Continual Knowledge Distillation for Neural Machine Translation

Yuanchi Zhang[1,3], **Peng Li**[*2,4], **Maosong Sun**[1,3], **Yang Liu**[*1,2,3,4]
[1]Dept. of Comp. Sci. & Tech., Institute for AI, Tsinghua University, Beijing, China
[2]Institute for AI Industry Research (AIR), Tsinghua University, Beijing, China
[3]Beijing National Research Center for Information Science and Technology
[4]Shanghai Artificial Intelligence Laboratory, Shanghai, China
yuanchi-21@mails.tsinghua.edu.cn; lipeng@air.tsinghua.edu.cn
{sms,liuyang2011}@tsinghua.edu.cn

Abstract

While many parallel corpora are not publicly accessible for data copyright, data privacy and competitive differentiation reasons, trained translation models are increasingly available on open platforms. In this work, we propose a method called continual knowledge distillation to take advantage of existing translation models to improve one model of interest. The basic idea is to sequentially transfer knowledge from each trained model to the distilled model. Extensive experiments on Chinese-English and German-English datasets show that our method achieves significant and consistent improvements over strong baselines under both homogeneous and heterogeneous trained model settings and is robust to malicious models.[1]

1 Introduction

Current neural machine translation (NMT) systems often face such a situation: parallel corpora are not publicly accessible but trained models are more readily available. On the one hand, many data owners are usually unwilling to share their parallel corpora with the public for data copyright, data privacy and competitive differentiation reasons, leading to recent interests in federated learning for NMT (Wang et al., 2021b; Roosta et al., 2021). On the other hand, trained NMT models are increasingly available on platforms such as Huggingface (`https://huggingface.co`) and Opus-MT (`https://opus.nlpl.eu/Opus-MT`) since these models can be directly used without public access to the original training data.

As a result, a question naturally arises: *can we take advantage of increasingly available trained NMT models to enhance one NMT model of interest?* In this work, we propose a method called **Continual Knowledge Distillation** (CKD) to address

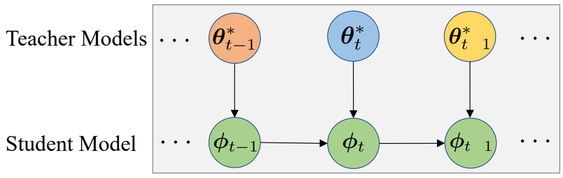

Figure 1: Continual knowledge distillation for neural machine translation. Knowledge is continually distilled from a sequence of teacher models to one student model. At each time step, the current student model (i.e., ϕ_t) fuses the knowledge transferred from both the current teacher model (i.e., $\boldsymbol{\theta}_t^*$) and the previous student model (i.e., ϕ_{t-1}). All teacher models are frozen and the student model is trainable. Different models are highlighted in different colors.

this problem for NMT. As shown in Figure 1, we assume that multiple trained NMT models (i.e., *teachers*) are available to "educate" one NMT model of interest (i.e., *student*) in a sequential manner, which means that teacher models to arrive in the future are not accessible at the current time step. We also assume that the training set of the student model, a transfer set, and a test set are available, but the training set of the teachers are unavailable. CKD aims to continually improve the translation performance of the student model on the test set by sequentially distilling knowledge from each incoming teacher model to the student model.

As its name suggests, CKD is an intersection of knowledge distillation (Hinton et al., 2015) and continual learning (Kirkpatrick et al., 2017). On the one hand, CKD differs from standard knowledge distillation in that the knowledge is transferred from teacher models to the student model asynchronously instead of synchronously. As a result, the knowledge transferred to the student model from previous teacher models can be overridden by an incoming teacher model, which is often referred to as the *catastrophic forgetting* problem (Kirkpatrick et al., 2017). The situation aggravates when not all teacher models convey knowledge benefi-

[*]Corresponding authors.
[1]The source code is available at `https://github.com/THUNLP-MT/CKD`.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 7978–7996

cial to the student model. On the other hand, CKD is different from conventional continual learning methods by focusing on learning one task (i.e., enhancing the student model) rather than learning many different tasks. The learning process is still very challenging as compared with standard continual learning because the original training data of teacher models is inaccessible to the student model. Consequently, we have to resort to knowledge distillation at each time step to make the most of teacher models.

To address these aforementioned challenges, we propose to fuse two knowledge sources for the student model at each time step: filtering the new knowledge from the current teacher model (i.e., *knowledge filtration*) and inheriting the old knowledge from the previous student model (i.e., *knowledge inheritance*) simultaneously. Experimental results show that our method significantly and consistently outperforms strong baselines under both homogeneous and heterogeneous teacher settings for Chinese-to-English and German-to-English translation. And it is also robust to malicious teachers.

2 Approach

2.1 Problem Statement

Let $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1^*, \boldsymbol{\theta}_2^*, \ldots\}$ be a sequence of frozen trained NMT models (i.e., *teacher models*), where $\boldsymbol{\theta}_t^*$ denotes the t-th teacher model. Let $\boldsymbol{\phi}_0$ be an NMT model of interest (i.e., *student model*) and $\boldsymbol{\phi}_t$ be the student model at time step t. We use $\mathbf{x} = x_1, \ldots, x_I$ to denote a *source-language sentence* and $\mathbf{y} = y_1, \ldots, y_J$ to denote a *target-language sentence*. We use $\mathbf{y}_{<j} = y_1, \ldots, y_{j-1}$ to denote a *partial translation*. $D_{\text{train}} = \{\langle \mathbf{x}^{(m)}, \mathbf{y}^{(m)} \rangle\}_{m=1}^{M}$ represents the *training set* of the student model. $D_{\text{trans}} = \{\langle \mathbf{x}^{(n)}, \mathbf{y}^{(n)} \rangle\}_{n=1}^{N}$ represents the *transfer set* that a teacher model uses to "educate" the student model. D_{test} is a *test set* used to evaluate the student model. We use $\text{BLEU}(D_{\text{test}}, \boldsymbol{\phi}_t)$ to denote the BLEU score the student model at time step t obtains on the test set.

Given an initial student model $\boldsymbol{\phi}_0$, our goal is to maximize $\text{BLEU}(D_{\text{test}}, \boldsymbol{\phi}_t)$ by taking advantage of $\boldsymbol{\Theta}$, D_{train}, and D_{trans}.

2.2 Training Objective

As shown in Figure 1, the student model $\boldsymbol{\phi}_t$ at time step t is determined by the current teacher model $\boldsymbol{\theta}_t^*$ that encodes new knowledge and the previous learned student model $\hat{\boldsymbol{\phi}}_{t-1}$ that encodes previously learned knowledge. Therefore, the overall training objective of CKD is composed of three loss functions:

$$\begin{aligned}\ell(&\boldsymbol{\phi}_t, \boldsymbol{\theta}_t^*, \hat{\boldsymbol{\phi}}_{t-1}, D_{\text{train}}, D_{\text{trans}}) \\ &= \ell_{\text{CE}}(\boldsymbol{\phi}_t, D_{\text{train}}) + \lambda \ell_{\text{KF}}(\boldsymbol{\phi}_t, \boldsymbol{\theta}_t^*, D_{\text{trans}}) \\ &\qquad + (1-\lambda)\ell_{\text{KI}}(\boldsymbol{\phi}_t, \hat{\boldsymbol{\phi}}_{t-1}, D_{\text{trans}}), \quad (1)\end{aligned}$$

where $\ell_{\text{CE}}(\boldsymbol{\phi}_t, D_{\text{train}})$ is the standard cross entropy loss defined as

$$\begin{aligned}\ell_{\text{CE}}&(\boldsymbol{\phi}_t, D_{\text{train}}) \\ &= -\sum_{m=1}^{M} \sum_{j=1}^{J^{(m)}} P(y_j^{(m)} | \mathbf{y}_{<j}^{(m)}, \mathbf{x}^{(m)}; \boldsymbol{\phi}_t), \quad (2)\end{aligned}$$

Note that $J^{(m)}$ is the length of the m-th target sentence $\mathbf{y}^{(m)}$. In Eq. 1, $\ell_{\text{KF}}(\boldsymbol{\phi}_t, \boldsymbol{\theta}_t^*, D_{\text{trans}})$ is a *knowledge filtration* loss (see Sec. 2.3) that filters the knowledge transferred from $\boldsymbol{\theta}_t^*$, $\ell_{\text{KI}}(\boldsymbol{\phi}_t, \hat{\boldsymbol{\phi}}_{t-1}, D_{\text{trans}})$ is a *knowledge inheritance* loss (see Sec. 2.4) that inherits the knowledge transferred from $\hat{\boldsymbol{\phi}}_{t-1}$, and λ is a hyper-parameter that balances the preference between receiving new and inheriting old knowledge.

Therefore, the learned student model at time step t can be obtained by

$$\hat{\boldsymbol{\phi}}_t = \underset{\boldsymbol{\phi}_t}{\text{argmin}} \Big\{ \ell(\boldsymbol{\phi}_t, \boldsymbol{\theta}_t^*, \hat{\boldsymbol{\phi}}_{t-1}, D_{\text{train}}, D_{\text{trans}}) \Big\}. \quad (3)$$

2.3 Knowledge Filtration

In standard knowledge distillation (Hinton et al., 2015), an important assumption is that the teacher model is "stronger" than the student model, which means that the teacher model contains knowledge that can help improve the student model. Unfortunately, this assumption does not necessarily hold in our problem setting because it is uncertain what the next incoming teacher model will be. As a result, there are two interesting questions:

1. How do we know whether the teacher model contains knowledge useful to the student model?

2. How do we locate and transfer the useful knowledge from the teacher model to the student model?

Intuitively, the teacher and the student can do the same "test paper" in order to find where the

Source	布朗 *bulang*	教授 *jiaoshou*	将 *jiang*	在 *zai*	会议 *huiyi*	上 *shang*	发表 *fabiao*	演讲 *yanjiang*	
Target	Prof.	Brown	will	make	a	speech	at	the	meeting
Teacher	**0.6**	**0.7**	0.3	**0.9**	**0.5**	0.2	**0.7**	**0.5**	**0.7**
Student	0.1	0.2	**0.5**	0.3	0.3	**0.6**	0.5	0.4	0.2

Teacher > Student   Teacher   Student

Figure 2: An example that illustrates how to find where a teacher model can help a student model. Given a sentence pair of the transfer set, both the teacher and student models try to predict a target word given the source sentence and the partial translation. How well a model predicts can be quantified as a real-valued number. The target words on which the teacher performs better than the student are highlighted in red. Other words are highlighted in blue.

teacher can help the student. Figure 2 shows an example. Given a (Romanized) Chinese sentence and its English translation, both the teacher and student models predict every target word y_j given the source sentence $\mathbf{x}$ and the partial translation $\mathbf{y}_{<y}$. The quality of a prediction can be quantified as a real-valued number. If the teacher model performs better than the student model on a target word (e.g., "Prof."), it is likely that the teacher model contains knowledge useful to the student model in this case. On the contrary, the teacher model is probably not more knowledgable than the student model regarding this case if its prediction is worse than that of the student (e.g., "will").

