Multi-Teacher Distillation with Single Model for Neural Machine Translation

Xiaobo Liang, Lijun Wu, Juntao Li, Tao Qin, *Senior Member, IEEE,* Min Zhang, and Tie-Yan Liu, *Fellow, IEEE*

***Abstract*—Knowledge distillation (KD) is an effective strategy for neural machine translation (NMT) to improve the performance of a student model. Usually, the teacher can guide the student to be better by distilling the soft label or data knowledge from the teacher itself. However, the data diversity and teacher knowledge are limited with only one teacher model. Though a natural solution is to adopt multiple randomized teacher models, one big shortcoming is that the model parameters and training costs are largely increased with the number of teacher models. In this work, we explore to mimic multiple teacher distillation from the sub-network space and permuted variants of one single teacher model. Specifically, we train a teacher by multiple sub-network extraction paradigms: sub-layer reordering, layer-drop, and dropout variants. In doing so, one teacher model can provide multiple outputs variants and causes neither additional parameters nor much extra training cost. Experiments on 8 IWSLT datasets: IWSLT14 En↔De, En↔Es and IWSLT17 En↔Fr, En↔Zh and the large WMT14 EN→DE translation tasks show that our method even achieves nearly comparable performance with multiple teacher models with different randomized parameters, both word-level and sequence-level knowledge distillation. Our code is available at GitHub[1].**

***Index Terms*—Neural Machine Translation, Knowledge Distillation, Sub-networks, Dropout, Multiple Teachers.**

I. Introduction

NEURAL machine translation (NMT) has witnessed significant progress with the powerful neural architectures [1], [2], [3], [4] and has been further improved by various knowledge distillation (KD) methods in recent years [5], [6], [7], [8]. Through distilling and gathering knowledge from the teacher model, the model performance of the student can be enhanced substantially. Ideally, multiple teachers [8], [9] are more preferable and effective where each teacher can capture diversified knowledge from different representation spaces, but it significantly increases model parameters and training costs. Thus, one single teacher model is a pragmatic choice and has been widely adopted [10], [11]. In this paper, we stick with the line of one teacher model in KD and focus on performance improvement rather than model compression.

Although utilizing one single teacher model is more applicable and feasible, it inherits the intrinsic shortage and limitation of one model, i.e., the teacher's representation capability and knowledge are limited. As a result, the teacher model can only generate data with less knowledge and low diversity under the data distillation paradigm, which is a common setting in KD. Unfortunately, data diversity has been proven to be a critical factor in the success of NMT system [12], thus one single teacher model can only yield a limited benefit for the student. Therefore, in terms of the effectiveness and applicability, one question is: *can we tap the potential of one single teacher model so as to capture rich knowledge and high diversity, and in turn to improve the student model?*

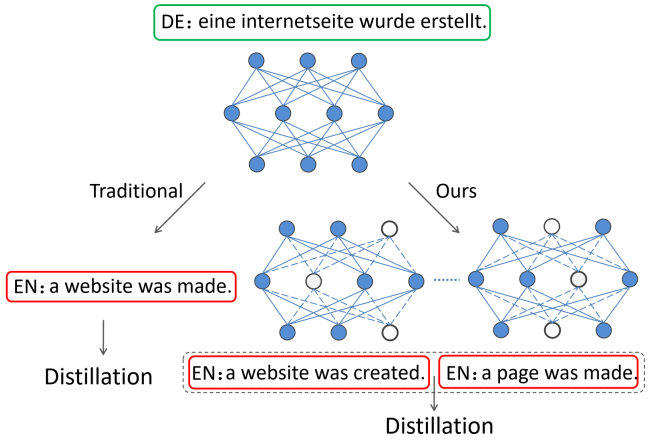


Fig. 1. The traditional teacher model (left) is a full model, which outputs stationary translation result. Instead, our teacher model (right) are sub-models, which outputs diversified and rich translation sequences.

Despite the fact that KD has obtained remarkable success in improving the student model, it neglects the side effects introduced from the teacher model. One single teacher model is only a sample point of the distribution of parameters, which is incomplete to describe or model the given data. To put it bluntly, one single teacher model is risky to represent the given data and can only capture limited knowledge. That is, a stationary teacher distillation objective contains biased and noisy signals for the knowledge distillation process, i.e., the teacher model may output the incorrect and over-confident label. In this case, the student model can either only capture incomplete knowledge or obtain unexpected information. Such risk will be amplified in the knowledge distillation process and restrict the effectiveness of knowledge distillation. The root of above problems is that one single model is incapable of correctly modeling data multi-view [13], especially in such multi-modal dataset like NMT corpus.

An interesting observation for the over-parameterized NMT

Xiaobo Liang and Lijun Wu contribute equally to this paper.

Xiaobo Liang, Juntao Li and Min Zhang are with Soochow University, Suzhou, China (e-mail: 20204027012@stu.suda.edu.cn; ljt@suda.edu.cn; minzhang@suda.edu.cn). Juntao Li is the corresponding author.

Lijun Wu, Tao Qin, and Tie-Yan Liu are with Microsoft Research, Beijing 100080, China (e-mail: lijuwu@microsoft.com; taoqin@microsoft.com; tyliu@microsoft.com).

This work is supported by the National Science Foundation of China (NSFC No. 62036004)

[1]https://github.com/dropreg/RLD

model is that some specific sub-models extracted from the full model can gain diversified data at the cost of tiny performance decreases. Fig. 1 shows that compared with the static and simplex output distribution of a full teacher model, the sub-models can represent dynamic and rich output distribution. More concretely, the traditional teacher full model can only output static and simplex translation results, but the sub-models can represent dynamic and rich results, i.e., "erstellt" (DE) can be translated as "created" or "made". Even if obtained an unfaithful translation, it can be corrected by considering multiple sub-models decisions. This manner can break through the abovementioned bottleneck of the traditional distillation method and efficiently obtain the sub-model knowledge. In particular, we explore variety extraction operations to find suitable sub-models, including sub-layer reordering [14], layer drop [15], and multiple dropout variants [16], i.e., attention dropout and activation dropout. Through these strategies, our single-model in "training" mode can mimic multiple teachers and provide more diversified knowledge.

We launch experiments on the widely acknowledged WMT14 English→German, and 8 IWLST translation tasks for evaluation, ranging from low-resource to rich-resource scenarios. Our proposed method achieves substantial performance improvement over Transformer by near 2.0 BLEU points and also consistently outperforms the conventional single teacher distillation of both word-level and sequence-level KD strategies, but with limited extra cost. Besides, with our proposed simple teacher model training, a single model can even achieve almost comparable performance with multiple models with different randomized parameters in the scenarios of both KD and model ensemble.

II. Preliminary

A. Knowledge Distillation

Knowledge distillation (KD) [17], [5] aims at transferring knowledge from the teacher model to the student and has long been studied in the era of neural networks, no matter for computer vision [18], [19], [20] or natural language processing [21], [22], [23], [24]. Specifically for NMT, [6] first apply KD method and introduce two sequence-level KD variants to improve the translation performance. Since then, various KD methods have been proposed [25], [26], [7]. One typical research line is to distill the knowledge from one single teacher model to a student model [10], [27], [28], [11], and another direction is to transfer the knowledge from multiple teacher models to the student model, where the teachers usually have the same architecture/size as the student [29], [9], [8]. We focus on the former one since one single teacher is more feasible in NMT, but we aim to tap the potential of one teacher model like multiple teachers to provide more knowledge and high diversity for data distillation.

The current method usually selects high probability sequences using beam search but remarkably reduces the diversity. It is not an appropriate approach to collecting the high-ranking sequence in a beam. Because the model tends to spread too much probability mass over the hypothesis space [30], leading high-quality candidates may not appear in a narrow beam. Model ensemble or multiple teacher ensemble distillations can alleviate the above problem by introducing numerous model hypothesis spaces. The drawbacks are also obvious that the cost of training and inference increases linearly with the number of models. But in fact, over-parameterized teacher model that contains rich knowledge has not been explored deeply [31]. Therefore, it is crucial to develop solutions to enable beam search to find high-quality candidates in the single teacher hypothesis space. In our work, we skip the full model hypothesis space to explore the approximately sub-model hypothesis space. For the views of the sub-model, It is avoidable that the problem of "probability mass" occurs in the full model beam.

