Continual Learning with Semi-supervised Contrastive Distillation for Incremental Neural Machine Translation

Anonymous ACL submission

Abstract

Incrementally expanding the capability of an existing translation model to solve new domain tasks over time is a fundamental and practical problem, which usually suffers from catastrophic forgetting. Generally, multi-domain learning can be seen as a good solution. However, there are two drawbacks: 1) it requires having the training data for all domains available at the same time, which may be unrealistic due to storage or privacy concerns; 2) it requires re-training the model on the data of all domains from scratch when adding a new domain and this is time-consuming and computationally expensive. To address these issues, we present a semi-supervised contrastive distillation framework for incremental neural machine translation. Specifically, to avoid catastrophic forgetting, we propose to exploit unlabeled data from the same distributions of the older domains through knowledge distillation. Further, to ensure the distinct domain characteristics in the model as the number of domains increases, we devise a cross-domain contrastive objective to enhance the distilled knowledge. Extensive experiments on domain translation benchmarks show that our approach, without accessing any previous training data or re-training on all domains from scratch, can significantly prevent the model from forgetting previously learned knowledge while obtaining good performance on the incrementally added domains. [1]

1 Introduction

In the real scenario, translating an out-of-domain sentence is a common situation while it usually cannot work well due to domain discrepancy. An effective solution is to incrementally expand the capability of the existing translation model, *i.e.*, continual learning (Silver et al., 2013). However, the biggest challenge is catastrophic forgetting when the model learns new knowledge and it would forget the previously acquired knowledge (Goodfellow et al., 2013; Gu and Feng, 2020). A theoretically good technique is multi-domain learning, which usually requires having all the training data available at the same time and re-training the model on all domains from scratch. Nevertheless, in practice, it may be unfeasible because we sometimes cannot access the previous data due to storage or privacy concerns, and re-training would bring more training and resource consumption.

To overcome these drawbacks, many efforts have been devoted that fall into three categories, *i.e.*, constructing pseudo data of previous domains/tasks, adding task-specific adapters, and regularization-based learning. (*i*) The first category aims to create pseudo data of the previous task and mix them with the new task data for joint training (Kim and Rush, 2016; Liu et al., 2021; Ko et al., 2021). Although intuitive and effective, they generally require obtaining a large training data of previous tasks and are not flexible in practice. (*ii*) The second category is to add additional task-specific layers for new tasks and only optimizes these parameters with the new task data, having achieved impressive performance (Bapna and Firat, 2019a; Aharoni and Goldberg, 2020; Escolano et al., 2021; Liang et al., 2021; Cao et al., 2021; Gu et al., 2019, 2021). However, the task-specific adapters may increase the difficulty of the model to be aware of which tasks the input belongs to and thus neglect the distinct task characteristics, which limits its application in practice. (*iii*) The third category essentially searches a trade-off between the new task and the previous ones through multi-objective training with an extra penalty item (*e.g.*, L2 or EWC regularization) on the parameters (Khayrallah et al., 2018; Thompson et al., 2019). Therefore, previous methods usually lead to under- or over-constraint problems and achieve a suboptimal performance. Besides, they typically require the parallel data of the previous tasks/domains (Gu et al., 2022) and the time and space cost for computing the penalty item is

[1] The code will be released upon acceptance.

expansive, especially with new tasks/domains appearing (Cao et al., 2021).

In this paper, to address the above issues, we present a Semi-supervised Contrastive Distillation (named SCD) framework for incremental neural machine translation. Specifically, to memorize the learned knowledge from previous domains, we propose to exploit unlabeled data from the same distributions of the older domains through knowledge distillation. To this end, we utilize the source-side data related to the previous domains, *e.g.*, the source-side data of validation set[2], which is small-scale and easy to obtain compared to requiring parallel data. Furthermore, to guarantee distinct domain characteristics in the model as new domains appear, we devise a cross-domain contrastive objective to enhance the distilled knowledge, which encourages the model to learn to keep different domain characteristics and thus benefits translation for various domains.

We validate our proposed SCD framework on the commonly-used machine translation benchmark (Aharoni and Goldberg, 2020), which contains five domains. We incrementally add a single domain at each time to simulate the real-world situation. Extensive experiments show that our model effectively addresses the catastrophic forgetting issue and significantly outperforms related strong methods in terms of BLEU (Papineni et al., 2002) scores, demonstrating its effectiveness.

In summary, our main contributions are:

- We propose a novel continual learning framework for incremental neural machine translation without accessing any previous training data or re-training on all domains from scratch. We also propose a cross-domain contrastive objective to enhance the distilled knowledge to guarantee distinct domain characteristics in the model.
- We conduct extensive experiments and systemic analysis on a more general scenario where m streams of data from different domains are fed to the model sequentially, and our approach can significantly prevent the model from forgetting previously acquired knowledge while obtaining good performance on the newly added domains.
- We show that our method can also achieve better performance only with a handful of unlabeled data than that using a large of parallel data.

[2]Note that the target-side data is not used.

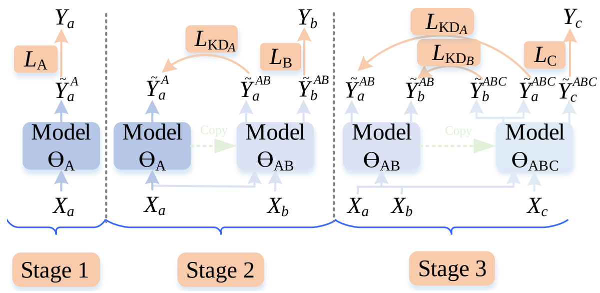

Figure 1: An illustration of incrementally learning three domains. Stage 1: A model Θ_A is trained on a domain A using labeled data with the cross-entropy loss $\mathcal{L}_A$. Here Y_a indicates the reference and $\hat{Y}_a^A$ indicates the translation in domain A by Θ_A. Stage 2: The trained model from Stage 1 is treated as a frozen teacher model. A trainable student Θ_{AB} is a copy from Θ_A and then trained with a loss $\mathcal{L}_B$ for domain B and distillation loss $\mathcal{L}_{\text{KD}_A}$ for domain A. To compute $\mathcal{L}_{\text{KD}_A}$, a set of unlabeled data is used: teacher model's predictions on such dataset for domain A are treated as soft labels and are used against the student model's predictions. In this way, the Θ_{AB} learns to perform domain B and meanwhile tries to keep domain A's knowledge by distillation from the Θ_A. Stage 3: The student Θ_{AB} from Stage 2 acts as the frozen teacher, and a student copy Θ_{ABC} is created to add domain C. The rest of the training process is similar to Stage 2.

2 Methods

In this section, we first describe the problem definition § 2.1. Secondly, we introduce the proposed semi-supervised distillation method in § 2.2, which prevents the model from forgetting the previously learned knowledge. Then, to further ensure the domain characteristics, we present a cross-domain contrastive objective to enhance the distilled knowledge § 2.3. Finally, we elaborate on the training and inference in § 2.4.

2.1 Problem Statement

Domain-incremental training (Cao et al., 2021) aims to simulate training of the NMT model on real-world time streaming data, where the training domain data come from different times and is fed to the model in chronological order. And we indicate (X_a, Y_a) and (X_b, Y_b) as the training translation pairs for domain A and B, respectively. For example, as shown in Fig. 1, the model Θ_A is firstly trained on a domain A. After a period of time, a new domain data B comes. Then, a model Θ_{AB}, which needs to deal with both domains, is trained incrementally based on Model Θ_A without accessing the previous domain data A. The rest of the training process is similar to adding domain B.