More formally, we use $Q(y_j, \mathbf{y}_{<j}, \mathbf{x}, \boldsymbol{\phi})$ to quantify how well a student model predicts a target token. It can be defined in the following ways: [2]

1. *Token entropy:* calculating the entropy of target tokens without using the ground truth token.

$$Q(y_j, \mathbf{y}_{<j}, \mathbf{x}, \boldsymbol{\phi}) = -\sum_{y \in \mathcal{Y}} P(y|\mathbf{y}_{<j}, \mathbf{x}; \boldsymbol{\phi}) \times \log P(y|\mathbf{y}_{<j}, \mathbf{x}; \boldsymbol{\phi}) \quad (4)$$

where $\mathcal{Y}$ is the vocabulary of the target language.

2. *Hard label matching*: checking whether the predicted token is identical to the ground truth.

$$Q(y_j, \mathbf{y}_{<j}, \mathbf{x}, \boldsymbol{\phi}) = \delta\Big(y_j, \operatorname*{argmax}_{y} P(y|\mathbf{y}_{<j}, \mathbf{x}; \boldsymbol{\phi})\Big) \quad (5)$$

where $\delta(y, y')$ returns 1 if y is identical to y' and 0 otherwise.

3. *Token-level cross entropy:* calculating token-level cross entropy using the given model.

$$Q(y_j, \mathbf{y}_{<j}, \mathbf{x}, \boldsymbol{\phi}) = -\log P(y_j|\mathbf{y}_{<j}, \mathbf{x}; \boldsymbol{\phi}) \quad (6)$$

The quantification function for a teacher model $Q(y_j, \mathbf{y}_{<j}, \mathbf{x}, \boldsymbol{\theta})$ can be defined likewise.

Since the transfer set D_{trans} can be equivalently seen as a collection of tuples

$$\Big\{\langle y_j^{(n)}, \mathbf{y}_{<j}^{(n)}, \mathbf{x}^{(n)}\rangle \Big| j \in [1, J^{(n)}], n \in [1, N]\Big\}, \quad (7)$$

it can be divided into two parts depending on the comparison between the predictions of teacher and student models: a *positive subset* D^+_{trans} and a *negative subset* D^-_{trans}. A tuple $\langle y_j, \mathbf{y}_{<j}, \mathbf{x}\rangle$ belongs to D^+_{trans} if $Q(y_j, \mathbf{y}_{<j}, \mathbf{x}, \boldsymbol{\theta}^*_t)$ is greater than $Q(y_j, \mathbf{y}_{<j}, \mathbf{x}, \boldsymbol{\phi}_t)$. Otherwise, it is a negative instance that belongs to D^-_{trans}.

After splitting the transfer set into two parts, it is natural to apply standard knowledge distillation using the positive subset D^+_{trans}:

$$\ell_{\text{KD}}(\boldsymbol{\phi}_t, \boldsymbol{\theta}^*_t, D^+_{\text{trans}}) = \sum_{\langle y_j, \mathbf{y}_{<j}, \mathbf{x}\rangle \in D^+_{\text{trans}}} \text{KL}\Big(P(y_j|\mathbf{y}_{<j}, \mathbf{x}; \boldsymbol{\theta}^*_t) \Big|\Big| P(y_j|\mathbf{y}_{<j}, \mathbf{x}; \boldsymbol{\phi}_t)\Big) \quad (8)$$

[2] Note that it is also possible to define sentence-level quantification functions $Q(\mathbf{y}, \mathbf{x}, \boldsymbol{\phi})$. If the teacher performs better than the student at sentence-level predictions, all target words with the sentence are considered positive instances for knowledge transfer. As our preliminary experiments show that the more fine-grained word-level quantification functions are much better than its sentence-level counterparts, we omit the discussion of sentence-level quantification functions due to the space limit.

However, one problem is that D^{+}_{trans} may be very small in most cases in practice, making training efficiency very low.

Therefore, instead of discarding the negative subset D^{-}_{trans}, we introduce a new loss function to make the most of negative instances. In analogy to humans, teachers can educate students by telling them what not to do. We expect that the student model can learn from D^{-}_{trans} in the same way. Our intuition is that erroneous tokens with a high probability in teacher model's output distribution are critical because the student is prone to make the same mistakes. Pushing the output distribution of the student model away from the poor target distribution may enable the student model to avoid making the same mistakes. As a result, D^{-}_{trans} can be leveraged effectively and the overall learning efficiency will be improved significantly. Accordingly, the negative KD loss function on the negative subset is defined as

$$\ell_{\text{NEG}}(\boldsymbol{\phi}_t, \boldsymbol{\theta}^*_t, D^{-}_{\text{trans}}) = \min\Big(0, \alpha - \ell_{\text{KD}}(\boldsymbol{\phi}_t, \boldsymbol{\theta}^*_t, D^{-}_{\text{trans}})\Big) \quad (9)$$

where α is a hyper-parameter that controls the activation of the loss.

Finally, the knowledge filtration loss is the combination of the two functions:

$$\ell_{\text{KF}}(\boldsymbol{\phi}_t, \boldsymbol{\theta}^*_t, D_{\text{trans}}) = \ell_{\text{KD}}(\boldsymbol{\phi}_t, \boldsymbol{\theta}^*_t, D^{+}_{\text{trans}}) + \ell_{\text{NEG}}(\boldsymbol{\phi}_t, \boldsymbol{\theta}^*_t, D^{-}_{\text{trans}}). \quad (10)$$

2.4 Knowledge Inheritance

To circumvent the catastrophic forgetting problem, we introduce a loss function to inherit knowledge learned from previous time step for the current student model:

$$\ell_{\text{KI}}(\boldsymbol{\phi}_t, \hat{\boldsymbol{\phi}}_{t-1}, D_{\text{trans}}) = \sum_{\langle y_j, \mathbf{y}_{<j}, \mathbf{x}\rangle \in D_{\text{trans}}} \text{KL}\Big(P(y_j|\mathbf{y}_{<j}, \mathbf{x}; \hat{\boldsymbol{\phi}}_{t-1}) || P(y_j|\mathbf{y}_{<j}, \mathbf{x}; \boldsymbol{\phi}_t)\Big). \quad (11)$$

3 Experiments

To evaluate the effectiveness of our method, we conduct experiments on Chinese-to-English and German-to-English translation under three representative settings including homogeneous, heterogeneous and malicious teacher settings.

Model	Domain	Training	Dev.	Test
A	News	1,250,000	4,000	13,000
B	Oral	2,500,000	4,000	12,000
C	Internet	750,000	4,000	13,000
D	Speech	220,000	4,000	5,000
E	Subtitle	300,000	4,000	4,000

Table 1: The domain, training and evaluation corpora of the five Transformer-base models used in the Chinese-to-English experiments. More details of the datasets are provided in Appendix A.

3.1 Setup

Configurations. For the Chinese-to-English translation experiments under the homogeneous teacher setting, both the teachers and the student are Transformer-base models (Vaswani et al., 2017). Besides model architecture, there are a few other factors that may affect performance, e.g., teacher performance, student performance, model domain, and the order that the teachers arrive. To investigate the impact of model performance and model domain, we leverage five parallel corpora of representative domains as shown in Table 1, among which two are in million scale, one is in middle scale, and the other two are in small scale. Correspondingly, five Transformer-base models are trained on these corpora, denoted as A, B, C, D, and E, respectively. Intuitively, A and B are well-trained while D and E are under-trained due to the training data sizes. To investigate the impact of the order of teachers, we enumerate all the six permutations of A, B and C. In addition, we append D and E to the end of each permutation to simulate the "weak" teacher scenario. Therefore, we have six configurations in total.

Specially, we use a string like "ABDE $\rightarrow$ C" to denote a configuration, which means C is the student, A, B, D and E are the teachers and A arrives first, then B and so on. For simplicity, we use the training set of C as both the training set D_{train} and the transfer set D_{trans} in CKD, and the test set of C is leveraged as D_{test}. The goal in this configuration is to improve the performance of C on D_{test}. In summary, the six configurations are "BCDE$\rightarrow$A", "CBDE$\rightarrow$A", "ACDE$\rightarrow$B", "CADE$\rightarrow$B", "ABDE$\rightarrow$C", and "BADE$\rightarrow$C".

For clarity, the differences of other aforementioned settings with this one will be given in the corresponding sections later.

