# Mixing and Matching: Combining Independently Trained Translation Model Components

**Taido Purason** and **Andre Tättar** and **Mark Fishel**
University of Tartu, Estonia
{taido,andre,mark}@tartunlp.ai

## Abstract

This paper investigates how to combine encoders and decoders of different independently trained NMT models. Combining encoders/decoders is not directly possible since the intermediate representations of any two independent NMT models are different and cannot be combined without modification. To address this, firstly, a dimension adapter is added if the encoder and decoder have different embedding dimensionalities, and secondly, representation adapter layers are added to align the encoder's representations for the decoder to process. As a proof of concept, this paper looks at many-to-Estonian translation and combines a massively multilingual encoder (NLLB) and a high-quality language-specific decoder. The paper successfully demonstrates that the sentence representations of two independent NMT models can be made compatible without changing the pre-trained components while keeping translation quality from deteriorating. Results show significant improvements in both translation quality and speed for many-to-one translation over the baseline multilingual model.

## 1 Introduction

As the availability of pre-trained models continuously increases, there is a growing need to investigate how to use them efficiently. Previous works have looked at effectively using pre-trained neural machine translation (NMT) models by effective fine-tuning (Bapna and Firat, 2019; Zhu et al., 2021) as well as using pre-trained language models in NMT model training (Zhu et al., 2020; Rothe et al., 2020; Chen et al., 2021; Sun et al., 2021; Chen et al., 2022).

This paper examines the feasibility of combining together components (like encoders and decoders) of independent pre-trained NMT models without any retraining or fine-tuning. We investigate how representations of independently trained models can be made compatible and evaluate the resulting translation quality and efficiency. Surprisingly, our evaluation shows that the resulting combined model can surpass the original models in translation quality and speed.

Combining any pre-trained encoder and decoder poses two problems. Firstly, their representation spaces will not be compatible, as the models are trained independently. Secondly, the embedding dimension of the representation can also differ across any two pre-trained models. We propose a method that solves both issues and allows the encoder and decoder of any pre-trained NMT models to be combined. Specifically, in our architecture (Figure 1), we use a small adapter to convert the dimensionality and representation space of the encoder to something the decoder is trained to process. In order for the adapter to learn its weights, the whole pipeline (Encoder A - adapter - Decoder B) is trained in an end-to-end fashion, except both the encoder and decoder are frozen. Thus, the only part changing the weights is the adapter itself while the original components remain intact.

As a proof of concept, we investigate combining encoders and decoders of multiple different pre-trained NMT models, focusing on an output language-specific scenario. In other words, a highly multilingual encoder is combined with a monolingual decoder, tuned to high performance on a single

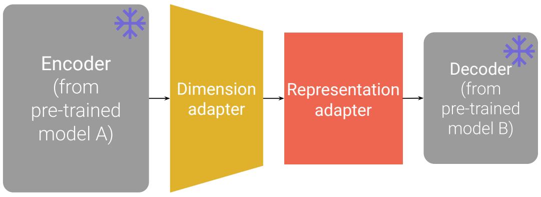

Figure 1: The proposed mix-and-match architecture. Dimension adapter is a component that takes input with the dimensionality of model A output and outputs with the dimensionality of model B (for example a linear transformation). Adapter layers are transformer encoder layers. Components from models A and B have frozen parameters.

*Proceedings of the 1st Workshop on Modular and Open Multilingual NLP (MOOMIN 2024)*, pages 44–56
March 21, 2024 ©2024 Association for Computational Linguistics

language. Since highly multilingual models often suffer from the capacity bottleneck (Johnson et al., 2017; Tan et al., 2019; Arivazhagan et al., 2019), we hypothesize that adding a high-quality language-specific decoder can improve the translation quality to the language of the decoder. Furthermore, translation to one language requires less capacity than many-to-many scenarios and thus would potentially require fewer parameters, resulting in faster translation.

Using NLLB (Team et al., 2022) as the multilingual model and MTee (Tättar et al., 2022) as the language-specific Estonian model, we demonstrate significant improvements in translation quality over the baseline NMT model for many-to-Estonian translation and show competitive results to pivoting and fine-tuning. Our method is not only effective to train compared to traditional fine-tuning but also provides a reduction in running costs of the translation model thanks to the number of parameters being reduced by 40% compared to the baseline NLLB model.

The main contributions of this work are:

- a novel method for combining pre-trained NMT models, which improves translation quality, is effective to train, and reduces the model's parameters (Section 3);
- a detailed ablation of the proposed method, exploring the effect of freezing or unfreezing different involved components, comparing simpler and more complicated adapter architectures, and involving more source languages in training (Section 4);
- an open-source implementation of our proposed method (see subsection 3.5).

## 2 Related Work

To the best of our knowledge, creating new NMT models by connecting encoders and decoders of different pre-trained NMT models has not been explored yet. Similar approaches have been tested in speech translation (Li et al., 2021; Gállego et al., 2021). Similarity between independently learned representations has been explored between linguistic, image representations as well as brain waves (Søgaard, 2023; Li et al., 2023), however we attempt direct conversion and exploitation of these representations.

