial LLMs by using training data less than other models and using a limited computational budget, i.e., 80 Tesla V100 GPUs. As a result, we present two pre-trained models, namely MITRE (**m**ulti**i**lingual **t**ranslation with **re**gisters) series of MITRE-466M and MITRE-913M. We also conduct fine-tuning experiments that improve the performance on specific translation directions with limited parallel data, a capability that is essential in practice, such as customizing a deployed multilingual system for a predefined usage scope (Cao et al. 2024) or targeting performance bottlenecks in weaker language pairs (Hudi et al. 2024).

## 5.1 Data Collection via Bridge Languages

**Overview** Building a robust MNMT model requires supervision across multiple translation directions rather than relying solely on English-centric data (Zhang et al. 2020; Eriguchi et al. 2022), i.e., translating into/from English. However, collecting data for all possible translation directions is infeasible, as the number of directions increases quadratically with the number of supported languages. To balance coverage and scalability, we adopt the *Bridge Language Strategy* (Fan et al. 2021), and extract data from the reproduced NLLB corpus.[11] In this strategy, all supported languages are first grouped by their linguistic relations. From each group, one or several high-resource languages are chosen as *bridge languages*, which serve as central hubs that link other languages within the same group and across groups to form supervised translation directions. Consequently, every supervised direction either originates from or translates into a bridge language, ensuring full language participation while keeping the number of directions tractable. This bridge-based design maximizes cross-lingual generalization, specifically, even when no direct parallel data exist between low-resource languages from different groups, and their representations are aligned through these bridge languages, enabling zero-shot translation. Compared to English-centric setups, where knowledge transfer often overfits to English representations, the multi-bridge configuration distributes supervision more evenly across languages, promoting more balanced and robust multilingual transfer.

**Language Grouping** We organize the 24 languages into six groups, consisting of four linguistic families (i.e., Germanic, Romance, Slavic, and Malayo-Polynesian) and two special groups (i.e., English and Asian), as shown in Table 2. English is treated as an individual group. These languages are selected based on two statistics: (1) whether the language can be paired with other languages within the same family to fit the bridge language strategy, and (2) the data of each translation pair should be sufficient, where we empirically set the threshold to more than 100k sentences per pair. We also control the overall scale of language coverage so that it approaches, but does not exceed, the upper bound permitted by our computational budget. The Germanic, Romance, and Slavic families each include five languages, while the Malayo-Polynesian family comprises four languages that differ substantially from these European families. In addition, we define an Asian group that includes four widely used Asian languages: `ja`, `zh`, `ko`, and `vi`. Although these Asian languages belong to different linguistic families, they share lexical and syntactic similarities[12] due to historical and cultural proximity. From each group, we select the

[11] `https://opus.nlpl.eu/NLLB/corpus/version/NLLB`

[12] For example, `ja` and `zh` share overlapping characters; `ja` and `ko` have comparable grammar; and `zh` and `vi` exhibit similar syntactic patterns.

| Group | ISO | Flores | Language | Script | Bridge | Family | ISO | Flores | Language | Script | Bridge |
|---|---|---|---|---|---|---|---|---|---|---|---|
| English* | en | eng_Latn | English | Latin | ✔ | Slavic | cs | ces_Latn | Czech | Latin | ✔ |
| Germanic | de | deu_Latn | German | Latin | ✔ | | pl | pol_Latn | Polish | Latin | |
| | nl | nld_Latn | Dutch | Latin | ✔ | | bg | bul_Cyrl | Bulgarian | Cyrillic | |
| | sv | swe_Latn | Swedish | Latin | | | uk | ukr_Cyrl | Ukrainian | Cyrillic | |
| | da | dan_Latn | Danish | Latin | | Malayo-Polynesian | id | ind_Latn | Indonesian | Latin | ✔ |
| | af | afr_Latn | Afrikaans | Latin | | | ms | zsm_Latn | Malay | Latin | ✔ |
| Romance | ro | ron_Latn | Romanian | Latin | ✔ | | jv | jav_Latn | Javanese | Latin | |
| | fr | fra_Latn | French | Latin | ✔ | | tl | fil_Latn | Filipino | Latin | |
| | es | spa_Latn | Spanish | Latin | | Asian* | ja | jpn_Jpan | Japanese | Kanji; Kana | ✔ |
| | it | ita_Latn | Italian | Latin | | | zh | cmn_Hans | Chinese | Chinese | ✔ |
| | pt | por_Latn | Portuguese | Latin | | | ko | kor_Hang | Korean | Hangul | |
| Slavic | ru | rus_Cyrl | Russian | Cyrillic | ✔ | | vi | vie_Latn | Vietnamese | Latin | |

**Table 2** Languages in data collection. For each language, we present its ISO 639-1 code, Flores code, language name, and script, corresponding to the columns *ISO*, *Flores*, *Language*, and *Script*, respectively. *Bridge* indicates whether the language serves as a bridge language. A * denotes a group that is not referred to as a linguistic family.

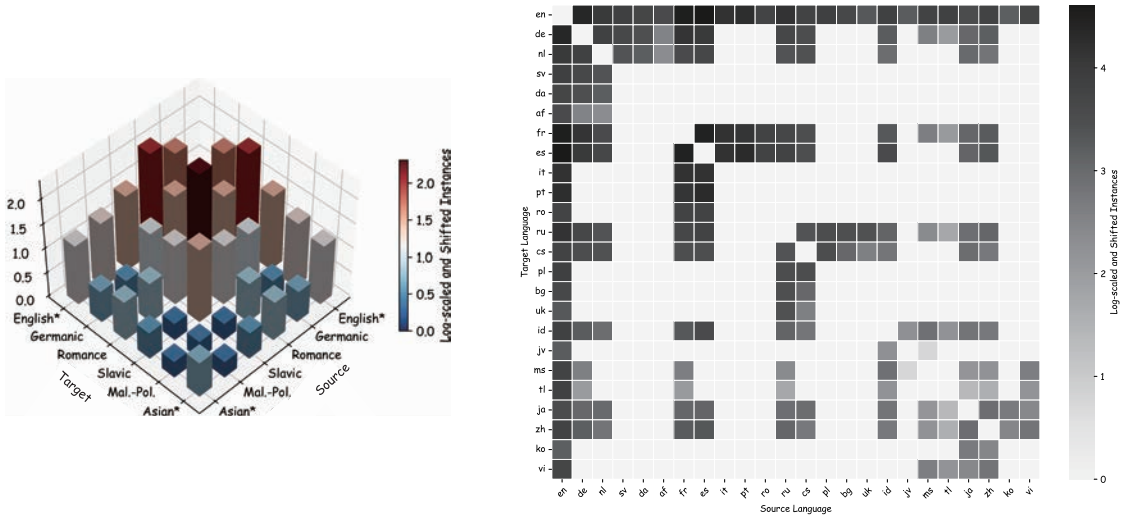
two most resource-rich languages as bridge languages.

**Linking Rules** Data collection follows three rules: (1) `en` links with all other languages, meaning that all translation directions involving English are included as the supervised directions in training; (2) All bridge languages link with one another, forming multilingual anchor links across groups; (3) Within each group, each bridge language links to the remaining languages to ensure sufficient intra-family supervision. Additionally, `ms` being relatively low-resource, cannot fully satisfy rules (2) and (3). Therefore, we collect as much additional data as possible for `ms` to strengthen its representation in pre-training.

**Dataset Summary** Out of 552 possible translation directions, we collect data for 194 supervised directions, yielding approximately 9.3 billion sentence pairs for pre-training. Figure 3a illustrates the data distribution across groups, and Figure 3b shows per-language coverage, which also exhibits the distribution of supervised and zero-shot directions.

## 5.2 Configuration and Baseline

We begin by training a vocabulary of 160,000 tokens using SentencePiece (Kudo and Richardson 2018) with a unigram language model (Kudo 2018) on 150 million sentences randomly sampled from the training set. We then pre-train our models under the same training settings with 466 million and 913 million parameters, respectively. Specifically, the hyperparameters differ (Table 3), but the training procedure is the same for both versions. MITRE-466M is configured by following the basic configurations of M2M (Fan et al. 2021) and NLLB (NLLB Team 2022). In contrast, MITRE-913M scales both width and depth, where the embedding size,




(a) Group Level (b) Language Level

**Figure 3** Data distribution of our pre-training dataset. 3a shows data size distribution at the group level, while (b) displays data size at the language level. In (a), non-zero values are scaled by log10 and adjusted by subtracting 7 for clearer visualization. In (b), non-zero values are also scaled by log10 and shifted by subtracting the minimum value to enhance illustration clarity.

| Model | Embedding dim | FFN inner dim | Attention heads | Layers | Params |
|---|---|---|---|---|---|
| MITRE-466M | 1,024 | 4,096 | 16 | 24 | 466M |
| MITRE-913M | 1,280 | 5,120 | 20 | 36 | 913M |

**Table 3** Modeling hyperparameters for MITRE series.

FFN dimension, and number of attention heads are scaled by 1.25×, and the layer count by 1.5× to reach the upper limit of our GPU memory. The two models were trained on 80 Tesla V100 GPUs, setting *memory-efficient-fp16* in Fairseq, with a maximum input of 1,408 source tokens per GPU and a gradient accumulation of 10 steps. In practice, this setup results in each batch containing approximately 0.91 million source tokens.