Evaluation. We leverage the following two metrics to evaluate our method:

Step	Method	BCDE→A BLEU↑	BCDE→A AD↓	ACDE→B BLEU↑	ACDE→B AD↓	ABDE→C BLEU↑	ABDE→C AD↓	Average BLEU↑	Average AD↓
0		42.84		27.53		18.06		29.48	/
1	KD	$46.19_{3.35}$	0.00	$24.32_{-3.21}$	3.21	$17.06_{-1.00}$	1.00	$29.19_{-0.29}$	1.40
	EWC	$46.09_{3.25}$	0.00	$24.32_{-3.21}$	3.21	$17.12_{-0.94}$	0.94	$29.18_{-0.30}$	1.38
	CL-NMT	$46.14_{3.30}$	0.00	$24.28_{-3.25}$	3.25	$17.09_{-0.97}$	0.97	$29.17_{-0.31}$	1.41
	Ours	$46.00_{3.16}$	0.00	$28.01_{0.48}$	0.00	$18.98_{0.92}$	0.00	$31.00_{1.52}$	0.00
2	KD	$44.62_{1.78}$	1.57	$26.11_{-1.42}$	3.21	$19.33_{1.27}$	1.00	$30.02_{0.54}$	1.93
	EWC	$45.80_{2.96}$	0.29	$25.28_{-2.25}$	3.21	$18.13_{0.07}$	0.94	$29.74_{0.26}$	1.48
	CL-NMT	$45.24_{2.40}$	0.90	$27.67_{0.14}$	3.25	$19.09_{1.03}$	0.97	$30.67_{1.19}$	1.71
	Ours	$45.89_{3.05}$	0.11	$28.28_{0.75}$	0.00	$19.18_{1.12}$	0.00	$31.12_{1.64}$	0.04
3	KD	$39.16_{-3.68}$	7.03	$21.76_{-5.77}$	7.56	$16.14_{-1.92}$	4.19	$25.69_{-3.79}$	6.26
	EWC	$43.88_{1.04}$	2.21	$24.48_{-3.05}$	4.01	$17.76_{-0.30}$	1.31	$28.71_{-0.77}$	2.51
	CL-NMT	$43.91_{1.07}$	2.23	$27.23_{-0.3}$	3.69	$18.45_{0.39}$	1.61	$29.86_{0.39}$	2.51
	Ours	$45.89_{3.05}$	0.11	$28.41_{0.88}$	0.00	$19.15_{1.09}$	0.03	$31.15_{1.67}$	0.05
4	KD	$30.57_{-12.27}$	15.62	$22.71_{-4.82}$	7.56	$13.88_{-4.18}$	6.45	$22.39_{-7.09}$	9.88
	EWC	$41.13_{-1.71}$	4.96	$24.89_{-2.64}$	4.01	$17.24_{-0.82}$	1.83	$27.75_{-1.72}$	3.60
	CL-NMT	$43.13_{0.29}$	3.01	$27.90_{0.37}$	3.69	$18.59_{0.53}$	1.61	$29.87_{0.40}$	2.77
	Ours	**45.89**$_{3.05}$	**0.11**	**28.49**$_{0.96}$	**0.00**	**19.15**$_{1.09}$	**0.03**	**31.18**$_{1.70}$	**0.05**

Table 2: Results of *Chinese-to-English* translation under *homogeneous* teacher setting. "BCDE→A" denotes A is the student model and B, C, D, and E are teacher models in step 1 to 4, respectively. "Δ" denotes ΔBLEU compared with step 0 (i.e., initial student model), and ΔBLEU scores are also reported as subscript numbers. "AD" is the accumulative degradation defined in Eq. 12, which is the lower the better. The last two columns are numbers averaged row-wise. Best results in step 4 are in **bold**.

- *BLEU* (Papineni et al., 2002) [3]: the most widely used evaluation metric for machine translation.
- *Accumulative Degradation* (AD): measuring the accumulative occasional quality degradation in all steps, which should be avoided as much as possible. AD from step 1 to t is defined as follows:

$$\mathrm{AD} = \sum_{k=1}^{t} \max(0, \mathrm{B}(\phi_{k-1}) - \mathrm{B}(\phi_k)), \quad (12)$$

where $\mathrm{B}(\cdot)$ denotes $\mathrm{BLEU}(D_{\text{test}}, \cdot)$.

Baselines. Our method is compared with the following baseline methods:

- *Knowledge Distillation* (KD) (Khayrallah et al., 2018) for NMT which applies vanilla knowledge distillation on each token trivially.
- *Elastic Weight Consolidation* (EWC) (Saunders et al., 2019; Thompson et al., 2019) which is a representative continual learning method that adds an EWC term as a penalty to alleviate catastrophic forgetting.
- *Continual Learning for NMT* (CL-NMT) (Cao et al., 2021) which is a representative work on *multi-step* continual learning in NMT.

[3] BLEU score is computed using `multi-bleu.perl` on the corresponding test set for each student model.

3.2 Implementation Details

We use byte pair encoding (BPE) (Sennrich et al., 2016) with the vocabulary size of 32k. The hyper-parameters of the Transformer-base models are set mostly following Vaswani et al. (2017). We use Adam (Kingma and Ba, 2014) optimizer, in which $\beta_1 = 0.9$, $\beta_2 = 0.98$. During training, learning rate is 7×10^{-4} and dropout rate is 0.1. Batch size is $6,000$. λ in Eq. 1 in step t is defined as $\lambda = 0.999 \frac{1-0.999^{t-1}}{1-0.999^t}$ following (Cao et al., 2021). More details of hyper-parameters are provided in Appendix B.

3.3 Quantification Function Selection

We first evaluate the three candidates for the quantification function Q defined in Sec. 2.3. A proper Q should correlate well with model performance and generalize well to a wide range of domains. To this end, we collect six widely used datasets of different domains and varying sizes and evaluate the correlations between the candidates and corpus-level BLEU scores on them. The Pearson correlation coefficients between token entropy (Eq. 4), hard label matching (Eq. 5) and token-level cross entropy (Eq. 6) are -0.5622, 0.8091 and 0.7792, respectively. Both hard label matching and token-level cross entropy are strongly correlated with corpus-level BLEU. However, hard label matching can not break a tie when both the teacher and student models' predictions are correct or incorrect. Therefore, we adopt token-level cross entropy as Q in the rest of this work. Examples and more discussions can be found in Appendix C.

3.4 Chinese-to-English Translation

Homogeneous Teacher Setting. In this setting, all the student and teacher models are of the same model architecture, which is Transformer-base. For space limitation, we only show results of three configurations in Table 2. The full results for all configurations can be found in Appendix D.1. From Table 2 we can observe that:

(1) Our method achieves improvements over the initial student model in all steps and configurations, and outperforms all baselines significantly. It indicates that our method is effective for leveraging diverse teacher models to continually improve the performance of the student model on its test dataset.

(2) Our method achieves zero or near-zero accumulative performance degradation (AD) scores in all configurationss, indicating our method is also effective to retain acquired knowledge. Especially, when encountering model D (step 3), nearly all baselines face severe quality degradation compared with step 2, while our method even achieves gain in $ACDE \rightarrow B$, which further justifies the effectiveness of our method.

(3) All baselines perform poorly after four steps of distillation, indicating that the problem we aim to resolve is challenging. Specifically, KD, the worst one, suffers from severe performance degradation as averaged ΔBLEU and AD scores are -7.09 and 9.88, respectively. We argue this is due to KD implicitly assumes that the teacher models are helpful such that it is prone to less beneficial knowledge provided by them. EWC is designed to alleviate catastrophic forgetting and achieves better ΔBLEU and AD scores than KD. However, EWC still fails to achieve improvement over the initial student model, i.e., all ΔBLEU scores are negative. CL-NMT is specially designed for multi-step continual learning in NMT and achieves the best ΔBLEU and AD scores among baselines. Nevertheless, its average ΔBLEU score is significantly smaller than ours (0.40 v.s. 1.70) and its average AD score is significantly worse than ours (2.77 v.s. 0.05). Overall, the problem to be resolved is challenging and our method is remarkably effective than baselines.

Method	Base→Base		RNN→Base		Big→Base	
	BLEU↑	AD↓	BLEU↑	AD↓	BLEU↑	AD↓
Original	29.48	/	29.48	/	29.48	/
KD	$29.58_{0.09}$	0.93	$28.29_{-1.19}$	1.21	$29.21_{-0.27}$	1.17
EWC	$29.63_{0.15}$	0.90	$27.83_{-1.65}$	1.65	$29.39_{-0.09}$	0.32
CL-NMT	$29.57_{0.09}$	0.92	$25.89_{-3.59}$	3.59	$28.00_{-1.48}$	1.91
Ours	$\mathbf{31.07}_{1.59}$	**0.00**	$\mathbf{29.44}_{-0.04}$	**0.12**	$\mathbf{30.82}_{1.34}$	**0.00**

Table 3: Results of *Chinese-to-English* translation under the *heterogeneous* teacher setting in step 1, averaged over six configurations. "Base" and "Big" denote Transformer-based and Transformer-big models, respectively. And "X→Y" denotes that X is the teacher and Y is the student.

Method	Base (M)→Base		RNN (M)→Base		Big (M)→Base	
	BLEU↑	AD↓	BLEU↑	AD↓	BLEU↑	AD↓
Original	29.48	/	29.48	/	29.48	/
KD	$18.36_{-11.12}$	11.12	$13.88_{-15.60}$	15.60	$18.53_{-10.95}$	10.95
EWC	$24.13_{-5.35}$	5.35	$22.84_{-6.64}$	6.64	$23.04_{-6.44}$	6.44
CL-NMT	$11.14_{-18.34}$	18.34	$3.16_{-26.32}$	26.32	$10.90_{-18.58}$	18.58
Ours	$\mathbf{29.48}_{0.00}$	**0.00**	$\mathbf{29.48}_{0.00}$	**0.00**	$\mathbf{29.48}_{0.00}$	**0.00**

Table 4: Results of *Chinese-to-English* translation under the *malicious* teacher setting in step 1, averaged over six configurations. "(M)" is short for "malicious".

(4) Despite the promising results, slight performance degradation can still be observed occasionally for our method. Therefore, there is still room for further improvement on retaining acquired knowledge.

Heterogeneous Teacher Setting. Using logits as the medium to transfer and retain knowledge, our approach is model-agnostic and scalable. To justify that, we replace the Transformer-base teacher models with RNN (Bahdanau et al., 2014) and Transformer-big (Vaswani et al., 2017) models, and repeat the experiments in Table 2 with other settings remaining identical. Table 3 shows similar results as Table 2 that our method outperforms all baselines significantly and also achieves zero or near-zero AD scores, indicating that our method is extensible to different model architectures. Interestingly, all the baselines encounter serious performance degradation while the ΔBLEU of our method is nearly zero, indicating that distilling knowledge from a teacher of a completely different architecture may be extremely difficult. It deserves more thoughtful investigation and we leave it as future work.