B. Sub-Model Extraction

Our work is also closely related to the recent works on architecture exploration, more concretely, sub-layer reordering in Transformer structures. [32] first explore the sub-layer reordering in Transformer and find a sandwich pattern for arranging the Transformer sub-layers for downstream tasks. They find that such sandwich Transformers can improve multiple language modeling benchmarks but fail on NMT tasks for strong BLEU scores. After that, [14] figure out a way to improve the NMT performance through an instance-wise sub-layer reordering for Transformer structure, motivated by the empirical observation that different instances favor different sub-layer orders of one single Transformer model in the training process. Different from them, we introduce sub-layer reordering only for the teacher model under the knowledge distillation setting, in order to enrich the knowledge and capability of the teacher model, and thus enhance the data distillation for student model.

Besides, we also explore dropout-based methods and demonstrates that dropout is a better method to extract sub-model. Dropout [16] is a technique widely used in neural network training, i.e., neuron dropout, attention dropout, activation dropout, layer drop [15], and so on [33]. In the over-parameterized scenario, dropout prevents neurons from overfitting when learning from data by reducing the co-adaptation between neurons. It is simple to use that keep dropout open for training and turning it off for inference. It is interesting that a large number of sub-models extracted by dropout do not keep unbiased learning in training time. That is, these distinct architecture sub-models can obtain different patterns from randomly observed data, and that makes sub-models obtain various representation abilities and generate diversified sentences. However, the full model will make consistent decisions to balance different sub-models in inference time, and a large amount of learned knowledge will be lost. This traditional approach is an information bottleneck for the teacher network in KD. To avoid this situation, we will use sub-models as teachers in distillation to maintain consistency in training and inference.

III. Methodology

In this section, we first present multiple training paradigm to extract sub-models of Transformer, which can enhance

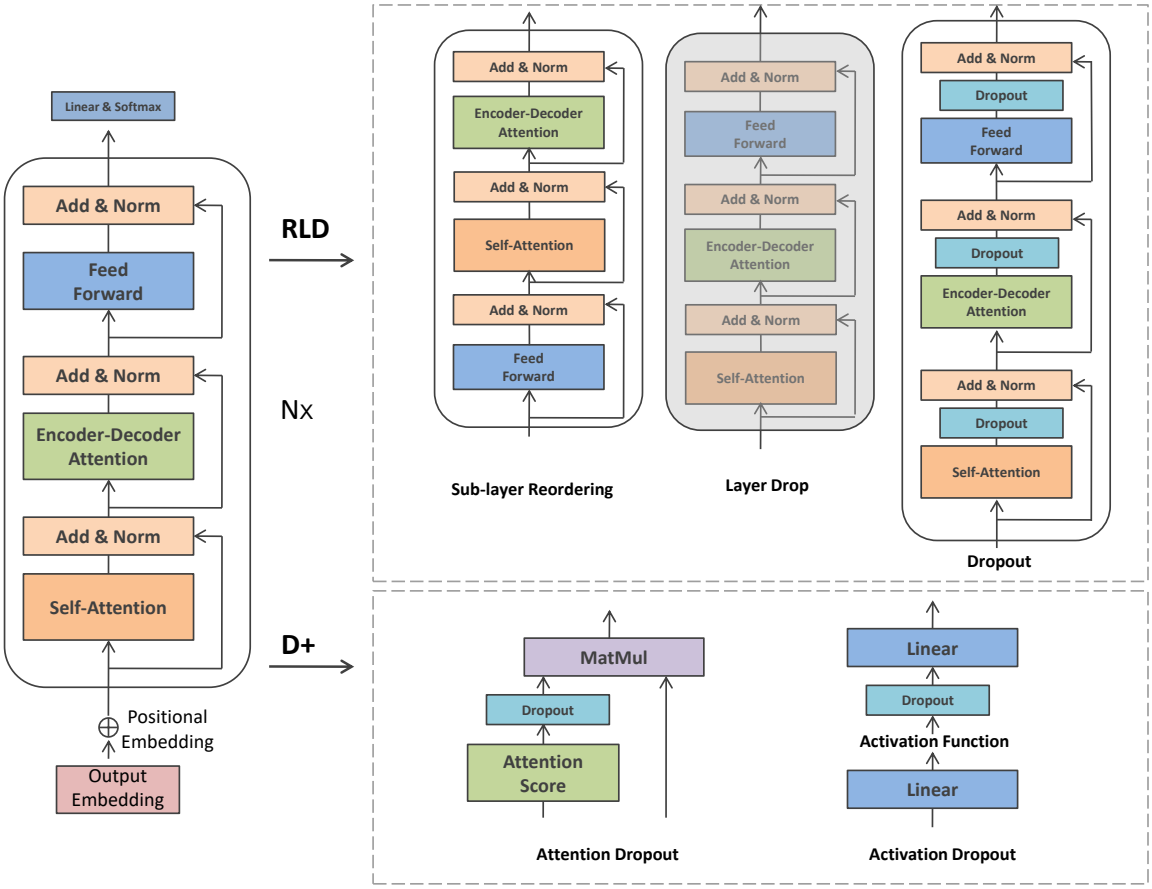


Fig. 2. The overview of our method, which simplifies the Transformer model with only decoder blocks.

the teacher knowledge and representation capability in Section III-A. To further facilitate the knowledge transfer from the teacher model, we also present an effective distillation strategy and apply it into word-level KD and sequence-level KD scenarios in Section III-B.

A. Training Transformer's Sub-Model

Transformer [3] has been the dominant architecture for the machine translation task with an encoder-decoder structure, where both encoder and decoder are composed of a stack of L identical blocks. Each encoder block comprises two sub-layers, i.e., multi-head self-attention layer (SA) and fully connected feed-forward layer (FF), with a fixed order SA→FF, while each decoder block has an extra encoder-decoder attention layer (ED) with the fixed sub-layer order of SA→ED→FF.

Given the training dataset D=$\{(x_i, y_i)\}_{i=1}^{M}$, the l-th encoder block takes the representation of a sequence x_i from the (l-1)-th block as input, denoted as $e_{x_i}^{l-1}$, and sequentially processes the input with self-attention sub-layer (SA) and feed-forward sub-layer (FF), which are formulated as:

$$\begin{aligned} \hat{e}_{x_i}^{l} &= SA(e_{x_i}^{l-1}) + e_{x_i}^{l-1}, \\ e_{x_i}^{l} &= FF(\hat{e}_{x_i}^{l}) + \hat{e}_{x_i}^{l}, \end{aligned} \tag{1}$$

where $1 \leq l \leq L$. Besides, there is a residual connection between the input and the output of each sub-layer, followed by a layer normalization operation[2]. Similarly, for the decoder block, the representation of a sequence y_i from the (l-1)-th decoder block is sequentially processed by the multi-head self-attention layer (SA), encoder-decoder attention layer (ED) and fully connected feed-forward layer (FF), written by:

$$\begin{aligned} \hat{e}_{y_i}^{l} &= SA(e_{y_i}^{l-1}) + e_{y_i}^{l-1}, \\ \bar{e}_{y_i}^{l} &= ED(\hat{e}_{y_i}^{l}, e_{x_i}^{L}) + \hat{e}_{y_i}^{l}, \\ e_{y_i}^{l} &= FF(\bar{e}_{y_i}^{l}) + \bar{e}_{y_i}^{l}, \end{aligned} \tag{2}$$

where $e_{x_i}^{L}$ is the output of the last encoder layer L, and there is also a residual connection and layer normalization for each sub-layer.

