Language-Aware Multilingual Machine Translation with Self-Supervised Learning

Haoran Xu[♠], **Jean Maillard**[♡], **Vedanuj Goswami**[♡]

[♠]Johns Hopkins University, [♡]Meta AI

hxu64@jhu.edu
{jeanm,vedanuj}@meta.com

Abstract

Multilingual machine translation (MMT) benefits from cross-lingual transfer but is a challenging multitask optimization problem. This is partly because there is no clear framework to systematically learn language-specific parameters. Self-supervised learning (SSL) approaches that leverage large quantities of monolingual data (where parallel data is unavailable) have shown promise by improving translation performance as complementary tasks to the MMT task. However, jointly optimizing SSL and MMT tasks is even more challenging. In this work, we first investigate how to utilize **intra-distillation** to learn more *language-specific* parameters and then show the importance of these language-specific parameters. Next, we propose a novel but simple SSL task, **concurrent denoising**, that co-trains with the MMT task by concurrently denoising monolingual data on both the encoder and decoder. Finally, we apply **intra-distillation** to this co-training approach. Combining these two approaches significantly improves MMT performance, outperforming three state-of-the-art SSL methods by a large margin, e.g., 11.3% and 3.7% improvement on an 8-language and a 15-language benchmark compared with MASS, respectively[1].

1 Introduction

Multilingual machine translation (MMT) (Aharoni et al., 2019; Arivazhagan et al., 2019) comes with the problem of designing architectures where certain parameters are shared and certain parameters are more language-specific. In order to mitigate negative interference across languages, recent studies have investigated language-specific parameters, including searching for more language-specific parameters (Lin et al., 2021), or adding extra language-specific components to the original model (Zhang et al., 2021; NLLB Team et al., 2022), or even utilizing language-specific pre-trained language models (Xu et al., 2021; Yarmohammadi et al., 2021). All these studies indicate the importance of language-specific parameters. *In this work, we first want to encourage parameters to have more language-specific attributes given a fixed model size.*

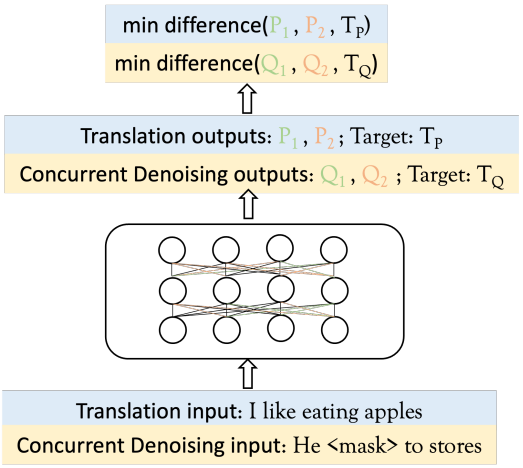


Figure 1: Concurrent denoising is a complementary task to the MMT task. Both tasks are applied with intra-distillation, where we forward pass model twice for the translation and masked inputs and each time we disable different subsets of parameters (illustrated by different colors). Then, for each task, we not only minimize the difference between the target and two outputs (e.g., minimize difference(P_1, T_P) and difference(P_2, T_P) in the MMT task), we also minimize the difference between two translated outputs as well as two denoised outputs (e.g., minimize difference(P_1, P_2) for MMT).

The difficulty of scaling MMT to low-resource and long-tail languages arises due to the scarcity of abundantly available parallel aligned data. Previous works (NLLB Team et al., 2022; Siddhant et al., 2022; Kim et al., 2021; Wang et al., 2020; Siddhant et al., 2020) try to tackle this by collecting massive amounts of monolingual data and using various types of self-supervised learning (SSL)

Work done during an internship at Meta AI Research

[1]Code is released at https://github.com/fe1ixxu/CD_ID_MMT.

Findings of the Association for Computational Linguistics: EACL 2023, pages 526–539

objectives, such as denoising AutoEncoder (DAE) (Liu et al., 2020) or Masked Sequence to Sequence (MASS) (Song et al., 2019) as auxiliary tasks to co-train with the MMT task, to compensate for the scarcity of parallel data for low-resource languages. *Following this line, we secondly aim to propose a more effective SSL objective.*

With the goal of learning language-aware MMT models and designing more effective SSL methods for MMT, we introduce two approaches. The first approach is **Intra-Distillation** (ID) (Xu et al., 2022), which performs a forward pass through the model K times[2], and in each pass disables a different set of parameters. This enforces consistent contributions between these disabled parameters by minimizing the difference between the K outputs. ID was originally proposed by Xu et al. (2022) to achieve a balanced parameter contribution in a model. Here, we study the effectiveness of ID in learning language-specific parameters for MMT models. Next, we introduce **Concurrent Denoising** (CD) which is an auxiliary self-supervised task jointly trained with the MMT task. CD predicts the same masked sentences both on the encoder and decoder side with a shared projection layer to facilitate the consistent understanding between encoder and decoder representations. We show that CD outperforms several state-of-the-art SSL methods for translation. Finally, we apply ID to our co-training scheme to further improve the MMT performance by learning more language-specific parameters. The overall framework is illustrated in Figure 1 and we summarize our main contributions below.

- We propose a method to quantify the degree of language-specificity of all parameters (Section 2) and perform a thorough analysis to demonstrate that intra-distillation helps the model learn more language-specific parameters. These parameters contribute more towards a specific language to improve the overall model generalization performance (Section 3).
- We propose the **concurrent denoising** SSL method and demonstrate its improvements over other existing SSL objectives for MMT. Moreover, we introduce a co-training method of MMT and CD with the help of intra-distillation and shows the strong effectiveness of ID in improving MMT+SSL multi-task optimization (Section 4).
- We conduct extensive experiments on a 8-language dataset and a larger 15-language multilingual dataset, and demonstrate that MMT with concurrent denoising and intra-distillation outperforms multiple strong state-of-the-art methods (Section 5).

[2]We use $K = 2$ in this work.