Method	BLEU↑	AD↓
Original	32.77	/
KD	$28.79_{-3.98}$	7.95
EWC	$31.54_{-1.23}$	2.56
CL-NMT	$31.01_{-1.76}$	3.42
Ours	$\mathbf{33.43}_{0.66}$	**0.03**

Table 5: Results of extending the training data of the *Chinese-to-English* teacher models to ten million scale under the *homogeneous* teacher setting, averaged over six configurations.

Method	Homogeneous		Heterogeneous		Malicious	
	BLEU↑	AD↓	BLEU↑	AD↓	BLEU↑	AD↓
Original	30.62	/	30.62	/	30.62	/
KD	$30.68_{0.06}$	0.13	$30.19_{-0.43}$	0.44	$17.87_{-12.75}$	12.75
EWC	$30.66_{0.04}$	0.18	$30.39_{-0.23}$	0.25	$24.26_{-6.36}$	6.36
CL-NMT	$30.85_{0.23}$	0.10	$26.20_{-4.42}$	4.42	$8.49_{-22.1}$	22.13
Ours	$\mathbf{31.19}_{0.57}$	**0.00**	$\mathbf{30.85}_{0.23}$	**0.05**	$\mathbf{30.62}_{0.00}$	**0.00**

Table 6: Results of *German-to-English* translation in step 1, averaged over all six setting groups.

	Method	Step 1	Step 4
1	Full Model	**31.07**	**31.18**
2	Removing ℓ_{NEG}	30.69	30.54
3	Replacing ℓ_{NEG} with ℓ_{KD}	30.60	30.31
4	Removing ℓ_{KI}	30.74	29.94

Table 7: Ablation study on *Chinese-to-English* translation under *homogeneous* teacher setting. BLEU scores averaged over six configurations are reported.

Malicious Teacher Setting. Robustness to malicious models is critical in our scenario as only the parameters rather than training data of teachers are available. We simulate malicious teacher models by shuffling the outputs of a well-trained model within a batch so that the model answers almost completely wrong with high confidence. We repeat the experiments in Table 2 with other settings remaining identical. As shown in Table 4, our approach is far less affected by the malicious model with three different teacher model architectures. Moreover, it could be further explored to detect and skip malicious models to save computational resources directly.

3.5 Larger Scale Chinese-to-English Translation

We scale up the dataset size of the Chinese-to-English translation experiment under the homogeneous teacher setting from one million to ten million. Other settings are similar to the original experiments and are detailed in Appendix D.2. As shown in Table 5, our method remains effective while all baseline methods fail to achieve positive quality gain (ΔBLEU). This demonstrates that the performance of the baseline methods does not improve as the size of the data and performance of the models increase, while our method remains valid. Thus, it shows that our method is scalable for corpus of different sizes.

3.6 German-to-English Translation

We also conduct experiments on German-to-English datasets. Models are trained on four different datasets from different domains. Other settings are similar to the Chinese-to-English experiments and are detailed in Appendix D.3. The average values among each of the homogeneous, heterogeneous, and malicious teacher settings are reported in Table 6. Due to the large domain differences of the datasets, only our method consistently obtains BLEU gains and zero or near zero AD scores, exceeding the baselines, demonstrating that our approach is effective for different language pairs.

3.7 Ablation Study

Table 7 shows the effect of the negative KD loss ℓ_{NEG} (Eq. 9) in knowledge filtration and the knowledge inheritance loss ℓ_{KI}. Results at the beginning ($t = 1$) and later step ($t = 4$) for Chinese-to-English translation under the homogeneous teacher setting are reported. We can observe that:

1. Removing either ℓ_{NEG} (row 2) or ℓ_{KI} (row 4) hurts the performance, indicating both of them are effective.

2. Comparing row 1 with row 2, we can conclude that the negative subset of the transfer set where the teacher performs worse than the student (D^{-}_{trans}) also contains valuable non-trivial knowledge. Furthermore, trivially applying vanilla KD loss ℓ_{KD} on D^{-}_{trans} (row 2 v.s. 3) brings no gain. Therefore, our proposed negative KD loss is effective for making less beneficial knowledge play a good role.

3. Without ℓ_{KI}, the performance drops severely, especially at a later step, verifying that knowledge inheritance is essential for retaining acquired knowledge.

3.8 Comparison with Multi-teacher Knowledge Distillation

Multi-teacher KD (Freitag et al., 2017), aka ensemble KD, generally requires all teachers available at the same time, which violates the definition of our problem and may result in enormous computational and memory cost as teacher number grows. Moreover, it is also non-trivial to adapt it to our scenarios due to potential unbeneficial knowledge provided by teachers. Therefore, we do not include it as a major baseline in the experiments above. Nevertheless, in this section, we still provide a comparison of our method with vanilla multi-teacher KD which averages the outputs of all teachers as the target distribution for analysis. The BLEU score of vanilla multi-teacher KD averaged over six configurations is 30.49, lower than our 31.18, indicating that our method is superior to vanilla multi-teacher KD although the comparison is more favorable to it. More details on comparison in terms of task definition, robustness and storage requirement are analyzed in Appendix D.4.

4 Related Work

Knowledge Distillation. Knowledge distillation (KD) is the most widely used technique for transferring knowledge between models (Hinton et al., 2015). Despite of their effectiveness, conventional KD methods usually implicitly assume that the teacher model is superior or complementary to the student model (Gou et al., 2021). Although recently Qin et al. (2022) allow a big model to learn from small models, they still require that the small models are better than the big model for the given tasks and datasets. However, the assumption does not necessarily hold in our scenario due to the diversity of teacher models. Multi-teacher KD (Freitag et al., 2017; You et al., 2017; Fukuda et al., 2017; Mirzadeh et al., 2020; Liu et al., 2020), which distills knowledge from multiple teachers simultaneously, is highly related to this work. Generally, multi-teacher KD requires all teachers to be available at the same time, which will result in enormous extra memory consummation as the number of teachers grows. More importantly, new teachers may be released constantly (Wolf et al., 2020), which can not be seen in advance. Therefore, multi-teacher KD methods are not feasible to our scenario. L2KD (Chuang et al., 2020) leverages sequential KD to continually learn new tasks, having different goal and challenges compared with our scenario. Another line of related work is selective distillation (Gu et al., 2020; Wang et al., 2021a; Shi and Radu, 2022), which selects data and losses to accelerate KD or enhance model robustness. In contrast, we select data for conducting different ways of distillation in our proposed method.

Continual Learning. Continual learning (CL) for neural machine translation (NMT) aims at learning knowledge of new domains (Thompson et al., 2019; Liang et al., 2021; Cao et al., 2021) or languages (Neubig and Hu, 2018; Garcia et al., 2021; Huang et al., 2022) without forgetting old knowledge. Our scenario also requires learning new knowledge but focuses on improving performance of the student on its test set instead. Moreover, alleviating the negative impact of the less beneficial knowledge conveyed by "weak" teachers is essential in our scenario, which is hardly explored in CL for NMT. While our scenario is a multi-step process, multi-step CL is less explored in NMT (Cao et al., 2021; Liang et al., 2021). Zeng et al. (2019) address a similar task of adapting from multiple out-of-domain models to a single in-domain model. Nevertheless, they assume the training data for the out-of-domain models are available, which is inaccessible in our scenario. Besides, leveraging high-resource language NMT models to improve low-resource language translation has also attracted intensive efforts (Neubig and Hu, 2018; Lakew et al., 2019; Liu et al., 2021; Huang et al., 2022), which can be a future extension of our method.

5 Conclusion and Future Work

To take advantage of increasingly available trained neural machine translation (NMT) models to improve one model of interest, we propose a novel method named continual knowledge distillation. Specially, knowledge from the trained models is transferred to the interested model via knowledge distillation in a sequential manner. Extensive experiments on two language pairs under homogeneous, heterogeneous, and malicious teacher settings show the effectiveness of our proposed method.

In the future, we will further explore the effect of the teacher model order. It is also worth involving more sophisticated methods in knowledge filtration, such as gradient-based and meta-learning-based methods. Moreover, it is also a promising research direction to exchange knowledge among all the models such that all of them achieve improvement.

Limitations

There are some limitations that have yet to be addressed. Since we use the predicted probability distributions of the model output as a medium for continual KD for NMT, the vocabulary of multiple models needs to be consistent. Overcoming it allows continual KD for NMT to be extended to models with different language pairs and different modalities. Also, although our approach is robust to malicious models, there are more diverse and sophisticated attacks in real-world that require more research on defense. In addition, the teacher and student models must be trained on the same language pair. Further studies can consider more general scenarios without the above limitations. There are other approaches worth exploring in order to address the transfer of knowledge from models rather than their training data besides sequential manner. For example, it is also possible to explore various distillation methods like organizing teacher models into batches or pipelines.

Ethics Statement

In practice, a provider may publicly release a model but may not wish its knowledge to be transferred into another one. Applying our method on such models will result in model stealing (He et al., 2022) related ethical concerns. How to detect this kind of misconduct still needs further exploration. Although sharing knowledge without exposing private data is one of the potential benefits of our method, models produced by our method are still vulnerable to attacks such as membership inference (Hisamoto et al., 2020), and the private training data could still be stolen from the model.

Acknowledgments

This work is supported by the National Key R&D Program of China (2022ZD0160502) and the National Natural Science Foundation of China (No. 61925601, 62276152, 62236011). We thank all the reviewers for their valuable and insightful comments. We also thank Weihe Gao, Kaiyu Huang and Shuo Wang for their help.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate.

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.

Yung-Sung Chuang, Shang-Yu Su, and Yun-Nung Chen. 2020. Lifelong language knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2914–2924, Online. Association for Computational Linguistics.

Markus Freitag, Yaser Al-Onaizan, and Baskaran Sankaran. 2017. Ensemble distillation for neural machine translation.

Takashi Fukuda, Masayuki Suzuki, Gakuto Kurata, Samuel Thomas, Jia Cui, and Bhuvana Ramabhadran. 2017. Efficient knowledge distillation from an ensemble of teachers. In *Proc. Interspeech 2017*, pages 3697–3701.

Xavier Garcia, Noah Constant, Ankur Parikh, and Orhan Firat. 2021. Towards continual learning for multilingual machine translation via vocabulary substitution. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1184–1192, Online. Association for Computational Linguistics.

Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *Int. J. Comput. Vision*, 129(6):1789–1819.

Yuxian Gu, Zhengyan Zhang, Xiaozhi Wang, Zhiyuan Liu, and Maosong Sun. 2020. Train no evil: Selective masking for task-guided pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6966–6974, Online. Association for Computational Linguistics.

Yingzhe He, Guozhu Meng, Kai Chen, Xingbo Hu, and Jinwen He. 2022. Towards security threats of deep learning systems: A survey. *IEEE Transactions on Software Engineering*, 48(05):1743–1770.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network.

Sorami Hisamoto, Matt Post, and Kevin Duh. 2020. Membership inference attacks on sequence-to-sequence models: Is my data in your machine translation system? *Transactions of the Association for Computational Linguistics*, 8:49–63.

Kaiyu Huang, Peng Li, Jin Ma, and Yang Liu. 2022. Entropy-based vocabulary substitution for incremental learning in multilingual neural machine translation. In *Proceedings of EMNLP 2022*, Online. Association for Computational Linguistics.

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.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X: Papers*, pages 79–86, Phuket, Thailand.

Surafel M. Lakew, Alina Karakanta, Marcello Federico, Matteo Negri, and Marco Turchi. 2019. Adapting multilingual neural machine translation to unseen languages. In *Proceedings of the 16th International Conference on Spoken Language Translation*, Hong Kong. Association for Computational Linguistics.

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

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).

Yuang Liu, Wei Zhang, and Jun Wang. 2020. Adaptive multi-teacher multi-level knowledge distillation. *Neurocomputing*, 415:106–113.

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.

Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. 2020. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5191–5198.

Graham Neubig and Junjie Hu. 2018. Rapid adaptation of neural machine translation to new languages. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 875–880, Brussels, Belgium. Association for Computational Linguistics.

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

Yujia Qin, Yankai Lin, Jing Yi, Jiajie Zhang, Xu Han, Zhengyan Zhang, Yusheng Su, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2022. Knowledge inheritance for pre-trained language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3921–3937, Seattle, United States. Association for Computational Linguistics.

Tanya G. Roosta, Peyman Passban, and Ankit Chadha. 2021. Communication-efficient federated learning for neural machine translation. *CoRR*, abs/2112.06135.

Roberts Rozis and Raivis Skadiņš. 2017. Tilde MODEL - multilingual open data for EU languages. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 263–265, Gothenburg, Sweden. Association for Computational Linguistics.

Danielle Saunders, Felix Stahlberg, Adrià de Gispert, and Bill Byrne. 2019. Domain adaptive inference for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 222–228, Florence, Italy. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Hongrui Shi and Valentin Radu. 2022. Data selection for efficient model update in federated learning. In *Proceedings of the 2nd European Workshop on Machine Learning and Systems*, EuroMLSys '22, page 72–78, New York, NY, USA. Association for Computing Machinery.

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.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

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

Fusheng Wang, Jianhao Yan, Fandong Meng, and Jie Zhou. 2021a. 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, Online. Association for Computational Linguistics.

Jianzong Wang, Zhangcheng Huang, Lingwei Kong, Denghao Li, and Jing Xiao. 2021b. Modeling without sharing privacy: Federated neural machine translation. In *Web Information Systems Engineering – WISE 2021*, pages 216–223, Cham. Springer International Publishing.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Jiahong Wu, He Zheng, Bo Zhao, Yixin Li, Baoming Yan, Rui Liang, Wenjia Wang, Shipei Zhou, Guosen Lin, Yanwei Fu, Yizhou Wang, and Yonggang Wang. 2017. AI challenger : A large-scale dataset for going deeper in image understanding. *CoRR*, abs/1711.06475.

Shan You, Chang Xu, Chao Xu, and Dacheng Tao. 2017. Learning from multiple teacher networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 1285–1294, New York, NY, USA. Association for Computing Machinery.

Jiali Zeng, Yang Liu, Jinsong Su, Yubing 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)*, pages 845–855, Hong Kong, China. Association for Computational Linguistics.

A Datasets for Chinese-to-English Translation

The statistics of the datasets have been shown in Table 1. The training data for the news, oral, internet, speech and subtitle domains are randomly sampled from LDC [4], AI Challenger 2018 (Wu et al., 2017), translation2019zh [5], TED transcripts (Tiedemann, 2012) and Subtitles (Lison and Tiedemann, 2016), respectively. Newstest 2018 and NIST 02-09 from LDC are used as the development and test set for the news domain. The AI Challenger 2017 dataset is used as the test set for the oral domain. For the other corresponding domains, the development and test sets provided along with the training sets are used as development and test sets accordingly.

B Hyper-parameter Search

α	BLEU
0.05	30.86
0.1	**31.07**
0.5	30.96
1	30.43
2	29.54
3	28.20

Table 8: Results under different α. The metrics are averaged over six configurations.

$\mathbf{k_a : k_b}$	BLEU
1 : 1	**31.07**
1 : 0.5	31.03
1 : 2	31.06

Table 9: Results under different ratio of $k_a : k_b$. The metrics are averaged over six configurations.

For hyper-parameter α in Eq. 9, we try multiple values in Table 8, and choose 0.1 as the default value.

Since α regulates how much the student output distribution is pushed away from the negative distribution, we also try to regulate the proportion of positive and negative KD losses that work in a similar way. We modify Eq. 10 to:

$$\ell_{\mathrm{KF}}(\boldsymbol{\phi}_t, \boldsymbol{\theta}_t^*, D_{\mathrm{trans}}) = k_a \ell_{\mathrm{KD}}(\boldsymbol{\phi}_t, \boldsymbol{\theta}_t^*, D^{+}_{\mathrm{trans}}) + k_b \ell_{\mathrm{NEG}}(\boldsymbol{\phi}_t, \boldsymbol{\theta}_t^*, D^{-}_{\mathrm{trans}}) \quad (13)$$

By adjusting k_a and k_b, we can regulate the weights of positive and negative losses. As shown in Table 9, we still use the original settings since no significant performance improvement is found when adjusting $k_a : k_b$.

C Exploring Knowledge Filtration Quantification Function

C.1 Examples

In Table 10, we show three examples to demonstrate how the default quantification function (token-level cross entropy) works in knowledge filtration.

- In the first case, we apply standard knowledge distillation because the teacher model assigns a higher probability of the ground truth token "decorations" than student, indicating a better distribution from the former.
- In the second case, the output from the teacher model is discarded because the negative KD loss exceeds the threshold. It might be a reasonable choice since the output of the teacher is too far from the ground truth token.
- In the third case, the teacher model have slightly worse predictions than students, motivating the student model not to make similar error-prone mistakes.

C.2 Alternatives to Quantification Function

The advantage of token-level cross entropy is that the predictions corresponding to the tokens in the transfer set D_{trans} can be divided into two mutually disjoint parts depending on the comparison between the predictions of teacher and student models. In contrast, hard label matching divides D_{trans} according to whether the teacher and student models predict the ground-truth token correctly, which will result in four parts due ties as shown in Table 11.

Are the advantages of these two metrics beneficial for our task? Is it possible to combine the beneficial properties? To answer the questions, we define several metrics in Table 11 to compare these two metrics at a fine-grained level. And the effects of these metrics are shown in Table 12. It could be found that token-level cross entropy always performs better because fewer samples are discarded such that more knowledge is transferred in knowledge distillation.

[4] LDC2002E18, LDC2003E07, LDC2003E14, part of LDC2004T07, LDC2004T08 and LDC2005T06

[5] https://github.com/brightmart/nlp_chinese_corpus, version 1.0.

Source	每棵圣诞树上都挂满琳琅目的装点，但每棵树的顶端必定有一特大的星星					
Target	*every christmas tree hung with dazzling* ***decorations*** *, but the top of each tree must have a tree big stars*					
Teacher	**Candidates:**	decorations $_{P=0.396}$	ornaments $_{P=0.125}$	costumes $_{P=0.033}$	ar@@ $_{P=0.032}$	jewelry $_{P=0.022}$
Student	**Candidates:**	display $_{P=0.023}$	car@@ $_{P=0.0022}$	displays $_{P=0.019}$	's $_{P=0.011}$	decorations $_{P=0.010}$
Loss	$Q = 0.396 > Q = 0.010 \Longrightarrow \langle y_j, \mathbf{y}_{<j}, \mathbf{x} \rangle$ belongs to $D^{+}_{\text{trans}} \Longrightarrow \ell_{\text{KD}} = 2.542$				// Teacher is informative.	
Source	无量跌停并非没有前兆，前一交易日它的证券股价表现已经显得出奇地疲弱					
Target	*measureless limit is a* ***precursor*** *, in the previous session its securities share price performance appears ...*					
Teacher	**Candidates:**	fore@@ $_{P=0.216}$	sign $_{P=0.144}$	single $_{P=0.104}$	good@@ $_{P=0.037}$	major $_{P=0.029}$
Student	**Candidates:**	pre@@ $_{P=0.533}$	precursor@@ $_{P=0.161}$	aug@@ $_{P=0.030}$	omen $_{P=0.017}$	pre $_{P=0.012}$
Loss	$Q = 0.003 < Q = 0.161 \Longrightarrow \langle y_j, \mathbf{y}_{<j}, \mathbf{x} \rangle$ belongs to $D^{-}_{\text{trans}} \Longrightarrow \ell_{\text{NEG}} = \min\{0, \alpha - (-3.921)\} = 0$				// Teacher is too unproductive, $\alpha = 0.1$.	
Source	这笔钱将提存立即中标人					
Target	*money will be escrowed immediately to winning* ***bidder*** .					
Teacher	**Candidates:**	the@@ $_{P=0.394}$	mark $_{P=0.260}$	be $_{P=0.034}$	pay@@ $_{P=0.034}$	save $_{P=0.019}$
Student	**Candidates:**	target@@ $_{P=0.331}$	get@@ $_{P=0.235}$	sign@@ $_{P=0.022}$	put $_{P=0.021}$	get $_{P=0.020}$
Loss	$Q = 0.001 < Q = 0.003 \Longrightarrow \langle y_j, \mathbf{y}_{<j}, \mathbf{x} \rangle$ belongs to $D^{-}_{\text{trans}} \Longrightarrow \ell_{\text{NEG}} = \min\{0, \alpha - 0.085\} = \alpha - 0.085$				// Teacher is somewhat informative, $\alpha = 0.1$.	