We follow NLLB (NLLB Team 2022) to set hyperparameters in the training: the learning rate of 0.002 with the inverse square root schedule and the warmup of 8,000 steps, a random seed of 42, the Adam optimizer (Kingma and Ba 2017), dropout of 0.1, attention dropout of 0.1, a label smoothing rate of 0.1, no weight decay, and the temperature sampling with $T = 1$ (Arivazhagan et al. 2019b). We train for 300,000 steps and save a checkpoint per 10,000 steps. Finally, we aver-

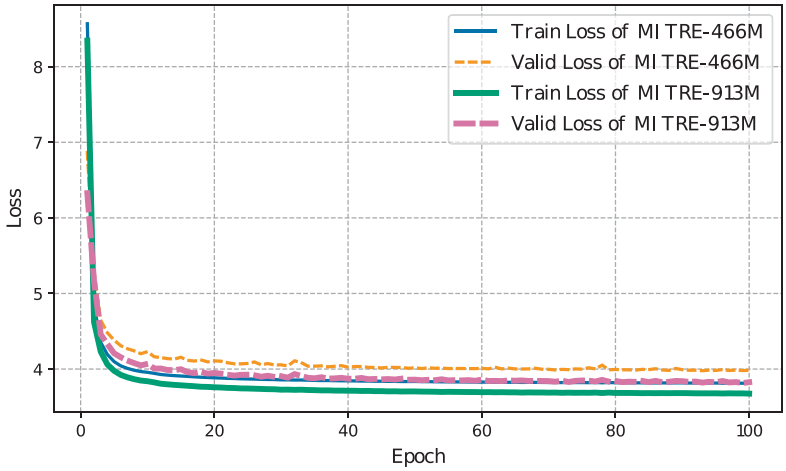



**Figure 4** The training and validation loss in pre-training MITRE. We report the first 100 epochs, each with approximately 2,262 steps.

age the last 5 checkpoints. The validation setting remains the same as described in Sections 4.1. Specifically, we extract and recombine data from Flores *dev* to cover 194 supervised directions described in Section 5.1. Figure 4 shows the training and validation loss curves, demonstrating that MITRE-466M and MITRE-913M are stable without almost no fluctuations and highly consistent under the two hyperparameter configurations, which also verifies the scalability of our approach demonstrated in Section 4.

The testing procedures also remain the same as described in Sections 4.1 and 4.3. Specifically, (1) we set the beam size to 5 during inference for our two MITRE models; (2) we use Flores *devtest* split as the test set, where testing covers all 522 possible translation directions; (3) we perform a comprehensive evaluation, including spBLEU, chrF++, COMET, and statistical significance testing as described in Section 4.2.

We compare our model against not only state-of-the-art MNMT-specific models, but also commercial LLMs, because commercial LLMs empirically present the upper bound of MNMT-specific models as described in Section 2. First, the MNMT models include three versions of M2M (483M[13], 615M, and 1.2B) (Fan et al. 2021) and three versions of NLLB (615M-distilled, 1.3B, and 3.3B) (NLLB Team 2022), where inference is conducted under the same configuration as MITRE to ensure fair comparison, specifically, beam search with a beam size of 5 is applied. Also, we

[13] The official name is M2M-418M, however, this model actually has 483M parameters.

include commercial LLMs, GPT-3.5 turbo[14] (Brown et al. 2020) and GPT-4o mini[15] (OpenAI 2024), and we list the setups and prompts used for GPT series in Appendix C. All of these public models are trained under different conditions, causing severe challenges for completely fair comparisons. Here, we strive to maintain a relatively fair comparison. For instance, although MITRE supports 24 languages fewer than M2M of 100 and NLLB of 200, MITRE is trained with smaller parameter sizes, less training data, and a limited computational budget. Therefore, comparing MITRE against larger-scale models is informative for evaluating performance under constrained resources. We will leave the further fairer comparisons to future studies, given our computational budget limitation.

In fine-tuning experiments, we compare the fine-tuning adaptability between MITRE and NLLB. Experiments are designed to include three scenarios by randomly selecting 5, 25, and 100 translation directions from all 552 supported ones by MITRE, i.e., approximately two-thirds of fine-tuning directions are not seen in the pre-training of MITRE. These experiments are not completely fair, rather relatively fair. Specifically, although NLLB supports languages more than MITRE, NLLB has a larger parameter size and is trained on more data. Moreover, the data collection strategy of NLLB is to cover all possible translation directions (NLLB Team 2022) by data mining and back translation (Edunov et al. 2018), instead of the Bridge Language Strategy. This indicates that most language pairs used in our fine-tuning experiments are seen during pre-training NLLB. Given these differences, the comparison can therefore be considered relatively fair.

In the setup of finetuning experiments, we use the Flores *dev* split as the fine-tuning dataset, which comprises 997 sentence pairs per direction. Each model is fine-tuned using both full-parameter updates and LoRA (Hu et al. 2022), but NLLB-3.3B is excluded from full-parameter fine-tuning due to the computational resources requirement, which exceeds the memory capacity of the Tesla V100 GPUs used in our experiments. The detailed configurations for fine-tuning are described in Appendix B. Finally, the performance is also measured by the Flores *devtest* split with a beam size of 5 in inference.

## 5.3 Main Results

In contrast to the EC-40 settings reported in Table 1, the pretraining data used in our pre-training setting does not exhibit systematic differences in resource size across languages after carefully designed bridging. Therefore, we report the results in Table 4 grouped by language

[14] Version is *gpt-3.5-turbo-0125*.
[15] Version is *gpt-4o-mini-2024-07-18*.

family rather than by resource tier. The experimental results comparing MITRE with the baselines are presented in Table 4. We report scores in spBLEU, chrF++, and COMET, where the off-target ratio is omitted due to its near-zero values in large-scale models with sufficient training

| | | English | | Germanic | | Romance | | Slavic | | Mal.-Polyn. | | Asian | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Model | | → | ← | → | ← | → | ← | → | ← | → | ← | → | ← | avg. |
| **spBLEU scores** | | | | | | | | | | | | | | |
| M2M | 483M | 30.36 | 31.92 | 24.40 | 22.58 | 24.01 | 25.81 | 22.59 | 23.40 | 17.94 | 16.50 | 18.30 | 18.37 | 22.10 |
| | 615M | 30.69 | 31.98 | 26.35 | 25.56 | 25.47 | 27.52 | 24.02 | 24.77 | 20.11 | 17.49 | 19.31 | 19.09 | 23.65 |
| | 1.2B | 35.92 | 35.14 | 29.51 | 26.82 | 28.40 | 28.38 | 26.58 | 26.19 | 18.09 | 17.57 | 15.48 | 20.07 | 24.69 |
| NLLB | 615M | 35.85 | 41.04 | 28.13 | 27.41 | 27.46 | 29.09 | 25.40 | 25.33 | 25.39 | 24.35 | 20.72 | 19.42 | 26.05 |
| | 1.3B | 38.08 | 43.42 | 30.52 | 30.17 | 29.63 | 31.42 | 27.84 | 28.25 | 28.08 | 26.87 | 23.50 | 21.06 | 28.51 |
| | 3.3B | 39.80 | **45.08** | 31.93 | 31.77 | 30.88 | 32.62 | 29.29 | 30.13 | 29.81 | **28.08** | 25.18 | 22.56 | 30.01 |
| GPT | 3.5 turbo | 38.27 | 42.37 | 31.01 | 31.02 | 30.09 | 32.73 | 28.56 | 27.85 | 26.75 | 22.81 | 23.61 | 24.08 | 28.66 |
| | 4o mini | **41.49** | 43.97 | 33.09 | 31.92 | 31.40 | 34.03 | 30.54 | 30.69 | **31.01** | 27.20 | **26.34** | **27.53** | 31.09 |
| MITRE | 466M | 40.20 | 42.60 | 32.14 | 31.51 | 31.32 | 33.26 | 29.36 | 29.80 | 28.46 | 26.16 | 24.05 | 23.56 | 29.77 |
| | 913M | 41.16 | 44.17 | **33.34** | **32.95** | **32.53** | **34.23** | **30.74** | **31.26** | 29.90 | 27.22 | 25.93 | 25.58 | **31.15** |
| **chrF++ scores** | | | | | | | | | | | | | | |
| M2M | 483M | 50.43 | 54.36 | 44.84 | 46.24 | 43.96 | 48.26 | 43.36 | 43.06 | 37.54 | 41.77 | 39.83 | 27.85 | 42.53 |
| | 615M | 49.97 | 54.74 | 46.77 | 48.62 | 45.34 | 49.83 | 44.70 | 44.42 | 40.17 | 43.59 | 41.04 | 28.84 | 44.12 |
| | 1.2B | 54.80 | 54.44 | 49.02 | 47.06 | 47.49 | 48.04 | 45.87 | 43.35 | 32.78 | 40.68 | 30.90 | 28.01 | 42.56 |
| NLLB | 615M | 55.54 | 61.73 | 48.48 | 50.39 | 47.38 | 51.54 | 46.38 | 45.34 | 45.98 | 50.96 | 42.56 | 29.73 | 46.70 |
| | 1.3B | 56.82 | 63.35 | 50.07 | 52.21 | 48.80 | 53.19 | 47.98 | 47.46 | 47.92 | 52.49 | 44.70 | 30.98 | 48.40 |
| | 3.3B | 57.88 | **64.27** | 50.95 | 53.16 | 49.63 | 53.92 | 48.91 | 48.76 | 49.06 | **53.28** | 45.71 | 31.97 | 49.35 |
| GPT | 3.5 turbo | 55.30 | 61.60 | 49.39 | 52.16 | 48.31 | 53.34 | 47.54 | 46.35 | 45.64 | 48.05 | 43.46 | 31.20 | 47.41 |
| | 4o mini | 58.03 | 62.85 | 51.26 | 52.90 | 49.33 | 54.33 | 49.12 | 48.62 | **49.39** | 52.25 | 45.86 | **34.11** | 49.48 |
| MITRE | 466M | 58.11 | 62.77 | 51.40 | 53.41 | 50.06 | 54.46 | 49.18 | 48.62 | 47.83 | 52.04 | 45.10 | 32.41 | 49.29 |
| | 913M | **58.84** | 64.01 | **52.40** | **54.59** | **51.03** | **55.36** | **50.32** | **49.84** | 48.97 | 52.88 | **46.65** | 33.88 | **50.42** |
| **COMET scores** | | | | | | | | | | | | | | |
| M2M | 483M | 81.63 | 81.40 | 78.90 | 77.03 | 80.48 | 78.49 | 79.85 | 80.31 | 68.34 | 72.75 | 78.67 | 78.57 | 77.74 |
| | 615M | 81.16 | 82.15 | 81.42 | 79.98 | 82.00 | 80.50 | 81.29 | 82.43 | 72.56 | 74.62 | 80.02 | 79.96 | 79.79 |
| | 1.2B | 85.93 | 85.17 | 84.15 | 82.87 | 84.46 | 83.12 | 83.87 | 85.41 | 75.91 | 77.83 | 82.95 | 82.58 | 82.66 |
| NLLB | 615M | 86.61 | 86.76 | 84.06 | 82.77 | 84.50 | 83.05 | 83.41 | 84.39 | 81.26 | 83.51 | 82.12 | 82.02 | 83.33 |
| | 1.3B | 87.76 | 87.63 | 85.72 | 84.76 | 85.93 | 84.76 | 85.07 | 86.82 | 83.57 | 84.93 | 84.36 | 83.52 | 85.13 |
| | 3.3B | 88.22 | 88.09 | 86.49 | 85.58 | 86.58 | 85.45 | 85.84 | 87.96 | 84.63 | 85.45 | 85.42 | 84.56 | 85.96 |
| GPT | 3.5 turbo | 87.67 | 88.02 | 86.26 | 85.80 | 86.62 | 85.96 | 85.91 | 87.50 | 83.09 | 82.09 | 85.45 | 85.77 | 85.66 |
| | 4o mini | **89.59** | **88.64** | **87.50** | **86.38** | **87.58** | **86.57** | **87.08** | **88.90** | **85.89** | **86.03** | **86.99** | **87.47** | **87.16** |
| MITRE | 466M | 87.87 | 87.29 | 85.99 | 84.98 | 86.49 | 85.14 | 85.58 | 87.19 | 82.24 | 83.41 | 84.38 | 84.29 | 85.19 |
| | 913M | 88.11 | 87.81 | 86.54 | 85.61 | 86.96 | 85.70 | 86.16 | 88.03 | 83.15 | 83.80 | 85.52 | 85.35 | 85.88 |