### 2.1 Pre-trained NMT models

There are many pre-trained NMT models already openly available for use. OpusMT provides over 1000 NMT models, most of which are bilingual, but some also multilingual (Tiedemann and Thottingal, 2020). Rothe et al. (2020) published NMT models which were initialized from BERT and trained on the NMT task. M2M-100 is a series of NMT models (varying in size) which were trained on 7.5B sentence pairs and support translation between 100 languages (Fan et al., 2020). The NLLB-200 NMT model further improves it and extends support to 200 languages with a training dataset of 18B sentence pairs (Team et al., 2022). Both M2M-100 and NLLB-200 are strong baselines in NMT research regarding translation quality. MTee provides an Estonian-centric (Estonian to/from English, German, Russian) NMT model with language-specific encoders-decoders (Tättar et al., 2022). The most recent contribution to massively multilingual models is MADLAD-400 (Kudugunta et al., 2023), with both decoder-only as well as sequence-to-sequence models with both the encoder and decoder released. Finally, large multilingual language models like GPT-3 and GPT-4 have demonstrated an ability to translate (Brown et al., 2020; Bubeck et al., 2023), however they only demonstrate highly competitive quality for high-resource languages.

### 2.2 Multilingual NMT

Recently, there have been numerous advancements in multilingual NMT. One of the most widely followed approaches is demonstrated by Johnson et al. (2017), where they use a single (universal) model with shared vocabulary for multilingual NMT, which enables transfer learning and zero-shot translation. Massively multilingual training has since been successfully demonstrated (Aharoni et al., 2019; Arivazhagan et al., 2019; Zhang et al., 2020). Additionally, fine-tuning methods of NMT models have been investigated, including lightweight fine-tuning methods such as adapters (Bapna and Firat, 2019; Zhu et al., 2021). In addition to universal models, there has been successful research into modular multilingual NMT using language-specific encoders and decoders (Escolano et al., 2021; Lyu et al., 2020). As an alternative to supporting all directions in the models, pivoting (translating through a pivot language) has also been used as a method for achieving higher quality multilingual translation (Habash and Hu, 2009).

### 2.3 Pre-trained Language Models for NMT

With many pre-trained language models (LMs) becoming available, making use of them in NMT has become an important topic.

The first line of works takes the approach of pre-training an encoder-decoder model for seq2seq tasks and then fine-tuning the model for MT, for example, mBART (Liu et al., 2020), and MASS (Song et al., 2019).

In the second approach, the encoder or the decoder can be trained independently and later used in an NMT model. Zhu et al. (2020) incorporates input sentence representations into an NMT model. Rothe et al. (2020) initializes NMT model's encoder and/or decoder weights from pre-trained language models. SixT (Chen et al., 2021) used XLM-R as the pre-trained encoder in combination with a randomly initialized decoder, trained using 2-stage training where first the decoder is trained (rest of the model frozen) and secondly, the rest of the model is tuned. This was further improved and expanded in SixT+ (Chen et al., 2022). Sun et al. (2021) combined a BERT-like encoder and a GPT-like decoder into a single model by adding extra layers to both the encoder and decoder.

Ma et al. (2021) uses aspects of both approaches by initializing an encoder-decoder model from an encoder-only language model and pre-training on seq2seq tasks before fine-tuning for MT.

Li et al. (2021) combines a pre-trained audio encoder and pre-trained decoder from mBART to create a speech translation model through fine-tuning.

## 3 Approach and Setup

### 3.1 Methodology

Our approach combines two pre-trained NMT models using an adapter placed "between" the encoder and decoder: see Figure 1). The adapter consists of a dimension adapter and representation adapter.

The dimension adapter is a linear transformation (feed-forward layer) with input dimensionality equal to the encoder embedding dimension and the output dimensionality to the decoder embedding dimension. We place the dimension adapter directly after the pre-trained encoder.

Representation adapter layers are implemented as randomly initialized transformer layers. They have the same embedding dimension as the decoder. We do not modify the decoder by adding extra layers or other parameters; thus it is kept lightweight, leading to fast translation using beam search since encoder embeddings are calculated once for a sentence, but the decoder is used repeatedly.

**Training:** when training the model, the adapter learns with the rest of the components in an end-to-end fashion. Training examples are passed through the whole pipeline (encoder, then adapter, then decoder), however both the encoder and decoder remain frozen. Thus the only weights that are allowed to change are the parts of the adapter.

We also perform reverse-ablation and compare our original approach of freezing all but the adapter to less efficient alternatives of also letting the decoder tune itself during training, randomly initializing the decoder as well as tuning the whole model. A combination of the originally proposed approach (tuning only the adapter) and then continuing training the adapter and an unfrozed pre-initialized decoder will be referred to as the 2-stage approach.

### 3.2 Translation models

We rely on *NLLB-1B-distilled* as the pre-trained model for encoders in our experiments (referred to in the further text as NLLB-1B or NLLB); Section 4.3.3 also includes a comparison to *NLLB-600M-distilled* as the base model. For the decoder, we use the Estonian decoder from MTee (Tättar et al., 2022) – a modular model with language-specific encoders and decoders (encoders/decoders follow transformer base architecture (Vaswani et al., 2017)).

The pre-trained NLLB-1B encoder has 24 layers with an embedding dimension of 1024 and a feed-forward dimension of 8192. In the main experiments, we add a linear dimension adapter that transforms the embedding dimension from 1024 to 512 and 4 representation adapter layers with the same embedding and feed-forward dimension as the decoder (512 and 2048 respectively) to the encoder.

### 3.3 Dataset

We use English-Estonian (22M, sentence pairs), German-Estonian (12.5M sentence pairs), French-Estonian (11.7M sentence pairs), and Polish-Estonian (7M sentence pairs) directions from CC-Matrix (Schwenk et al., 2019). In Ablation Section 4.3.3 we use Europarl (Tiedemann, 2012).