Table 10: Three representative examples for illustrating the effectiveness of the knowledge filtration. Ground truth token y_j and candidates matching with y_j are highlighted in yellow. "Student" and "Teacher" show the top 5 predicted candidate tokens and their corresponding probabilities. $\alpha = 0.1$ is the threshold here.

Metric	$\mathbb{1}\{y_j = k\}$ $\mathbb{1}\{y_j = k^*\}$	$\mathbb{1}\{y_j \neq k\}$ $\mathbb{1}\{y_j \neq k^*\}$	$\mathbb{1}\{y_j = k\}$ $\mathbb{1}\{y_j \neq k^*\}$	$\mathbb{1}\{y_j \neq k\}$ $\mathbb{1}\{y_j = k^*\}$
Trivial	+ KD loss	+ KD loss	+ KD loss	+ KD loss
Hard Label Matching	Discarded	Discarded	Discarded	+ KD loss
Hard Label Matching (With Filtration)	Discarded	Discarded	- KD loss	+ KD loss
Token-level CE	$\begin{cases} \text{+ KD loss if } \Delta Q > 0 \\ \text{Discarded if } \Delta Q \leq 0 \end{cases}$	$\begin{cases} \text{+ KD loss if } \Delta Q > 0 \\ \text{Discarded if } \Delta Q \leq 0 \end{cases}$	Discarded	+ KD loss
Token-level CE (With Filtration)	$\begin{cases} \text{+KD loss if } \Delta Q > 0 \\ \text{- KD loss if } \Delta Q \leq 0 \end{cases}$	$\begin{cases} \text{+KD loss if } \Delta Q > 0 \\ \text{- KD loss if } \Delta Q \leq 0 \end{cases}$	- KD loss	+ KD loss
Hybrid Metric 1	$\begin{cases} \text{+KD loss if } \Delta Q > 0 \\ \text{- KD loss if } \Delta Q \leq 0 \end{cases}$	$\begin{cases} \text{+KD loss if } \Delta Q > 0 \\ \text{Discarded if } \Delta Q \leq 0 \end{cases}$	- KD loss	+ KD loss
Hybrid Metric 2	$\begin{cases} \text{+KD loss if } \Delta Q > 0 \\ \text{Discarded if } \Delta Q \leq 0 \end{cases}$	$\begin{cases} \text{+KD loss if } \Delta Q > 0 \\ \text{- KD loss if } \Delta Q \leq 0 \end{cases}$	- KD loss	+ KD loss
Hybrid Metric 3	$\begin{cases} \text{Discarded if } \Delta Q > 0 \\ \text{- KD loss if } \Delta Q \leq 0 \end{cases}$	$\begin{cases} \text{+KD loss if } \Delta Q > 0 \\ \text{- KD loss if } \Delta Q \leq 0 \end{cases}$	- KD loss	+ KD loss

Table 11: Different metrics have different behaviors depending on the correctness of student's and teacher's prediction k and k^* for a given token y_j. "+ KD loss" and "- KD loss" mean positive and negative KD loss. "ΔQ" denotes the difference in metric f between the student and teacher model. $\mathbb{1}$ is an indicator function.

Metric	**BLEU**
Trivial	27.36
Hard Label Matching	29.13
Hard Label Matching (With Filtration)	30.37
Token-level CE	30.69
Token-level CE (With Filtration)	**31.07**
Hybrid Metric 1	31.05
Hybrid Metric 2	30.89
Hybrid Metric 3	30.22

Table 12: Results for different metrics. The metrics are averaged over six configurations.

D Detailed Results

D.1 Chinese-to-English Translation

The full results for the Chinese-to-English homogeneous model setting for all configurations are shown in Table 13. Our experiments are conducted on NVIDIA A100 GPUs. Each distillation process requires 48 GPU hours and is run only once due to computational budgets.

D.2 Larger Scale Chinese-to-English Translation

To illustrate the scalability of our method, we repeat Chinese-to-English translation experiments under homogeneous model setting (Sec. 3.4) using larger and complete corpora without sampling. As shown in Table 15, we choose five full datasets for news, oral, internet, speech and subtitle domains, respectively: WMT20, AI Challenger 2018, translation2019zh, TED transcripts and Subtitles. Compared with Table 1, only the training set of model F,G and H are replaced with bigger datasets. And the validation sets, test sets and other settings are kept identical. F, G, H, I, J are all Transformer-base models. The three models F, G and H are combined in different orders to form six groups of experiments. To simulate the most challenging scenario, I and J with weaker performance are added at the end of these experiments to test the performance of our method on poor models. According

Step	Method	BCDE→A		CBDE→A		ACDE→B		CADE→B		ABDE→C		BADE→C		Average	
		BLEU↑	AD↓	BLEU↑	AD↓	BLEU↑	AD↓	BLEU↑	AD↓	BLEU↑	AD↓	BLEU↑	AD↓	BLEU↑	AD↓
0		42.84		42.84		27.53		27.53		18.06		18.06		29.48	/
1	KD	$46.19_{3.35}$	0.00	$44.47_{1.63}$	0.00	$24.32_{-3.21}$	3.21	$26.17_{-1.36}$	1.36	$17.06_{-1.00}$	1.00	$19.22_{1.16}$	0.00	$29.57_{0.09}$	0.93
	EWC	$46.09_{3.25}$	0.00	$44.59_{1.75}$	0.00	$24.32_{-3.21}$	3.21	$26.26_{-1.27}$	1.27	$17.12_{-0.94}$	0.94	$19.35_{1.29}$	0.00	$29.57_{0.15}$	0.90
	CL-NMT	$46.14_{3.30}$	0.00	$44.53_{1.69}$	0.00	$24.28_{-3.25}$	3.25	$26.21_{-1.32}$	1.32	$17.09_{-0.97}$	0.97	$19.17_{1.11}$	0.00	$29.62_{0.09}$	0.92
	Ours	$46.00_{3.16}$	0.00	$45.88_{3.04}$	0.00	$28.01_{0.48}$	0.00	$28.17_{0.64}$	0.00	$18.98_{0.92}$	0.00	$19.36_{1.30}$	0.00	$31.07_{1.59}$	0.00
2	KD	$44.62_{1.78}$	1.57	$46.28_{3.44}$	0.00	$26.11_{-1.42}$	3.21	$23.96_{-3.57}$	3.57	$19.33_{1.27}$	1.00	$17.25_{-0.81}$	1.97	$29.59_{0.11}$	1.89
	EWC	$45.80_{2.96}$	0.29	$46.26_{3.42}$	0.00	$25.28_{-2.25}$	3.21	$25.92_{-1.61}$	1.61	$18.13_{0.07}$	0.94	$18.52_{0.46}$	0.83	$30.65_{0.51}$	1.15
	CL-NMT	$45.24_{2.40}$	0.90	$45.48_{2.64}$	0.00	$27.67_{0.14}$	3.25	$27.70_{0.17}$	1.32	$19.09_{1.03}$	0.97	$18.69_{0.63}$	0.48	$29.99_{1.17}$	1.15
	Ours	$45.89_{3.05}$	0.11	$46.08_{3.24}$	0.00	$28.28_{0.75}$	0.00	$28.50_{0.97}$	0.00	$19.18_{1.12}$	0.00	$19.36_{1.30}$	0.00	$31.22_{1.74}$	0.02
3	KD	$39.16_{-3.68}$	7.03	$39.11_{-3.73}$	7.17	$21.76_{-5.77}$	7.56	$21.69_{-5.84}$	5.84	$16.14_{-1.92}$	4.19	$16.12_{-1.94}$	3.10	$25.66_{-3.81}$	5.82
	EWC	$43.88_{1.04}$	2.21	$44.02_{1.18}$	2.24	$24.48_{-3.05}$	4.01	$24.27_{-3.26}$	3.26	$17.76_{-0.30}$	1.31	$18.09_{0.03}$	1.26	$29.91_{-0.73}$	2.38
	CL-NMT	$43.91_{1.07}$	2.23	$43.83_{0.99}$	1.65	$27.23_{-0.3}$	3.69	$27.32_{-0.21}$	1.70	$18.45_{0.39}$	1.61	$18.71_{0.65}$	0.48	$28.75_{0.43}$	1.89
	Ours	$45.89_{3.05}$	0.11	$46.08_{3.24}$	0.00	$28.41_{0.88}$	0.00	$28.48_{0.95}$	0.02	$19.15_{1.09}$	0.03	$18.98_{0.92}$	0.38	$31.17_{1.69}$	0.09
4	KD	$30.57_{-12.27}$	15.62	$30.31_{-12.53}$	15.97	$22.71_{-4.82}$	7.56	$22.66_{-4.87}$	5.84	$13.88_{-4.18}$	6.45	$13.99_{-4.07}$	5.23	$22.35_{-7.12}$	9.45
	EWC	$41.13_{-1.71}$	4.96	$37.41_{-5.43}$	8.85	$24.89_{-2.64}$	4.01	$24.96_{-2.57}$	3.26	$17.24_{-0.82}$	1.83	$17.59_{-0.47}$	1.76	$29.85_{-2.27}$	4.11
	CL-NMT	$43.13_{0.29}$	3.01	$42.99_{0.15}$	2.49	$27.90_{0.37}$	3.69	$27.93_{0.4}$	1.70	$18.59_{0.53}$	1.61	$18.58_{0.52}$	0.61	$27.20_{0.38}$	2.19
	Ours	**45.89**$_{3.05}$	**0.11**	**46.08**$_{3.24}$	**0.00**	**28.49**$_{0.96}$	**0.00**	**28.51**$_{0.98}$	**0.02**	**19.15**$_{1.09}$	**0.03**	**18.98**$_{0.92}$	**0.38**	**31.18**$_{1.71}$	**0.09**

Table 13: Results of *Chinese-to-English* translation under *homogeneous* teacher setting. "BCDE→A" denotes A is the student model and B, C, D, and E are teacher models in step 1 to 4, respectively. "Δ" denotes ΔBLEU compared with step 0 (i.e., initial student model), and ΔBLEU scores are also reported as subscript numbers. "AD" is the accumulative degradation defined in Eq. 12, which is the lower the better. The last two columns are numbers averaged row-wise. Best results in step 4 are in **bold**.