**Table 4** Averaged spBLEU, chrF++, and COMET scores comparing MITRE with baselines. Mal.-Polyn. abbreviates Malayo-Polynesian, and other abbreviations follow Table 1. → indicates translation directions from the corresponding group to languages outside this group, and ← indicates the inverse directions of →. Additionally, we use light gray boxes and dark gray boxes to respectively highlight scores significantly better than NLLB-3.3B and GPT-4o mini. When a score meets both conditions, it is displayed using the dark gray box, as GPT-4o mini typically serves as a stronger baseline.

data, as the off-target problem is largely resolved by increased computational capacity as discussed in Section 1. We first observe that NLLB-3.3B surpasses MITRE-913M by 0.91 spBLEU points for translations into English and by 0.86 points for Malayo-Polynesian languages. However, MITRE-913M consistently achieves higher scores in all other translation directions, with an overall average gain of 1.14 points. Given that NLLB shows extremely strong performance in English translation, i.e., surpassing GPT-4o mini by 1.11 BLEU points, but performs relatively worse than MITRE in other directions, we can conclude that MITRE, a Dec-only model with registering, achieves better cross-lingual generalization than NLLB, an Enc-dec architecture. Specifically, although NLLB is trained on all possible translation directions in the NLLB dataset, non-English languages do not benefit from its strong English translation capability. The tendency has already been pointed out in the prior studies, which indicate that the strong English performance accompanied by weaker results in non-English languages has been regarded as a form of overfitting (Gu et al. 2019; Liu et al. 2021; Qu and Watanabe 2022). In terms of parameter scaling efficiency, scaling NLLB from 1.3B to 3.3B parameters improves spBLEU from 28.51 to 30.01 (+1.50 points) with 2B additional parameters. In contrast, MITRE improves spBLEU from 29.77 to 31.15 (+1.38 points) with approximately 450M additional parameters, achieving a similar improvement with substantially fewer added parameters and reaching a higher final score. The gain demonstrates the parameter scaling efficiency of MITRE given that the pretrained performance of MITRE-466M is already higher than that of NLLB-1.3B, and the prior studies indicate that obtaining further improvements on already better-performing MNMT models remains challenging (Zhang et al. 2020; Eriguchi et al. 2022; NLLB Team 2022). Meanwhile, M2M-1.2B, which is also trained with the Bridge Language Strategy, performs significantly worse than MITRE-466M. Although it is difficult to draw any concrete conclusions given that M2M-1.2B supports more languages than MITRE-466M, our approach scales better than M2M under the same training strategy.

As shown in Table 4, chrF++, a character-level similarity metric, exhibits a trend similar to spBLEU, a lexical-level similarity metric, with MITRE-913M outperforming NLLB-3.3B and even significantly exceeding GPT-4o mini. In contrast, the scores of COMET, which measures semantic-level similarity, show a different trend, where MITRE-913M underperforms GPT-4o mini in COMET. This is expected, as commercial models are aligned with human-like styles (Wang et al. 2023), whereas MITRE follows the training data's style. To support this, GPT-4o mini shows a significant improvement over GPT-3.5 turbo on COMET. Therefore, based on the three evaluation metrics and their respective implications as discussed in Section 4.2, we conclude that MITRE-466M performs comparably to NLLB-3.3B, while MITRE-913M not only

outperforms NLLB-3.3B in most cases but also achieves performance close to GPT-4o mini, demonstrating strong practical potential.

## 5.4 Fine-tuning Results

Table 5 shows the fine-tuning results. By comparing NLLB and MITRE, we observe that MITRE outperforms NLLB in both scenarios: fine-tuning on a few translation directions and fine-tuning on multiple directions simultaneously. Specifically, we find that performance gains from fine-tuning increase with model size, and our MITRE-913M shows the highest improvement in both full-parameter fine-tuning and LoRA-based fine-tuning. Since both pre-trained NLLB and MITRE have already achieved near-zero off-target ratios before fine-tuning, these substantial gains shown in fine-tuned MITRE can be attributed to increased quality rather than addressing the off-target problem, which reaffirms the practical potential of MITRE. Previous empirical studies in MNMT have shown that achieving improvements through fine-tuning on already strong models is particularly challenging (Zhang et al. 2020; Moslem et al. 2023; Cao et al. 2024). Our models have already surpassed NLLB-3.3B in the pre-trained setting, and fine-tuning MITRE leads to even larger improvements across all three scenarios. Moreover, Table 5 reflects the parameter utilization efficiency of MITRE. Under comparable conditions (Section 5.2), larger models

| | | 5-direction | | 25-direction | | 100-direction | |
|---|---|---|---|---|---|---|---|
| model | | spBLEU | COMET | spBLEU | COMET | spBLEU | COMET |
| NLLB-615M | pre-trained | 24.00 | 82.91 | 25.88 | 83.71 | 25.37 | 83.35 |
| | lora | 24.68 (+0.68) | 83.23 (+0.32) | 26.84 (+0.96) | 84.10 (+0.39) | 26.41 (+1.04) | 84.00 (+0.65) |
| | fine-tuned | 25.59 (+1.59) | 83.80 (+0.89) | 27.32 (+1.44) | 84.29 (+0.58) | 26.70 (+1.33) | 84.17 (+0.82) |
| NLLB-1.3B | pre-trained | 26.59 | 84.86 | 28.33 | 85.41 | 27.82 | 85.17 |
| | lora | 27.39 (+0.80) | 85.20 (+0.34) | 29.49 (+1.16) | 85.87 (+0.46) | 29.18 (+1.36) | 85.87 (+0.70) |
| | fine-tuned | 28.50 (+1.91) | 85.78 (+0.92) | 30.13 (+1.8) | 86.09 (+0.68) | 29.61 (+1.79) | 86.05 (+0.88) |
| NLLB-3.3B | pre-trained | 27.95 | 85.60 | 29.70 | 86.16 | 29.31 | 85.98 |
| | lora | 29.05 (+1.10) | 86.08 (+0.48) | 31.23 (+1.53) | 86.63 (+0.47) | 30.98 (+1.67) | 86.71 (+0.73) |
| MITRE-466M | pre-trained | 24.51 | 83.73 | 28.71 | 85.41 | 29.07 | 85.26 |
| | lora | 26.37 (+1.86) | 84.47 (+0.74) | 29.97 (+1.26) | 86.09 (+0.68) | 30.42 (+1.35) | 86.27 (+1.01) |
| | fine-tuned | 28.19 (+3.68) | 85.28 (+1.55) | 30.61 (+1.90) | 86.36 (+0.95) | 30.81 (+1.74) | 86.45 (+1.19) |
| MITRE-913M | pre-trained | 25.52 | 84.56 | 29.95 | 86.07 | 30.37 | 85.92 |
| | lora | 28.14 (+2.62) | 85.59 (+1.03) | 31.68 (+1.73) | 86.90 (+0.83) | 32.33 (+1.96) | 87.15 (+1.23) |
| | fine-tuned | **30.09 (+4.57)** | **86.45 (+1.89)** | **32.47 (+2.52)** | **87.23 (+1.16)** | **32.73 (+2.36)** | **87.35 (+1.43)** |