We use SentencePiece (SP) (Kudo and Richardson, 2018) models from the respective pre-trained NMT models for segmenting the data. For example when we use NLLB encoder and MTee decoder, we use NLLB SP model for processing the source and MTee SP model for processing the target.

The models are evaluated using FLORES-200 (Team et al., 2022) *devtest* as the test set and *dev* as the validation set. The same directions the model is trained on are used for validation. The best checkpoint, according to the validation loss, is used for test set evaluation. Test set evaluation is carried out on all 201 many-to-Estonian directions. We confirmed that the test set was not present in the training data of MTee and also trust that since FLORES-200 was the main test set of NLLB (Team et al., 2022), it would be properly cleaned from their training dataset.

### 3.4 Evaluation

For evaluation we mainly rely on chrF++[1] (Popović, 2017), but also report chrF[2] (Popović, 2015) for comparison with previous research. We use the sacreBLEU (Post, 2018) implementation.

Although BLEU (Papineni et al., 2002) is a widely adopted metric, several evaluation campaigns (Barrault et al., 2021; Koehn et al., 2022) have shown its weaker correlation with human judgements of translation quality compared to chrF/chrF++ and neural metrics like COMET (Rei et al., 2020). However, we still include BLEU scores for comparison in Appendix A. Additionally, we provide COMET scores (Rei et al., 2020) for a selection of languages in Appendix B.

For the main experiments, we conduct 5 random restarts for each model and report the mean score with a confidence interval ($p = 0.01$, t-distribution). We also report the Win Rate with Significance (WRS) – the percentage of language pairs where the model outperforms the baseline (NLLB-1B) with significance $p = 0.01$. The significance is tested using a one-sample one-tailed t-test for experiments with 5 seeds. Additionally, we report WRS based on a single seed with significance calculated with paired bootstrap resampling (PBR) (Koehn, 2004).

### 3.5 Implementation and training

We use Fairseq (Ott et al., 2019) for implementing training. Additionally, we made our specific implementation of training and models public[3].

For the main experiments, all models are trained for a total of 100k updates. If 2-stage training is used, the first stage is trained for 50k updates and the second stage for 50k updates. The learning rate used is 0.0005 for the first stage and 0.0001 for the second stage. We use Adam optimizer (Kingma and Ba, 2015). An inverse square root learning rate scheduler with 4000 warm-up steps is used for all experiments. We use dropout and attention dropout of 0.1. Models are trained with mixed precision (*fp16*). All translations are acquired using beam search with beam size 4.

The models were trained on 8 GPUs for the main experiments. The batch size was 4096 tokens per GPU. The training was performed on the LUMI supercomputer[4], utilizing 4 AMD Instinct MI250X 128GB HBM2e (each acting as 2 GPUs).

## 4 Results

### 4.1 Main Results

The main results are reported in Table 1. *NLLB-1B-distilled* is used as a baseline. Additionally, results of the largest publicly available NLLB model (NLLB-MoE) with 54.5B parameters reported by Team et al. (2022) are used for comparison. The table lists average chrF++ scores over all many-to-Estonian translation directions and all official EU languages[5]. The EU language averages are reported to highlight the translation quality for languages more closely related to Estonian and also more frequently translated from. We analyze the quantitative results of pivoting, fine-tuning, and our mixing and matching approach of combining the encoder and the decoder of different pre-trained models.

#### 4.1.1 Pivoting

NLLB-1B English pivoting for many-to-Estonian translation results in an average 1.2 chrF++ point improvement across all directions, significantly outperforming the baseline NLLB-1B model on 84.6% of directions (see (3) in Table 1). When NLLB-1B is used to translate to English and MTee is used for English-to-Estonian translation (see (4) in Table 1), the translation quality is improved by 3.2 chrF++

[1] sacreBLEU signature: `nrefs:1|case:mixed|eff:yes|nc:6|nw:2|space:no|version:2.3.1`

[2] sacreBLEU signature: `nrefs:1|case:mixed|eff:yes|nc:6|nw:0|space:no|version:2.3.1`

[3] https://anonymous.4open.science/r/mix-and-match-nmt

[4] https://www.lumi-supercomputer.eu/

[5] Bulgarian, Croatian, Czech, Danish, Dutch, English, Estonian, Finnish, French, German, Greek, Hungarian, Irish, Italian, Latvian, Lithuanian, Maltese, Polish, Portuguese, Romanian, Slovak, Slovenian, Spanish, and Swedish

| | Model | Parameters | | | Train. | average chrF++ ↑ | | WRS (%) ↑ | |
|---|---|---|---|---|---|---|---|---|---|
| | | train | total | eff. | time | full | EU | t-test | PBR |
| (1) | NLLB-1B | - | 1.37B | 1.37B | - | 40.2 | 46.7 | - | - |
| (2) | NLLB-MoE[†] | - | 54.5B | 54.5B | - | 43.0 | 49.6 | - | 99.5 |
| | **Pivot**, m2en: NLLB-1B | | | | | | | | |
| (3) | en2et NLLB-1B | - | 1.37B | 2.74B | - | 41.4 | 47.5 | - | 84.6 |
| (4) | en2et: MTee | - | 1.42B | 1.42B | - | **43.4** | 50.2 | - | **100.0** |
| | **Fine-tune** NLLB-1B | | | | | | | | |
| (5) | - | 1.37B | 1.37B | 1.37B | 22.3 | 42.5 ± 0.1 | 50.1 ± 0.3 | 91.0 | 86.6 |
| (6) | freeze enc | 604M | 1.37B | 1.37B | 15.0 | 43.0 ± 0.1 | 50.3 ± 0.2 | **98.0** | 98.5 |
| | **Ours:** NLLB-1B enc + | | | | | | | | |
| (7) | rand dec | 51M | 817M | 817M | 4.4 | 42.6 ± 0.3 | 50.2 ± 0.3 | 93.5 | 97.5 |
| (8) | MTee dec | 13M | 817M | 817M | 3.9 | 42.5 ± 0.1 | 50.4 ± 0.1 | 92.0 | 89.1 |
| (9) | MTee dec, 2-stage | 51M | 817M | 817M | 4.1 | 43.1 ± 0.1 | **50.9 ± 0.1** | 93.0 | 96.5 |