Step	Method	GHIJ→F		HGIJ→F		FHIJ→G		HFIJ→G		FGIJ→H		GFIJ→H		Average	
		BLEU↑	AD↓	BLEU↑	AD↓	BLEU↑	AD↓	BLEU↑	AD↓	BLEU↑	AD↓	BLEU↑	AD↓	BLEU↑	AD↓
0		43.16		43.16		31.35		31.35		23.8		23.8		32.77	/
1	KD	$43.90_{0.74}$	0.00	$41.54_{-1.62}$	1.62	$29.92_{-1.43}$	1.43	$29.70_{-1.65}$	1.65	$23.70_{-0.10}$	0.10	$23.69_{-0.11}$	0.11	$32.08_{-0.70}$	0.82
	EWC	$43.99_{0.83}$	0.00	$41.65_{-1.51}$	1.51	$29.90_{-1.45}$	1.45	$29.73_{-1.62}$	1.62	$23.70_{-0.10}$	0.10	$23.66_{-0.14}$	0.14	$32.17_{-0.66}$	0.80
	CL-NMT	$44.08_{0.92}$	0.00	$41.75_{-1.41}$	1.41	$29.92_{-1.43}$	1.43	$29.77_{-1.58}$	1.58	$23.70_{-0.10}$	0.10	$23.76_{-0.04}$	0.04	$32.11_{-0.60}$	0.76
	Ours	$44.25_{1.09}$	0.00	$44.05_{0.89}$	0.00	$31.51_{0.16}$	0.00	$31.35_{0.00}$	0.00	$23.81_{0.01}$	0.00	$23.77_{-0.03}$	0.03	$33.12_{0.35}$	0.01
2	KD	$42.19_{-0.97}$	1.71	$44.06_{0.90}$	1.62	$28.91_{-2.44}$	2.44	$28.94_{-2.41}$	2.41	$22.86_{-0.94}$	0.94	$22.84_{-0.96}$	0.96	$31.63_{-1.14}$	1.68
	EWC	$43.16_{0.00}$	0.83	$43.03_{-0.13}$	1.51	$27.83_{-3.52}$	3.52	$30.21_{-1.14}$	1.62	$23.01_{-0.79}$	0.79	$23.71_{-0.09}$	0.14	$32.40_{-0.94}$	1.40
	CL-NMT	$44.05_{0.89}$	0.04	$44.09_{0.93}$	1.41	$29.46_{-1.89}$	1.89	$29.64_{-1.71}$	1.71	$23.58_{-0.22}$	0.22	$23.60_{-0.20}$	0.20	$31.83_{-0.37}$	0.91
	Ours	$44.45_{1.29}$	0.00	$44.44_{1.28}$	0.00	$31.51_{0.16}$	0.00	$31.44_{0.09}$	0.00	$24.14_{0.34}$	0.00	$24.11_{0.31}$	0.03	$33.35_{0.58}$	0.01
3	KD	$35.13_{-8.03}$	8.77	$34.86_{-8.30}$	10.82	$22.82_{-8.53}$	8.53	$22.78_{-8.57}$	8.57	$18.30_{-5.50}$	5.50	$18.28_{-5.52}$	5.52	$25.36_{-7.41}$	7.95
	EWC	$41.58_{-1.58}$	2.42	$40.93_{-2.23}$	3.61	$27.03_{-4.32}$	4.32	$29.28_{-2.07}$	2.55	$22.05_{-1.75}$	1.75	$23.16_{-0.64}$	0.69	$29.89_{-2.10}$	2.56
	CL-NMT	$41.30_{-1.86}$	2.79	$43.38_{0.22}$	2.11	$29.47_{-1.88}$	1.89	$24.25_{-7.10}$	7.10	$22.68_{-1.12}$	1.12	$18.25_{-5.55}$	5.55	$30.67_{-2.88}$	3.42
	Ours	$44.63_{1.47}$	0.00	$44.47_{1.31}$	0.00	$31.48_{0.13}$	0.03	$31.50_{0.15}$	0.00	$24.04_{0.24}$	0.10	$24.12_{0.32}$	0.03	$33.37_{0.60}$	0.03
4	KD	$39.45_{-3.71}$	8.77	$39.12_{-4.04}$	10.82	$26.67_{-4.68}$	8.53	$26.79_{-4.56}$	8.57	$20.39_{-3.41}$	5.50	$20.33_{-3.47}$	5.52	$28.79_{-3.98}$	7.95
	EWC	$41.99_{-1.17}$	2.42	$41.40_{-1.76}$	3.61	$29.66_{-1.69}$	4.32	$30.77_{-0.58}$	2.55	$22.04_{-1.76}$	1.76	$23.37_{-0.43}$	0.69	$31.01_{-1.23}$	2.56
	CL-NMT	$42.69_{-0.47}$	2.79	$43.60_{0.44}$	2.11	$30.96_{-0.39}$	1.89	$27.08_{-4.27}$	7.10	$22.99_{-0.81}$	1.12	$18.75_{-5.05}$	5.55	$31.54_{-1.76}$	3.42
	Ours	**44.65**$_{1.49}$	**0.00**	**44.54**$_{1.38}$	**0.00**	**31.53**$_{0.18}$	**0.03**	**31.55**$_{0.20}$	**0.00**	**24.15**$_{0.35}$	**0.10**	**24.14**$_{0.34}$	**0.03**	$33.43_{0.66}$	**0.03**

Table 14: Results on larger Chinese-to-English datasets. "GHIJ→F" denotes F is the student model and model G to J are teacher models in step 1 to 4, respectively.

Model	Domain	Training	Dev.	Test
F	News	20,000,000	4,000	13,000
G	Oral	12,500,000	4,000	12,000
H	Internet	5,200,000	4,000	13,000
I	Speech	220,000	4,000	5,000
J	Subtitle	300,000	4,000	4,000

Table 15: The domain, training and evaluation corpora of Chinese-to-English translation large-scale experiments. Compare with Table 1, news, oral and Internet datasets are replaced with larger datasets to test the scalability. Other settings are kept identical.

to the task definition illustrated above, the transfer set of each set of experiments is the training set of the according student model.

The results are shown in Table 14. Similar to results in Sec. 3.4, our method outperforms baselines significantly and consistently, justifying that our method is scalable and generalizes to different corpus sizes. For example, without using any extra data, our method yield up to +1.49, +0.20, +0.35 BLEU scores on WMT20, AI Challenger 2018 and translation2019zh datasets, respectively.

Step	Method	LMN→K		MLN→K		KMN→L		MKN→L		KLN→M		LKN→M		Average	
		BLEU↑	**AD↓**	**BLEU↑**	**AD↓**	**BLEU↑**	**AD↓**	**BLEU↑**	**AD↓**	**BLEU↑**	**AD↓**	**BLEU↑**	**AD↓**	**BLEU↑**	**AD↓**
0		31.10		31.10		30.04		30.04		30.72		30.72		30.62	/
1	KD	$31.18_{0.08}$	0.00	$31.57_{0.47}$	0.00	$29.62_{-0.42}$	0.42	$29.84_{-0.20}$	0.20	$31.33_{0.61}$	0.00	$30.53_{-0.19}$	0.19	$30.68_{0.06}$	0.13
	EWC	$31.34_{0.24}$	0.00	$31.58_{0.48}$	0.00	$29.61_{-0.43}$	0.43	$29.85_{-0.19}$	0.19	$31.33_{0.61}$	0.00	$30.24_{-0.48}$	0.48	$30.85_{0.04}$	0.18
	CL-NMT	$31.50_{0.40}$	0.00	$31.59_{0.49}$	0.00	$29.62_{-0.42}$	0.42	$29.87_{-0.17}$	0.17	$31.33_{0.61}$	0.00	$31.18_{0.46}$	0.00	$30.66_{0.23}$	0.10
	Ours	$31.78_{0.68}$	0.00	$31.82_{0.72}$	0.00	$30.36_{0.32}$	0.00	$30.43_{0.39}$	0.00	$31.38_{0.66}$	0.00	$31.34_{0.62}$	0.00	$31.19_{0.57}$	0.00
2	KD	$30.96_{-0.14}$	0.22	$30.76_{-0.34}$	0.81	$30.22_{0.18}$	0.42	$30.34_{0.30}$	0.20	$30.54_{-0.18}$	0.79	$31.30_{0.58}$	0.19	$30.69_{0.07}$	0.44
	EWC	$31.26_{0.16}$	0.08	$31.26_{0.16}$	0.32	$30.34_{0.30}$	0.43	$29.78_{-0.26}$	0.26	$30.64_{-0.08}$	0.69	$30.42_{-0.30}$	0.48	$30.79_{0.00}$	0.37
	CL-NMT	$31.55_{0.45}$	0.00	$30.86_{-0.24}$	0.73	$29.95_{-0.09}$	0.42	$30.08_{0.04}$	0.17	$31.02_{0.30}$	0.31	$31.30_{0.58}$	0.00	$30.62_{0.17}$	0.27
	Ours	$31.87_{0.77}$	0.00	$31.93_{0.83}$	0.00	$30.46_{0.42}$	0.00	$30.54_{0.50}$	0.00	$31.38_{0.66}$	0.00	$31.33_{0.61}$	0.01	$31.25_{0.63}$	0.00
3	KD	$23.74_{-7.36}$	7.44	$23.64_{-7.46}$	7.93	$23.11_{-6.93}$	7.53	$24.62_{-5.42}$	5.92	$26.00_{-4.72}$	5.33	$27.81_{-2.92}$	3.68	$24.82_{-5.80}$	6.31
	EWC	$29.52_{-1.58}$	1.82	$29.63_{-1.47}$	1.95	$29.84_{-0.20}$	0.92	$28.88_{-1.16}$	1.16	$29.88_{-0.84}$	1.45	$30.10_{-0.62}$	0.80	$28.77_{-0.98}$	1.35
	CL-NMT	$28.64_{-2.46}$	2.91	$30.30_{-0.80}$	1.29	$30.00_{-0.04}$	0.42	$25.07_{-4.97}$	5.18	$31.15_{0.43}$	0.31	$27.47_{-3.25}$	3.83	$29.64_{-1.85}$	2.32
	Ours	$\mathbf{31.89}_{0.79}$	**0.00**	$\mathbf{31.94}_{0.84}$	**0.00**	$\mathbf{30.47}_{0.43}$	**0.00**	$\mathbf{30.55}_{0.51}$	**0.00**	$\mathbf{31.52}_{0.80}$	**0.00**	$\mathbf{31.54}_{0.82}$	**0.01**	$\mathbf{31.32}_{0.70}$	**0.00**

Table 16: Results on German-to-English datasets. "LMN→K" denotes K is the student model and model L, M and N are teacher models in step 1 to 3 respectively.