2.2 Semi-supervised Distillation

Motivation. To continually learn new domains for translation, we exploit the knowledge distillation (KD) (Hinton et al., 2015) framework. Without loss of generality, we assume that we have already trained a model Θ_A to solve domain A in stage 1 and we want to update it to learn how to also solve a new domain B. As illustrated in Fig. 1, we start by creating a copy of Θ_A for domains A and B, i.e., Θ_{AB}. The original Θ_A and Θ_{AB} models act as the teacher and the student in the KD framework, respectively. During training, we fix the model Θ_A and only update Θ_{AB} with the objective of (1) learning the new domain from the training data of domain B and (2) preserving the older domain's knowledge by minimizing the loss function:

$$\begin{aligned} \mathcal{L}_{\mathrm{AB}}^{\mathrm{KD}} &= \mathcal{L}_{\mathrm{B}} + \alpha \mathcal{L}_{\mathrm{KD}_A}, \\ \mathcal{L}_{\mathrm{B}} &= \mathrm{CE}(Y_b, \Theta_{AB}(X_b)), \\ \mathcal{L}_{\mathrm{KD}_A} &= \mathrm{CE}(\Theta_A(X_a), \Theta_{AB}(X_a)), \end{aligned} \tag{1}$$

where CE denotes the cross-entropy loss and $\mathcal{L}_{\mathrm{KD}_A}$ denotes the CE loss between the token probability distribution of the student on domain A and the soft targets of the teacher Θ_A, and α is the balancing coefficient. Here $\mathcal{L}_{\mathrm{B}}$ serves to let the student learn how to solve a new domain and $\mathcal{L}_{\mathrm{KD}_A}$ helps it in preventing catastrophic forgetting of the old one. In the standard application of KD to continual learning, $\mathcal{L}_{\mathrm{KD}_A}$ is computed on the new domain data (Shmelkov et al., 2017; Cao et al., 2021): this assumes that the old and new domains have the same data distribution (Dakwale and Monz, 2017).

However, the assumption does not satisfy the real-world machine translation where different domains are typically defined on extremely different data distributions. If we use the new domain data to compute the distillation loss, the model will bias the translation toward the new domain style when translating the sentence of the old domain. Therefore, preventing catastrophic forgetting when using only the new domain data can be challenging.

Dealing with Different Domain Distributions. To address this issue, we propose to augment the KD learning process with a data distribution resembling the one used to train the teacher model to solve domain A. Our assumption is that while the original training material for domain A may no longer be available, we can still observe a stream of unlabeled data (X_a) from the same distribution, which is easy to obtain, e.g., the validation set of domain A.

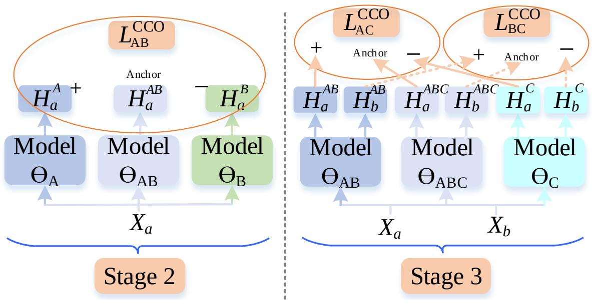

Figure 2: When incrementally learning a new domain, we propose cross-domain contrastive learning objectives to enhance the distilled knowledge to keep distinct domain characteristics.

By this way, the loss function $\mathcal{L}_{\mathrm{KD}_A}$ represents the discrepancy between the teacher and student predictions for the old domains on a set of unlabeled data. In practice, the unlabeled data X_a are automatically labeled by the teacher model Θ_A to produce the soft targets dataset of domain A. This data will be used to compute the loss $\mathcal{L}_{\mathrm{KD}_A}$. Meanwhile, a new labeled dataset for domain B is used to compute $\mathcal{L}_{\mathrm{B}}$. By doing so, the student model should be able to minimize the discrepancy with the teacher on the old domains (*i.e.*, minimizing the catastrophic forgetting) while learning the new domains.

This methodology can be trivially extended to the general case where the teacher is already trained on n domains and the student needs to solve a new domain. In this setting, we need to prevent the catastrophic forgetting of n different domains. We assume the availability of an unlabeled stream of data for each of the old domains to compute the individual distillation losses. For example, for three domains as the stage 3 shown in Fig. 1, the total loss is written as:

$$\begin{aligned} \mathcal{L}_{\mathrm{ABC}}^{\mathrm{KD}} &= \mathcal{L}_{\mathrm{C}} + \alpha(\mathcal{L}_{\mathrm{KD}_A} + \mathcal{L}_{\mathrm{KD}_B}), \\ \mathcal{L}_{\mathrm{C}} &= \mathrm{CE}(Y_c, \Theta_{ABC}(X_c)), \\ \mathcal{L}_{\mathrm{KD}_A} &= \mathrm{CE}(\Theta_{AB}(X_a), \Theta_{ABC}(X_a)), \\ \mathcal{L}_{\mathrm{KD}_B} &= \mathrm{CE}(\Theta_{AB}(X_b), \Theta_{ABC}(X_b)). \end{aligned} \tag{2}$$

In this way, the student model will maintain the relevant knowledge to solve the n domains by distilling it from the teacher on the unlabeled data stream, while also learning how to solve the new domain on the labeled data.

2.3 Cross-domain Contrastive Objective

In domain-incremental NMT, we require the model to simultaneously handle multiple domains and generate domain-aware translations. To guarantee the domain characteristics, we further propose a cross-domain contrastive objective to enhance the distilled knowledge. Particularly, as the stage 2 shown in Fig. 2, we use the output feature of the student model as an anchor feature $\mathbf{H}_{a,i}^{AB}$, and push it close to its original domain representation $\mathbf{H}_{a,i}^{A}$ provided by the teacher model. In contrast, we push apart the irrelevant pairs, *e.g.*, the random one in the mini-batch $\mathbf{H}_{a,j}^{AB}$, $j \neq i$. However, the simple negative sample cannot work well in distinguishing domain characteristics because they are different instances but come from the same domain. Therefore, we design a hard negative that is the same instance but encoded with another model for domain B. In this way, the only difference is that they are encoded by different domain models and thus we can distinguish the domain characteristics between domain A and B. That is, our negative samples include two parts: 1) Easy Negatives X_a^j ($j \neq i$) randomly sampled and encoded by domain model Θ_B; 2) Hard Negative X_a^i encoded with domain model Θ_B. This forces the model Θ_{AB} to capture and distinguish well domain A and domain B. Formally, the cross-domain contrastive training objective is defined by (N is the batch size):

$$\mathcal{L}_{\text{AB}}^{\text{CCO}} = -\log \frac{e^{\text{sim}(\mathbf{H}_{a,i}^{AB}, \mathbf{H}_{a,i}^{A})/\tau}}{e^{\text{sim}(\mathbf{H}_{a,i}^{AB}, \mathbf{H}_{a,i}^{A})/\tau} + \sum_{j=1}^{N} e^{\text{sim}(\mathbf{H}_{a,i}^{AB}, \mathbf{H}_{a,j}^{B})/\tau}}, \tag{3}$$

where $\text{sim}(\cdot, \cdot)$ is the cosine similarity and τ denotes a temperature hyperparameter.

Similarly, as the number of domains increases, we can easily extend Eq. 3 to a general setting. For example, for three domains as the stage 3 shown in Fig. 2, we require two cross-domain contrastive objectives $\mathcal{L}_{\text{AC}}^{\text{CCO}}$ and $\mathcal{L}_{\text{BC}}^{\text{CCO}}$ for domains $A\&C$ and $B\&C$, respectively.

2.4 Training and Inference

At training, we train our model with the following objective at stage 2:

$$\mathcal{J} = \mathcal{L}_{\text{AB}}^{\text{KD}} + \beta \mathcal{L}_{\text{AB}}^{\text{CCO}}, \tag{4}$$

where β is the balancing hyper-parameter.

Note that when training model Θ_{AB}, the model Θ_A and Θ_B are frozen. During inference, only the model Θ_{AB} is used to generate translations for domains A and B. The rest of the training process is similar to the stage 2.

3 Experiments

3.1 Datasets

We use the domain translation dataset proposed by Koehn and Knowles (2017) to simulate the incremental multi-domain setting. The dataset mainly covers five diverse domains: IT, Koran, Law, Medical, and Subtitles, which are available in OPUS (Aulamo and Tiedemann, 2019). Following previous work (Gu and Feng, 2020; Gu et al., 2022), we use the new data splitting released by Aharoni and Goldberg (2020), and perform German to English translation (De→En). Please refer to Tab. 7 of Appendix C for detailed data statistics.