Table 1: Many-to-Estonian translation average chrF++ scores. Additionally model training, total and effective parameters and training time (hours) is reported. Effective parameter count represents the number of parameters used during translation. For experiments involving model training, the average of 5 random seeds is reported with confidence intervals ($p = 0.01$). Average chrF++ is reported for all directions and official EU languages separately. WRS (Win Rate with significance, $p = 0.01$) reports what percentage of directions outperform the baseline with both significance based on t-test on 5 seeds and significance based on paired bootstrap resampling t-test (PBR). † - Scores reported by (Team et al., 2022).

points on average compared to the baseline (1), significantly outperforming it on all directions. These results demonstrate that pivoting can enhance translation quality without additional training. However, pivoting requires passing through two models, which increases the time required for translation and reduces long-term cost efficiency.

### 4.1.2 Fine-tuning

We experimented with two different fine-tuning strategies: full fine-tuning (5) and fine-tuning only the decoder of the baseline NLLB model with the encoder frozen (6). We found that both approaches lead to significant improvements over the baseline: 2.3 and 2.8 chrF++ points, respectively. Moreover, fine-tuning exhibited superior performance compared to the baseline across more language pairs, as confirmed by the t-test WRS scores: 98.0% for the frozen encoder method vs. 91.0% for full fine-tuning.

### 4.1.3 Mixing and Matching

When NLLB encoder and MTee decoder are combined with adapter layers, by only training the adapter (13M parameters) and freezing the pre-trained components, the resulting model (NLLB enc + MTee dec model (8)) significantly outperforms the baseline on 92.0% of the directions according to the t-test (89.1% according to PBR), with an average improvement of 2.3 chrF++ points. The 2-stage training approach (9) – training the adapter first (13M parameters), followed by training the adapter with the decoder (51M parameters) – achieved the best results. This method (9) outperforms the baseline by 2.9 chrF++ points on average across all directions and achieves similar average chrF++ scores to the 54B parameter NLLB model. It is only slightly behind the best-performing pivoting model in terms of average chrF++ scores. Additionally, we observed that the 2-stage training approach significantly outperforms the baseline on 93% of the language pairs according to the t-test (96.5% according to the PBR). However, the fine-tuning method with a frozen encoder showed significant improvements over the baseline in 5% more directions than our approach.

We also evaluated a decoder that was randomly initialized with the same architecture and vocabulary as MTee (7), and trained in a single stage with a frozen encoder, only training the adapter and decoder. It outperformed the baseline by 2.4 chrF++ points on average. This method performs similarly to the initialized model with no decoder training. Although it is still slightly outperformed by the 2-stage model with the pre-initialized decoder in terms of the average chrF++ score, it can be useful when a high-quality pre-trained decoder model is unavailable.

Average BLEU scores are presented in Appendix A Table 6, since they support the same conclusions as the chrF++ scores.

| Model | eng_Latn | deu_Latn | rus_Cyrl | zho_Hans | arb_Arab |
|---|---|---|---|---|---|
| NLLB-1B | 52.6 | 48.5 | 46.6 | 40.2 | 45.8 |
| NLLB-MoE† | 56.1 | 51.8 | 49.5 | 43.8 | 49.1 |
| MTee | 56.9 | 52.2 | 49.9 | - | - |
| **Pivot**, m2en: NLLB-1B | | | | | |
| en2et NLLB-1B | 52.6 | 48.7 | 47.2 | 42.4 | 46.8 |
| en2et: MTee | 56.9 | 52.4 | 49.8 | **45.5** | **49.5** |
| **Fine-tune** NLLB-1B | | | | | |
| - | 56.6 ± 0.3 | 52.3 ± 0.5 | 50.1 ± 0.2 | 44.5 ± 0.2 | 48.8 ± 0.2 |
| freeze enc | 56.2 ± 0.4 | 52.3 ± 0.3 | 50.1 ± 0.2 | 44.6 ± 0.2 | 48.8 ± 0.2 |
| **Ours:** NLLB-1B enc + | | | | | |
| rand dec | 56.1 ± 0.4 | 52.0 ± 0.5 | 49.8 ± 0.5 | 44.1 ± 0.3 | 48.6 ± 0.3 |
| MTee dec | 56.7 ± 0.5 | 52.4 ± 0.4 | 49.9 ± 0.3 | 43.5 ± 0.3 | 48.6 ± 0.2 |
| MTee dec 2-stage | **57.3 ± 0.3** | **52.8 ± 0.2** | **50.4 ± 0.3** | 44.6 ± 0.4 | 49.1 ± 0.3 |

Table 2: Many-to-Estonian translation chrF++ scores for selected directions. Confidence intervals are based on 5 random seeds. † - Scores reported by Team et al. (2022). Language abbreviations following Team et al. (2022).