**Table 5** Averaged spBLEU and COMET scores of results on three fine-tuning scenarios, where the specific translation directions are listed in Appendix B. The term, fine-tuned, means fine-tuning with full parameters. However, full-parameter fine-tuning is not performed for NLLB-3.3B due to the computational resources requirement. The best score is in bold, dark gray boxes highlights the largest gain in fine-tuned relative to pre-trained, and light gray boxes highlights the largest gain in LoRA.

consistently achieve higher fine-tuning gains than smaller ones, for example, in the 100-direction scenario, lora gains of NLLB-3.3B, 1.3B, and 615M are 1.04, 1.36, and 1.67 spBLEU points, respectively, and the same trend is observed in MITRE-913M versus MITRE-466M. Notably, MITRE-913M uses only one quarter of the parameters of NLLB-3.3B while delivering greater and higher-quality improvements, highlighting its superiority in fine-tuning and its practicality as a publicly released model. In addition, considering our cost-effective data collection strategy, the substantial improvements in fine-tuning suggest that the pre-trained performance of MITRE can be further improved if more data are available in pre-training.

## 6 Discussion

### 6.1 Ablation Study

We conduct two ablation studies to measure the impact of registering. As mentioned in Section 3.2, registering integrates the advantages of both explicit and implicit strategies through two steps: (1) explicitly adding registers and (2) modifying the attention mask to implicitly optimize the source token representation. As shown in Table 6, merely adding registers reduces the performance of vanilla Dec-only models; registering only becomes effective after modifying the attention mask. This result aligns with expectations, i.e., the model without constraints on generation defaults to relying directly on source tokens instead of registers.

In Section 3.2, we introduced that the lengths of $\boldsymbol{r}^v$ and $\boldsymbol{x}'$ are matched to ensure a one-to-one correspondence between registers and source tokens. To validate this design, we vary the length of $\boldsymbol{r}^v$ while keeping $\boldsymbol{x}'$ fixed to observe performance trends. Specifically, a ratio less than 1.0 means that $\boldsymbol{r}^v$ augments $\boldsymbol{x}'$, while a ratio greater than 1.0 means $\boldsymbol{r}^v$ compresses $\boldsymbol{x}'$. The trend illustrated in Figure 5, where the length ratio of 1.0 is the optimal case, empirically supports our

| #layer | register | mask | spB.↑ | chrf↑ | com.↑ | off.↓ |
|---|---|---|---|---|---|---|
| 12 | ✘ | ✘ | 8.34 | 22.34 | 55.71 | 22.21 |
| | ✔ | ✘ | 8.19 | 22.96 | 55.72 | 32.11 |
| | ✔ | ✔ | 11.92 | 29.02 | 61.19 | 4.65 |
| 24 | ✘ | ✘ | 10.82 | 26.04 | 59.95 | 19.01 |
| | ✔ | ✘ | 8.91 | 24.06 | 57.60 | 34.22 |
| | ✔ | ✔ | 13.12 | 30.51 | 63.54 | 3.65 |

**Table 6** Averaged scores of ablation study on EC-40. Register means adding registers, and mask means modifying the attention mask.

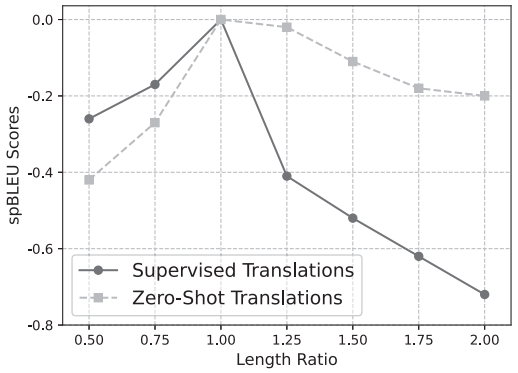



**Figure 5** Variations of spBLEU scores on EC-40 with different length ratios of $\boldsymbol{r}^v$ to $\boldsymbol{x}'$, where the length of $\boldsymbol{x}'$ is fixed.

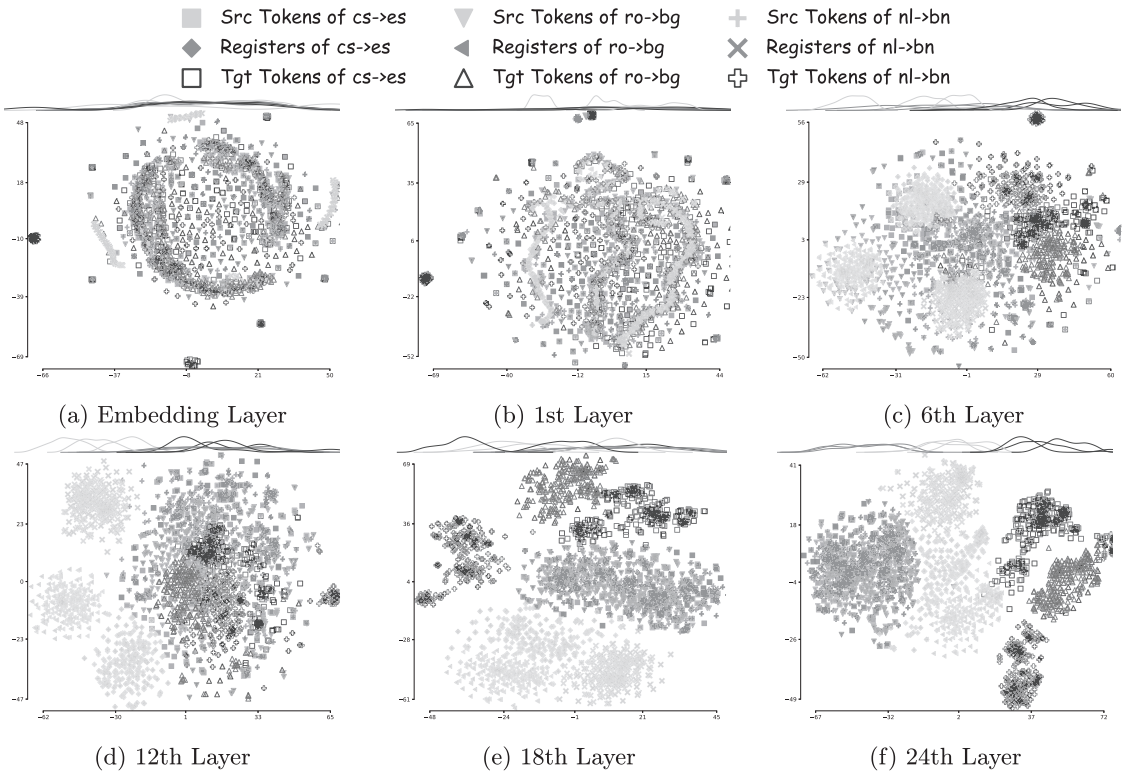
design.

## 6.2 Mechanism: registering source tokens to target language spaces

We analyze and validate the registering mechanism from two perspectives: (1) the representation of registers is located in the intended target language space, and (2) the register carries the semantics of the positionally-aligned source token.

We analyze token representations by randomly selecting 100 translation instances from three translation directions and applying t-SNE (van der Maaten and Hinton 2008) to reduce them to two dimensions in Figure 6. Although registers lack inherent semantics and are initially distinguished only by their positional embeddings in the initial step (Figures 6a and 6b), the distribution shown in the final (24th) layer follows our design as shown in Figure 6f. Specifically, source tokens, registers, and target tokens are clearly separated into distinct spaces, indicating that the model can distinguish their different functions. Moreover, source token representations from different languages are clustered together, suggesting that the model processes them in a language-agnostic manner. Most importantly, for registers and target tokens, token representations for the three translation directions are clustered in separate spaces. This supports our design, where the register representation is located in the intended target language space. Appendix D presents two additional scenarios: (1) register representations converge when sharing the same target language and (2) register representations separate when target languages differ, further confirming that registers align with the intended target language space.