Then, we introduce various ways to train Transformer, as shown in Fig. 2. There are some ways to slightly affect the Transformer performance: sub-layer Reordering, LayerDrop, and Dropout, denoted as RLD method. In contrast, attention Dropout and activation Dropout dramatically changes the Transformer performance, namely RLD+ method in short.

a) Sub-layer Reordering: As introduced before, we are interested in discovering diverse and rich knowledge from one single teacher model so that the resulted data distillation can largely boost the student model. To achieve this goal, inspired by [32], we perform sub-layer reordering for the

[2]We omit the layer normalization formula for a clear representation of later sub-layer reordering.

Transformer to break up the fixed order in the training process. Sub-layer reordering can reduce the co-adaption between sub-layers and make the model robust. More accurately, we only launch sub-layer reordering for the decoder blocks, insomuch as reordering sub-layers for both encoder and decoder blocks does not achieve better performance [14]. For the 3 different sub-layers in the decoder block, we can obtain $A_3^3 = 6$ different sub-layer orders, e.g., SA→ED→FF, FF→SA→ED. Take a specific order SA→FF→ED for example, it produces the representation of a sequence y_i from the (l-1)-th decoder block by computing:

$$\begin{aligned} \hat{e}^l_{y_i} &= SA(e^{l-1}_{y_i}) + e^{l-1}_{y_i}, \\ \widetilde{e}^l_{y_i} &= FF(\hat{e}^l_{y_i}) + \hat{e}^l_{y_i}, \\ e^l_{y_i} &= ED(\widetilde{e}^l_{y_i}, e^L_{x_i}) + \widetilde{e}^l_{y_i}. \end{aligned} \tag{3}$$

b) LayerDrop: LayerDrop [15] targets to regularize deep Transformers and stabilizes their training using groups dropping. However, LayerDrop does not explicitly provide a way to select which groups to prune. It considers several different pruning strategies, and we apply "Every Other" in our work, which is a straightforward strategy to simply drop every other block. Each block is dropped independently with a hyper-parameter p_l>0 that controls the drop rate.

c) Dropout: As opposed to structured pruning, dropout is an effective method for weight pruning, which is to randomly drop neurons of neural networks during training with a pre-defined probability p_d>0. In practice, dropout is applied to each sub-layer for Transformer, both encoder and decoder blocks by default:

$$\begin{aligned} \hat{e}^l_{x_i} &= Dropout(SA(e^{l-1}_{x_i})) + e^{l-1}_{x_i}, \\ e^l_{x_i} &= Dropout(FF(\hat{e}^l_{x_i})) + \hat{e}^l_{x_i}, \end{aligned} \tag{4}$$

d) Attention Dropout: An attention function maps a query and a set of key-value pairs to output in the SA and ED layer, computed as a weighted sum of the values. Attention scores are equals to multiply the query by key. Attention Dropout randomly drops the attention similarity score, which is an efficient way to reduce the co-adaptation between query and key. It is an easy way to implement but effective to improve the model performance.

e) Activation Dropout: The feed-forward layer contains two linear layers and an activation function layer, which includes many neurons and is over-parameterized. To overcome the above limitation, activation dropout is crucial to the current layer by adding a dropout layer behind the activation function. Due to the improvement for model performance, we can utilize these two methods (Attention Dropout and Activation Dropout) to build an influential teacher to explore the factor of teacher ability.

f) Training and Inference: With the paired dataset D and the Transformer model, the training objective of the NMT task is to minimize the following cross-entropy loss:

$$\begin{aligned} L_{NLL}(D;\theta) &= -\sum_{i=1}^{M} \log p(y_i|x_i;\theta), \\ \log P(y_i|x_i;\theta) &= \sum_{j=0}^{|y_i|} \log p(y_i^j|y_i^{<j}, x_i;\theta), \end{aligned} \tag{5}$$

where θ refers to the parameters of the Transformer model and M is the size of the training dataset. $|y_i|$ represents the length of the i-th ground-truth sentence y_i and $y_i^{<j}$ refers to the tokens before the j-th time step.

In training time, all of the above mentioned methods are adopted with probability to extract the sub-models from the full teacher model for training. Specifically, at each step, LayerDrop will decide if any layer will be dropped, sub-layer reordering will decide one specific order of the decoder block, dropout (including attention and activation dropout) will decide the neurons to be dropped, then the data in this step will be trained with decided sub-model. Therefore, different sub-models are trained at different training steps. But overall, we only have a full teacher model. RLD based teacher model can effectively increase the diversity of the sub-models, and RLD+ based teacher model can significantly improve the presentation ability and performance of the sub-models.

In inference time, similarly, we still use these methods to extract the sub-models as in teacher training phrase to generate multiple translation candidates, representing more knowledge that can distill the student. Specifically, all extraction strategy will be applied to obtain sub-models, and the randomness of each strategy will affect the diversity of the generated data. One question is how many sub-models we will extract, which we will discuss separately according to different distillation methods in the following section. Furthermore, one may concern the training cost is increased with multiple paradigms. As we present in Section V-C, the training cost of our model is actually not increased due to faster convergence.

B. Sub-Model Knowledge Distillation

The vanilla KD method is to transfer knowledge from teacher to student through word-level soft label distillation. But for sequence prediction tasks like machine translation, sequence-level distillation is more effective than word-level distillation [6]. This section will show how to apply our methods in two different KD scenarios, and our student model is single vanilla transformer unless otherwise specified.

a) Word-level KD: The traditional word-level distillation training objective is to minimize the following cross-entropy term and Kullback-Leibler Divergence (KL) loss:

$$\begin{aligned} L(D;\theta) &= L_{NLL}(D;\theta) + \alpha L_{KD}(D;\theta), \\ L_{KD}(D;\theta) &= \sum_{i=1}^{M}\sum_{j=0}^{|y_i|}\sum_{k=0}^{|V|} q(y_i^j = k|y_i^{<j}, x_i;\theta_t)\times \\ &\quad \log p(y_i^j = k|y_i^{<j}, x_i;\theta), \end{aligned} \tag{6}$$

where $|V|$ refers to the target vocabulary set, α is the Co-efficient to balance the cross entropy loss and KL loss. θ_t denotes the teacher model parameters, q represents the teacher output distribution, and p corresponds to the student output distribution. The only difference is how to generate distribution q. In our work, we will use sub-models instead of the full model by dynamic RLD extraction strategy.

The whole word-level distillation training process is shown in Algorithm 1. Line 1 to 7 is training the teacher model. Each sentence is trained by extracting sub-networks randomly, as given in line 5. Line 6 represents that the model accumulates

Algorithm 1 Word-level KD Algorithm

Input: Training data $D = \{(x_i, y_i)\}_{i=1}^{M}$.
Output: model parameter θ_S.

1: ▷ Train Transformer variants as teacher.
2: Initialize teacher M_T with parameters θ_T.
3: **while** not converged **do**
4: $(x_i, y_i) \sim D$
5: $\hat{\theta_T} \leftarrow$ExtractSubModel$(\theta_T)$
6: $\hat{\theta_T} \leftarrow$ GradientUpdate $(\hat{\theta_T}, \bigtriangledown_{\hat{\theta_T}} L_{NLL})$
7: **end while**
8: ▷ Train standard Transformer as student.
9: **while** not converged **do**
10: $(x_i, y_i) \sim D$
11: $\hat{\theta_T} \leftarrow$ExtractSubModel$(\theta_T)$
12: $\hat{y}_i = M_T(x_i; \hat{\theta_T})$
13: $\bigtriangledown_{\theta_S} L = \bigtriangledown_{\theta_S} L_{NLL}(x_i, y_i) + \bigtriangledown_{\theta_S} L_{KD}(x_i, \hat{y}_i)$
14: $\theta_S \leftarrow$ GradientUpdate $(\theta_S, \bigtriangledown_{\theta_S} L)$
15: **end while**

Algorithm 2 Sequence-level KD Algorithm

Input: Training data $D = \{(x_i, y_i)\}_{i=1}^{M}$, predefined sampling number N.
Output: model parameter θ_S.