For EU languages, NLLB-enc+MTee-dec, 2-stage (9) achieves the highest average chrF++ score and outperforms the baseline by 4.2 chrF++ points. This shows that our method achieves the best result for more closely related languages, whereas the pivoting approach of combining two models was better for more distant languages. A possible explanation could be the training data being composed of EU languages. Furthermore, the pre-trained decoder was also trained with two EU languages and Russian as input, which could contribute to the high performance on translating EU languages.

In Table 2, we present the chrF++ scores for translations from a selection of languages to Estonian, serving as an example. It also shows the comparison with the MTee model for the languages supported by the pre-trained MTee model. The mix-and-match models (ours) perform similarly to the MTee model, with the 2-stage model outperforming MTee slightly. It can also be seen that for Chinese and Arabic, our approach is outperformed by pivoting with NLLB and MTee. This further suggests that our method produces better translation quality for closer related languages. We also provide COMET scores for these directions in Appendix B, which support mostly the same conclusions, except for NLLB-MoE scores, which rank the highest among the models.

### 4.1.4 Efficiency

The mix-and-match method (NLLB-1B enc. + MTee dec.) reduces the number of parameters by 40% compared to the baseline model and the default fine-tuning approach. Even though we add 13M trainable parameters to the encoder (adapter layers), we use a significantly smaller decoder than NLLB-1B, leading to fewer trained and total parameters. This makes the training time of our method (4.1 hours for NLLB-enc+MTee-dec, 2-stage) 5.4 times faster than the full fine-tuning (22.3 hours). Furthermore, the inference with NLLB-enc+MTee-dec is approximately 6.5 times faster than with NLLB-1B. This demonstrates that our approach offers an efficient and cost-effective alternative to fine-tuning and pivoting that delivers comparable or better translation quality, with the added benefit of faster training (compared to fine-tuning), fewer parameters, and faster inference.

## 4.2 Ukrainian-Estonian Translation

| Model | chrF ↑ |
|---|---|
| NLLB-1B | 50.9 |
| NLLB-MoE† | 54.0 |
| NLLB-MTee EN pivot | 54.5 |
| NLLB-enc+MTee-dec | 54.6 ± 0.2 |
| NLLB-enc+MTee-dec, 2-stage | **55.0 ± 0.1** |
| Bergmanis and Pinnis (2022) | 53.5 |

Table 3: Ukrainian (Cyrillic) to Estonian (Latin) translation chrF scores on FLORES-101 *devtest*. NLLB-1B model was used for all experiments, except for NLLB-MoE (54B). † - calculated from translations reported by (Team et al., 2022).

We demonstrate that without needing Ukrainian-Estonian data, we can rapidly create a model with competitive translation quality. We compare scores of our best model with work by Bergmanis and Pinnis (2022) and report chrF to be compatible with their evaluation. We can see that our best model

(NLLB-enc+MTee-dec, 2-stage) outperforms their Ukrainian to Estonian model by 1.5 chrF points (see Table 3). It also outperforms the NLLB-1B baseline by 4.1 chrF points and achieves a slightly higher score than NLLB-MoE and pivoting with NLLB-1B and MTee.

## 4.3 Ablation

### 4.3.1 Effect of multi-stage training

We look at additional training strategies in addition to training adapter or adapter and decoder. It can be seen in Table 4 that training only the adapter and decoder yields the best results both in single-stage and multi-stage training strategies. Strategies involving encoder training take longer to train due to more trained parameters and do not yield any visible benefit. We can hypothesize that it is because the encoder is already trained for the domain of the test set. We can see that the 2-stage training, which trains the adapter in the first stage and the adapter and decoder in the second stage, produces the best scoring model and is also the second fastest behind the single-stage model, which trains only the adapter. While encoder training did not yield improvements for the current pre-trained models, training and test datasets, it might yield different results if these elements differ. For example, when pre-trained models are trained for a domain different from the training and test datasets, fine-tuning the encoder might be necessary.

| Training setup<br>dec. init. | | stage | Trained<br>params | Time<br>(hrs) | chrF++<br>avg |
|---|---|---|---|---|---|
| | single | | | | |
| random | A+D | | 51M | 4.3 | 42.8 |
| MTee | A+D | | 51M | 4.4 | 42.9 |
| MTee | A | | 13M | 3.8 | 42.4 |
| | I | II | | | |
| random | A+D | E+A+D | 817M | 5.5 | 42.7 |
| MTee | A | A+D | 51M | 4.0 | 43.2 |
| MTee | A | E+A | 779M | 7.5 | 42.1 |
| MTee | A | E+A+D | 817M | 7.2 | 42.8 |

Table 4: Comparison of training strategies. chrF++ scores as calculated on FLORES200 *devtest*. All models listed have 817M total parameters. Trained parameters are based on the last stage and models follow the NLLB-1B+MTee mix-and-match model structure. The stage column describes which parameters are trained. A - dim. adapter and adapter layers, D - decoder, E - encoder. The results are based on a single seed.

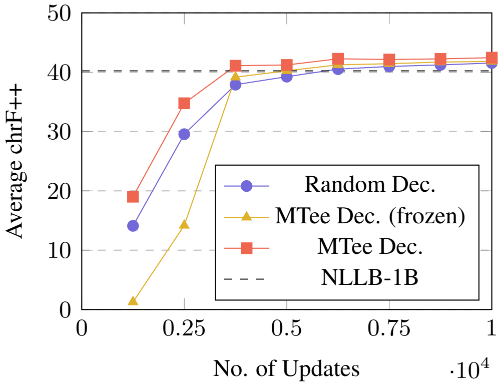

Figure 2: Average test chrF++ score for NLLB+MTee models for first 10,000 training updates (evaluated every 1250 updates). Decoder and adapter (dimensional and layers) are trained, with the rest of the encoder frozen, unless specified with frozen.