3.2 Metric

For a fair comparison, we follow previous work (Gu et al., 2022) and adopt the 4-gram case-sensitive BLEU with the SacreBLEU tool[3] (Post, 2018) and report the statistical significance test (Koehn, 2004).

3.3 Implementation Details

Following Gu et al. (2022), we use the mBART50-nn (Tang et al., 2020) as our baseline model. Please refer to Appnedix A for detailed settings.

3.4 Comparison Models

Our comparison models consist of two parts: non-continual learning methods and continual learning methods. Please refer to Appendix B for details.

3.5 Main Results

3.5.1 Adding a Second Domain

We investigate different methods for adding a new domain to a model already trained on one domain. In detail, we first fine-tune the mBART50-nn model on one domain. Then, we add another domain to the model through the proposed approach without accessing any training labels for the first domain. The results of all models are shown in Tab. 1.

As hypothesized, when adding the Koran domain to a model fine-tuned on the IT domain, in the regularization-based setting (mBART50-nn (L2-Reg or EWC)) the models are not able to learn the IT domain by only adjusting the model weights with constraint (the BLEU of old domain is about 4 points below the single-domain fine-tuning). Alternatively, the mBART50-nn (TKD) method also cannot prevent the catastrophic forgetting of the

[3]BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a+version.1.4.13

Setting	IT	Koran	Law	Medical	Subtitles	Avg.
Scratch	39.87	18.80	53.96	53.88	27.71	38.84
mBART50-nn	35.65	16.41	41.81	37.21	27.14	31.64
mBART50-nn (Adapter)	37.15	19.38	55.01	56.13	30.89	39.71
mBART50-nn (FT)	39.48	24.04	59.49	58.95	30.78	42.54
mBART50-nn (MDL) [Five Domains]	39.01	23.37	59.37	59.18	30.18	42.22
mBART50-nn (MDL) [IT + Koran]	38.77	23.53	-	-	-	31.15
mBART50-nn (L2-Reg) [IT→Koran]	35.67	23.52	-	-	-	29.60
mBART50-nn (EWC) [IT→Koran]	35.55	23.54	-	-	-	29.55
mBART50-nn (TKD) [IT→Koran]	36.69	**23.57**	-	-	-	30.13
mBART50-nn (LFR-OM) [IT→Koran]	37.47	23.55	-	-	-	30.51
SCD [IT→Koran]	**39.87**†	22.03	-	-	-	**30.95**
mBART50-nn (L2-Reg) [Koran→IT]	38.78	16.57	-	-	-	27.88
mBART50-nn (EWC) [Koran→IT]	38.71	17.04	-	-	-	27.88
mBART50-nn (TKD) [Koran→IT]	**39.40**	19.40	-	-	-	29.40
mBART50-nn (LFR-OM) [Koran→IT]	39.21	20.13	-	-	-	29.67
SCD [Koran→IT]	39.28	**23.15**†	-	-	-	**31.22**†
mBART50-nn (MDL) [IT + Law]	39.45	-	59.92	-	-	49.68
mBART50-nn (L2-Reg) [IT→Law]	29.47	-	59.12	-	-	44.30
mBART50-nn (EWC) [IT→Law]	29.35	-	59.05	-	-	44.20
mBART50-nn (TKD) [IT→Law]	30.70	-	**59.26**	-	-	44.98
mBART50-nn (LFR-OM) [IT→Law]	31.74	-	59.07	-	-	45.41
SCD [IT→Law]	**37.70**†	-	57.33	-	-	**47.52**†
mBART50-nn (L2-Reg) [Law→IT]	38.61	-	50.71	-	-	45.16
mBART50-nn (EWC) [Law→IT]	38.67	-	50.15	-	-	44.41
mBART50-nn (TKD) [Law→IT]	**38.69**	-	51.55	-	-	45.12
mBART50-nn (LFR-OM) [Law→IT]	38.42	-	53.02	-	-	45.72
SCD [Law→IT]	37.89	-	**56.90**†	-	-	**47.40**†
mBART50-nn (MDL) [IT + Medical]	38.91	-	-	59.63	-	49.27
mBART50-nn (L2-Reg) [IT→Medical]	30.87	-	-	58.87	-	44.87
mBART50-nn (EWC) [IT→Medical]	30.13	-	-	59.01	-	45.57
mBART50-nn (TKD) [IT→Medical]	31.35	-	-	**59.07**	-	45.21
mBART50-nn (LFR-OM) [IT→Medical]	32.59	-	-	58.91	-	45.75
SCD [IT→Medical]	**37.70**†	-	-	57.14	-	**47.42**†
mBART50-nn (L2-Reg) [Medical→IT]	38.95	-	-	49.23	-	44.09
mBART50-nn (EWC) [Medical→IT]	38.83	-	-	49.01	-	43.92
mBART50-nn (TKD) [Medical→IT]	**39.72**	-	-	50.24	-	44.98
mBART50-nn (LFR-OM) [Medical→IT]	39.08	-	-	51.04	-	45.06
SCD [Medical→IT]	38.05	-	-	**56.96**†	-	**47.51**†
mBART50-nn (MDL) [IT + Subtitles]	39.66	-	-	-	30.48	35.07
mBART50-nn (L2-Reg) [IT→Subtitles]	29.97	-	-	-	30.33	30.15
mBART50-nn (EWC) [IT→Subtitles]	30.25	-	-	-	30.28	30.27
mBART50-nn (TKD) [IT→Subtitles]	31.54	-	-	-	30.41	30.94
mBART50-nn (LFR-OM) [IT→Subtitles]	32.18	-	-	-	30.71	31.45
SCD [IT→Subtitles]	**38.52**†	-	-	-	**31.00**	**34.76**†
mBART50-nn (L2-Reg) [Subtitles→IT]	38.38	-	-	-	24.77	31.58
mBART50-nn (EWC) [Subtitles→IT]	38.75	-	-	-	24.71	31.73
mBART50-nn (TKD) [Subtitles→IT]	38.89	-	-	-	25.19	32.04
mBART50-nn (LFR-OM) [Subtitles→IT]	38.91	-	-	-	25.48	32.19
SCD [Subtitles→IT]	**39.04**	-	-	-	**30.01**†	**34.52**†

Table 1: Comparison of different continual learning strategies to learn two domains in different orders. "[IT + Koran]" means we mixed both training data to jointly train the model. "[IT→Law]" means Law is added to an IT model. The "SCD" indicates the proposed semi-supervised contrastive distillation method. The best results are in bold. "†" indicates that statistically significant better than "mBART50-nn (LFR-OM)" with t-test $p < 0.01$. The results of the other orders (*e.g.*, [Law→Medical]) are shown in Tab. 8 of Appendix.

Setting	IT	Koran	Law	Medical	Avg.
Scratch	39.87	18.80	53.96	53.88	41.63
mBART50-nn	35.65	16.41	41.81	37.21	32.77
mBART50-nn (Adapter)	37.15	19.38	55.01	56.13	41.92
mBART50-nn (FT)	39.48	24.04	59.49	58.95	45.49
mBART50-nn (MDL) IT + Koran + Law	38.85	23.63	59.19	-	40.55
+ Medical	38.75	23.83	59.49	58.75	45.21
mBART50-nn (LFR-OM) IT→Koran→Law	33.78	19.12	54.25	-	35.71
→Medical	31.50	18.52	41.55	53.37	36.24
SCD IT→Koran→Law	37.88	21.06	56.89	-	**38.61**†
→Medical	34.01	22.93	45.53	55.56	**39.51**†
mBART50-nn (LFR-OM) IT→Law→Koran	31.72	21.27	56.91	-	36.63
→Medical	32.85	15.90	42.56	53.56	36.22
SCD IT→Law→Koran	39.35	21.33	52.31	-	**37.66**†
→Medical	37.91	21.19	46.48	56.30	**40.47**†
mBART50-nn (LFR-OM) Koran→IT→Law	32.56	18.43	54.66	-	35.21
→Medical	31.37	18.05	41.67	53.56	36.16
SCD Koran→IT→Law	37.97	21.58	57.32	-	**38.96**†
→Medical	34.46	23.10	45.58	56.01	**39.79**†
mBART50-nn (LFR-OM) Koran→Law→IT	38.19	17.74	51.74	-	35.89
→Medical	35.46	16.67	44.23	54.45	37.70
SCD Koran→Law→IT	38.47	21.10	56.42	-	**38.66**†
→Medical	36.67	22.31	46.71	56.88	**40.64**†
mBART50-nn (LFR-OM) Law→IT→Koran	31.72	21.27	51.91	-	34.97
→Medical	31.24	20.78	45.78	54.78	38.14
SCD Law→IT→Koran	38.35	21.33	52.31	-	**37.33**†
→Medical	37.22	20.83	47.90	56.92	**40.72**†
mBART50-nn (LFR-OM) Law→Koran→IT	38.19	17.74	51.74	-	35.89
→Medical	35.52	16.56	50.15	54.69	39.23
SCD Law→Koran→IT	38.47	23.10	56.42	-	**39.33**†
→Medical	38.69	21.16	53.49	56.88	**42.56**†