1: ▷ Train Transformer with RLD as teacher.
2: Initialize teacher M_T with parameters θ_T.
3: **while** not converged **do**
4: $(x_i, y_i) \sim D$
5: $\hat{\theta_T} \leftarrow$ExtractSubModel$(\theta_T)$
6: $\hat{\theta_T} \leftarrow$GradientUpdate $(\hat{\theta_T}, \bigtriangledown_{\hat{\theta_T}} L)$
7: **end while**
8: ▷ Distill data from teacher.
9: Initialize distilled datasets $D_{dis} = \{\emptyset\}$
10: **for** k from 1 to N **do**
11: $\hat{\theta_T} \leftarrow$ExtractSubModel$(\theta_T)$
12: **for** $(x_i, y_i) \in D$ **do**
13: $\hat{y}_i = M_T(x_i; \hat{\theta_T})$
14: $D_{dis} \leftarrow D_{dis} \cup (x_i, \hat{y}_i)$
15: **end for**
16: **end for**
17: ▷ Train standard Transformer as student.
18: **while** not converged **do**
19: $(x_i, y_i) \sim (D \cup D_{dis})$
20: $\theta_S \leftarrow$ GradientUpdate $(\theta_S, \bigtriangledown_{\theta_S} \lambda_i L)$
21: **end while**

all instance gradients to update the parameters based on loss function L. Next, the teacher model distills the soft labels to the student model, which are generated by teacher sub-models in line 11. The student is trained directly by minimizing the NMT loss and KD term in line 14.

b) Sequence-level KD (data distillation): Since the dominant NMT model is an autoregressive model, generating current translation word is based on the previously generated context. The soft label can not apply to distill sequence-level knowledge, in which the length of the ground truth sequence is usually not equal to the translation sequence. Therefore, we alternatively employ the data distillation strategy, i.e., the teacher first generates translation sequences and then adds these data to the training. More specifically, we first train a Transformer model as a teacher, where the sub-networks are constantly extracted to fit the bilingual data D in training time. After training, we can repeat extracting the sub-network several times to generate the target translations. With different sub-networks, the generated translations bring more diversified and feasible data knowledge to be used in distillation. We then collect the teacher predictions and add them to the bilingual data as a new dataset D_{dis}.

The whole training process is shown in Algorithm 2. Line 1 to 7 is the teacher model training process like word-level KD. Subsequently, the teacher model generates a corresponding dataset from line 9 to 16. Furthermore, the teacher extracts sub-network from the whole model with RLD for generating translation candidates, similar to the training stage at line 11. The remaining lines show how to train the student model and make use of the distilled data.

In our sub-model KD settings, RLD teacher need to dynamically extract sub-models to guide student learning. Therefore, one question is about the number of extracted sub-models. There are some differences between word-level and sequence-level distillation settings. (1) For word-level KD, the teacher model generates soft label in an online way. Hence, at each student training step, one specific teacher sub-model will be randomly chosen to generate the soft label. Along the student model training process, there will be plenty of teacher sub-models. (2) For sequence-level KD, we choose a fixed sampling number N for teacher sub-models to generate translations for distillation, which will affect the distillation performance.

IV. Experiments

We conduct experiments on both low-resource and rich-resource machine translation tasks to evaluate the effectiveness of our approach.

A. Experimental Design

a) Dataset: For the low-resource scenario, we perform on four language pairs, which are IWSLT14 [34] English↔German (En↔De), English↔Spanish (En↔Es), and IWSLT17 [35] English↔French (En↔Fr), English↔Chinese (En↔Zh) translation pairs, with total 8 translation tasks. Following previous works [3], [14], after the common data processing, there are about $160k$, $183k$, $236k$, $235k$ bilingual sentence pairs for training on En↔De, En↔Es, En↔Fr, En↔Zh translation datasets. For vocabulary, we use a joint source and target dictionary with $10k$ merge operations by byte-pair-encoding (BPE) [36] algorithm, except for IWSLT17 En↔Zh task, the source and target vocabulary are separated. As for the rich-resource scenario, we work on the widely acknowledged WMT14 English→German (En→De) task. After filtering, we obtain about $4.5M$ sentence pairs for training on the WMT14 English→German (En→De) corpus, and we concatenate `newstest2012` and `newstest2013` as dev

TABLE I
BLEU SCORES OF OUR RLD METHOD AND DIFFERENT TRAINING STRATEGIES ON 8 IWSLT TRANSLATION TASKS.

Model		En→De	De→En	En→Fr	Fr→En	En→Zh	Zh→En	En→Es	Es→En	Avg	△
V Transformer		28.68	34.72	36.2	36.7	25.8	18.7	39.0	40.5	32.54	-
RLD Transformer		28.79	35.25	36.0	36.9	26.0	18.3	39.2	40.9	32.67	+0.13
RLD+ Transformer		29.12	35.59	36.6	37.2	26.1	18.7	39.7	41.4	33.05	+0.51
WD	$\text{FT}_{RLD}\text{S}_V$	29.26	35.72	36.7	37.4	26.3	19.2	39.6	41.1	33.16	+0.62
	$\text{FT}_{RLD+}\text{S}_V$	29.68	35.94	37.1	37.4	26.9	19.2	39.6	41.6	33.43	+0.89
	$\text{ST}_{RLD}\text{S}_V$	29.70	36.25	37.1	37.5	26.6	19.2	39.9	42.1	33.54	+1.0
	$\text{ST}_{RLD}\text{S}_{D+}$	30.15	36.43	37.2	37.9	27.1	19.7	40.8	42.2	33.94	+1.4
	$\text{ST}_{RLD+}\text{S}_V$	30.01	36.21	37.4	37.8	26.8	19.4	40.0	42.0	33.70	+1.16
	$\text{ST}_{RLD+}\text{S}_{D+}$	30.19	36.73	37.4	37.9	26.9	19.7	40.7	42.5	34.00	+1.46
SD	$\text{FT}_{RLD}\text{S}_V$	29.83	36.00	37.4	37.6	27.2	19.2	40.1	41.2	33.57	+1.03
	$\text{FT}_{RLD+}\text{S}_V$	30.26	36.35	37.1	37.9	27.4	19.4	40.4	42.4	33.90	+1.36
	$\text{ST}_{RLD}\text{S}_V$	30.57	36.70	37.5	38.3	27.5	19.6	40.3	42.6	34.13	+1.59
	$\text{ST}_{RLD}\text{S}_{D+}$	30.60	36.89	37.8	38.4	27.6	19.7	40.5	42.6	34.26	+1.72
	$\text{ST}_{RLD+}\text{S}_V$	30.52	36.89	37.5	38.2	27.2	19.6	40.7	42.7	34.16	+1.62
	$\text{ST}_{RLD+}\text{S}_{D+}$	30.55	37.20	37.9	38.5	27.9	19.7	40.8	42.8	34.41	+1.87

set, `newstest2014` as test set. The WMT sentences are also encoded by BPE operations, which results in a joint source and target vocabulary with about $32k$ subword tokens.

b) Training: We adopt the default optimization algorithm and learning rate schedule as in [3], that is Adam [37] optimizer with initial learning rate 0.0005, learning rate schedule `inverse_sqrt` with $4,000$ warmup steps. Label smoothing is utilized in loss function with value 0.1.

c) Evaluation: We use `multi-bleu.perl` to evaluate IWSLT14 En↔De and all WMT translation tasks for fair comparison with previous works. For the remaining tasks, we use a more advanced implementation of BLEU evaluation, `sacre-bleu` toolkit. To generate the translation sentences, we follow [3] to use beam search with width 4 and length penalty 0.6 for WMT14 En→De, beam size 5 and penalty 1.0 for other translation tasks.

d) Models: We adopt Transformer [3] architecture for all the experiments. Specifically, for all low-resource IWSLT datasets, we use `Transformer_iwslt_de_en` setting for model configuration, which contains 6 layers in both encoder and decoder, the embedding size is 512 and the FFN layer dimension is $1,024$, the dropout p_d and weight decay is 0.3 and 0.0001 respectively. For WMT experiments, the model configuration is `Transformer_vaswani_wmt_en_de_big`, with 6 blocks, embedding size $1,024$ and FFN layer size $4,096$.