2 Preliminary

2.1 Quantify Language-Specific Parameters

Parameter sensitivity is a measure of the impact on the loss when a specific parameter of a model is zeroed-out. It is widely used in pruning as importance score (Ding et al., 2019; Molchanov et al., 2019; Lubana and Dick, 2021). A parameter can express different sensitivities depending on the language of the input data. Those parameters that have high sensitivity to a specific language but low sensitivity to others, are language-specific parameters. We define the i^{th} parameter in a model parameterized by $\mathbf{\Theta}$ as $\theta_i \in \mathbb{R}$. We further define $\mathbf{\Theta}_i = [0, \cdots, 0, \theta_i, 0, \cdots, 0] \in \mathbb{R}^{|\mathbf{\Theta}|}$ and $\mathbf{\Theta}_{-i} = [\theta_1, \cdots, \theta_{i-1}, 0, \theta_{i+1}, \cdots, \theta_{|\mathbf{\Theta}|}] \in \mathbb{R}^{|\mathbf{\Theta}|}$. The sensitivity of the i^{th} parameter given input batch b_l from language l is formulated as

$$\mathcal{S}(\theta_i, b_l) = |\mathcal{L}(\mathbf{\Theta}, b_l) - \mathcal{L}(\mathbf{\Theta}_{-i}, b_l)|, \quad (1)$$

where $\mathcal{L}(\cdot)$ is the loss function given the input batch and parameters. Then, we use a first-order Taylor decomposition to approximate the sensitivity of any arbitrary parameters. Equation 1 then becomes

$$\mathcal{S}(\theta_i, b_l) \approx |\mathbf{\Theta}_i^T \nabla_{\mathbf{\Theta}} \mathcal{L}(\mathbf{\Theta}, b_l)|, \quad (2)$$

where $\nabla_{\mathbf{\Theta}} \mathcal{L}(\mathbf{\Theta}, b_l)$ is the gradient of the loss with respect to the model parameters. In our implementation, we randomly pick 500 batches and feed them to the model to retrieve the gradients and compute the average sensitivity. We then have

$$\mathcal{S}(\theta_i, \mathcal{B}_l) \approx \frac{1}{|\mathcal{B}_l|} \sum_{b_l \in \mathcal{B}_l} |\mathbf{\Theta}_i^T \nabla_{\mathbf{\Theta}} \mathcal{L}(\mathbf{\Theta}, b_l)|, \quad (3)$$

where $\mathcal{B}_l$ is a set containing 500 random b_l batches.

Now, we propose to quantify the degree of language-specificity of θ_i with respect to language l by measuring the relative sensitivity difference between language l and the other languages as

$$D(\theta_i, l) = \frac{\mathcal{S}(\theta_i, \mathcal{B}_l) - \mathcal{S}(\theta_i, \mathcal{B}_{-l})}{\mathcal{S}(\theta_i, \mathcal{B}_{-l}) + \sigma}, \quad (4)$$

where $\mathcal{B}_{-l}$ represents the set composed of mixed batches from all training languages except for the language l, and σ is a very small positive constant[3]. The larger $D(\theta_i, l)$ is, the more language-specific θ_i is to language l.

2.2 Intra-Distillation

A model with more balanced parameter sensitivity distribution shows better generalization (Liang et al., 2022). Xu et al. (2022) propose intra-distillation (ID) as an effective task-agnostic training method, aiming to encourage all parameters to contribute equally, which improves performance when model size is fixed. However we argue that, in the multilingual setting, ID actually helps the model learn more language-specific parameters resulting in improved performance. Given an input batch, ID needs to forward pass the model K times to obtain K outputs and each time a random subset of parameters is zeroed out. The core idea of ID is to minimize the difference of these K outputs to approximate minimizing the contribution gap of the parameters that are zeroed-out, because the K outputs are forced to be the same with different zeroed parameters. Let $\{p_1, \cdots, p_i, \cdots, p_K\}$ denote the K outputs. Note that the outputs are probability distributions in the translation and denoising task. The ID loss is then formulated by the X-divergence (Xu et al., 2022) to minimize the difference of K outputs as

$$\mathcal{L}_{id} = \frac{1}{K}\sum_{i=1}^{K} \mathbb{KL}(p_i \parallel \bar{p}) + \mathbb{KL}(\bar{p} \parallel p_i) \quad \text{where } \bar{p} = \frac{1}{K}\sum_{i=1}^{K} p_i \tag{5}$$

Let the original task loss be $\mathcal{L}_i$ for the i^{th} pass. Then, the total loss is a combination of the original task losses and ID loss, given as

$$\min \frac{1}{K}\sum_{i=1}^{K} \mathcal{L}_i + \alpha \mathcal{L}_{id} \tag{6}$$

where α is a hyper-parameter to control the strength of ID. Similar to Xu et al. (2022), we use dropout to simulate *zeroed-out* parameters in all experiments.

Although the explanation for better performance after using ID is that the model parameters become more balanced, it is unclear how parameter contributions to different languages change after applying ID in a multilingual (multitask) setting. For instance, do parameters become more language-agnostic and shareable across all languages, or do they become more language-specific? We investigate this in more details in Section 3.2.

3 Language-Aware MMT Models

In this section, we study how parameters can be prompted to be more language-specific by applying **intra-distillation**, which improves the model generalization performance. Specifically, certain parameters become more language-specific and tend to contribute more to their specific language and less to others. We demonstrate the importance of language-specific parameters by showing how much they can contribute in pruning experiments. We begin our analysis from a case study on MMT experiments with an 8-language dataset (M8), and then scale up our experiments to 15 languages (M15) with larger data size in Section 5. Here, we show results and analysis on `xxx→eng` directions. Similar discussions for `eng→xxx` directions are shown in Appendix A.

3.1 Experiments on Intra-Distillation

Dataset and Training We train MMT models with and without ID on the M8 dataset[4]. M8 is composed of Nigerian Fulfulde (`fuv`, 18K parallel sentences), Kimbundu (`kmb`, 82K), Ganda (`lug`, 278K), Chewa (`nya`, 693K), Swahili (`swh`, 2.1M), Umbundu (`umb`, 193K), Wolof (`wol`, 9K) and Zulu (`zul`, 1.2M). Datasets are extracted from the primary bitext used by the NLLB-200 model (NLLB Team et al., 2022). For ID, we pass the model twice ($K = 2$) considering the computational cost, and set α as 5 suggested by Xu et al. (2022). We use FLORES-200 as our dev and test sets (NLLB Team et al., 2022). Our model training is based on the Transformer$_{\text{big}}$ architecture (Vaswani et al., 2017) with 32K vocabulary jointly trained by SentencePiece (Kudo

[3] σ is 1e-8 in our implementation.