Table 2: mBART50-nn (LFR-OM) and SCD performances when incrementally learning three and four domains. "$D_1 \rightarrow D_2 \rightarrow D_3$" means the mBART50-nn model was fine-tuned for D_1 first. Then D_2 and D_3 were added incrementally. "→Subtitles" rows show the result after further adding the Subtitles domain.

previous domain, as demonstrated by the drop of about 2.5 points in terms of the BLEU score. This is happening to various degrees to all the old domains in all the pairs. We note that the same pattern can also be found for the other domain pairs (*e.g.*, [IT→Law]). Compared with them, the mBART50-nn (LFR-OM) method, to some extent, can keep the performance of the previous domain because they only update these parameters which does not harm the performance of the previous domain. However, this method first needs some parallel data to search such parameters. Given that the drop we observe for mBART50-nn (L2-Reg)&mBART50-nn (EWC)&mBART50-nn (TKD) is generally higher than mBART50-nn (LRF-OM), we will not report their results in the following sections.

In sum, computing the distillation loss with our proposed semi-supervised distillation and cross-domain contrastive objective largely mitigates the catastrophic forgetting issue and keeps the capability of the model to learn the new domain. When adding Koran to an IT-trained model, our model even surpasses the MDL or single-domain fine-tuning methods after the second stage when we use the unlabeled development set (we only use source-side data) of IT domain for distillation (the drop of mBART50-nn (LFR-OM) is about 1.3%). Additionally, the BLEU scores of all models on the Koran when added as a new domain are comparable with each other. *1) This means that the model is able to retain the general linguistic knowledge required to learn the new domain, while also preserving its knowledge of the older domain.* Meanwhile, we observe a similar trend in the reverse setting, where we add IT to a model fine-tuned on the Koran. Finally, this pattern is consistent in other domain pairs as well (*e.g.*, adding IT to Medical or Subtitles).

3.5.2 Adding Third and Fourth Domains

We further investigate the effectiveness of SCD by incrementally learning three and four domains, and

	IT	Koran	Law	Medical	Subtitles	Avg.
Scratch	39.87	18.80	53.96	53.88	27.71	38.84
mBART50-nn	35.65	16.41	41.81	37.21	27.14	31.64
mBART50-nn (Adapter)	37.15	19.38	55.01	56.13	30.89	39.71
mBART50-nn (FT)	39.48	24.04	59.49	58.95	30.78	42.54
Stage 2: Koran added to IT						
mBART50-nn (MDL)	38.77	23.53				31.15
mBART50-nn (LFR-OM)	37.47	23.55				30.51
SCD	39.87	22.03				**30.95**
Stage 3: Law added to [IT→Koran]						
mBART50-nn (MDL)	38.85	23.63	59.19			40.56
mBART50-nn (LFR-OM)	33.78	19.12	54.25			35.75
SCD	37.88	21.06	56.89			**38.61**†
Stage 4: Medical added to [IT→Koran→Law]						
mBART50-nn (MDL)	38.75	23.83	59.49	58.75		45.21
mBART50-nn (LFR-OM)	31.50	18.52	41.55	53.37		36.24
SCD	34.01	22.93	45.53	55.56		**39.51**†
Stage 5: Subtitles added to [IT→Koran→Law→Medical]						
mBART50-nn (MDL)	39.01	23.37	59.37	59.18	30.18	42.22
mBART50-nn (LFR-OM)	30.33	16.98	40.41	50.44	28.72	33.38
SCD	33.15	22.60	44.68	53.21	28.78	**36.48**†
Other orders: Stage 5: IT added to [Koran→Law→Medical→Subtitles]						
mBART50-nn (MDL)	39.01	23.37	59.37	59.18	30.18	42.22
mBART50-nn (LFR-OM)	38.03	16.17	39.12	50.19	23.88	33.48
SCD	38.21	20.10	42.39	52.94	26.41	**36.01**†
Other orders: Stage 5: Koran added to [Law→Medical→Subtitles→IT]						
mBART50-nn (MDL)	39.01	23.37	59.37	59.18	30.18	42.22
mBART50-nn (LFR-OM)	30.33	22.82	38.54	49.87	25.66	33.44
SCD	33.15	22.96	42.19	51.93	27.93	**35.63**†

Table 3: Results of incrementally learning five domains. We first fine-tune a mBART50-nn model on IT. Then we incrementally add Koran, Law, Medical, and Subtitles to that model. The last two groups are the results of other orders.

Setting: Stage 2	**IT**	**Koran**
mBART50-nn (MDL) [IT + Koran]	38.77	23.53
SCD [IT→Koran]	39.87	22.03
w/o semi-supervised distillation	36.69	23.57
w/o $\mathcal{L}_{\text{AB}}^{\text{CCO}}$	37.94	21.82
SCD [Koran→IT]	39.28	23.15
w/o semi-supervised distillation	39.40	19.40
w/o $\mathcal{L}_{\text{AB}}^{\text{CCO}}$	38.93	22.12

Table 4: Ablation Study. "*w/o* semi-supervised distillation" denotes that we do not use unlabeled data of the same distribution of previous domains, *i.e.*, vanilla knowledge distillation.

we report the results with different domain orders in Tab. 2. Results show that our SCD is able to provide useful information to retain the knowledge in the model. For instance, when adding Law to IT and Koran (*i.e.*, It→Koran→Law setting), the BLEU score of IT drops about 5.70% with the mBART50-nn (LFR-OM), while using SCD the drop is only about 1.60% compared to the single-domain fine-tuning model. Notice that this pattern is consistent in almost every domain combination we experimented with.

When adding the fourth domain, we also observe a similar trend to adding the third domain. Besides, we find that *2) the performance of the first domain gets lower with the domain increases, including all methods.* This shows that there is much room for further improvement using other more advanced continuing learning methods.

3.5.3 Incremental Addition of Five Domains

In this section, we explore the effectiveness of SCD by incrementally adding five domains. We also list the results of adding the second, third, and fourth domains for comparison in Tab. 3.

The results show a similar pattern that we observed in Tab. 1 and Tab. 2. That is, our SCD still outperforms mBART50-nn (LFR-OM) in this setting. Incrementally adding a new domain gradually contributes to the forgetting of older domains for both mBART50-nn (LFR-OM) and SCD methods, especially for mBART50-nn (LFR-OM). For example, IT performance drops at each stage, resulting, at the last stage, in a total drop of about 9% drop in BLEU. The reason may be that *3) it is difficult for the mBART50-nn (LFR-OM) method to search such regions that can be freely updated for the previous four domains.* That is the updatable parameters for several domains may be conflicting or none. Even for our proposed SCD, the drop still is 6% BLEU scores, showing that incrementally learning many domains still remains a challenge and is worth studying in the future.

Besides, we report the results of two additional task ordering in the last two blocks of Tab. 3, *i.e.*, [Koran→Law→Medical→Subtitles→IT] and [Law→Medical→Subtitles→IT→Koran]. We observe that despite changing the order of the domain, the outcome is the same. We also find a similar pattern when we experimented with another domain order different from the mentioned ones. Our proposed model has the ability to limit catastrophic forgetting happening to some extent in the continual learning setting.