e) Distillation Setting: We adopt different Transformer model settings for our experiments to analyze the influence of sub-models, whether as a teacher or a student. Here we briefly define the notation that occurs in our experiments:

- V Transformer: Vanilla Transformer with dropout.
- D+ Transformer: Vanilla Transformer with Dropout, Attention Dropout and Activation Dropout.
- RLD Transformer: Transformer model trained by sublayer Reordering, LayerDrop, and Dropout.
- RLD+ Transformer: RLD Transformer with Attention Dropout and Activation Dropout.
- WD: Word-level Knowledge Distillation.
- SD: Sequence-level Knowledge Distillation.
- $\text{FT}_{RLD}\text{S}_V$: Vanilla KD with RLD teacher Full-model and Vanilla Transformer student.
- $\text{FT}_{RLD+}\text{S}_V$: Vanilla KD with RLD+ teacher Full-model and Vanilla Transformer student.
- $\text{ST}_{RLD}\text{S}_V$: Sub-Model KD with multiple RLD teacher Sub-models and Vanilla Transformer student.
- $\text{ST}_{RLD}\text{S}_{D+}$: Sub-Model KD with multiple RLD teacher Sub-models and D+ student.
- $\text{ST}_{RLD+}\text{S}_V$: Sub-Model KD with multiple RLD+ teacher Sub-models and Vanilla Transformer student.
- $\text{ST}_{RLD+}\text{S}_{D+}$: Sub-Model KD with multiple RLD+ teacher Sub-models and D+ student.

B. Main Results

a) IWSLT Results: We first present the results of our approach on IWSLT translation tasks. The numbers are shown in Table I. From the results, we can first observe that RLD Transformer and RLD+ Transformer outperform the vanilla Transformer (baseline) by averagely 0.13 and 0.51 BLEU scores, respectively, which demonstrates the RLD+ Transformer is a more powerful teacher compare to RLD Transformer. In the subsequent experiments, we compare the influence of teachers on different distillation settings by using RLD Transformer as a weak teacher and RLD+ Transformer as a strong teacher. For word-level KD, we can directly perceive through the BLEU score 33.16 for RLD full model teacher ($\text{FT}_{RLD}\text{S}_V$). Our advanced $\text{ST}_{RLD}\text{S}_V$ training strategy further increases the performance of the translation model and obtains an average BLEU score 33.54, which clearly verifies the sub-model superiority than the full model. Besides, the additional dropout training strategy (Attention Dropout and Activation Dropout) can improve model performance consistently when applying to whether teacher (WD ($\text{ST}_{RLD+}\text{S}_V$) v.s. WD ($\text{ST}_{RLD}\text{S}_V$)) or student (WD ($\text{ST}_{RLD}\text{S}_V$) v.s. WD ($\text{ST}_{RLD}\text{S}_{D+}$)) model. And if we applied the RLD+ strategy to teachers and the D+ strategy to students ($\text{ST}_{RLD+}\text{S}_{D+}$), we can obtain the best average BLEU score 34.00 in the word-level KD settings.

TABLE II
RESULTS ON IWSLT14 DE→EN TRANSLATION.

Method	De→En
Transformer [3]	34.64
DynamicConv [38]	35.20
Macaron Network [39]	35.40
MADL [40]	35.56
IOT [14]	35.62
AutoDropout [41]	35.80
MAT [42]	36.22
$ST_{RLD}S_V$ (WD)	36.25
$ST_{RLD}S_V$ (SD)	**36.70**

TABLE III
RESULTS ON WMT14 EN→DE TRANSLATION TASK.

Method	En→De
Transformer [3]	29.12
Relative Position [43]	29.20
Scaling NMT [44]	29.30
DynamicConv [38]	29.70
Evolved Transformer [45]	29.80
IOT [14]	30.03
$ST_{RLD}S_V$ (WD)	30.02
$ST_{RLD}S_V$ (SD)	**30.25**

For sequence-level KD, we employ sampling number $N = 7$ for all language task to enhance the distillation process, although it may not be optimal for every language. For the selection of sampling number N, the analysis experiment will be presented in the next section V-A0b. We can observe that it outperforms the conventional word-level KD strategy by an average of 0.41 BLEU scores (SD ($FT_{RLD}S_V$) v.s. WD ($FT_{RLD}S_V$)), which demonstrates that the sequence information is more crucial. From Table I, the results show that our method is better than the vanilla KD method (SD ($ST_{RLD}S_V$) v.s. SD ($FT_{RLD}S_V$) and SD ($ST_{RLD+}S_V$) v.s. SD ($FT_{RLD+}S_V$)). The above studies come to the same conclusion, i.e., the sub-models extracted by RLD and RLD+ strategy are beneficial to KD than the full model. Specifically, on several translation tasks, for example, IWSLT14 De→En, IWSLT17 Fr↔En, and IWSLT17 ES↔En translations, we improve the translation performance above Transformer baseline by nearly 2.0 BLEU points.

To further show the advantage of our method, we compare the results of our method among existing works on the widely acknowledged IWSLT14 De→En task. Table II presents the detailed comparisons. We can see that $ST_{RLD}S_V$ (SD) reaches a 36.70 BLEU score, which performs better than several strong baselines, such as the multi-agent dual learning [40] framework and the multi-branch Transformer [42], both with more model parameters.

b) WMT Results: In Table III, we report the results of our method on the rich-resource task, the WMT14 En→De translation. We also compare with several existing works, such as Evolved Transformer [45] by neural architecture search and IOT [14] by instance-wise model training. Among the compared baseline methods, our method achieves a superior performance of 30.25 BLEU score. These results all support that our method can perform well both on low-resource and rich-resource translation tasks. In contrast, our method has more obvious advantages in low-resource scenarios.

TABLE IV
DIFFERENT WORD-LEVEL KNOWLEDGE DISTILLATION LOSS FUNCTION COEFFICIENTS.

α	0.5	0.7	0.9	1.0	1.1	1.3	1.5
BLEU	35.73	35.95	36.15	**36.25**	36.16	36.09	36.20

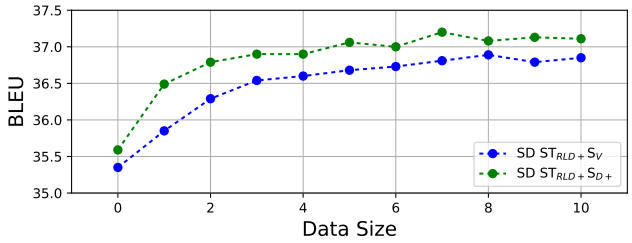


Fig. 3. BLUE score of our method trained with different amount of data.

V. ANALYSIS AND DISCUSSION

We have shown the strong performance of our approach. To give a better understanding of the relationship between the techniques of sub-model extraction and knowledge distillation, we conduct several detailed analyses and discussions in this section. These studies are mainly taken on IWSLT14 De→En translation, except there is a clear explanation.

A. Ablation study

Our framework is based on knowledge distillation, specifically, the teacher training and distilled student model training with extracted sub-models. Therefore, it is critical to investigate the importance of hyper-parameters: word-level KD loss coefficients, the dropout rate to extract sub-models, and the numbers of generated data. Beyond that, the most critical factor is how to extract the architecture and size of the sub-models. We can obtain different architecture sub-models using the abovementioned sub-model extraction methods, and we can get sub-models of various sizes by dynamically setting the dropout rate.

a) Effect of Word-level KD Loss Coefficients: It is crucial to balance NMT cross-entropy loss and KL loss. The KL loss function can constantly regularize the relationship between the student prediction probability distribution and the teacher distribution. We test α from 0.5 to 1.5 according to Eq. 6 and we show the results in Table IV, α = 1.0 is the best choice.

b) Effect of Sequence-Level Data Size: Since our distillation approach is performed on corpus combined with synthetic data and generated data, one natural and interesting study is to reveal the effect of the different numbers of data. In this subsection, we make an attempt to enumerate the number of data in the teacher model from 0 to 10. That is, for each size n, we randomly set a seed for our teacher to

TABLE V
RESULTS OF DIFFERENT TEACHER DATA GENERATION METHODS ON IWSLT14 DE→EN TRANSLATION TASK.