[4] The languages were selected in order to have a realistic dataset reflecting a specific use case. Multilingual training is crucial for languages that are low-resource, as is the case for many languages of Africa. We chose two different language groupings from the African continent: Benue-Congo languages (Kimbundu, Ganda, Chewa, Swahili, Umbundu, Zulu) and North-Central Atlantic languages (Nigerian Fulfulde, Wolof). While these languages may all belong to the Atlantic-Congo family, this is an extremely large, varied, and under-researched family, with Glottolog recording over 1,400 languoids in it – compare this to under 590 languoids recorded for the Indo-European family.

and Richardson, 2018). We report sacreBLEU scores (spm tokenizer) (Post, 2018).

Results Following NLLB Team et al. (2022), we categorized a language as *low-resource* if there are fewer than 1M parallel sentences, and as *very low-resource* if fewer than 100K (very low-resource is not the subset of low-resource). Otherwise, the language is considered as *high-resource*. We report the average BLEU scores for each of the three categories. In Table 1, we show that MMT with ID outperforms the regular MMT model by a large margin on all three categorizes by +1.21 BLEU averaged across all languages.

Method	High	Low	Very Low	All
Regular	31.70	12.57	6.92	15.94
Intra-Distillation	**33.30**	**13.63**	**8.05**	**17.15**

Table 1: M8 results on xxx→eng comparing regular MMT and MMT with ID. We observe that MMT with ID outperforms regular MMT by a significant margin.

3.2 Language-Specific or Language-Agnostic?

Next, we study whether parameter contributions are more language-specific or just shareable across all languages after ID. Given the i^{th} language l_i, we compute the sensitivities (Equation 3) of all parameters and flatten them into a list. Then, we calculate the Pearson correlation coefficients (PCC) p_{ij} between sensitivity lists of any arbitrary pair of languages l_i and l_j. A lower p_{ij} indicates that there are more contribution (sensitivity) disagreements between languages l_i and l_j. We plot a heat map to visualize p_{ij} for every language pair. Taking into account that the top 10% parameters usually dominate the contribution (Xiao et al., 2019; Sanh et al., 2020), we consider the performance of two groups of parameters, high-sensitive (top 10% most sensitive) and low-sensitive (the remaining 90%) parameters, respectively. Figure 2 shows that all p_{ij} in both groups become lower, indicating there is lower sensitivity similarity between different languages for the same parameters, which means the model becomes more language-specific after ID. For instance, sensitivity similarity between zul and wol drops from 0.67 to 0.57 in the low-sensitive group. However, the p_{ij} of low-sensitive parameters drops much more than high-sensitive ones, and high-sensitive parameters still hold high similarity (over 0.9). Thus, low-sensitive parameters mostly have language-specific properties while high-sensitive parameters tend to play 'language-agnostic' roles. Overall, parameters are more language-specific after ID[5]. In fact, learning more language-specific parameters through ID in MMT leads to better performance as seen in Section 3.1. These findings align with the results of recent studies which investigate language-specific parameters (Lin et al., 2021; Zhang et al., 2021; NLLB Team et al., 2022), indicating the importance of language-specific parameters.

3.3 The Importance of Language-Specific Parameters

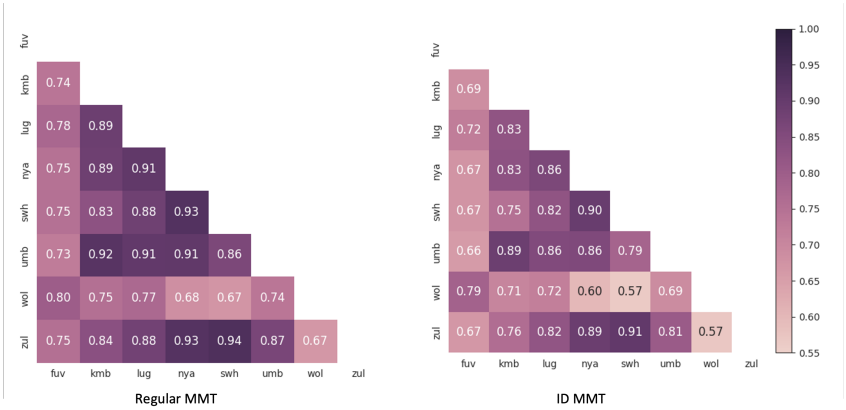
Here, we study the reason *why language-specific parameters are important and how much they contribute*. To investigate this, we first measure the degree of language-specificity of all parameters based on Equation 4. We explore the contribution of language-specific parameters with respect to the BLEU scores. Then, we conduct one-shot unstructured pruning with respect to BLEU scores in order of the degree of language-specificity for both models with and without ID, starting with the least language-specific parameters[6]. As more parameters are pruned, a slower performance drop means that a higher contribution comes from the remaining more language-specific parameters. Figure 3 shows the average BLEU drop across 8 languages versus the percentage of parameters pruned. After pruning the less language-specific parameters, the rest of the more language-specific parameters in the model with ID are able to preserve better performance, indicating the importance of more language-specific parameters.

4 Proposed Self-Supervision Method

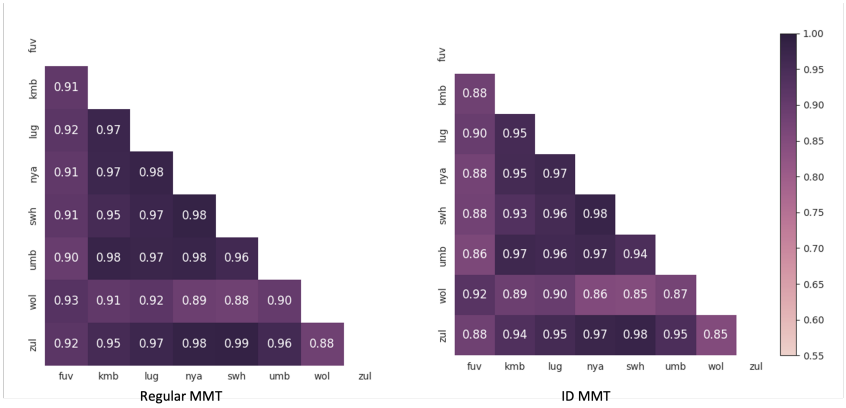
We extend our study of language-awareness to MMT models co-trained with self-supervised objectives that have been shown to improve translation performance. We first propose a simple but effective self-supervised learning objective, **concurrent denoising** (CD), and then investigate the effectiveness of ID in helping improve multi-task optimization challenges of co-training CD and

[5]The overall parameter contribution is still more balanced as claimed in Xu et al. (2022). We leave further discussion on this to Appendix B.