4 Analysis

4.1 Ablation Study

We conduct ablation studies to investigate how well semi-supervised distillation and cross-domain contrastive objective of SCD works. We conclude two findings from the results in Tab. 4.

(1) "*w/o* semi-supervised distillation": *i.e.*, without using the unlabeled data of the same distribution of the previous domain and using the data of the current domain, the model performance greatly

Models	xx→En	En→xx	El→En	En→El	Sk→En	En→Sk
mBART50-nn (MDL)	18.96	5.88	**30.56**	**26.42**	33.21	**33.75**
mBART50-nn (LFR-OM)	26.94	19.16	28.41	19.98	35.88	30.37
SCD (Ours)	**27.33**	**19.82**	29.15	20.87	**36.81**	31.96

Table 5: Results of Language Adaption. xx→En denotes other languages (*i.e.*, 49 languages supported by mBART50-nn) to English translation.

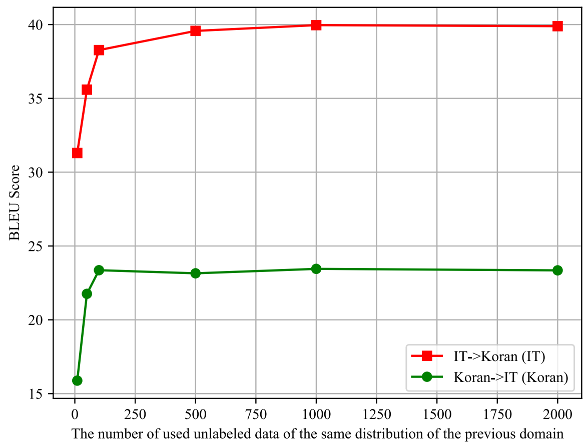

Figure 3: Effect of using different scales of unlabeled data with the same distribution of the previous domain.

Setting: Stage 2	IT	Koran
mBART50-nn (MDL) [IT + Koran]	38.77	23.53
FT [IT → Koran]	30.26	23.49
VKD [IT→Koran]	35.68	23.45
VCL [IT→Koran]	36.72	23.08
SCD [IT→Koran] (Ours)	39.87	22.03
FT [Koran→IT]	39.76	19.15
VKD[Koran→IT]	39.55	21.23
VCL[Koran→IT]	39.79	21.46
SCD [Koran→IT] (Ours)	39.28	23.15

Table 6: Effect of different model variants.

degrades on the older domain and slightly improves the result of the current domain. It shows the necessity of using the data of the same distribution of the previous domain to prevent catastrophic forgetting. Besides, we also find that there is a performance trade-off between older domains and new domains, where the phenomenon is introduced by the hyper-parameter α in Eq. 1. We investigated it in Tab. 9 of the Appendix and actually different hyperparameters have different impacts, which mainly affect the trade-off between older and new domains.

(2) "*w/o* $\mathcal{L}_{\text{AB}}^{\text{CCO}}$": the model performance becomes worse on both domains. This shows that our cross-domain contrastive learning indeed can enhance the distilled knowledge and guarantee the distinct domain characteristics, which thus benefits the model performance on both domains.

4.2 Analysis of Adaptation to New Languages

To investigate whether our approach can apply to new language pairs, we follow Gu et al. (2022) and conducted such experiments on introducing new language pairs, i.e., Greek (El)↔English (En) and Slovak (sk)↔English (En). The results are shown in Tab. 5.

The results show that our approach can significantly surpass the continual learning method, *i.e.*, mBART50-nn (LFR-OM), demonstrating the effectiveness and generality of our method.

4.3 Analysis of Model Variants

In our work, the additional domain model on N_{k+1} is used to provide a hard sample representation for cross-domain contrastive learning. In this setting, we have tried additional three settings: 1) fine-tuning on the N_{k+1} domain with the previously learned domain model (denoted as FT); 2) utilizing vanilla knowledge distillation (VKD), *i.e.*, using the arbitrary unlabeled data; 3) using vanilla contrastive learning (VCL; *i.e.*, only using the sample in the batch as the negative).

The results in 6 show that directly fine-tuning on the target domain without considering previous domains (FT), using vanilla knowledge distillation (VKD) or vanilla contrastive learning (VCL) cannot fully exert their advantage for domain translation. In comparison, cross-domain contrastive distillation has a positive impact on the model performance.

4.4 Effect of Using a Little Unlabeled Data

To further find out how much unlabeled data can achieve a good performance, we randomly sample 10, 50, 100, 500, 1000, and 2000 unlabeled examples from the validation set and use the remaining validation data to choose model checkpoints for evaluating on the test set. In Fig. 3, we observe that the model performance gradually improves and reaches stability as the used unlabeled data increases. Interestingly, we find that the model rapidly achieves a higher result only with a handful of unlabeled data, *i.e.*, 50 and 100 instances for Koran and IT, respectively. It even surpasses the mBART50-nn (LFR-OM) which uses all labeled data in the Koran→IT setting. This shows the superiority of using unlabeled data of the same distribution of the older domain, which can largely help the model retain the learned knowledge of the older domain and prevent catastrophic forgetting. It again indicates the effectiveness of our approach. We also provide a case study to intuitively show how it works in Appendix F.

5 Related Work

Continual Learning of Translation. Recent studies on continual learning of machine translation mainly includes data memory-based method, task-specific adapters, and regularization-based method. Specifically, (1) the data memory-based methods (Chu et al., 2017; Bapna and Firat, 2019a; Xu et al., 2020; Liu et al., 2021) usually require maintaining part or all of the training data of the previous domains/task, which is not flexible in practice and maybe not realistic in the real world due to data privacy. For example, Liu et al. (2021) produce many mixed-language sentences via a bilingual dictionary. (2) The task-specific adapter methods (Bapna and Firat, 2019a; Zeng et al., 2018, 2019; Gu et al., 2019; Cao et al., 2021; Gu et al., 2021; Liang et al., 2021) typically require assigning additional model parameters to different domains/tasks, which requires the model to know which task the input comes from. (3) The regularization-based methods (Khayrallah et al., 2018; Thompson et al., 2019; Dakwale and Monz, 2017) reduce forgetting by introducing an additional penalty term in the learning objective, which may suffer from under- or over-constraint issues. For example, Gu et al. (2022) firstly utilize the previous parallel data to search the low forgetting risk regions and then only update these parameters within the region to largely maintain the performance of the previous domain. Unlike the above work, our method is flexible and free to the requirement of parallel data of the previous domains compared with (1) and (3). Besides, our model does not explicitly lead to model separation against (2).

Knowledge Distillation. KD (Hinton et al., 2015) is to transfer the knowledge (*e.g.*, soft targets outputs) of the stronger model (*aka.* the teacher model) to the small model (*aka.* the student model), which has achieved impressive results in the literature (Kim and Rush, 2016; Wu et al., 2020; Wang et al., 2021; Lee et al., 2019). In neural machine translation, the KD-related work mainly focuses on how to effectively distill the knowledge of the teacher to the student. For example, Zhang et al. (2023) investigate where the knowledge comes from and then carefully design a method to contrapuntally distill the target knowledge. In this work, we aim to utilize the unlabeled development data of the previous domain to prevent catastrophic forgetting of the previous tasks via KD.

Contrastive Learning. The idea of contrastive learning aims to learn effective representation by pulling semantically close neighbors together and pushing apart non-neighbors (Hadsell et al., 2006), which has verified its superiority in many fields, such as model compression (Sun et al., 2020), sentence embedding (Gao et al., 2021), summary (Liu and Liu, 2021; Liang et al., 2023), pretraining (Zhou et al., 2023), and translation (Pan et al., 2021; Lee et al.; Cheng et al., 2022). For example, in neural machine translation, Cheng et al. (2022) propose a contrastive translation memory to enhance the model performance and Pan et al. (2021) utilize the contrastive learning to improve the multilingual neural machine translation. Differently, we introduce a cross-domain contrastive objective to enhance the distilled knowledge, which further guarantees the distinct domain characteristics and thus improves the model performance for several domains. To our knowledge, we are the first that introduce it to prevent catastrophic forgetting.