We have qualitatively analyzed the relation between source tokens and registers in terms of

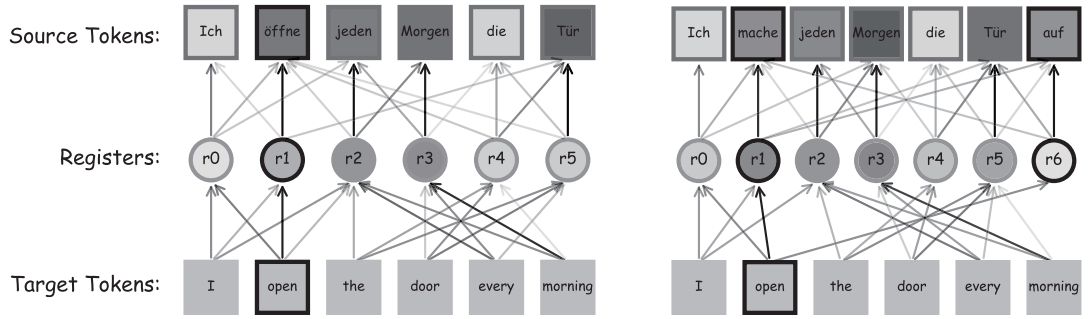



(a) Embedding Layer (b) 1st Layer (c) 6th Layer

(d) 12th Layer (e) 18th Layer (f) 24th Layer

**Figure 6** 2D distributions of token-level representations extracted from the different layers of a model trained on EC-40. Each class listed in the legend contains 300 randomly sampled tokens. These illustrations indicate that register representations evolve through layers to progressively encode semantics aligned with the intended target language.

attention weights using a specific grammatical item of German. Figure 7 shows two translation instances translated from German to English, where they share the same semantics. We observe that the target tokens in both examples are identical, while their source tokens differ only in the verb form highlighted with a purple border. In Figure 7a, the verb is a single token, "öffne", whereas in Figure 7b, it consists of two distant tokens, "mache ... auf". Despite this difference, "öffne" and "mache ... auf" share the same semantics and can be replaced by each other. The arrows in Figure 7 represent the attention weights for each token representation. In Figure 7a, the highest attention weight of the verb in the target tokens, i.e., "open", comes from $r_1$; in Figure 7b, "open" pays the highest attention weights to $r_1$ and $r_6$. Meanwhile, $r_1$ in Figure 7a aligns positionally with "öffne", and in Figure 7b, $r_1$ and $r_6$ align positionally with "mache ... auf". Additionally, we observe that the highest attention weight of a register typically comes from




(a) Inseparable Verb (b) Separable Verb

**Figure 7** Token-level attention weights illustration of two translation instances from German to English, where the weight of each token is averaged across all heads of a model trained on EC-40. The top-3 attention directions for each token are labeled, with darker colors indicating higher attention weights. Note that while the target tokens for these two instances are identical, their source tokens are not, because the verbs in 7a and 7b are semantically equivalent but have different forms. To aid understanding, we highlight the verbs with black borders: "öffne" in 7a and "mache ... auf" in 7b both correspond to the target verb "open". Then, the registers with the highest attention weights associated with these verbs are also marked with purple borders.

its positionally-aligned source token.[16] Based on the above, we conclude that the mechanism of registering aligns with our design, namely, the register's activation represents the target language and carries the semantics of the positionally-aligned token.

## 6.3 Efficiency

**Method Efficiency** As introduced in Section 3.2, we introduce $\boldsymbol{r}^v$ to reinforce the language transfer ability in the Dec-only MNMT model. Although introducing $\boldsymbol{r}^v$ doubles the length of the input sequence, this design does not practically incur noticeable cost in either training efficiency or inference efficiency since the registers do not introduce any additional parameters, based on our measurements on EC-40 using a single Tesla V100 GPU. Specifically, the training time for models with registers is 1.34 times that of the vanilla Dec-only model, 1.63 times that of the vanilla Enc-dec model, and approximately comparable (1.01 times) to the previous state-of-the-art method, i.e., CL (Pan et al. 2021). During inference, thanks to the KV Cache, the model with registers merely incurs a 1.1 times increase in computational cost compared to the vanilla

[16] We compute attention weights for 1,000 randomly selected sentences in MITRE-466M and show that 85.71% of registers followed this pattern. Registers that do not conform are often associated with source tokens carrying relatively weak semantic content. Additional examples are provided in Appendix E.

Dec-only model. However, in practical usage, i.e., MITRE, the inference time is substantially lower than that of NLLB and LLMs due to the smaller number of parameters.

**Parameter Efficiency** In Section 4, we show our method, registering, achieves significantly better translation quality than baseline methods under the same conditions. This reflects the parameter utilization efficiency of our models. Also, the fine-tuning experiments in Section 5.4 support this efficiency again, where MITRE performs better than NLLB under comparable conditions. Nevertheless, not only the scaling process from 12-layer models to 24-layer models in EC-40 (Section 4.4), but also comparing two versions of MITRE to M2M, which has the same training strategy, in Section 5.3, supports the parameter scaling efficiency of our proposed method.

## 7 Conclusion

In this work, we present registering, a simple, cost-efficient yet effective method for improving multilingual neural machine translation (MNMT). By introducing a set of registers and modifying the attention mask, our method ensures that the generation of target tokens relies solely on register activations. Analysis experiments demonstrate that these registers encode the semantics of source tokens within the target language space. Building on this methodology, we pre-train and open-source two MNMT-specific models, MITRE-466M and MITRE-913M, supporting translation across 24 languages. Experimental results show that MITRE achieves performance close to commercial large language models while establishing a new state-of-the-art in MNMT.

## Limitations

A key concern is our limited computational resources to support the more comprehensive experiments. Given that the training of MITRE-913M has already required 80 Tesla V100 GPUs for one month, we are unable to run fine-grained experiments or pre-train models without our method as a comparison. As a compromise, we measure our method at the methodology level via a large-scale benchmark in advance (Section 4), and then we pre-train MITRE to rival other public models (Section 5).

We are unable to continuously scale up MITRE due to the limited computational resources. Although we discussed the parameter scaling efficiency in Section 6.3, it is not clear whether this could be scaled with more parameters and more training data. We leave the exploration at the larger scale for future study.

We have several statements regarding our efforts in keeping the relatively fair comparison in

Section 5. First, MITRE supports only 24 languages, fewer than the supported languages of M2M (100 languages), NLLB (around 200 languages), and GPT-4o mini (over 50 languages). MITRE is substantially smaller than M2M and NLLB, yet MITRE-913M performs significantly better than NLLB-3.3B as shown in Section 5.3. Second, the limited training data: (1) Our training data is collected from the reproduced version of the NLLB dataset, which includes fewer samples per translation direction than those used in training NLLB models. (2) As described in Section 5.1, our Bridge Language strategy results in fewer supervised translation directions, whereas NLLB is trained on as many directions as possible. (3) As described in Section 2, the training of NLLB involves data augmented by back-translation (Edunov et al. 2018), whereas MITRE is trained on the training set only. We will leave it to a future study to train MITRE on the dataset of NLLB and M2M for a fairer comparison.

## Ethical Considerations

MITRE models have not been evaluated for toxicity or have not undergone detoxification. Thus, while we open-source MITRE, we recommend its use primarily for research purposes or in applications only after thorough appropriate processing.

## Acknowledgement

This paper is an extended version of our preliminary work (Qu et al. 2025), which has been accepted by the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL 2025 Main) as an oral presentation. Compared with the preliminary version, we have made the following major revisions and extensions: (1) We have revised the title to better reflect the focus on our pre-trained model, whereas the original paper emphasized methodological validation. (2) Based on this shift in focus, we have completely rewritten Sections 1 and 2 to more thoroughly convey our motivation. (3) We have added additional experimental results in Section 4. (4) We have rewritten Section 5 to provide a comprehensive description of our pre-trained models. (5) We have expanded Section 6 with more in-depth analyses. We also sincerely thank the anonymous reviewers for their constructive comments and suggestions, which have led to substantial improvements in this manuscript. We are grateful to the editor for the helpful guidance and encouragement to revise and resubmit this work.

## References

Aharoni, R., Johnson, M., and Firat, O. (2019). "Massively Multilingual Neural Machine Translation." *CoRR*, **abs/1903.00089**.

Alves, D. M., Pombal, J., Guerreiro, N. M., Martins, P. H., Alves, J., Farajian, A., Peters, B., Rei, R., Fernandes, P., Agrawal, S., Colombo, P., de Souza, J. G. C., and Martins, A. (2024). "Tower: An Open Multilingual Large Language Model for Translation-Related Tasks." In *1st Conference on Language Modeling*.

Arivazhagan, N., Bapna, A., Firat, O., Aharoni, R., Johnson, M., and Macherey, W. (2019a). "The Missing Ingredient in Zero-Shot Neural Machine Translation." *CoRR*, **abs/1903.07091**.

Arivazhagan, N., Bapna, A., Firat, O., Lepikhin, D., Johnson, M., Krikun, M., Chen, M. X., Cao, Y., Foster, G., Cherry, C., Macherey, W., Chen, Z., and Wu, Y. (2019b). "Massively Multilingual Neural Machine Translation in the Wild: Findings and Challenges." *CoRR*, **abs/1907.05019**.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). "Language Models are Few-Shot Learners." In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (Eds.), *Advances in Neural Information Processing Systems*, Vol. 33, pp. 1877–1901. Curran Associates, Inc.

Bu, M., Gu, S., and Feng, Y. (2024). "Improving Multilingual Neural Machine Translation by Utilizing Semantic and Linguistic Features." In Ku, L.-W., Martins, A., and Srikumar, V. (Eds.), *Findings of the Association for Computational Linguistics ACL 2024*, pp. 10410–10423, Bangkok, Thailand and Virtual Meeting. Association for Computational Linguistics.

Cao, Z., Qu, Z., Kamigaito, H., and Watanabe, T. (2024). "Exploring Intrinsic Language-specific Subspaces in Fine-tuning Multilingual Neural Machine Translation." In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N. (Eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 21142–21157, Miami, Florida, USA. Association for Computational Linguistics.

Chen, L., Ma, S., Zhang, D., Wei, F., and Chang, B. (2023). "On the Off-Target Problem of Zero-Shot Multilingual Neural Machine Translation." In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 9542–9558, Toronto, Canada. Association for

Computational Linguistics.

Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., Gao, J., Zhou, M., and Hon, H.-W. (2019). "Unified Language Model Pre-training for Natural Language Understanding and Generation." In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (Eds.), *Advances in Neural Information Processing Systems*, Vol. 32. Curran Associates, Inc.