[6]Note that, as shown in Figure 2b, the 10% most sensitive parameters are highly language-agnostic. They are easy to classify as less language-specific and can be pruned, but pruning them would lead to near-random performance (BLEU≈0), making it hard to evaluate the importance of more language-specific parameters. Thus, we keep the top 10% sensitive parameters and prune the rest of parameters that display a more language-specific behavior.



(a) 90% low-sensitive parameters



(b) 10% high-sensitive parameters

Figure 2: PCC between the lists of parameter sensitivity of every language (left for regular MMT and right for MMT with ID). We show contribution similarity of two groups of parameters, i.e., top 10% high-sensitive parameters and the remaining 90% parameters. The lower score between two languages represents the less similarity of parameter contributions for these two languages, which means more contribution disagreements and parameters are more language-specific.

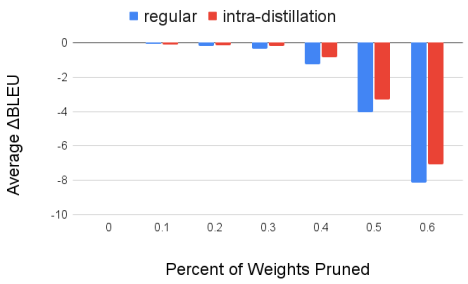


Figure 3: Change in average xxx→eng translation performance across 8 languages versus pruning ratio. Models are pruned starting with the least language-specific parameters.

MMT tasks together by learning more language-specific parameters.

4.1 Concurrent Denoising

Self-supervised learning objectives usually involve sentence denoising either on the encoder side, such as MLM (Devlin et al., 2019), or on the decoder side, such as DAE (Liu et al., 2020). Jointly denoising sentences on both the encoder and the decoder sometimes is better than a single denoising objective (Wang et al., 2020; Kim et al., 2021) for MMT, but the training cost is doubled as we need to calculate the loss for the same monolingual sentence twice (masked in two different ways). We propose **concurrent denosing**, a self-supervised task that denoises a single masked sentence both on the encoder and decoder sides, which not only reduces the training time but also improves the language understanding of the model to result in better MMT performance.

We add noise to the monolingual data by whole-word masking (Devlin et al., 2019), where we randomly replace $r_m\%$ words with the special token <mask>. During the replacement process, each word has a 10% chance not to be masked, and another 10% chance to be replaced with other

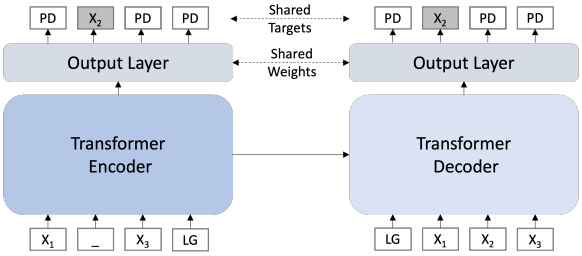


Figure 4: Concurrent denoising. In the example input sentence 'X_1 X_2 X_3', the token X_2 is masked. The encoder and decoder share the same output projection layer and target tokens to predict the masked token. We only calculate the loss for the masked token prediction. `PD` represents the target token loss padding and `LG` is a special language token.

random tokens. The encoder and decoder use a shared output layer to reconstruct the original sentence. The loss for the encoder and decoder side are denoted as $\mathcal{L}_e$ and $\mathcal{L}_d$ respectively[7]. The total training loss combining translation loss $\mathcal{L}_{MMT}$ and two self-supervised losses is

$$\mathcal{L} = \mathcal{L}_{MMT} + \mathcal{L}_e + \mathcal{L}_d. \quad (7)$$

Concurrent denoising is illustrated in Figure 4. Two key differences between our concurrent denoising method and regular MLM or DAE methods are worth highlighting.

Shared Output Projection Since the decoder has an output projection layer while the encoder does not, Wang et al. (2020) train the encoder with MLM by using an additional projection layer. However, we utilize the decoder projection layer as a shared layer for both encoder and decoder to reconstruct the sentence, which significantly reduces model parameters. This is because the projection layer is usually large when we have a large vocabulary size. We show the effect of using a shared projection layer in Appendix C.

Shared Target Tokens Since the output representations of the encoder and decoder are fed to the same projection layer, we want them to predict the same target token at the same position for the stability of the projection layer training. To achieve this, we carefully design our language token positions. Instead of only prepending a special language token at the beginning of the source sentence (Johnson et al., 2017), we append the special language token on the source side and also prepend it on the decoder side (As shown in Figure 4). This design also applies to MMT. In this way, we can avoid the encoder and decoder from predicting the same token at different positions.

[7]Unlike DAE training on the decoder side, we zero out the losses which predict non-masked tokens.

4.2 Concurrent Denoising with Intra-Distillation

We investigate whether ID helps concurrent denoising to improve overall performance. We apply ID to the co-training of CD and MMT tasks. Following Equation 6 and 7, our final loss is

$$\mathcal{L} = \frac{1}{K}(\sum_{i=1}^{K}\mathcal{L}_{MMT_i} + \sum_{i=1}^{K}\mathcal{L}_{e_i} + \sum_{i=1}^{K}\mathcal{L}_{d_i}) + \alpha(\mathcal{L}_{id\_MMT} + \mathcal{L}_{id\_e} + \mathcal{L}_{id\_d}), \quad (8)$$

where $\mathcal{L}_{id\_MMT}$, $\mathcal{L}_{id\_e}$ and $\mathcal{L}_{id\_d}$ respectively represent the ID loss for translation, encoder denoising and decoder denoising (i.e., $\mathcal{L}_{id\_e}$ minimizes the difference of the K encoder outputs based on Equation 5, etc.). The i index in $\mathcal{L}_{e_i}$, $\mathcal{L}_{MMT_i}$ and $\mathcal{L}_{d_i}$ indicates that these losses are for the i^{th} forward pass.