6 Conclusion

In this paper, we propose a new continual learning framework for incremental neural machine translation without accessing any previous training data or re-training on all domains from scratch. To maintain the performance of the previous domain, we propose to utilize small-scale source-side development data of the previous domain via knowledge distillation. To further ensure distinct domain characteristics in a model, we devise a cross-domain contrastive objective to enhance the distilled knowledge. Extensive experiments on a more general scenario show that our method can achieve significant improvements over several strong baselines.

Limitations

While we show that the SCD achieves significant performance in continual learning of domain adaption translation, there are some limitations worth considering to study in future work: (1) In this study, we only conduct experiments on sequentially five domains, and future work could extend our method to more domains; (2) This work does not conduct experiments on more real-world applications, *e.g.*, sequentially adding different translation tasks (first sentence-level machine translation and then document-level machine translation and more) or multilingual translation task.

References

Roee Aharoni and Yoav Goldberg. 2020. Unsupervised domain clusters in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7747–7763, Online. 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.

Mikko Aulamo and Jörg Tiedemann. 2019. The OPUS resource repository: An open package for creating parallel corpora and machine translation services. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics, NoDaLiDa 2019, Turku, Finland, September 30 - October 2, 2019*, pages 389–394.

Ankur Bapna and Orhan Firat. 2019a. Non-parametric adaptation for 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 1921–1931, Minneapolis, Minnesota. Association for Computational Linguistics.

Ankur Bapna and Orhan Firat. 2019b. Simple, scalable adaptation for neural machine translation. In *Proceedings of EMNLP*, pages 1538–1548, Hong Kong, China. Association for Computational Linguistics.

Antonio Valerio Miceli Barone, Barry Haddow, Ulrich Germann, and Rico Sennrich. 2017. Regularization techniques for fine-tuning in neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 1489–1494.

Yue Cao, Hao-Ran Wei, Boxing Chen, and Xiaojun Wan. 2021. Continual learning for neural machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3964–3974, Online. Association for Computational Linguistics.

Xin Cheng, Shen Gao, Lemao Liu, Dongyan Zhao, and Rui Yan. 2022. Neural machine translation with contrastive translation memories. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3591–3601, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Chenhui Chu, Raj Dabre, and Sadao Kurohashi. 2017. An empirical comparison of simple domain adaptation methods for neural machine translation. *arXiv preprint arXiv:1701.03214*.

Praveen Dakwale and Christof Monz. 2017. Fine-tuning for neural machine translation with limited degradation across in-and out-of-domain data. *Proceedings of the XVI Machine Translation Summit*, page 117.

Carlos Escolano, Marta R. Costa-Jussà, and José A. R. Fonollosa. 2021. From bilingual to multilingual neural-based machine translation by incremental training. *Journal of the Association for Information Science and Technology*, 72(2):190–203.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. 2013. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*.

Shuhao Gu and Yang Feng. 2020. Investigating catastrophic forgetting during continual training for neural machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4315–4326, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Shuhao Gu, Yang Feng, and Qun Liu. 2019. Improving domain adaptation translation with domain invariant and specific information. 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 3081–3091, Minneapolis, Minnesota. Association for Computational Linguistics.

Shuhao Gu, Yang Feng, and Wanying Xie. 2021. Pruning-then-expanding model for domain adaptation of neural machine translation. In *Proceedings* *of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3942–3952, Online. Association for Computational Linguistics.

Shuhao Gu, Bojie Hu, and Yang Feng. 2022. Continual learning of neural machine translation within low forgetting risk regions. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1707–1718, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

R. Hadsell, S. Chopra, and Y. LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742.

Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7).

Huda Khayrallah, Brian Thompson, Kevin Duh, and Philipp Koehn. 2018. Regularized training objective for continued training for domain adaptation in neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 36–44, Melbourne, Australia. Association for Computational Linguistics.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Wei-Jen Ko, Ahmed El-Kishky, Adithya Renduchintala, Vishrav Chaudhary, Naman Goyal, Francisco Guzmán, Pascale Fung, Philipp Koehn, and Mona Diab. 2021. Adapting high-resource NMT models to translate low-resource related languages without parallel data. 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 802–812, Online. Association for Computational Linguistics.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP*, pages 388–395.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation, NMT@ACL 2017, Vancouver, Canada, August 4, 2017*, pages 28–39.

Kibok Lee, Kimin Lee, Jinwoo Shin, and Honglak Lee. 2019. Overcoming catastrophic forgetting with unlabeled data in the wild.

Seanie Lee, Dong Bok Lee, and Sung Ju Hwang. Contrastive learning with adversarial perturbations for conditional text generation. In *International Conference on Learning Representations*.

Jianze Liang, Chengqi Zhao, Mingxuan Wang, Xipeng Qiu, and Lei Li. 2021. Finding sparse structures for domain specific neural machine translation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15):13333–13342.

Yunlong Liang, Fandong Meng, Jiaan Wang, Jinan Xu, Yufeng Chen, and Jie Zhou. 2023. D2tv: Dual knowledge distillation and target-oriented vision modeling for many-to-many multimodal summarization. *arXiv preprint arXiv:2305.12767*.

Yixin Liu and Pengfei Liu. 2021. SimCLS: A simple framework for contrastive learning of abstractive summarization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 1065–1072, Online. Association for Computational Linguistics.

Zihan Liu, Genta Indra Winata, and Pascale Fung. 2021. Continual mixed-language pre-training for extremely low-resource neural machine translation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2706–2718, Online. Association for Computational Linguistics.

Minh-Thang Luong and Christopher D Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 76–79.

Alexander Ly, Maarten Marsman, Josine Verhagen, Raoul PPP Grasman, and Eric-Jan Wagenmakers. 2017. A tutorial on fisher information. *Journal of Mathematical Psychology*, 80:40–55.

Xiao Pan, Mingxuan Wang, Liwei Wu, and Lei Li. 2021. Contrastive learning for many-to-many multilingual neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 244–258, Online. 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 ACL*, pages 311–318.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of WMT*, pages 186–191.

Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. 2017. Incremental learning of object detectors without catastrophic forgetting. In *Proceedings of the IEEE international conference on computer vision*, pages 3400–3409.

Daniel L Silver, Qiang Yang, and Lianghao Li. 2013. Lifelong machine learning systems: Beyond learning algorithms. In *2013 AAAI spring symposium series*.

Siqi Sun, Zhe Gan, Yuwei Fang, Yu Cheng, Shuohang Wang, and Jingjing Liu. 2020. Contrastive distillation on intermediate representations for language model compression. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 498–508, Online. Association for Computational Linguistics.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning. *CoRR*, abs/2008.00401.

Brian Thompson, Jeremy Gwinnup, Huda Khayrallah, Kevin Duh, and Philipp Koehn. 2019. Overcoming catastrophic forgetting during domain adaptation of 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 2062–2068, Minneapolis, Minnesota. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, pages 5998–6008.

Fusheng Wang, Jianhao Yan, Fandong Meng, and Jie Zhou. 2021. Selective knowledge distillation for neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6456–6466.

Yimeng Wu, Peyman Passban, Mehdi Rezagholizadeh, and Qun Liu. 2020. Why skip if you can combine: A simple knowledge distillation technique for intermediate layers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1016–1021, Online. Association for Computational Linguistics.

Jitao Xu, Josep Maria Crego, and Jean Senellart. 2020. Boosting neural machine translation with similar translations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1580–1590.

Jiali Zeng, Yang Liu, Jinsong Su, Yubin Ge, Yaojie Lu, Yongjing Yin, and Jiebo Luo. 2019. Iterative dual domain 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 2019, Hong Kong, China, November 3-7, 2019*, pages 845–855.

Jiali Zeng, Jinsong Su, Huating Wen, Yang Liu, Jun Xie, Yongjing Yin, and Jianqiang Zhao. 2018. Multi-domain neural machine translation with word-level domain context discrimination. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 447–457.