Edunov, S., Ott, M., Auli, M., and Grangier, D. (2018). "Understanding Back-Translation at Scale." In Riloff, E., Chiang, D., Hockenmaier, J., and Tsujii, J. (Eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 489–500, Brussels, Belgium. Association for Computational Linguistics.

Eriguchi, A., Xie, S., Qin, T., and Hassan, H. (2022). "Building Multilingual Machine Translation Systems That Serve Arbitrary XY Translations." In Carpuat, M., de Marneffe, M.-C., and Meza Ruiz, I. V. (Eds.), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 600–606, Seattle, United States. Association for Computational Linguistics.

Fan, A., Bhosale, S., Schwenk, H., Ma, Z., El-Kishky, A., Goyal, S., Baines, M., Celebi, O., Wenzek, G., Chaudhary, V., Goyal, N., Birch, T., Liptchinsky, V., Edunov, S., Grave, E., Auli, M., and Joulin, A. (2021). "Beyond English-centric Multilingual Machine Translation." *The Journal of Machine Learning Research*, **22** (1).

Galiano-Jiménez, A., Pérez-Ortiz, J. A., Sánchez-Martínez, F., and Sánchez-Cartagena, V. M. (2025). "Beyond the Mode: Sequence-Level Distillation of Multilingual Translation Models for Low-Resource Language Pairs." In Chiruzzo, L., Ritter, A., and Wang, L. (Eds.), *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 6661–6676, Albuquerque, New Mexico. Association for Computational Linguistics.

Gao, Y., Herold, C., Yang, Z., and Ney, H. (2022). "Is Encoder-Decoder Redundant for Neural Machine Translation?" In He, Y., Ji, H., Li, S., Liu, Y., and Chang, C.-H. (Eds.), *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 562–574, Online only. Association for Computational Linguistics.

Goyal, N., Gao, C., Chaudhary, V., Chen, P.-J., Wenzek, G., Ju, D., Krishnan, S., Ranzato, M., Guzman, F., and Fan, A. (2021). "The FLORES-101 Evaluation Benchmark for Low-Resource and Multilingual Machine Translation." *CoRR*, **abs/2106.03193**.

Gu, J., Wang, Y., Cho, K., and Li, V. O. (2019). "Improved Zero-shot Neural Machine Transla-

tion via Ignoring Spurious Correlations." In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1258–1268, Florence, Italy. Association for Computational Linguistics.

Hinton, G., Vinyals, O., and Dean, J. (2015). "Distilling the Knowledge in a Neural Network." *CoRR*, **abs/1503.02531**.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2022). "LoRA: Low-Rank Adaptation of Large Language Models." In *International Conference on Learning Representations*.

Hudi, F., Qu, Z., Kamigaito, H., and Watanabe, T. (2024). "Disentangling Pretrained Representation to Leverage Low-Resource Languages in Multilingual Machine Translation." In Calzolari, N., Kan, M.-Y., Hoste, V., Lenci, A., Sakti, S., and Xue, N. (Eds.), *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 4978–4989, Torino, Italia. ELRA and ICCL.

Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F., Wattenberg, M., Corrado, G., Hughes, M., and Dean, J. (2017). "Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation." *Transactions of the Association for Computational Linguistics*, **5**, pp. 339–351.

Kingma, D. P. and Ba, J. (2017). "Adam: A Method for Stochastic Optimization." *CoRR*, **abs/1412.6980**.

Koehn, P. (2004). "Statistical Significance Tests for Machine Translation Evaluation." In Lin, D. and Wu, D. (Eds.), *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pp. 388–395, Barcelona, Spain. Association for Computational Linguistics.

Kudo, T. (2018). "Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates." In Gurevych, I. and Miyao, Y. (Eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 66–75, Melbourne, Australia. Association for Computational Linguistics.

Kudo, T. and Richardson, J. (2018). "SentencePiece: A Simple and Language Independent Subword Tokenizer and Detokenizer for Neural Text Processing." In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 66–71, Brussels, Belgium. Association for Computational Linguistics.

Liu, D., Niehues, J., Cross, J., Guzmán, F., and Li, X. (2021). "Improving Zero-Shot Translation by Disentangling Positional Information." In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on*

*Natural Language Processing (Volume 1: Long Papers)*, pp. 1259–1273, Online. Association for Computational Linguistics.

Moslem, Y., Haque, R., Kelleher, J. D., and Way, A. (2023). "Adaptive Machine Translation with Large Language Models." In Nurminen, M., Brenner, J., Koponen, M., Latomaa, S., Mikhailov, M., Schierl, F., Ranasinghe, T., Vanmassenhove, E., Vidal, S. A., Aranberri, N., Nunziatini, M., Escartín, C. P., Forcada, M., Popovic, M., Scarton, C., and Moniz, H. (Eds.), *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pp. 227–237, Tampere, Finland. European Association for Machine Translation.

Mullov, C., Pham, Q., and Waibel, A. (2024). "Decoupled Vocabulary Learning Enables Zero-Shot Translation from Unseen Languages." In Ku, L.-W., Martins, A., and Srikumar, V. (Eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6693–6709, Bangkok, Thailand. Association for Computational Linguistics.

NLLB Team (2022). "No Language Left Behind: Scaling Human-Centered Machine Translation." *CoRR*, **abs/2207.04672**.

OpenAI (2024). "GPT-4 Technical Report." *CoRR*, **abs/2303.08774**.

Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., and Auli, M. (2019). "fairseq: A Fast, Extensible Toolkit for Sequence Modeling." In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pp. 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Pan, X., Wang, M., Wu, L., and Li, L. (2021). "Contrastive Learning for Many-to-many Multilingual Neural Machine Translation." In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 244–258, Online. Association for Computational Linguistics.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). "Bleu: a Method for Automatic Evaluation of Machine Translation." In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Popović, M. (2015). "chrF: Character n-gram F-score for Automatic MT Evaluation." In Bojar, O., Chatterjee, R., Federmann, C., Haddow, B., Hokamp, C., Huck, M., Logacheva, V., and Pecina, P. (Eds.), *Proceedings of the 10th Workshop on Statistical Machine Translation*, pp. 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Popović, M. (2017). "chrF++: Words Helping Character n-grams." In Bojar, O., Buck, C., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huck, M., Yepes, A. J., Koehn, P., and Kreutzer, J. (Eds.), *Proceedings of the 2nd Conference on Machine Translation*, pp. 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Post, M. (2018). "A Call for Clarity in Reporting BLEU Scores." In *Proceedings of the 3rd Conference on Machine Translation: Research Papers*, pp. 186–191, Brussels, Belgium. Association for Computational Linguistics.

Proietti, L., Perrella, S., and Navigli, R. (2025). "Has Machine Translation Evaluation Achieved Human Parity? The Human Reference and the Limits of Progress." In Che, W., Nabende, J., Shutova, E., and Pilehvar, M. T. (Eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 790–813, Vienna, Austria. Association for Computational Linguistics.

Qu, Z., Ding, C., and Watanabe, T. (2025). "Languages Transferred Within the Encoder: On Representation Transfer in Zero-Shot Multilingual Translation." In Bouillon, P., Gerlach, J., Girletti, S., Volkart, L., Rubino, R., Sennrich, R., Farinha, A. C., Gaido, M., Daems, J., Kenny, D., Moniz, H., and Szoc, S. (Eds.), *Proceedings of Machine Translation Summit XX: Volume 1*, pp. 81–98, Geneva, Switzerland. European Association for Machine Translation.

Qu, Z., Wang, Y., Ding, C., Tanaka, H., Utiyama, M., and Watanabe, T. (2024). "Improving Language Transfer Capability of Decoder-only Architecture in Multilingual Neural Machine Translation." *CoRR*, **abs/2412.02101**.

Qu, Z., Wang, Y., Mao, J., Ding, C., Tanaka, H., Utiyama, M., and Watanabe, T. (2025). "Registering Source Tokens to Target Language Spaces in Multilingual Neural Machine Translation." In Che, W., Nabende, J., Shutova, E., and Pilehvar, M. T. (Eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21687–21706, Vienna, Austria. Association for Computational Linguistics.

Qu, Z. and Watanabe, T. (2022). "Adapting to Non-Centered Languages for Zero-shot Multilingual Translation." In *Proceedings of the 29th International Conference on Computational Linguistics*, pp. 5251–5265, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). "Improving Language Understanding by Generative Pre-training." *OpenAI*.

Rei, R., C. de Souza, J. G., Alves, D., Zerva, C., Farinha, A. C., Glushkova, T., Lavie, A., Coheur, L., and Martins, A. F. T. (2022). "COMET-22: Unbabel-IST 2022 Submission for the Metrics Shared Task." In Koehn, P., Barrault, L., Bojar, O., Bougares, F., Chatterjee,

R., Costa-jussà, M. R., Federmann, C., Fishel, M., Fraser, A., Freitag, M., Graham, Y., Grundkiewicz, R., Guzman, P., Haddow, B., Huck, M., Jimeno Yepes, A., Kocmi, T., Martins, A., Morishita, M., Monz, C., Nagata, M., Nakazawa, T., Negri, M., Névéol, A., Neves, M., Popel, M., Turchi, M., and Zampieri, M. (Eds.), *Proceedings of the 7th Conference on Machine Translation (WMT)*, pp. 578–585, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Rei, R., Stewart, C., Farinha, A. C., and Lavie, A. (2020). "COMET: A Neural Framework for MT Evaluation." In Webber, B., Cohn, T., He, Y., and Liu, Y. (Eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2685–2702, Online. Association for Computational Linguistics.