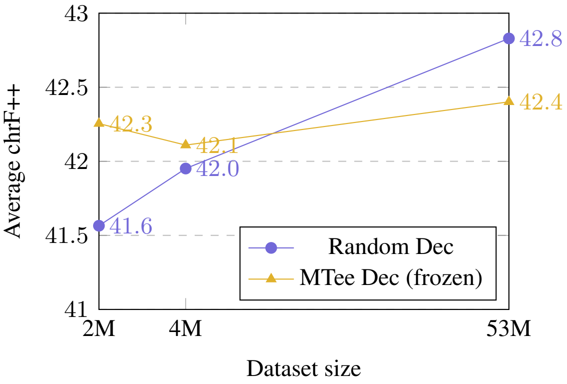

Figure 3: Average test chrF++ score for NLLB+MTee models for three dataset sizes: 500k sentence pairs per direction (2M in total), 1M per direction (4M in total) and the whole dataset (53M in total) trained for 100k updates. For MTee Dec model only dimensional adapter and adapter layers are trained, while the decoder and encoder remain frozen.

### 4.3.2 Effect of the pre-trained decoder

Since we saw that using a pre-trained decoder had a result close to using a randomly initialized decoder, we investigated further how fast the models converge and how the results would compare using less training data.

From Figure 2, we can see that surprisingly for the first 2500 updates the model with a pre-trained encoder and decoder, which trains only the adapter converges the slowest, even being behind the randomly initialized decoder. However, when the decoder is not frozen, we can see that it converges faster than with an uninitialized decoder.

For the dataset size, we can see on Figure 3 that the model with pre-trained encoder and decoder models is less affected by the dataset size, compared to the model that only uses a pre-trained encoder.

### 4.3.3 Effect of adapter structure and the number of languages

| | Model | | | chrF++↑ |
|---|---|---|---|---|
| | NLLB-600M baseline | | | 36.6 |
| | NLLB-600M + MTee | | | |
| | adapter config | DA type | src langs | |
| (1) | DA | MLP | 2 | 35.7 ± 0.2 |
| (2) | DA | linear | 2 | 34.6 ± 0.3 |
| (3) | DA + AL | MLP | 2 | 35.7 ± 2.3 |
| (4) | DA + AL | linear | 2 | 38.2 ± 0.3 |
| (5) | DA + 2 AL | MLP | 2 | 38.0 ± 1.9 |
| (6) | DA + 2 AL | linear | 2 | 38.7 ± 0.3 |
| (7) | 2 AL + DA | linear | 2 | 38.3 ± 0.9 |
| (8) | AL + DA + AL | linear | 2 | 38.5 ± 0.2 |
| (9) | DA + 2 AL | linear | 4 | 38.9 ± 0.1 |
| (10) | DA + 2 AL | linear | 6 | 38.9 ± 0.1 |
| (11) | DA + 3 AL | linear | 4 | 39.0 ± 0.1 |
| (12) | DA + 4 AL | linear | 4 | 39.1 ± 0.1 |
| (13) | DA + 5 AL | linear | 4 | 39.0 ± 0.2 |

Table 5: Many-to-Estonian translation average chrF++ scores of ablation models trained on Europarl evaluated on FLORES200 *devtest*. DA - dimension adapter, AL - adapter layer, DA + $n$ AL means dimension adapter followed by $n$ adapter layers. Training set source languages used are EN, DE, FR, PL, LV, FI, added in the same order when number of languages is increased.

Experiments in this section are performed on the Europarl dataset with results reported in Table 5. The models are trained for 20 epochs on 1 GPU.

It can be seen that using only a dimension adapter without any added layers does not yield as good results and adding layers significantly increases the chrF++ score (see experiments 1–6 in Table 5). Additionally, we see that using the MLP dimension adapter instead of linear yields better results when only using the dimension adapter, but when adding layers it is less stable, resulting in higher variance in average chrF++ scores and lower scores in general.

We can also see that changing the position of the dimension adapter in relation to the adapter layers (to the middle or to the end) does not result in any benefit (see experiments 7 – 9 vs 6).

Using 4 languages results in slightly higher scores than 2 languages (experiments 8 vs 9), however, there is no significant difference when using 6 languages compared to 4 (experiments 9 vs 10). The increase in chrF++ scores could also be caused by the larger dataset and not require different languages to be achieved.

Using 4 layers yields the best result, although the difference in chrF++ scores is small and might not be significant when compared to other numbers of layers (see experiments 11 – 13).

## 5 Conclusion

We have demonstrated that different pre-trained models can be successfully combined even if they have different architectures that wouldn't be directly compatible. With our method, the pre-trained models can remain unchanged while the added dimension adapter and adapter layers align the embeddings. However, in our experiments, the best results were obtained by continuing decoder training after initial adapter training. This might differ in other scenarios depending on the dataset, pre-trained models, and desired translation domain. Our method allowed for a 40% reduction in parameters, efficient training, fast translation, and increased translation quality compared to the original models. With this in mind, we can think of pre-trained translation model encoders and decoders as modules that can be combined depending on the desired outcome.