Method	De→En
Transformer [3]	34.74
Dropout (SD)	36.48
LayerDrop + (SD)	36.13
Sub-layer Reordering + (SD)	36.09
$ST_{RLD}S_V$ (SD)	**36.70**
Attention Dropout (SD)	36.32
Activation Dropout (SD)	36.19
$ST_{RLD+}S_V$ (SD)	**36.89**

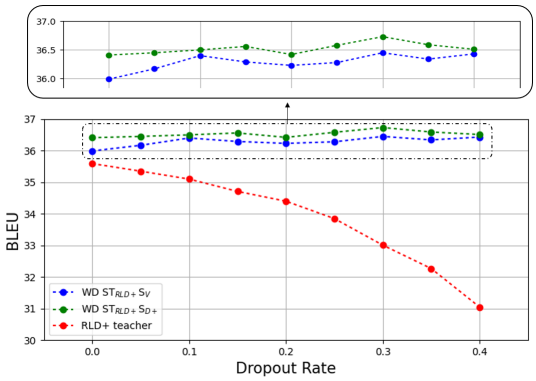


Fig. 4. BLUE score of our method trained with different dropout rate.

TABLE VI
RESULTS OF DIFFERENT TEACHER DATA GENERATION METHODS ON IWSLT14 DE→EN TRANSLATION TASK.

Method	BDM	BLEU
SD + [Beam Search]	0.3180	35.89
SD + [sampling]	**0.6527**	35.41
MTDD	0.4127	**36.81**
$ST_{RLD}S_V$ (SD)	0.3181	**36.70**

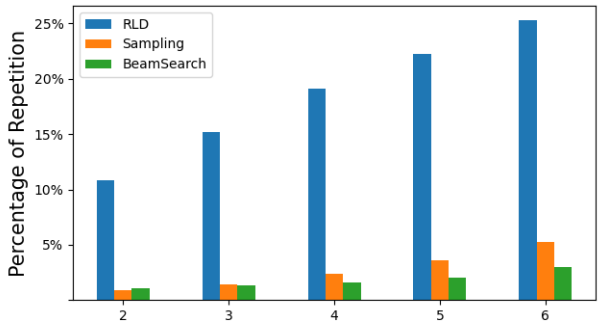


Fig. 5. The number of Percentage of Repetition Data on IWSLT14 De→En Training Set.

generate data and perform the student model training from the diverse data. This experiment is repeated 3 times for each data size, and we report the averaged results in Fig. 3. We can see that the different set of data does affect the data scale and data diversity, and the student model has better performance with more data. This is intuitive that more data can better amplify the knowledge from teacher model and thus better guide students [29]. However, it does not bring more significant performance improvements when too much data is provided. Therefore, in order to trade off the training time and performance, we chose 7 as the hyper-parameter for all tasks.

c) Effectiveness of Sub-Model Architecture: In order to verify the effect of different components, we use different components to conduct knowledge distillation. It can be seen that the effect of Dropout is significant in Table V, and sub-layer reordering along with LayerDrop can also improve the data distillation. This is reasonable that the more subspace can be chosen, the greater performance can be obtained. That is, the number of sub-networks that can be extracted at the width is much greater than the depth. It also demonstrates that the diversity representation ability of teacher can be improved by increasing the number of sub-models.

d) Effectiveness of Sub-Model Size: We analyze the experimental results of two settings $ST_{RLD+}S_V$ (WD) and $ST_{RLD+}S_{D+}$ (WD) with different dropout rates in Fig. 4, in which teachers can maintain model performance within an extensive range of dropout rates. According to the curves, teacher performance decreases with the dropout rate increases linearly, which means that the distance of distributions of sub-models and the diversity of the translation sequences are far away from each other. However, student performance raises as the diversity of the teacher's output increases. The best result for student model is with the dropout = 0.3.

B. Data Diversity & Quality

In this section, we will individually explore the relationship between sub-model distribution diversity and KD performance. In order to facilitate the exploration of specific examples corresponding to the model distribution, we conduct the experiment with the data distillation method. We first study the different ways to generate data and analyse the diversity and quality. We use beam search with beam size 6 to obtain the distilled data and keep all the 6 translations (SD [Beam Search]). Besides, we also investigate the random sampling strategy to obtain the data. Therefore, we also sample 6 translations for each source sentence and use them for student model training (SD [Sampling]) [46]. Results are shown in Table VI. From the Table VI, we can observe that (1) Though the data scale is larger (SD [Beam Search]), the translation performance is slightly decreased; (2) Though the data scale and data diversity are both increased compared with SD [Sampling], the student model performance is significantly reduced. Therefore, we turn to explore one question: *What kind of data does the model need in data distillation and how to generate it?*

It is commonly acknowledged that data diversity is beneficial for the NMT task [47], [48], [49]. Data diversity refers to translation diversity that the source sentence can translate to multiple targets in machine translation. Here we introduce the BLEU-based Discrepancy Metric (BDM) [50] to compute the

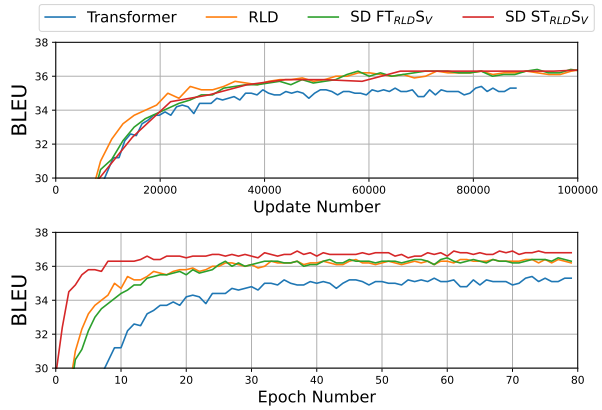


Fig. 6. Convergence curves of different model on IWSLT14 De→En development set.

diversity score:

$$\mathrm{BDM}(Y) = \frac{1}{|Y|(|Y|-1)} \sum_{y \in Y} \sum_{\hat{y} \in Y, \hat{y} \neq y} 1 - \Delta(y, \hat{y}), \quad (7)$$

where $\Delta(y, \hat{y})$ is the BLEU score of two candidates, and Y is a list of candidates translations. The higher BDM score represents more diverse. We can see the SD [Sampling] gets a higher BDM score than other methods, which proves that random sampling can produce translations with better diversity. SD [Beam Search] has neither good diversity nor improves the performance of distillation, indicating that there are not enough high-quality candidates to improve performance in narrow beam search space. Furthermore, we count the percentage of repetition data for the generated data in Fig. 5, The repetition data generated by the RLD method increases linearly with the number of sub-models, demonstrating that the sub-models search space can make consistent decisions instead of only considering the diversity factor.

From the above experiments, we try to explain the phenomena and the reasons behind them from two views, i.e., diversity and quality. The model tends to spread too much probability mass over the hypothesis space, leading good candidates distributed sparsely in huge space [30]. For beam search, the top-ranking candidates are in high quality but not diverse enough. Instead, sampling-based distillation usually generates diverse data but with lower quality. Our method alleviates these problems to generate the data with both diversity and high quality. We break the limitation for searching from the entire model hypothesis space and turn to search from sub-networks. With the RLD training paradigm, different sub-networks can construct different hypothesis spaces in which the top-ranking candidates have high quality and enough distinction.

C. Training Analysis & Visualization

To better show the advantage of our method, we study the model training from different aspects. As aforementioned, our method does not cause much extra training cost, so we first give detailed comparisons on the training time of the RLD teacher and the vanilla Transformer model (baseline). The training curves are in Fig. 6. Specifically, the RLD was slightly lower than the baseline at the early training stage due to the strong regularization effect by RLD. But there was no significant difference when the model is almost converged after 40 epochs. So the RLD approach does not add extra computational cost and without any performance loss.