5 MMT+SSL Experiments

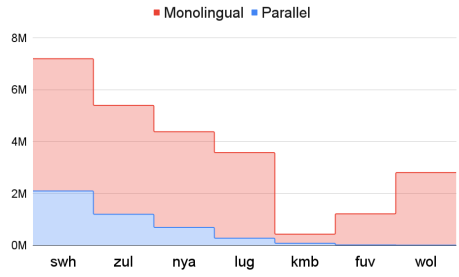
5.1 Baselines

We consider three strong baselines. All baselines are our own implementation following the settings from the original papers.

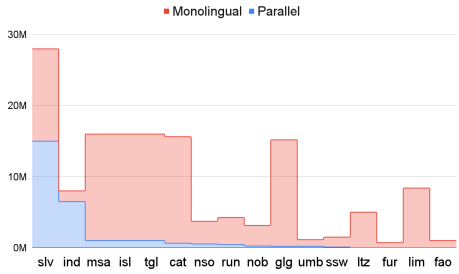
DAE NLLB Team et al. (2022) learn the effects of the causal language modeling (LM) and DAE objectives (Liu et al., 2020). Since they find that DAE performs better than LM or LM+DAE, we only compare our methods with the DAE objective.

DAE+MLM Wang et al. (2020) study a multi-task learning framework which jointly trains the MMT, MLM and DAE objectives, where MLM and DAE reconstruct sentences noised by different masking methods. Kim et al. (2021) also investigate the effectiveness of ELECTRA (Clark et al., 2020). They conclude that DAE+MLM is better than DAE+ELECTRA.

MASS Siddhant et al. (2020) and Siddhant et al. (2022) utilize MASS (masked sequence to sequence pre-training) (Song et al., 2019) to improve the MMT performance. Similar to MLM which predicts masked tokens on the encoder side, MASS masks a fragment of a sentence and predicts the masked fragment but on the decoder side.



(a) M8 dataset



(b) M15 dataset

Figure 5: The statistics of monolingual and parallel data for M8 and M15 are presented. The languages are arranged in descending order of parallel data size.

5.2 Datasets

In addition to the M8 dataset described in Section 3.1, we also build a larger dataset (M15), covering 15 languages. In composing this dataset, we take into account linguistic diversity and data size. The resulting dataset has languages from 6 linguistic families and a balanced number of high-resource, low-resource and very low-resource languages. Detailed information on this dataset is in Appendix D. We randomly sample at most 3M monolingual samples per language for M8, and 15M for M15. The distribution of monolingual data and parallel data for M8 and M15 is shown in Figure 5. Note that we also use parallel data for self-supervised learning, so the true monolingual data size includes bitext data. We use the FLORES-200 dataset for evaluation. All datasets come from the primary bitext and monolingual data used for the NLLB-200 model (NLLB Team et al., 2022).

5.3 Data Sampling

We use a data sampling temperature of $T = 1$ suggested by NLLB Team et al. (2022) to train on the MMT objective. For monolingual data, we use a temperature of $\frac{10}{7}$ to balance the SSL training, as suggested by Liu et al. (2020). During co-training, we mix the two sources in an equal ratio (50% monolingual data (including bitext used for SSL training) with self-supervision and 50% parallel data).

Method	High	Low	Very Low	All
M8 results				
Regular MMT	31.70	12.57	6.92	15.94
+DAE (NLLB Team et al., 2022)	32.69	13.38	7.57	16.75
+DAE+MLM (Wang et al., 2020)	33.05	13.93	8.20	17.27
+MASS (Siddhant et al., 2020)	32.64	13.03	6.79	16.37
+CD (ours)	32.92	13.94	8.38	17.29
+CD+ID (ours)	**35.16**	**15.18**	**9.18**	**18.69**
M15 results				
Regular MMT	**39.87**	35.20	24.45	33.17
+DAE (NLLB Team et al., 2022)	38.46	34.05	26.23	32.91
+DAE+MLM (Wang et al., 2020)	38.60	34.00	25.49	32.70
+MASS (Siddhant et al., 2020)	38.53	33.93	22.79	31.75
+CD (ours)	39.23	34.88	28.21	34.11
+CD+ID (ours)	39.58	**35.53**	**29.43**	**34.85**

Table 2: Overall xxx→eng BLEU for M8 and M15.

5.4 Training and Evaluation Details

All experiments consider both the eng→xxx and xxx→eng directions and use the Transformer architecture (Vaswani et al., 2017). We use $\text{Transformer}_{\text{big}}$ (242M parameters, 6 layers, 16 heads, 1,024 hidden dimension, 4,096 FFN dimension) for M8 experiments. For M15 experiments, we double the layers of $\text{Transformer}_{\text{big}}$ (418M parameters). We use a vocabulary of size 32k for both M8 and M15 with SentencePiece (Kudo and Richardson, 2018). The batch size is 30K tokens. We warm-up for the first 8K steps. We set the total training steps to 100K and 300k for M8 and M15 respectively, with patience set to 10 for early stopping. We forward pass the model twice (K=2) to conduct ID. We set the ID weight $\alpha = 5$. During concurrent denoising, the masking ratio is set to $r_m = 30\%$. We also show the effect of masking ratio in Appendix E. During generation, we use beam search with a beam size of 5 and a length penalty of 1.0. All models are evaluated with sacreBLEU (spm tokenizer).