Model	**Domain**	**Training**	**Dev.**	**Test**
K	News	4,500,000	4,000	8,000
L	Multiple	4,300,000	4,000	8,000
M	Europarl	1,300,000	4,000	8,000
N	Tanzil	500,000	4,000	8,000

Table 17: The domain, training and evaluation corpora of the four Transformer-base models used in the German-to-English translation experiments. Other settings are kept identical to Chinese-to-English translation.

D.3 German-to-English Translation

As shown in Table 17, the training data for German-to-English translation are from WMT16, Tilde-MODEL v2018 (Rozis and Skadiņš, 2017), Tanzil v1 and Europarl (Koehn, 2005), respectively. We sample $4,000$ sentences from the original corpus as the development set and $8,000$ sentences as the test set. Model N has weaker performance. Other settings are kept identical to the Chinese-to-English translation experiments. K, L, M, N are all Transformer-base models. The three models K, L and M are combined in different orders to form six groups of experiments. To simulate the most challenging scenario, N with weaker performance is added at the end of these experiments to test the performance of our method on poor models. According to the task definition illustrated above, the transfer set of each set of experiments is the training set of the according student model.

As shown in Table 16, our method performs similarly on the German-to-English language pair as it does on the Chinese-to-English language pair under the homogeneous model setting. We also repeat the experiments under the heterogeneous and malicious model settings. As shown in Table 18 and Table 19, our method is also superior to the baselines under these settings for German-to-English translation.

Method	**Transformer-base**		**RNN**		**Transformer-big**	
	BLEU↑	**AD↓**	**BLEU↑**	**AD↓**	**BLEU↑**	**AD↓**
Original	30.62	/	30.62	/	30.62	/
KD	$30.68_{0.06}$	0.13	$29.91_{-0.71}$	0.71	$30.46_{-0.16}$	0.17
EWC	$30.66_{0.04}$	0.18	$30.19_{-0.43}$	0.43	$30.60_{-0.02}$	0.06
CL-NMT	$30.85_{0.23}$	0.10	$25.53_{-5.09}$	5.09	$26.87_{-3.75}$	3.75
Ours	$\mathbf{31.19}_{0.57}$	**0.00**	$\mathbf{30.61}_{-0.01}$	**0.09**	$\mathbf{31.10}_{0.48}$	**0.00**

Table 18: Results for different architecture models in step 1 on German-to-English datasets, averaged over six configurations.

Method	**Transformer-base**		**RNN**		**Transformer-big**	
	BLEU↑	**AD↓**	**BLEU↑**	**AD↓**	**BLEU↑**	**AD↓**
Original	30.62	/	30.62	/	30.62	/
KD	$20.03_{-10.59}$	10.59	$16.17_{-14.45}$	14.45	$19.56_{-11.06}$	11.06
EWC	$24.29_{-6.33}$	6.33	$23.40_{-7.22}$	7.22	$25.13_{-5.49}$	5.49
CL-NMT	$11.67_{-18.95}$	18.95	$5.09_{-25.53}$	25.53	$11.90_{-18.72}$	18.72
Ours	$\mathbf{30.62}_{0.00}$	**0.00**	$\mathbf{30.62}_{0.00}$	**0.00**	$\mathbf{30.62}_{0.00}$	**0.00**

Table 19: Results for malicious models in step 1 on German-to-English datasets, averaged over six configurations.

D.4 Comparison with Multi-teacher Knowledge Distillation

Multi-teacher distillation differs with our method in three aspects:

- **Applicable scenario.** Vanilla multi-teacher distillation averages the outputs of all teacher models as the target distribution, which requires all teacher models available at the same time, violating the task definition that a se-

Method	BCDE→A	CBDE→A	ACDE→B	CADE→B	ABDE→C	ABDE→C
Multi-teacher KD	45.72	45.72	26.94	26.94	18.82	18.82
Ours	**45.89**	**46.08**	**28.49**	**28.51**	**19.15**	**18.98**

Table 20: BLEU scores of multi-teacher KD and our method. Multi-teacher KD violates the task definition and is not applicable in our scenario.

quence of teacher models are distilled in many steps. It is impossible to get all teacher models at early steps. Thus, vanilla multi-teacher distillation cannot be used as a baseline method.

- **Robustness.** Vanilla multi-teacher distillation averages the output of all teacher models and is vulnerable to D^{-}_{trans}.
- **Storage requirement.** The memory footprint of vanilla multi-teacher distillation will exceed the available memories of GPUs as the number of teacher models increases. A straightforward way to alleviate the problem is storing the output of teachers in a similar way as the aforementioned knowledge inheritance. However, it is non-trivial to achieve a good balance between storage requirement and performance. Let $|D_{\text{trans}}|$ be token numbers of target sentences in D_{trans}, N_{step} be the number of steps, and $N_{\mathcal{V}}$ be the output vocabulary size. The following three high-potential methods all face problems:
  - *Storing logits*: It leaves room for integrating knowledge filtration. However, its storage requirement is$|D_{\text{trans}}| \cdot N_{\mathcal{V}} \cdot N_{\text{step}}$, which is impractical since N_{step} can be arbitrarily large.
  - *Storing top-1 tokens*: It requires constant storage of $|D_{\text{trans}}|$. However, knowledge filtration is hard if not impossible to be developed.
  - *Storing moving average of the logits*: Its storage requirement is also constant, i.e., $|D_{\text{trans}}| \cdot N_{\mathcal{V}}$. However, it is prone to D^{-}_{trans} and knowledge filtration is also hard to be integrated.

  In contrast, our knowledge inheritance has been shown to be effective and requires a constant size of storage ($|D_{\text{trans}}| \cdot N_{\mathcal{V}}$). Moreover, we still try our best to apply multi-teacher distillation on continual KD for NMT by storing the output logits of all teacher models. The results in Table 20 show no performance advantage over our method.

Overall, our proposed method is superior to vanilla multi-teacher distillation for continual KD for NMT.

E Exploring Loss Function of Knowledge Filtration

In this section, we will study the effects of inverse KL loss and modifying sources of knowledge filtration loss. All experiments are conducted under the homogeneous model setting on the Chinese-to-English language pair.

E.1 Inverse KL Loss

Trivial KL loss is zero-avoiding and concentrates on a single mode, while inverse KL loss covers the broad range and is zero-pursing.

$$\mathrm{KL}(P_1||P_2) = -\sum_i P_1(i) \ln \frac{P_2(i)}{P_1(i)} \quad (14)$$

$$\mathrm{InvKL}(P_1||P_2) = -\sum_i P_2(i) \ln \frac{P_1(i)}{P_2(i)} \quad (15)$$

The BLEU scores of the trivial KL loss and inverse KL loss are 31.07 and 30.29, respectively. Therefore, the trivial KL loss performs better in our task.

E.2 Effect of Knowledge Filtration Loss

In the knowledge filtration loss ℓ_{KF}, ℓ_{KD} motivates the model to learn from student, while ℓ_{NEG} motivates the model to learn against student. Intuitively, ℓ_{NEG} is calculated from poor predictions from teacher models, and these error-prone distributions might provide empirical knowledge that motivates the student model not to make the same mistakes as the teachers. The output distributions from the student model could be improved by pushing them away from poor distributions of corresponding tokens output by teacher models. Conversely, random distributions are not beneficial to the student model. To verify this, we replace ℓ_{NEG} with the following noise:

Source of Noise Sample	Sample Size	BLEU
Uniform distribution	1	30.91
Normal distribution	1	30.86
Shuffled Batch (Attached)	1	29.98
Shuffled Batch (Attached)	5	29.99
Shuffled Batch (Detached)	1	30.52
Shuffled Batch (Detached)	5	30.62
Negative KD Loss		**31.07**

Table 21: Results for replacing negative KD loss with noise sample. The metrics are averaged over six configurations.

- **Noise sampled from uniform distribution.** The probability distribution that replaces the original negative loss is obtained by sampling from a uniform distribution and passing through a softmax layer.
- **Noise sampled from normal distribution.** The probability distribution that replaces the original negative loss is obtained by sampling from a normal distribution and passing through a softmax layer.
- **Noise from shuffled batch.** We randomly pick up a prediction distribution as the noise distribution from the batch from which the original negative sample is. It is worth noting that this kind of noise is usually single-peaked, high-confidence, and more similar to the original negative sample compared to the above two noises.
  - **Attached** The negative KD loss here is included when calculating the gradient of the sampled negative noise sample;
  - **Detached** The negative KD loss here is excluded when calculating the gradient of the sampled negative noise sample.

As shown in Table 21, performance is degraded when negative KD loss is derived from noise, thus illustrating that negative samples assist student models in avoiding errors and improving the efficiency of knowledge distillation.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
"Limitation" Section

☑ A2. Did you discuss any potential risks of your work?
"Ethics Statement" Section

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☒ Did you use or create scientific artifacts?

Left blank.

☐ B1. Did you cite the creators of artifacts you used?
No response.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
No response.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
No response.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
No response.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
No response.

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
No response.

C ☑ Did you run computational experiments?

Section 3, Appendix B, Appendix C, Appendix D and Appendix E

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 3.2, Appendix B and Appendix D

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 3.2, Appendix A, Appendix B and Appendix D

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 3.2, Appendix A, Appendix B, caption of Table 2, caption of Table 3, caption of Table 4, caption of Table 5, caption of Table 6 and caption of Table 7

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 3.2, Appendix A and Appendix B

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.