Songming Zhang, Yunlong Liang, Shuaibo Wang, Wenjuan Han, Jian Liu, Jinan Xu, and Yufeng Chen. 2023. Towards understanding and improving knowledge distillation for neural machine translation. *arXiv preprint arXiv:2305.08096*.

Chulun Zhou, Yunlong Liang, Fandong Meng, Jinan Xu, Jinsong Su, and Jie Zhou. 2023. Rc3: Regularized contrastive cross-lingual cross-modal pretraining. *arXiv preprint arXiv:2305.07927*.

A Implementation Details

Following Gu et al. (2022), we use the mBART50-nn (Tang et al., 2020) as our baseline model. The mBART50-nn is a many-to-many multilingual NMT model that can support the translation among 50 different languages. The layer number of its encoder and decoder are both 12, whose attention heads are set as 16. The size of the embedding layer and hidden states is set as 1024, while the layer size of the feed-forward network is 4096. Please refer to Tang et al. (2020) for more details.

At training, we employ the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.98$. We use the inverse square root learning scheduler and set the $warmup\_steps = 4000$. We set $lr = 5e-5$ and train the model 10k steps. All the systems are trained on 8 V100 GPUs with the update frequency 2. The max token is 1024 for each GPU. Besides, we use beam search with the size of 4 and length penalty as 0.6 during decoding. We investigate the factor α and β in Appendix D, which are both set to 0.5.

B Comparison Models

Our comparison models consist of two parts: non-continual learning methods and continual learning methods.

1) non-continual learning methods :
• **Scratch**: We train a vanilla transformer (Vaswani et al., 2017) from scratch only with the training data from the new domain task.
• **mBART50-nn** (Tang et al., 2020) is a large scale pre-trained NMT model. All the following systems are implemented based on this model.
• **mBART50-nn (FT)** (Luong and Manning, 2015): We fine-tune the mBART50-nn model only on individual domain training data.
• **mBART50-nn (MDL)** fine-tune the mBART50-nn model with all domain training data, which is considered the **upper bound** in the field of continual learning. We use the temperature-based sampling function to oversample the validation datasets (Arivazhagan et al., 2019).

2) Continual learning methods , which aim to get a good balance between previous and new domains.
• **mBART50-nn (TKD)** (Dakwale and Monz, 2017): Besides minimizing the training loss of the new domain, this method also minimizes the distillation loss for the previous domain, which is computed on the new domain's training data, *i.e.*, without using any previous data. The training objective based on the mBART50-nn model is:

$$\mathcal{L}_{\mathrm{AB}}^{\mathrm{TKD}} = \mathcal{L}_{\mathrm{B}} + \alpha \mathrm{CE}(\Theta_A(X_b), \Theta_{AB}(X_b)). \quad (5)$$

• **mBART50-nn (L2-Reg)** (Barone et al., 2017) adds an L2-norm regularization on the mBART50-nn model to alleviate the catastrophic forgetting when adding a new domain.
• **mBART50-nn (EWC)** (Thompson et al., 2019) first models the importance of the parameters of the mBART50-nn model with Fisher information matrix (Ly et al., 2017) and then puts more constraints on the important parameters to let them stay close to the original values.
• **mBART50-nn (Adapter)** (Bapna and Firat, 2019b) inject the domain-specific adapter layers into the mBART50-nn model and only update the adapters for different domains.
• **mBART50-nn (LFR-OM)** (Gu et al., 2022) aims to update the parameters within the low forgetting risk regions with the output-based method, which requires the parallel data of the previous domain to search the low forgetting risk regions first.

C Additional Results of Adding a Second Domain

In Tab. 8, we find the same trend as observed in Tab. 1. Besides, we also find that our model always achieves the best results on the older domains while sometimes performing slightly worse on the newly added domain compared with some baselines, e.g., mBART50-nn (TKD). The reason may be that our proposed method (knowledge distillation on the unlabeled data with the same distribution as previous domains and contrastive learning) aims to prevent catastrophic forgetting and does not obtain a better trade-off between previous and new tasks to some extent. Through tuning different hyper-parameters, α and β in the training loss, we observe a further improvement on previous domains without sacrificing the performance on new domains (see Tab. 9). Actually, with more domains added, the advantages of our approach are more evident (Tab. 2 and Tab. 3). Anyway, our method can always achieve the best average results, showing its effectiveness.

D Effect of Hyperparameters α and β

We have investigated the impact of hyperparameters, *i.e.*, α and β. Indeed, different hyperparam-

		Train	Valid	Test
Domain Translation Dataset (De→En)	IT	0.22M	2000	2000
	Koran	18K		
	Law	0.47M		
	Medical	0.25M		
	Subtitles	0.5M		
Language Adaptation Dataset	xx↔En	/	997	1012
	El↔En	1M		
	Sk↔En	1M		

Table 7: The data statistic of the domain translation dataset and language adaptation dataset. The number in Train/Valid/Test columns denotes the number of sentence pairs in each domain/language pair.

