Multi-Task Learning with Shared Encoder for Non-Autoregressive Machine Translation

Yongchang Hao[†*]   **Shilin He**[‡*]   **Wenxiang Jiao**[‡]

Zhaopeng Tu[††]   **Michael R. Lyu**[‡]   **Xing Wang**[††]

[†]School of Computer Science and Technology, Soochow University
[‡]Department of Computer Science and Engineering, The Chinese University of Hong Kong
[††]Tencent AI Lab
[†]`yongchanghao.w@gmai.com`
[‡]`{slhe,wxjiao,lyu}@cse.cuhk.edu.hk`
[††]`{zptu,brightxwang}@tencent.com`

Abstract

Non-Autoregressive machine Translation (NAT) models have demonstrated significant inference speedup but suffer from inferior translation accuracy. The common practice to tackle the problem is transferring the Autoregressive machine Translation (AT) knowledge to NAT models, e.g., with knowledge distillation. In this work, we hypothesize and empirically verify that AT and NAT encoders capture different linguistic properties of source sentences. Therefore, we propose to adopt multi-task learning to transfer the AT knowledge to NAT models through encoder sharing. Specifically, we take the AT model as an auxiliary task to enhance NAT model performance. Experimental results on WMT14 English⇔German and WMT16 English⇔Romanian datasets show that the proposed MULTI-TASK NAT achieves significant improvements over the baseline NAT models. Furthermore, the performance on large-scale WMT19 and WMT20 English⇔German datasets confirm the consistency of our proposed method. In addition, experimental results demonstrate that our MULTI-TASK NAT is complementary to knowledge distillation, the standard knowledge transfer method for NAT. [1]

1 Introduction

Neural machine translation (NMT), as the state-of-the-art machine translation paradigm, has recently been approached with two different sequence decoding strategies. The first type *autoregressive translation* (AT) models generate output tokens one by one following the left to right direction (Vaswani et al., 2017; Bahdanau et al., 2015), but it is often criticized for its slow inference speed (Gu et al., 2018). The second type *non-autoregressive translation* (NAT) models adopt a parallel decoding algorithm to produce output tokens simultaneously (Gu et al., 2019; Ghazvininejad et al., 2019; Ma et al., 2020), but the translation quality of it is often inferior to auto-regressive models (Gu et al., 2018).

Many researchers have investigated the collaboration between AT and NAT models. For instance, ENCODER-NAD-AD (Zhou et al., 2020) leverages NAT models to improve the performance of AT. Specifically, their method inserts a NAT decoder between the conventional AT encoder and decoder to generate coarse target sequences for the final autoregressive decoding. A line of research (Wang et al., 2019b; Guo et al., 2020; Ding et al., 2020) holds the opinion that the lack of contextual dependency on target sentences potentially leads to the deteriorated performance of NAT models. To boost the NAT translation performance, many recent works resort to the knowledge transfer from a well-trained AT model. Typical knowledge transfer methods include sequence-level knowledge distillation with translation outputs generated by strong AT models (Gu et al., 2019; Ghazvininejad et al., 2019), word-level knowledge distillation with AT decoder representations (Wei et al., 2019; Li et al., 2019), and fine-tuning on AT model by curriculum learning (Guo et al., 2020), etc.

In this work, we first verify our our hypothesis that AT and NAT encoders – although they belong to the same sequence-to-sequence learning task – capture different linguistic properties of source sentences. We conduct our verification by evaluating the encoder on a set of probing tasks (Conneau

---

[1] Code is publicly available at `https://github.com/yongchanghao/multi-task-nat`

Task		AT	NAT
Surface	SeLen	91.7	**93.4**
	WC	76.0	**79.1**
Syntactic	TrDep	45.8	46.0
	ToCo	78.3	**79.7**
	BShif	**74.8**	73.4
Semantic	Tense	89.2	89.2
	SubN	86.2	**87.5**
	ObjN	85.2	85.3
	SoMo	**54.0**	53.0
	CoIn	**64.9**	62.8