Method	High	Low	Very Low	All
M8 results				
Regular MMT	34.14	11.47	5.75	15.71
+DAE (NLLB Team et al., 2022)	34.35	11.41	5.79	15.74
+DAE+MLM (Wang et al., 2020)	34.48	11.45	5.20	15.64
+MASS (Siddhant et al., 2020)	34.02	11.53	4.75	15.46
+CD (ours)	34.87	11.50	**5.90**	15.94
+CD+ID (ours)	**35.83**	**11.90**	5.81	**16.37**
M15 results				
Regular MMT	**38.44**	31.62	16.46	28.84
+DAE (NLLB Team et al., 2022)	37.46	30.86	18.39	28.90
+DAE+MLM (Wang et al., 2020)	37.99	30.98	18.05	29.01
+MASS (Siddhant et al., 2020)	38.19	31.20	17.88	29.09
+CD (ours)	37.74	30.94	19.04	29.24
+CD+ID (ours)	38.29	**31.71**	**19.43**	**29.81**

Table 3: Overall eng→xxx BLEU for M8 and M15.

5.5 Results

The overall results for the xxx→eng and eng→xxx directions are shown in Tables 2 and 3. For both M8 and M15, and both translation directions, concurrent denosing is better than all aforementioned baselines, and combining it with ID further improves upon the baselines by an even larger margin. For instance, our method outperforms MASS by 11.3% and 3.7% on M8 and M15 respectively, averaged across all languages and directions. We also show the effectiveness of ID on other objectives like DAE in Section 6.1, but the results are subpar compared to CD+ID.

Aligned with the findings of Wang et al. (2020); Kim et al. (2021), we observe that DAE+MLM is better than DAE alone in M8 xxx→eng, but the improvements become very minor when it comes to M8 eng→xxx or when scaling to 15 languages. MASS performs similarly or better than DAE in the eng→xxx but worse in the xxx→eng.

In M15, high-resource languages perform slightly worse with SSL methods compared to the MMT only baseline, but improves other categories, similar to the observations of NLLB Team et al. (2022). It does not occur on M8, possibly due to the smaller dataset size allowing for sufficient model capacity to learn from additional monolingual data.

Note that the effectiveness of SSL such as DAE and MASS is not as pronounced as reported by Wang et al. (2020) and Siddhant et al. (2022). However, it is necessary to consider the for domain mismatch between the training and evaluation data. As demonstrated by Siddhant et al. (2022), a significant decline in performance can occur when either monolingual or bitexts diverge from the evaluation domain. In our study, the training data is sourced from NLLB-200 and FLORES-200, which encompasses a wide range of domains. we hypothesize that this contributes to the observed lessened effectiveness of SSL techniques in our experiments.

6 Analysis

6.1 Ablation Study

The final loss, described in Equation 8, has 6 loss terms. Except for the translation loss, we ablate the relative contribution of all the other 5 loss terms to the translation task performance. In Table 4, we show the results of this ablation study on M8 xxx→eng directions. Method ① is the regular MMT model and method ② is ID training only for MMT (the same result as in Section 3.1). Method ③ is the same as the MMT+DAE method. With the help of ID for the decoder denoising (method ④) and an additional ID for translation (method ⑤), translation performance can respectively obtain +0.41 and +0.98 BLEU on average compared to ③. Note that method ⑤ is the MMT+DAE+ID method. Compared to our MMT+CD+ID method, it substantially underperforms our method (17.73 vs. 18.69), which shows that our method could better stimulate the potential of ID. The results for methods ⑥, ⑦ and ⑧ indicate the effectiveness of encoder denoising with CD and applying ID. Overall, the translation performance improves by including all the loss terms.

	Method	Avg. BLEU
①	$\mathcal{L}_{MMT}$	15.94
②	$\mathcal{L}'_{MMT} + \mathcal{L}_{id\_MMT}$	17.15
③	$\mathcal{L}_{MMT} + \mathcal{L}_{d}$	16.75
④	$\mathcal{L}'_{MMT} + \mathcal{L}'_{d} + \alpha\mathcal{L}_{id\_d}$	17.16
⑤	$\mathcal{L}'_{MMT} + \mathcal{L}'_{d} + \alpha(\mathcal{L}_{id\_d} + \mathcal{L}_{id\_MMT})$	17.73
⑥	$\mathcal{L}_{MMT} + \mathcal{L}_{e} + \mathcal{L}_{d}$	17.29
⑦	$\mathcal{L}'_{MMT} + \mathcal{L}'_{e} + \mathcal{L}'_{d} + \alpha(\mathcal{L}_{id\_d} + \mathcal{L}_{id\_e})$	17.59
⑧	$\mathcal{L}'_{MMT} + \mathcal{L}'_{e} + \mathcal{L}'_{d} + \alpha(\mathcal{L}_{id\_d} + \mathcal{L}_{id\_e} + \mathcal{L}_{id\_MMT})$	**18.69**

Table 4: Ablation study on loss terms. For simplicity, we use $\mathcal{L}'$ to represent the mean loss of K forward pass, e.g., $\mathcal{L}'_e = \frac{1}{K}\sum_{i=1}^{K}\mathcal{L}_{e_i}$.

6.2 Language-Specific Parameters for SSL

In Section 3, we observed that ID helps MMT learn more language-specific parameters and improve model generalization. We are also interested in understanding 1) whether the model also learns more language-specific parameters for the SSL task (here we investigate CD), and 2) what is the relationship of parameter contribution between MMT and SSL tasks for the same language. We use the xxx→eng direction of the M8 dataset as an example to study these questions.

In Figure 6, we plot a heat map to illustrate the PCC of all parameter sensitivities between every language pair. As expected, parameter sensitivity similarity becomes lower for all languages, which means there are more language-specific parameters when we train SSL methods with ID. For the second question, in Figure 7 we show the parameter sensitivity similarity between the MMT and CD tasks for each language. The contribution similarity becomes higher between the two tasks for every language with ID. This is expected, since the losses of MMT and CD have the same objective on the decoder side, i.e., text generation conditioned on

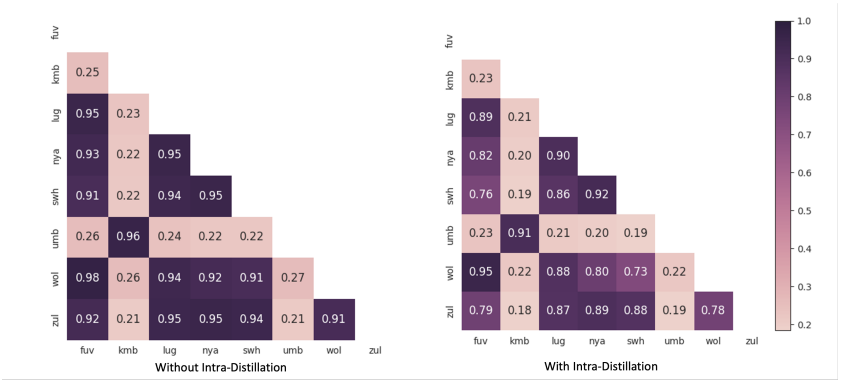


Figure 6: Parameter contribution similarity among all language pairs, evaluated by PCC for the CD task before (left) and after (right) ID.