Furthermore, we also study the training process and visualize the performance improvements of the student model. In Fig. 6, we plot the curves of training and valid BLEU score based on different timelines (Update number and Epoch number) since FT$_{RLD}$S$_V$ has 2 times data and ST$_{RLD}$S$_V$ has 7 times data than Baseline. From the update number, ST$_{RLD}$S$_V$ gets similar results compared with other models under the same update steps. From the epoch number, we can see that convergence of our method is faster than other methods with better performance.

TABLE VII
RESULTS OF MODEL ENSEMBLE WITH DIFFERENT MODEL NUMBER, MULTI-TEACHER DATA DISTILLATION (MTDD) WITH DIFFERENT TEACHER NUMBER, AND OUR METHOD WITH DIFFERENT DATA SIZE ON IWSLT14 DE→EN TRANSLATION.

Model	1	2	3	4	5	6
Ensemble	34.74	36.06	36.37	36.59	36.80	36.92
MTDD	35.91	36.24	36.49	36.65	36.77	36.81
ST$_{RLD}$S$_V$ (SD)	35.85	36.29	36.54	36.60	36.68	36.70

TABLE VIII
RESULTS OF OUR MODEL AND MTDD ON IWSLT14 DE→EN TRANSLATION.

Model	P	R	F1	BLEU
Vanilla Transformer	0.9382	0.9403	0.9390	34.72
MTDD	0.9400	0.9419	0.9407	36.81
WD ST$_{RLD+}$S$_{D+}$	0.9405	0.9425	0.9412	36.73
SD ST$_{RLD+}$S$_{D+}$	0.9406	0.9429	0.9415	37.20

D. Ensemble & Multiple Teachers

To further show the power of our methods, we compare our method with ensemble method and multi-teacher data distillation (MTDD). Specifically, we vary the models (parameters with random seeds) from 1 to 6 and perform ensemble and data distillation. The results are shown in Table VII. We can see that with more diverse data generated from multiple teacher models, the student model performances increase gradually. From the perspective of the diversity of generated samples, MTDD exactly generates more diverse data (MTDD (0.4127) v.s. ST$_{RLD}$S$_V$ (0.3181)) according to Table VI. It is worth noting that the model ensemble method requires 6 times as much parameter sizes and inference time, and MTDD requires $6+1$ (teacher and student) times parameters for training. Our method can achieve a similar performance compared to 4 teacher models, but we only need $1+1$ times parameters for training without additional inference cost.

To comprehensively evaluate our method, we introduce the BERTScore [51], which leverages the pre-trained contextual embeddings from BERT and matches words in candidate and reference sentences by cosine similarity. It has been shown to

better correlate with human judgment on sentence-level and system-level evaluation. The results are shown in Table VIII, our method $\text{ST}_{RLD+}\text{S}_{D+}$ (WD) achieves the better F1 score than MTDD (0.9412 v.s. 0.9407) though BLEU score is lower than it, and $\text{ST}_{RLD+}\text{S}_{D+}$ (SD) gets best F1 score 0.9415 and best BLEU score 37.20.

VI. Conclusion

In this paper, we propose a simple yet effective knowledge distillation method with one single teacher model. Through introducing three types of randomness in teacher model training and inference, our method can obtain model variants of one teacher from the sub-network space by introducing dropout & sub-layer dropout and as the permutation of model parameters by launching sub-layer reordering. Experimental results on 9 translation tasks well prove the superiority of our method.

In the near future, we intend to explore more simple and effective strategies to obtain better model variants and conduct efficient and fine-grained knowledge distillation.

Acknowledgment

The authors would like to thank the efforts and suggestions of editors and anonymous reviewers in improving this paper.

References

[1] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in Neural Information Processing Systems*, vol. 27, pp. 3104–3112, 2014.

[2] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *International Conference on Learning Representations*, 2015.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[4] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1243–1252.

[5] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[6] Y. Kim and A. M. Rush, "Sequence-level knowledge distillation," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 1317–1327.

[7] H. Sun, R. Wang, K. Chen, M. Utiyama, E. Sumita, and T. Zhao, "Knowledge distillation for multilingual unsupervised neural machine translation," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 3525–3535.

[8] F. Saleh, W. Buntine, and G. Haffari, "Collective wisdom: Improving low-resource neural machine translation using adaptive knowledge distillation," in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 3413–3421.

[9] X. Tan, Y. Ren, D. He, T. Qin, Z. Zhao, and T.-Y. Liu, "Multilingual neural machine translation with knowledge distillation," *International Conference on Learning Representations*, 2019.

[10] H.-R. Wei, S. Huang, R. Wang, X. Dai, and J. Chen, "Online distilling from checkpoints 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)*, 2019, pp. 1932–1941.

[11] A. Lin, J. Wohlwend, H. Chen, and T. Lei, "Autoregressive knowledge distillation through imitation learning," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 6121–6133.

[12] X.-P. Nguyen, J. Shafiq, K. Wu, and A. T. Aw, "Data diversification: A simple strategy for neural machine translation," in *Advances in Neural Information Processing Systems*, 2020.

[13] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural network," in *International Conference on Machine Learning*. PMLR, 2015, pp. 1613–1622.

[14] J. Zhu, W. Lijun, X. Yingce, X. Shufang, T. Qin, W. Zhou, H. Li, and T.-Y. Liu, "Iot: Instance-wise layer reordering for transformer structures," in *International Conference on Learning Representations*, 2021.

[15] A. Fan, E. Grave, and A. Joulin, "Reducing transformer depth on demand with structured dropout," *arXiv preprint arXiv:1909.11556*, 2019.

[16] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[17] C. Buciluǎ, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2006, pp. 535–541.

[18] L. J. Ba and R. Caruana, "Do deep nets really need to be deep?" in *Proceedings of the 27th International Conference on Neural Information Processing Systems-Volume 2*, 2014, pp. 2654–2662.

[19] A. A. Rusu, S. G. Colmenarejo, C. Gulcehre, G. Desjardins, J. Kirkpatrick, R. Pascanu, V. Mnih, K. Kavukcuoglu, and R. Hadsell, "Policy distillation," *International Conference on Learning Representations*, 2016.

[20] M.-C. Wu, C.-T. Chiu, and K.-H. Wu, "Multi-teacher knowledge distillation for compressed video action recognition on deep neural networks," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 2202–2206.

[21] S. Arora, M. M. Khapra, and H. G. Ramaswamy, "On knowledge distillation from complex networks for response prediction," 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)*, 2019, pp. 3813–3822.

[22] Z. Yang, L. Shou, M. Gong, W. Lin, and D. Jiang, "Model compression with two-stage multi-teacher knowledge distillation for web question answering system," in *Proceedings of the 13th International Conference on Web Search and Data Mining*, 2020, pp. 690–698.

[23] A. Adhikari, A. Ram, R. Tang, W. L. Hamilton, and J. Lin, "Exploring the limits of simple learners in knowledge distillation for document classification with docbert," in *Proceedings of the 5th Workshop on Representation Learning for NLP*, 2020, pp. 72–77.

[24] X. Wang, Y. Jiang, N. Bach, T. Wang, F. Huang, and K. Tu, "Structure-level knowledge distillation for multilingual sequence labeling," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 3317–3330.

[25] S. Hahn and H. Choi, "Self-knowledge distillation in natural language processing," in *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, 2019, pp. 423–430.

[26] C. Zhou, J. Gu, and G. Neubig, "Understanding knowledge distillation in non-autoregressive machine translation," in *International Conference on Learning Representations*, 2019.

[27] Y. Wu, P. Passban, M. Rezagholizadeh, and Q. Liu, "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)*, 2020, pp. 1016–1021.

[28] M. Gordon and K. Duh, "Distill, adapt, distill: Training small, in-domain models for neural machine translation," in *Proceedings of the Fourth Workshop on Neural Generation and Translation*, 2020, pp. 110–118.