## 6 Future Works

Our focus is on many-to-one translation. However, it should also be investigated how the mix-and-match approach could be used in one-to-many or many-to-many (or many-to-few) scenarios. The proposed method should also be investigated for other more specific domains and other languages apart from Estonian. Additionally, it should be investigated how other parameter-efficient methods compare to this approach and how they could be incorporated into this method. Further comparisons with pre-trained language models and a combination of using LM and NMT models need exploring as well. Finally, this approach of making sequence representations compatible is not limited to NMT and could be applied to other tasks and modalities.

## 7 Acknowledgements

This work was partially supported by the Estonian Research Council grant PRG2006 as well as the National Programme of Estonian Language Technology grant EKTB67. All computations were

performed on the LUMI Supercomputer through the University of Tartu's HPC center.

## 8 Limitations

One potential limiting factor of the proposed approach is the evaluation process. To ensure accurate and fair evaluation of the models, it is necessary to possess knowledge of the data on which the model was trained to avoid issues with leaky test data. The evaluation of our results relied primarily on automatic metrics, and we mainly utilized the FLORES-200 *devtest* due to the limited availability of test sets for Estonian and non-English languages. Additionally, we were unable to confirm that other available test sets were not part of the original models' training data, so we could not use them for a fair evaluation.

Moreover, the applicability of the mix-and-match method is dependent on the availability of pre-trained models in the target language. For instance, while Estonian models were readily available, other languages may not have such models, rendering the proposed method inapplicable. However, as an alternative, we proposed training the decoder from scratch and demonstrated its competitive performance.

It should also be noted that the translation quality results for Estonian cannot be generalized to all other languages. For example, English already exhibits high translation quality in most multilingual pre-trained NMT models, hence our method may not significantly improve performance as it would for Estonian. However, this limitation does not detract from other positive aspects of our method, including reduced parameter count and efficient training.

## Ethics Statement

From an environmental standpoint, our method reduces the training time, giving a significant one-time reduction. Since our scenario also created a smaller model with faster translation, it reduces long-term computation costs.

From the social standpoint, the resulting models might still be suffering from the same kind of biases as the original models and this aspect is yet to be evaluated. However, with our methods, we can make the use of pre-trained models accessible to more people in terms of computational costs.

## References

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884, Minneapolis, Minnesota. Association for Computational Linguistics.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges.

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1538–1548, Hong Kong, China. Association for Computational Linguistics.

Loic Barrault, Ondrej Bojar, Fethi Bougares, Rajen Chatterjee, Marta R. Costa-jussa, Christian Federmann, Mark Fishel, Alexander Fraser, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Paco Guzman, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Tom Kocmi, Andre Martins, Makoto Morishita, and Christof Monz, editors. 2021. *Proceedings of the Sixth Conference on Machine Translation*. Association for Computational Linguistics, Online.

Toms Bergmanis and Marcis Pinnis. 2022. From zero to production: Baltic-ukrainian machine translation systems to aid refugees. *Baltic Journal of Modern Computing*, 10(3):271–282.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4.

Guanhua Chen, Shuming Ma, Yun Chen, Li Dong, Dongdong Zhang, Jia Pan, Wenping Wang, and Furu Wei. 2021. Zero-shot cross-lingual transfer of neural machine translation with multilingual pretrained encoders. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 15–26, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Guanhua Chen, Shuming Ma, Yun Chen, Dongdong Zhang, Jia Pan, Wenping Wang, and Furu Wei. 2022. Towards making the most of cross-lingual transfer for zero-shot neural machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 142–157, Dublin, Ireland. Association for Computational Linguistics.

Carlos Escolano, Marta R. Costa-jussà, José A. R. Fonollosa, and Mikel Artetxe. 2021. Multilingual machine translation: Closing the gap between shared and language-specific encoder-decoders. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 944–948, Online. Association for Computational Linguistics.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. Beyond English-Centric Multilingual Machine Translation.

Gerard I. Gállego, Ioannis Tsiamas, Carlos Escolano, José A. R. Fonollosa, and Marta R. Costa-jussà. 2021. End-to-end speech translation with pre-trained models and adapters: UPC at IWSLT 2021. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 110–119, Bangkok, Thailand (online). Association for Computational Linguistics.

Nizar Habash and Jun Hu. 2009. Improving Arabic-Chinese statistical machine translation using English as pivot language. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 173–181, Athens, Greece. Association for Computational Linguistics.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Diederik P. Kingma and Jimmy Lei Ba. 2015. Adam: A method for stochastic optimization.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Philipp Koehn, Loïc Barrault, Ondřej Bojar, Fethi Bougares, Rajen Chatterjee, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Alexander Fraser, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Paco Guzman, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Tom Kocmi, André Martins, Makoto Morishita, Christof Monz, Masaaki Nagata, Toshiaki Nakazawa, Matteo Negri, Aurélie Névéol, Mariana Neves, Martin Popel, Marco Turchi, and Marcos Zampieri, editors. 2022. *Proceedings of the Seventh Conference on Machine Translation (WMT)*. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (Hybrid).

Taku Kudo and John Richardson. 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*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Sneha Kudugunta, Isaac Caswell, Biao Zhang, Xavier Garcia, Christopher A. Choquette-Choo, Katherine Lee, Derrick Xin, Aditya Kusupati, Romi Stella, Ankur Bapna, and Orhan Firat. 2023. Madlad-400: A multilingual and document-level large audited dataset.

Jiaang Li, Yova Kementchedjhieva, and Anders Søgaard. 2023. Implications of the convergence of language and vision model geometries.