Rios, A., Müller, M., and Sennrich, R. (2020). "Subword Segmentation and a Single Bridge Language Affect Zero-Shot Neural Machine Translation." In Barrault, L., Bojar, O., Bougares, F., Chatterjee, R., Costa-jussà, M. R., Federmann, C., Fishel, M., Fraser, A., Graham, Y., Guzman, P., Haddow, B., Huck, M., Yepes, A. J., Koehn, P., Martins, A., Morishita, M., Monz, C., Nagata, M., Nakazawa, T., and Negri, M. (Eds.), *Proceedings of the 5th Conference on Machine Translation*, pp. 528–537, Online. Association for Computational Linguistics.

Stap, D., Niculae, V., and Monz, C. (2023). "Viewing Knowledge Transfer in Multilingual Machine Translation Through a Representational Lens." In Bouamor, H., Pino, J., and Bali, K. (Eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 14973–14987, Singapore. Association for Computational Linguistics.

Sun, Z., Liu, Y., Meng, F., Xu, J., Chen, Y., and Zhou, J. (2024). "LCS: A Language Converter Strategy for Zero-Shot Neural Machine Translation." In Ku, L.-W., Martins, A., and Srikumar, V. (Eds.), *Findings of the Association for Computational Linguistics ACL 2024*, pp. 9201–9214, Bangkok, Thailand and Virtual Meeting. Association for Computational Linguistics.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). "Sequence to Sequence Learning with Neural Networks." In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pp. 3104–3112, Cambridge, MA, USA. MIT Press.

Tan, S. and Monz, C. (2023). "Towards a Better Understanding of Variations in Zero-Shot Neural Machine Translation Performance." In Bouamor, H., Pino, J., and Bali, K. (Eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13553–13568, Singapore. Association for Computational Linguistics.

van der Maaten, L. and Hinton, G. (2008). "Visualizing Data using t-SNE." *Journal of Machine Learning Research*, **9** (86), pp. 2579–2605.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). "Attention Is All You Need." In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (Eds.), *Advances in Neural Information Processing Systems*, Vol. 30. Curran Associates, Inc.

Wang, T., Roberts, A., Hesslow, D., Scao, T. L., Chung, H. W., Beltagy, I., Launay, J., and Raffel, C. (2022). "What Language Model Architecture and Pretraining Objective Work Best for Zero-Shot Generalization?" *CoRR*, **abs/2204.05832**.

Wang, Y., Zhong, W., Li, L., Mi, F., Zeng, X., Huang, W., Shang, L., Jiang, X., and Liu, Q. (2023). "Aligning Large Language Models with Human: A Survey." *CoRR*, **abs/2307.12966**.

Wu, D., Tan, S., Meng, Y., Stap, D., and Monz, C. (2024). "How Far Can 100 Samples Go? Unlocking Zero-Shot Translation with Tiny Multi-Parallel Data." In Ku, L.-W., Martins, A., and Srikumar, V. (Eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 15092–15108, Bangkok, Thailand. Association for Computational Linguistics.

Wu, L., Cheng, S., Wang, M., and Li, L. (2021). "Language Tags Matter for Zero-Shot Neural Machine Translation." In Zong, C., Xia, F., Li, W., and Navigli, R. (Eds.), *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 3001–3007, Online. Association for Computational Linguistics.

Xu, H., Kim, Y. J., Sharaf, A., and Awadalla, H. H. (2024). "A Paradigm Shift in Machine Translation: Boosting Translation Performance of Large Language Models." In *The 12th International Conference on Learning Representations*.

Xu, H., Murray, K., Koehn, P., Hoang, H., Eriguchi, A., and Khayrallah, H. (2025). "X-ALMA: Plug & Play Modules and Adaptive Rejection for Quality Translation at Scale." In *The 13th International Conference on Learning Representations*.

Yang, W., Li, C., Zhang, J., and Zong, C. (2023). "BigTranslate: Augmenting Large Language Models with Multilingual Translation Capability over 100 Languages." *CoRR*, **abs/2305.18098**.

Zhang, B., Bapna, A., Sennrich, R., and Firat, O. (2021). "Share or Not? Learning to Schedule Language-Specific Capacity for Multilingual Translation." In *International Conference on Learning Representations*.

Zhang, B., Ghorbani, B., Bapna, A., Cheng, Y., Garcia, X., Shen, J., and Firat, O. (2022). "Examining Scaling and Transfer of Language Model Architectures for Machine Translation." In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (Eds.), *Proceedings of the 39th International Conference on Machine Learning*, Vol. 162 of *Proceedings*

*of Machine Learning Research*, pp. 26176–26192. PMLR.

Zhang, B., Williams, P., Titov, I., and Sennrich, R. (2020). "Improving Massively Multilingual Neural Machine Translation and Zero-Shot Translation." In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1628–1639, Online. Association for Computational Linguistics.

Zhu, W., Liu, H., Dong, Q., Xu, J., Huang, S., Kong, L., Chen, J., and Li, L. (2024). "Multilingual Machine Translation with Large Language Models: Empirical Results and Analysis." In Duh, K., Gomez, H., and Bethard, S. (Eds.), *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 2765–2781, Mexico City, Mexico. Association for Computational Linguistics.

Zou, W., Yang, S., Bao, Y., Huang, S., Chen, J., and Cheng, S. (2025). "TRANS-ZERO: Self-Play Incentivizes Large Language Models for Multilingual Translation Without Parallel Data." In Che, W., Nabende, J., Shutova, E., and Pilehvar, M. T. (Eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 12337–12347, Vienna, Austria. Association for Computational Linguistics.

# Appendix

## A Details of Evaluation Metrics

In evaluating models trained on EC-40, some languages are not supported by COMET (*Unbabel/wmt22-comet-da*) or the off-target ratio (*fast-langdetect*). Since *fast-langdetect* relies on word-level recognition, we also exclude several languages that show low recognition success rates. For clarity, Table 7 summarizes the supported languages for each metric, grouped by their resource tiers in EC-40.

## B Training Details of Fine-tuning

**Selecting Directions** We use *random.sample* in Python to randomly select translation directions for fine-tuning by setting the random seed to 0 for reproducibility. We define three scenarios of 5, 25, and 100 translation directions. It is important to note that *random.sample* draws the 5 and 25 directions as subsets of the 100 directions. Specifically, the first 5 and first 25 directions in the 100-direction set correspond to the other two scenarios. The 100 directions are: `jv`→`sv`,

| Resource Tier | Supported Languages |
|---|---|
| High | en, es, hi, de, cs, nl, ar, fr, ru, he, bn |
| Med | bg, da, pt, mr, ha, sv, it, pl, kn |
| Low | uk, am, ro, gu, sr, sd, af |
| Extra-Low | so, ca, be, bs, no, is, ne, ur |

**(a) COMET (*Unbabel/wmt22-comet-da*)**

| Resource Tier | Supported Languages |
|---|---|
| High | en, es, hi, de, cs, nl, ar, fr, ru, he, bn |
| Med | bg, da, pt, mr, sv, it, pl, kn, mt |
| Low | uk, ro, gu, am, sd |
| Extra-Low | ur, be, is |

**(b) Off-target Ratio (*fast-langdetect*)**

**Table 7** Languages supported by COMET and the off-target ratio, grouped by EC-40 resource tiers. Table 8 lists all EC-40 languages.

ms→id, de→tl, ru→pl, ko→jv, zh→bg, ms→en, pl→ru, zh→af, uk→ko, pt→jv, ko→ro, fr→da, cs→pl, fr→af, da→fr, ru→sv, fr→pl, pl→tl, da→ro, sv→es, bg→jv, zh→en, da→cs, uk→ms, tl→es, bg→de, pt→nl, vi→bg, tl→id, ru→bg, nl→ms, en→uk, da→sv, jv→ms, en→nl, zh→vi, bg→ja, ro→ja, bg→ru, nl→tl, vi→es, ja→pt, cs→uk, da→ko, af→it, jv→zh, zh→cs, sv→da, ko→pt, cs→nl, pt→vi, nl→en, vi→ja, es→nl, tl→ru, ru→es, ja→jv, ro→zh, nl→ro, fr→jv, cs→fr, fr→cs, uk→jv, ko→bg, cs→da, es→ro, ms→sv, ja→cs, cs→en, da→pl, jv→tl, pl→pt, zh→sv, pl→de, fr→ro, pt→zh, zh→id, pl→fr, ko→ru, it→bg, es→de, cs→tl, af→pt, fr→ru, da→nl, da→af, ms→fr, ko→cs, en→jv, pl→uk, bg→uk, af→tl, ro→bg, de→pl, de→vi, uk→nl, id→ja, nl→zh, zh→pl

**Fine-tuning Configurations** All fine-tuning experiments use the same settings. We conduct experiments on 8 Tesla V100 GPUs, with a maximum of 1,024 tokens per GPU and a gradient accumulation of 2 steps. Based on the pre-trained model, we set the learning rate to 0.0001 with a warmup step of 1 (for launching the inverse square root schedule), and train for 10 epochs. Finally, we use the last epoch for testing.

**LoRA Configurations** We adopt the setting of Hu et al. (2022) to implement LoRA components for pre-trained models. Specifically, LoRA is only implemented for Query and Value in the attention mechanism with a rank of 8. As a result, the learnable parameters of NLLB series are 1.18M, 2.36M, and 4.72M, respectively, and the learnable parameters of MITRE series are 0.78M and 1.47M, respectively.