Setting	IT	Koran	Law	Medical	Subtitles	Avg.
Scratch	39.87	53.96	53.88	27.71	18.80	38.84
mBART50-nn	35.65	41.81	37.21	27.14	16.41	31.64
mBART50-nn (Adapter)	37.15	19.38	55.01	56.13	30.89	39.71
mBART50-nn (FT)	39.48	59.49	58.95	30.78	24.04	42.54
mBART50-nn (MDL) [Five Domains]	39.01	59.37	59.18	30.18	23.37	42.22
mBART50-nn (MDL) [Koran + Law]	-	23.92	59.97	-	-	41.94
mBART50-nn (L2-Reg) [Koran→Law]	-	16.51	59.21	-	-	37.86
mBART50-nn (EWC) [Koran→Law]	-	17.41	59.33	-	-	38.37
mBART50-nn (TKD) [Koran→Law]	-	17.90	59.39	-	-	38.64
mBART50-nn (LFR-OM) [Koran→Law]	-	18.55	**59.41**	-	-	38.98
SCD [Koran→Law]	-	**22.71**†	58.63	-	-	**40.67**†
mBART50-nn (L2-Reg) [Law→Koran]	-	22.95	54.74	-	-	38.85
mBART50-nn (EWC) [Law→Koran]	-	23.12	55.39	-	-	39.25
mBART50-nn (TKD) [Law→Koran]	-	**23.28**	55.88	-	-	39.58
mBART50-nn (LFR-OM) [Law→Koran]	-	23.09	56.11	-	-	39.60
SCD [Law→Koran]	-	22.07	**58.87**†	-	-	**40.47**†
mBART50-nn (MDL) [Koran + Medical]	-	23.96	-	58.94	-	41.45
mBART50-nn (L2-Reg) [Koran→Medical]	-	15.44	-	58.93	-	37.19
mBART50-nn (EWC) [Koran→Medical]	-	16.05	-	58.99	-	37.52
mBART50-nn (TKD) [Koran→Medical]	-	16.60	-	**59.13**	-	37.87
mBART50-nn (LFR-OM) [Koran→Medical]	-	17.38	-	59.01	-	38.20
SCD [Koran→Medical]	-	**22.97**†	-	58.04	-	**40.51**†
mBART50-nn (L2-Reg) [Medical→Koran]	-	23.11	-	54.96	-	39.04
mBART50-nn (EWC) [Medical→Koran]	-	23.24	-	55.05	-	39.15
mBART50-nn (TKD) [Medical→Koran]	-	**23.65**	-	55.59	-	39.62
mBART50-nn (LFR-OM) [Medical→Koran]	-	23.50	-	55.91	-	39.70
SCD [Medical→Koran]	-	21.57	-	**58.17**†	-	**39.87**
mBART50-nn (MDL) [Koran + Subtitles]	-	23.84	-	-	30.61	27.23
mBART50-nn (L2-Reg) [Koran→Subtitles]	-	16.71	-	-	30.18	23.45
mBART50-nn (EWC) [Koran→Subtitles]	-	16.26	-	-	30.21	23.24
mBART50-nn (TKD) [Koran→Subtitles]	-	15.14	-	-	30.68	22.91
mBART50-nn (LFR-OM) [Koran→Subtitles]	-	18.11	-	-	30.54	24.33
SCD [Koran→Subtitles]	-	**21.85**†	-	-	**30.91**	**26.38**†
mBART50-nn (L2-Reg) [Subtitles→Koran]	-	22.75	-	-	21.19	21.97
mBART50-nn (EWC) [Subtitles→Koran]	-	22.87	-	-	21.47	22.12
mBART50-nn (TKD) [Subtitles→Koran]	-	**23.78**	-	-	19.23	21.51
mBART50-nn (LFR-OM) [Subtitles→Koran]	-	23.45	-	-	24.58	24.02
SCD [Subtitles→Koran]	-	22.44	-	-	**30.09**†	**26.27**†
mBART50-nn (MDL) [Law + Medical]	-	-	59.21	58.50	-	58.85
mBART50-nn (L2-Reg) [Law→Medical]	-	-	46.87	58.67	-	52.77
mBART50-nn (EWC) [Law→Medical]	-	-	47.92	58.79	-	53.34
mBART50-nn (TKD) [Law→Medical]	-	-	45.71	**59.09**	-	52.40
mBART50-nn (LFR-OM) [Law→Medical]	-	-	49.88	59.03	-	54.46
SCD [Law→Medical]	-	-	**55.02**†	56.90	-	**55.96**†
mBART50-nn (L2-Reg) [Medical→Law]	-	-	59.45	47.94	-	53.70
mBART50-nn (EWC) [Medical→Law]	-	-	59.39	49.23	-	54.31
mBART50-nn (TKD) [Medical→Law]	-	-	**59.58**	46.42	-	53.05
mBART50-nn (LFR-OM) [Medical→Law]	-	-	59.31	49.19	-	54.25
SCD [Medical→Law]	-	-	57.37	**54.05**†	-	**55.71**†
mBART50-nn (MDL) [Law + Subtitles]	-	-	59.49	-	30.70	45.09
mBART50-nn (L2-Reg) [Law→Subtitles]	-	-	49.48	-	30.37	39.92
mBART50-nn (EWC) [Law→Subtitles]	-	-	49.87	-	30.39	40.13
mBART50-nn (TKD) [Law→Subtitles]	-	-	47.90	-	**30.65**	39.28
mBART50-nn (LFR-OM) [Law→Subtitles]	-	-	51.06	-	30.41	40.74
SCD [Law→Subtitles]	-	-	**56.33**†	-	**30.65**	**43.49**†
mBART50-nn (L2-Reg) [Subtitles→Law]	-	-	58.83	-	24.38	41.61
mBART50-nn (EWC) [Subtitles→Law]	-	-	59.01	-	24.84	41.92
mBART50-nn (TKD) [Subtitles→Law]	-	-	**59.34**	-	22.18	40.76
mBART50-nn (LFR-OM) [Subtitles→Law]	-	-	59.11	-	25.02	42.06
SCD [Subtitles→Law]	-	-	57.80	-	**29.59**†	**43.70**†
mBART50-nn (MDL) [Medical + Subtitles]	-	-	-	58.67	30.51	44.59
mBART50-nn (L2-Reg) [Medical→Subtitles]	-	-	-	48.03	30.46	39.25
mBART50-nn (EWC) [Medical→Subtitles]	-	-	-	47.92	30.62	39.27
mBART50-nn (TKD) [Medical→Subtitles]	-	-	-	46.18	**30.67**	38.42
mBART50-nn (LFR-OM) [Medical→Subtitles]	-	-	-	51.23	30.51	40.87
SCD [Medical→Subtitles]	-	-	-	**56.04**†	30.60	**43.32**†
mBART50-nn (L2-Reg) [Subtitles→Medical]	-	-	-	58.14	23.15	40.65
mBART50-nn (EWC) [Subtitles→Medical]	-	-	-	58.16	23.61	40.88
mBART50-nn (TKD) [Subtitles→Medical]	-	-	-	**58.48**	21.69	40.08
mBART50-nn (LFR-OM) [Subtitles→Medical]	-	-	-	58.31	25.12	41.72
SCD [Subtitles→Medical]	-	-	-	57.17	**29.24**†	**43.21**†

Table 8: Comparison of different continual learning strategies to learn two domains in different orders. "[Law + Medical]" means we mixed law and medical training data to jointly train the model. "[Law→Medical]" means Medical is added to a Law model. The best results are in bold. "†" indicates that statistically significant better than "mBART50-nn (LFR-OM)" with t-test $p < 0.01$.

eters have different impacts, which mainly affect the trade-off between older and new domains. For example, in IT→Koran direction, the results are shown in Tab. 9. In our experiments, we choose $\alpha = 0.5$ and $\beta = 0.5$ to achieve a better trade-off performance between older and new domains.

E Training Efficiency

All our experiments are conducted on 8 V100 GPUs. The average running time is listed as follows (corresponding to different models in the Koran→IT setting of Table 1 with 10 epochs).

The results show that our method consumes slightly more time to train our model while achieving a significantly better performance. The inference time of all models costs the nearly same time due to the same model architecture.

F Case Study

We listed an example here and will add more case studies in the new version. In the IT→Koran setting, we first trained a model on the IT domain denoted as model-1. Then, we fine-tune model-1 on the Koran domain denoted as Model-2. Model-3 and Model-4 indicate mBART50-nn (LFR-OM) and our proposed method, respectively. The instance below is from the test set of the IT domain.

We can observe that model-1 can translate the domain word "Speicher" well after training on the IT domain. Unfortunately, after further fine-tuning on the Koran domain, the model forgets the previously learned domain knowledge and incorrectly translates "Speicher" to "storage". Besides, Model-3, which aims to update the parameters within the low forgetting risk regions with

α	β	IT	Koran
0.1	0.1	38.98	23.31
0.3	0.3	39.21	22.96
0.5	0.5	39.87	22.03
0.7	0.7	39.91	21.88
0.9	0.9	39.97	21.65
1.0	1.0	39.94	21.72

Table 9: Effect of Hyperparameters.

Models	Training Time (h: hour; m: minute)
mBART50-nn (MDL)	8h36m
mBART50-nn (L2-reg)	9h6m
mBART50-nn (EWC)	9h31m
mBART50-nn (TKD)	9h10m
mBART50-nn (LFR-OM)	8h55m + 20m preprocessed search time.
SCD (Ours)	9h22m

Table 10: Training time of different models.

Setting: Stage 2	**IT**	**Koran**
mBART50-nn (MDL) $_{\text{[IT + Koran]}}$	38.77	23.53
baseline $_{\text{[IT}\rightarrow\text{Koran]}}$	33.45	23.33
w/ semi-supervised distillation	37.94	21.82
w/ $\mathcal{L}_{\text{AB}}^{\text{CCO}}$	36.69	23.57
w/ both	39.87	22.03
baseline $_{\text{[Koran}\rightarrow\text{IT]}}$	38.82	17.23
w/ semi-supervised distillation	38.93	22.12
w/ $\mathcal{L}_{\text{AB}}^{\text{CCO}}$	39.40	19.40
w/ both	39.28	23.15

Table 11: Ablation Study. We add our approach one by one to show their performance.

the output-based method to prevent forgetting, still cannot address this case. However, our model can accurately translate it, which demonstrates that our model indeed can prevent from forgetting of previously learned domain knowledge and alleviate the forgetting problem compared to other methods.

Source (German)	Wenn der optionale Parameter small TRUE ist, wird ein alternative Dekomprimierungsalgorithmus verwendet, der weniger Speicher benötigt, jedoch nur halb so schnell läuft.
Reference (English)	If the optional parameter small is TRUE, an alternative decompression algorithm will be used which uses less memory (the maximum memory requirement drops to around 2300K) but works at roughly half the speed.
Model-1	If the optional parameter is small TRUE, an alternative decompression algorithm is used, which uses less memory but is only half as fast.
Model-2	If the optional parameter is small TRUE, an alternative decompression algorithm is used, which requires less storage, but runs half as fast.
Model-3	If the optional parameter is small, then an alternative decompression algorithm is used, which takes less storage but is half as fast.
Model-4 (Ours)	If the optional parameter is small TRUE, an alternative decompression algorithm is used, which consumes less memory but is only half as fast.

Table 12: Case Study.