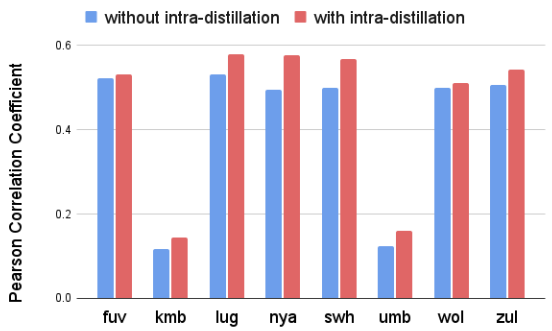


Figure 7: Parameter contribution similarity between MMT and CD for each language with and without ID.

another text. This is also another reason why SSL tasks can help multilingual translation.

7 Conclusions

We show extensive analysis that intra-distillation training helps multilingual translation by learning more language-specific parameters. We propose concurrent denoising, improving upon multiple state-of-the-art self-supervised learning methods. Moreover, we demonstrate that applying intra-distillation to the above co-training scheme offers further improvements to translation performance.

Limitations

Although we show improvements using our methods on multiple languages from diverse language families on multilingual machine translation, it should be noted that the generalizability of our findings to other multi-task learning settings, such as those involving the combination of tasks such as named entity recognition, part-of-speech tagging, and question answering, remains uncertain. This is due to the fact that our study primarily focused on the utilization of intra-distillation to learn task-specific parameters on multilingual machine translation and did not investigate the aforementioned tasks. Furthermore, with intra-distillation we need to perform more than one forward pass, leading to a trade-off between higher performance and increased training time – which, for many use-cases, could be arguably acceptable.

Acknowledgements

We would like to thank anonymous reviewers for their valuable comments. We also thank Alex Guo, Simeng Sun, and Weiting Tan for their helpful suggestions.

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 F. Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *CoRR*, abs/1907.05019.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186.

Xiaohan Ding, Xiangxin Zhou, Yuchen Guo, Jungong Han, Ji Liu, et al. 2019. Global sparse momentum sgd for pruning very deep neural networks. *Advances in Neural Information Processing Systems*, 32.

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.

Young Jin Kim, Ammar Ahmad Awan, Alexandre Muzio, Andres Felipe Cruz Salinas, Liyang Lu, Amr Hendy, Samyam Rajbhandari, Yuxiong He, and Hany Hassan Awadalla. 2021. Scalable and efficient moe training for multitask multilingual models. *arXiv preprint arXiv:2109.10465*.

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.

Chen Liang, Haoming Jiang, Simiao Zuo, Pengcheng He, Xiaodong Liu, Jianfeng Gao, Weizhu Chen, and Tuo Zhao. 2022. No parameters left behind: Sensitivity guided adaptive learning rate for training large transformer models. In *International Conference on Learning Representations*.

Zehui Lin, Liwei Wu, Mingxuan Wang, and Lei Li. 2021. Learning language specific sub-network for multilingual 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)*, pages 293–305, 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.

Ekdeep Singh Lubana and Robert Dick. 2021. A gradient flow framework for analyzing network pruning. In *International Conference on Learning Representations*.

Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. 2019. Importance estimation for neural network pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11264–11272.

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. *arXiv preprint arXiv:2207.04672*.

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.

Victor Sanh, Thomas Wolf, and Alexander Rush. 2020. Movement pruning: Adaptive sparsity by fine-tuning. *Advances in Neural Information Processing Systems*, 33:20378–20389.

Aditya Siddhant, Ankur Bapna, Yuan Cao, Orhan Firat, Mia Chen, Sneha Kudugunta, Naveen Arivazhagan, and Yonghui Wu. 2020. Leveraging monolingual data with self-supervision for multilingual neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2827–2835, Online. Association for Computational Linguistics.

Aditya Siddhant, Ankur Bapna, Orhan Firat, Yuan Cao, Mia Xu Chen, Isaac Caswell, and Xavier Garcia. 2022. Towards the next 1000 languages in multilingual machine translation: Exploring the synergy between supervised and self-supervised learning. *arXiv preprint arXiv:2201.03110*.

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. PMLR.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Yiren Wang, ChengXiang Zhai, and Hany Hassan. 2020. Multi-task learning for multilingual neural machine translation. In *Proceedings of the 2020*

Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1022–1034, Online. Association for Computational Linguistics.

Xia Xiao, Zigeng Wang, and Sanguthevar Rajasekaran. 2019. Autoprune: Automatic network pruning by regularizing auxiliary parameters. *Advances in neural information processing systems*, 32.

Haoran Xu, Philipp Koehn, and Kenton Murray. 2022. The importance of being parameters: An intra-distillation method for serious gains. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 170–183, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Haoran Xu, Benjamin Van Durme, and Kenton Murray. 2021. BERT, mBERT, or BiBERT? a study on contextualized embeddings for neural machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6663–6675, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mahsa Yarmohammadi, Shijie Wu, Marc Marone, Haoran Xu, Seth Ebner, Guanghui Qin, Yunmo Chen, Jialiang Guo, Craig Harman, Kenton Murray, Aaron Steven White, Mark Dredze, and Benjamin Van Durme. 2021. Everything is all it takes: A multipronged strategy for zero-shot cross-lingual information extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1950–1967, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Biao Zhang, Ankur Bapna, Rico Sennrich, and Orhan Firat. 2021. Share or not? learning to schedule language-specific capacity for multilingual translation. In *International Conference on Learning Representations*.