[29] M. Freitag, Y. Al-Onaizan, and B. Sankaran, "Ensemble distillation for neural machine translation," *arXiv preprint arXiv:1702.01802*, 2017.

[30] M. Ott, M. Auli, D. Grangier, and M. Ranzato, "Analyzing uncertainty in neural machine translation," in *International Conference on Machine Learning*. PMLR, 2018, pp. 3956–3965.

[31] J. Frankle and M. Carbin, "The lottery ticket hypothesis: Finding sparse, trainable neural networks," *arXiv preprint arXiv:1803.03635*, 2018.

[32] O. Press, N. A. Smith, and O. Levy, "Improving transformer models by reordering their sublayers," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 2996–3005.

[33] Z. Wu, L. Wu, Q. Meng, Y. Xia, S. Xie, T. Qin, X. Dai, and T.-Y. Liu, "Unidrop: A simple yet effective technique to improve transformer without extra cost," *arXiv preprint arXiv:2104.04946*, 2021.

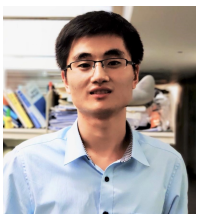
[34] M. Cettolo, J. Niehues, S. Stüker, L. Bentivogli, and M. Federico, "Report on the 11th iwslt evaluation campaign, iwslt 2014," in *Proceedings of the International Workshop on Spoken Language Translation, Hanoi, Vietnam*, vol. 57, 2014.

[35] M. Cettolo, M. Federico, L. Bentivogli, N. Jan, S. Sebastian, S. Katsuitho, Y. Koichiro, and F. Christian, "Overview of the iwslt 2017 evaluation campaign," in *International Workshop on Spoken Language Translation*, 2017, pp. 2–14.

[36] R. Sennrich, B. Haddow, and A. Birch, "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), 2016, pp. 1715–1725.
[37] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[38] F. Wu, A. Fan, A. Baevski, Y. Dauphin, and M. Auli, "Pay less attention with lightweight and dynamic convolutions," in *International Conference on Learning Representations*, 2019.
[39] Y. Lu, Z. Li, D. He, Z. Sun, B. Dong, T. Qin, L. Wang, and T.-Y. Liu, "Understanding and improving transformer from a multi-particle dynamic system point of view," *arXiv preprint arXiv:1906.02762*, 2019.
[40] Y. Wang, Y. Xia, T. He, F. Tian, T. Qin, C. X. Zhai, and T. Y. Liu, "Multi-agent dual learning," in *7th International Conference on Learning Representations, ICLR 2019*, 2019.
[41] H. Pham and Q. V. Le, "Autodropout: Learning dropout patterns to regularize deep networks," *arXiv preprint arXiv:2101.01761*, 2021.
[42] Y. Fan, S. Xie, Y. Xia, L. Wu, T. Qin, X.-Y. Li, and T.-Y. Liu, "Multi-branch attentive transformer," *arXiv preprint arXiv:2006.10270*, 2020.
[43] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 2018, pp. 464–468.
[44] M. Ott, S. Edunov, D. Grangier, and M. Auli, "Scaling neural machine translation," in *Proceedings of the Third Conference on Machine Translation: Research Papers*, 2018, pp. 1–9.
[45] D. So, Q. Le, and C. Liang, "The evolved transformer," in *International Conference on Machine Learning*. PMLR, 2019, pp. 5877–5886.
[46] S. Edunov, M. Ott, M. Auli, and D. Grangier, "Understanding back-translation at scale," *arXiv preprint arXiv:1808.09381*, 2018.
[47] J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan, "A diversity-promoting objective function for neural conversation models," *arXiv preprint arXiv:1510.03055*, 2015.
[48] D. Alihosseini, E. Montahaei, and M. S. Baghshah, "Jointly measuring diversity and quality in text generation models," in *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, 2019, pp. 90–98.
[49] M. Ghazvininejad, O. Levy, Y. Liu, and L. Zettlemoyer, "Mask-predict: Parallel decoding of conditional masked language models," *arXiv preprint arXiv:1904.09324*, 2019.
[50] G. Tevet and J. Berant, "Evaluating the evaluation of diversity in natural language generation," *arXiv preprint arXiv:2004.02990*, 2020.
[51] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," *arXiv preprint arXiv:1904.09675*, 2019.

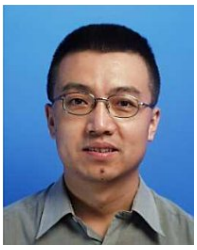
Xiaobo Liang is now a Ph.d student of Soochow University, supervised by Prof. Min Zhang. Before that, he obtained an MS degree from Northeastern University, China, and used to be a research intern at Westlake University, supervised by Prof. Yue Zhang. His current interests lie in neural machine translation and consistency learning. He also has published papers about name entity recognition and relation extraction on ACL and COLING. He served as the reviewer of ACL.



Lijun Wu is a Researcher of Machine Learning Group in Microsoft Research Asia (MSRA). He got the Ph.D. degree from Sun Yat-sen University (SYSU) in 2020, a member of joint Ph.D. program between SYSU and MSRA. He received MSRA Ph.D. Fellowship in 2018. His researches focus on Deep Learning, Natural Language Processing, Multimodality Learning, Medical Health. He has published more than 20 papers in top-tier conferences, e.g., ICML, ICLR, NeurIPS, ACL, EMNLP, AAAI, IJCAI and so on. He served as reviewers of these conferences and SPC/AC for ACL-21/IJCAI-21/AAAI-22.

Juntao Li is now an associate professor at Institute of Artificial Intelligence, Soochow University. Before that, he obtained the doctoral degree from Peking University in 2020. He is now working on text generation, dialogue systems, and artistic writing (e.g., poetry generation, story generation). He has published over 10 papers as leading authors on TOIS, ACL, EMNLP, AAAI, IJCAI and had given two tutorials on IJCAI and AAAI. He also serves as the reviewer of Computational Linguistics, TALLIP, TKDE, International Journal of Intelligent Systems, ACL(Area Chair), EMNLP, NAACL, AAAI, IJCAI(Senior PC), etc.

Tao Qin received the Ph.D. degree and Bachelor degree both from Tsinghua University. He is a Senior Member of ACM and IEEE, and an Adjunct Professor (Ph.D. advisor) in the University of Science and Technology of China. He is a Senior Principal Research Manager in Machine Learning Group, Microsoft Research Asia. His research interests include machine learning (with the focus on deep learning and reinforcement learning), artificial intelligence (with applications to language understanding and computer vision), game theory and multi-agent systems (with applications to cloud computing, online and mobile advertising, e-commerce), information retrieval and computational advertising.



Min Zhang is a Distinguished Professor at Soochow University (China). He received his bachelor's degree and Ph.D. degree from the Harbin Institute of Technology in 1991 and 1997, respectively. His current research interests include machine translation, natural language processing, information extraction, large-scale text processing, intelligent computing, and machine learning. He has authored 150 papers in leading journals and conferences and has co-edited 10 books that were published by Springer and IEEE. He has been actively contributing to the research community by organizing many conferences as a chair, program chair and organizing chair and by giving talks at many conferences and lectures.

Tie-Yan Liu received the Ph.D. degree and Bachelor degree both from Tsinghua University. He is an Assistant managing director of Microsoft Research Asia, leading the machine learning research area. His seminal contribution to the field of learning to rank and computational advertising has been widely recognized, and his recent research interests include deep learning, reinforcement learning, and distributed machine learning. In particular, he and his team have proposed a few new machine learning concepts, such as dual learning, learning to teach, and deliberation learning. He is an adjunct/honorary professor at Carnegie Mellon University (CMU), the University of Nottingham, and several other universities in China. He has published 200+ papers in refereed conferences and journals, e.g., SIGIR, WWW, ICML, KDD, NeurIPS, IJCAI, AAAI, ACL, and so on, with 30000+ citations. He is a fellow of IEEE, and a distinguished member of ACM.