Xian Li, Changhan Wang, Yun Tang, Chau Tran, Yuqing Tang, Juan Pino, Alexei Baevski, Alexis Conneau, and Michael Auli. 2021. Multilingual speech translation from efficient finetuning of pretrained models. 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)*, pages 827–838, Online. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Sungwon Lyu, Bokyung Son, Kichang Yang, and Jaekyoung Bae. 2020. Revisiting Modularized Multilingual NMT to Meet Industrial Demands. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5905–5918, Online. Association for Computational Linguistics.

Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, Alexandre Muzio, Saksham Singhal, Hany Hassan Awadalla, Xia Song, and Furu Wei. 2021. DeltaLM: Encoder-Decoder Pre-training for

Language Generation and Translation by Augmenting Pretrained Multilingual Encoders.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 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)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2020. Leveraging pre-trained checkpoints for sequence generation tasks. *Transactions of the Association for Computational Linguistics*, 8:264–280.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, and Armand Joulin. 2019. Ccmatrix: Mining billions of high-quality parallel sentences on the web. *arXiv preprint arXiv:1911.04944*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. In *International Conference on Machine Learning*, pages 5926–5936.

Zewei Sun, Mingxuan Wang, and Lei Li. 2021. Multilingual translation via grafting pre-trained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2735–2747, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Anders Søgaard. 2023. Grounding the vector space of an octopus: Word meaning from raw text. *Minds & Machines*, 33:33—54.

Xu Tan, Yi Ren, Di He, Tao Qin, and Tie-Yan Liu. 2019. Multilingual neural machine translation with knowledge distillation. In *International Conference on Learning Representations*.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT — Building open translation services for the World. In *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Andre Tättar, Taido Purason, Hele-Andra Kuulmets, Agnes Luhtaru, Liisa Rätsep, Maali Tars, Mārcis Pinnis, Toms Bergmanis, and Mark Fishel. 2022. Open and competitive multilingual neural machine translation in production. in: Proceedings of baltic hlt 2022. *Baltic Journal of Modern Computing*, 10(3):422434.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639, Online. Association for Computational Linguistics.

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tieyan Liu. 2020. Incorporating bert into neural machine translation. In *International Conference on Learning Representations*.

Yaoming Zhu, Jiangtao Feng, Chengqi Zhao, Mingxuan Wang, and Lei Li. 2021. Counter-interference adapter for multilingual machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2812–2823, Punta Cana, Dominican Republic. Association for Computational Linguistics.

## A BLEU Scores

Average BLEU scores are presented in Table 6

## B COMET Scores for Selected Directions

COMET scores of selected directions are displayed in Table 7.

|  | Model | average BLEU ↑ |  |
|---|---|---|---|
|  |  | full | EU |
| (1) | NLLB-1B | 12.8 | 16.9 |
| (2) | NLLB-MoE$^{\dagger}$ | 15.5 | 20.1 |
|  | **Pivot**, m2en: NLLB-1B |  |  |
| (3) | en2et NLLB-1B | 13.5 | 17.3 |
| (4) | en2et: MTee | **15.7** | 20.4 |
|  | **Fine-tune** NLLB-1B |  |  |
| (5) | - | 15.4 ± 0.1 | 20.8 ± 0.2 |
| (6) | freeze enc | 15.5 ± 0.1 | 20.8 ± 0.1 |
|  | **Ours:** NLLB-1B enc + |  |  |
| (7) | rand dec | 14.5 ± 0.1 | 19.8 ± 0.1 |
| (8) | MTee dec | 15.1 ± 0.1 | 20.6 ± 0.2 |
| (9) | MTee dec, 2-stage | 15.6 ± 0.1 | **21.3 ± 0.1** |

Table 6: Many-to-Estonian translation average BLEU scores. For experiments involving model training, the average of 5 random seeds are reported with confidence intervals ($p = 0.01$). † - Scores reported by (Team et al., 2022).

| Model | eng_Latn | deu_Latn | rus_Cyrl | zho_Hans | arb_Arab |
|---|---|---|---|---|---|
| NLLB-1B | 0.8967 | 0.8805 | 0.8700 | 0.8435 | 0.8492 |
| NLLB-MoE$^{\dagger}$ | **0.9144** | **0.9031** | **0.8904** | **0.8826** | **0.8781** |
| MTee | 0.8916 | 0.8908 | 0.8819 | - | - |
| **Pivot**, m2en NLLB-1B |  |  |  |  |  |
| en2et NLLB-1B | 0.8967 | 0.8808 | 0.8705 | 0.8673 | 0.8583 |
| en2et MTee | 0.8916 | 0.8899 | 0.8782 | 0.8788 | 0.8615 |
| **Fine-tune** NLLB-1B |  |  |  |  |  |
| - | 0.8954 | 0.8878 | 0.8825 | 0.8775 | 0.8631 |
| freeze enc | 0.8974 | 0.8912 | 0.8812 | 0.8772 | 0.8552 |
| **Ours:** NLLB-1B enc + |  |  |  |  |  |
| rand dec | 0.9001 | 0.8902 | 0.8793 | 0.8688 | 0.8561 |
| MTee dec | 0.9049 | 0.8953 | 0.8831 | 0.8659 | 0.8586 |
| MTee dec 2-stage | 0.9060 | 0.8929 | 0.8857 | 0.8724 | 0.8607 |

Table 7: Many-to-Estonian translation COMET scores for selected directions. Underlined results indicate a significant gain over the baseline NLLB-1B with $p = 0.01$ according to Paired Bootstrap Resampling t-test. † - Scores calculated from translations reported by Team et al. (2022). Language abbreviations are following Team et al. (2022).