A Analysis of Intra-Distillation for eng→xxx

Method	High	Low	Very Low	All
Regular	34.14	11.47	**5.75**	15.71
Intra-Distillation	**35.05**	**13.79**	5.69	**16.07**

Table 5: M8 eng→xx results of regular MMT and MMT with intra-distillation.

Similar to Section 3, the model with intra-distillation outperforms the regular MMT model by a large margin in the eng→xxx direction, as shown in Table 5. We still use a heat map to visualize the PCC of parameter sensitivity lists among every language pair in the eng→xxx direction. In Figure 8, we show that contribution similarity becomes lower as well, which means that the model also learns more language-specific parameters.

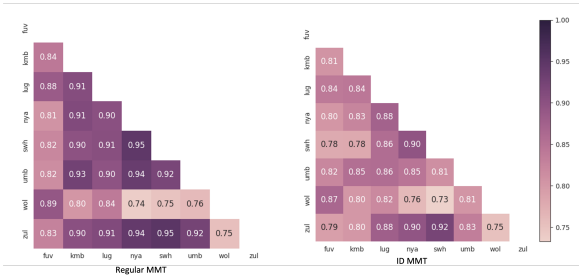


Figure 8: PCC between the list of all parameter sensitivities across every language in the M8 eng→xxx experiments. We compare the similarity between MMT with and without intra-distillation.

We also evaluate the importance of these language-specific parameters by following the same settings in Section 3.3. We conduct one-shot unstructured pruning, starting with the least language-specific parameters. We again see that the average BLEU scores of 8 languages from the model trained with intra-distillation drop slower after more parameters are pruned, indicating that these language-specific parameters learned by intra-distillation are able to preserve more performance.

B More Balanced Parameter Contribution

We compute the sensitivity of all parameters by feeding a set of batches $\mathcal{B}$ that contains all language data in the M8 xxx→eng experiment. We illustrate parameter sensitivity distribution in Figure 10. Aligned with the findings in Xu et al. (2022), the distribution of parameter sensitivity becomes more balanced after using ID.

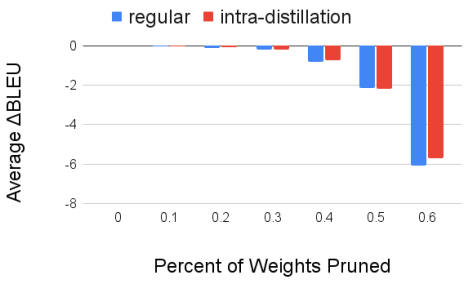


Figure 9: Change of model performance averaged across 8 languages against increasing pruning ratio for the eng→xxx translation task. Models are pruned starting with the least language-specific parameters.

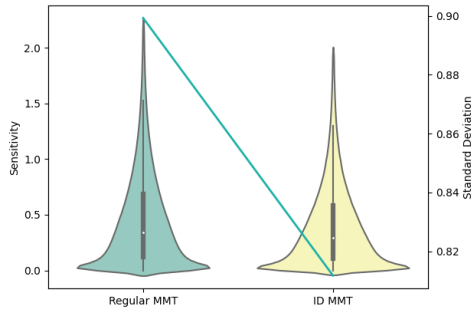


Figure 10: Sensitivity distribution (violin plots aligned with left y-axis) along with their standard deviation (green curve aligned with right y-axis, lower means more balanced parameter contribution). Note that we also remove the top 1% highest-sensitive parameters to ease the illustration.

C Ablation Study on Shared Projection Layer

Since we use a shared projection layer for both encoder and decoder denoising as well as for translation to reduce the model size and save memory, we investigate whether this sharing leads to a performance drop. We conduct experiments on M8 xxx→eng dataset. Table 6 shows that our method with shared layer slightly outperforms the one with separate output projection layers on average.

D M15 Language Information

We give a full account of the 15 languages in the M15 dataset in Table 7.

Method	High	Low	Very Low	All
CD+ID (shared layer)	**35.16**	**15.18**	9.23	**18.69**
CD+ID (NOT shared layer)	35.09	15.04	**9.28**	18.61

Table 6: Comparison of concurrent denoising + intra-distillation with and without using a shared projection layer.

E Effect of Masking Ratio

We take MMT+CD+ID as our study case to investigate the effect of masking ratio $r_m\%$ on the MMT performance. We conduct experiments on M8 xxx→eng. Figure 11 shows that there is no big performance change when we set mask ratio between 0.3 and 0.6.

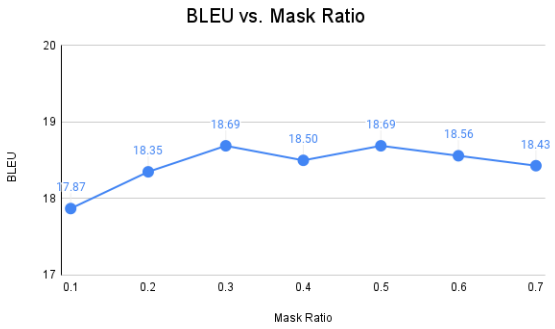


Figure 11: MMT performance change along with masking ratio on the MMT+CD+ID method.

Language	Language id	Parallel Data Size	Resource Level	Language family	Monolingual Data Size
Northern Sotho	nso	526K	Low	Central Narrow Bantu	3.2M
Rundi	run	454K	Low	Central Narrow Bantu	3.8M
Swati	ssw	94K	Very Low	Central Narrow Bantu	1.4M
Indonesian	ind	6.5M	High	Malayio-Polynesian	1.5M
Malay	msa	1M	High	Malayio-Polynesian	15M
Tagalog	tgl	1M	High	Malayo-Polinesian	15M
Bokmål (Norwegian)	nob	238K	Low	North Germanic	2.9M
Icelandic	isl	1M	High	North Germanic	15M
Faroese	fao	4K	Very Low	North Germanic	1.2M
Slovene	slv	15M	High	Southwestern Slavic	13M
Luxembourgish	ltz	8K	Very Low	Western Germanic	5M
Limburgish	lim	5K	Very Low	Western Germanic	8.4M
Catalan	cat	634K	Low	Western Romance	15M
Galician	glg	195K	Low	Western Romance	15M
Friulian	fur	6K	Very Low	Western Romance	730K

Table 7: The information of 15 languages in M15 dataset.