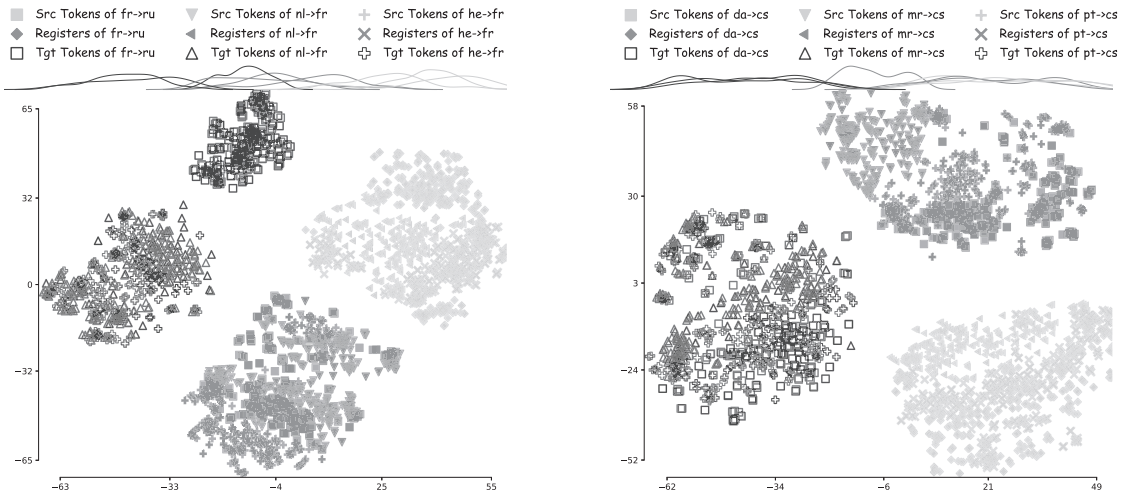
## C Setups and Prompts for GPT

We call the API of GPT series with default parameters. Our prompts for GPT series follow: Translating the following sentence from [SRC] to [TGT]: [INPUT]. Here, [SRC] and [TGT] are

the source and target language names following Table 2, and [INPUT] is the source sentence. We find that GPT occasionally repeats [INPUT] in the output. Once it happens, we manually remove the [INPUT] before evaluation.

## D Additional Scenarios for Token Representation Distribution

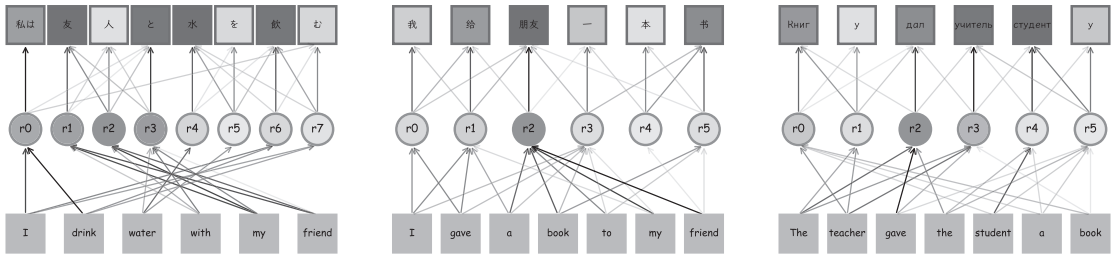
In Section 6.2 and Figure 6, we compared three randomly selected translation directions, each with a different target language. To further validate our findings, we present two additional scenarios in this section. Specifically, in Figure 8a, we select three random translation directions, where two share the same target language and one has a different target. The resulting distribution shows that register representations corresponding to the same target language form overlapping clusters, while the one associated with a different target language is clearly separated in the representation space. In Figure 8b, all three directions share the same target language. In this case, all register representations converge into a single, unified cluster. These observations reinforce the conclusion drawn in Section 6.2, namely, the register representation is located in the intended target language space.




(a) 2 target languages in 3 directions

(b) Same target language in 3 directions

**Figure 8** 2D distributions of token-level representations extracted from the output layer, i.e., 24th layer, of a model trained on EC-40. The illustration process keeps the same as Figure 6.




(a) `ja`→`en` (b) `zh`→`en` (c) `ru`→`en`

**Figure 9** Three cases of attention analysis on MITRE-466M. Details of this illustration follow that of Figure 7.

## E Additional Cases for Attention Analysis

To further support our analysis in Section 6.2, we examine additional cases in MITRE-466M where the source and target sentences exhibit significant structural differences. In all cases, the attention relationship between registers and source tokens remains consistent, i.e., one-to-one attention weights being the most prominent. Moreover, similar to the case of "die" in Figure 7, the token "を" in Figure 9a, which does not receive the highest attention weight from the register, functions as a grammatical particle. Likewise, in Figure 9c, the token "y" that fails to receive the highest weight is a suffix split from a content word. These observations further support our conclusion. Next, we observe the following patterns: (1) As shown in Figure 9a, in Japanese, the attention for "drink" points to $r_6$, while "friend" points to $r_1$ and $r_2$. (2) As shown in Figure 9b, "friend" points to $r_2$, and "book" points to $r_5$. (3) As shown in Figure 9c, "book" points to $r_0$, and "student" points to $r_4$. Given that the attention weights between registers and target tokens highlight the structural differences between source and target sentences, we can state again that registers mirror the corresponding source tokens.

## F Description of EC-40

EC-40 is an English-centric dataset introduced by Tan and Monz (2023) as a training set to benchmark MNMT-specific methodologies. In addition to English, it includes 40 languages spanning five language families, with each family containing eight languages. These languages are categorized into four tiers based on data availability: High, Medium, Low, and Extra Low. Each non-English language is paired with English, resulting in 80 supervised translation directions used

| | Germanic | | | Romance | | | Slavic | | | Indo-Aryan | | | Afro-Asiatic | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | code | Language | Script | code | Language | Script | code | Language | Script | code | Language | Script | code | Language | Script |
| High | de | German | Latin | fr | French | Latin | ru | Russian | Cyrillic | hi | Hindi | Devanagari | ar | Arabic | Arabic |
| (5 million) | nl | Dutch | Latin | es | Spanish | Latin | cs | Czech | Latin | bn | Bengali | Bengali | he | Hebrew | Hebrew |
| Med | sv | Swedish | Latin | it | Italian | Latin | pl | Polish | Latin | kn | Kannada | Devanagari | mt | Maltese | Latin |
| (1 million) | da | Danish | Latin | pt | Portuguese | Latin | bg | Bulgarian | Cyrillic | mr | Marathi | Devanagari | ha | Hausa* | Latin |
| Low | af | Afrikaans | Latin | ro | Romanian | Latin | uk | Ukrainian | Cyrillic | sd | Sindhi | Arabic | ti | Tigrinya | Ethiopic |
| (100 thousand) | lb | Luxembourgish | Latin | oc | Occitan | Latin | sr | Serbian | Latin | gu | Gujarati | Devanagari | am | Amharic | Ethiopic |
| Extra Low | no | Norwegian | Latin | ast | Asturian | Latin | be | Belarusian | Cyrillic | ne | Nepali | Devanagari | kab | Kabyle* | Latin |
| (50 thousand) | ic | Icelandic | Latin | ca | Catalan | Latin | bs | Bosnian | Latin | ur | Urdu | Arabic | so | Somali | Latin |

**Table 8** Details of non-English languages in EC-40. This table is duplicated from the original paper. Numbers in the table represent the number of sentences paired with English. Two exceptions are Hausa and Kabyle, where their data sizes are 334,000 and 18,448, respectively.

for training and 1,560 zero-shot translation directions. Details of this dataset are summarized in Table 8.

**Zhi Qu**: Ph.D. candidate at the Nara Institute of Science and Technology (NAIST). He received his Master's degree from NAIST in 2023 and his Bachelor's degree from Chongqing Normal University, China, in 2018. He also works as a technical researcher at Advanced Translation Technology Laboratory of National Institute of Information and Communications Technology (NICT), Japan.

**Yiran Wang**: The researcher of Advanced Translation Technology Laboratory of NICT, Japan. He received his Ph.D. degree from NAIST, Japan.

**Jiannan Mao**: The researcher of Advanced Translation Technology Laboratory of NICT, Japan. He received his Ph.D. degree from Gifu University, Japan.

**Jin Tei (a.k.a. Chenchen Ding)**: He is a senior researcher of the Advanced Translation Technology Laboratory of NICT, Japan, since 2022. He is also an associate professor of the Multilingual Natural Language Processing Laboratory, a collaborative laboratory between NICT and NAIST, from 2024. He received an M.E. and Ph.D. from the University of Tsukuba in 2012 and 2015, respectively.

**Hideki Tanaka**: He is a Research Executive Director of the Advanced Translation Technology Laboratory of NICT in Japan. He received a BS., MS., and Ph.D. from Kyushu University in 1982, 1984, and 1995, respectively.

**Masao Utiyama**: He is the director of the Advanced Translation Technology Laboratory, NICT, Japan, since 2023. He received his Ph.D. degree from the

University of Tsukuba in 1997. His main research field is machine translation.

**Taro Watanabe**: Taro Watanabe received his B.E. and M.E. degrees in information science from Kyoto University in 1994 and 1997, respectively, and obtained an M.S. degree in Language and Information Technologies from the School of Computer Science, Carnegie Mellon University in 2000. In 2004, he received a Ph.D. in informatics from Kyoto University. After working as a researcher at ATR, NTT and NICT, and as a software engineer at Google, he is a professor at the Nara Institute of Science and Technology starting in 2020. His research interests include natural language processing, machine learning, language modeling and machine translation.

(Received October 29, 2025)
(Revised January 8, 2026)
(Accepted March 5, 2026)