Table 1: Performance on the probing tasks of evaluating linguistic properties embedded in the learned representations of AT and NAT models.

et al., 2018; Raganato and Tiedemann, 2018) for AT and NAT models. Further, by leveraging the linguistic differences, we then adopt a multi-task learning framework with a shared encoder (i.e., MULTI-TASK NAT) to transfer the AT model knowledge into the NAT model. Specifically, we employ an additional AT task as the auxiliary task of which the encoder parameters are shared with the NAT task while parameters of the decoder are exclusive. Since many works (Cipolla et al., 2018; Liu et al., 2019) suggest that the weights for each task are critical to the multi-task learning, in this work, the multi-task weight assigned to the AT task is dynamically annealed from 1 to 0. We name this scheme *importance annealing*. We empirically show the benefit of *importance annealing* in both directions of the original WMT14 English⇔German dataset.

Further with knowledge distillation, our proposed MULTI-TASK NAT achieves significant improvements on WMT14 English⇔German and WMT16 English⇔Romanian datasets. This confirms the effectiveness of our proposed model on machine translation tasks.

Our contributions are as follows:

- We propose a multi-task learning framework to boost NAT translation quality by transferring the AT knowledge to the NAT model.
- Our analyses reveal that the encoder sharing is necessary for capturing more linguistic and semantic information.
- Experiments on standard benchmark datasets demonstrate the effectiveness of the proposed MULTI-TASK NAT.

2 Why Shared Encoder?

To verify our hypothesis that AT and NAT encoders capture different linguistic properties of source sentences and can thereby complement each other, we probe the linguistic knowledge (Conneau et al., 2018) that embedded in the AT and NAT encoders on a set of tasks to investigate to what extent an encoder captures the linguistic properties. We present the detail for each probing tasks in Appendix B. Moreover, in Appendix C, we also provide a qualitative investigation to capture the difference between high-dimensional representations of AT and NAT encoders from another perspective.

The AT and NAT models referred to in the following experiments are TRANSFORMER and MASK-PREDICT. We train the models on the WMT14 English⇒German dataset, and the details of the experiments are introduced in the Appendix.

Probing Tasks Probing tasks (Conneau et al., 2018) can quantitatively measure the linguistic knowledge embedded in the model representation. We follow Wang et al. (2019a) to set model configurations. The experimental results are depicted in Table 1.

Table 1 shows the AT and NAT encoders capture different linguistic properties of source sentences. We observe that on average, the NAT model captures more surface features but less semantic features than the AT model. For example, on the sentence length prediction (SeLen) task, NAT models significantly outperform AT models since the sentence length prediction is a key component in NAT models. However, for sentence modification (SoMo) and coordinate clauses inversion (CoIn) tasks, the AT model outperforms the NAT model by a large margin. The linguistic probing results reveal that AT and NAT models capture different linguistic properties, which thereby leaves space for the encoder sharing structure.

3 Approach

In this section, we introduce that our shared encoder structure between AT and NAT models under the multi-task learning framework.

Multi-Task NAT Given the AT and NAT models under the standard encoder-decoder structure, we employ the hard parameter sharing method (Ruder, 2017) to share their encoder parameters.

Therefore, as shown in Figure 1, the proposed model MULTI-TASK NAT consists of three com-
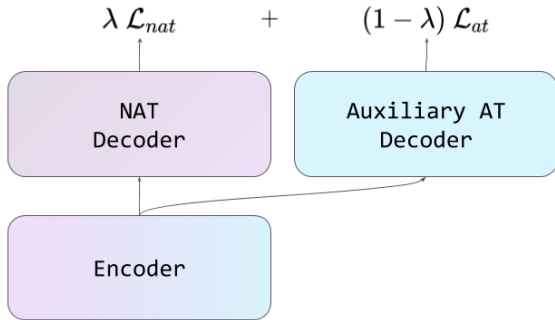
Figure 1: The architecture of our proposed model. We introduce an extra AT decoder to boost the training at the beginning, and gradually lower the importance weight of AT task by increasing λ.

ponents: shared encoder, AT decoder, and NAT decoder. Their parameters are jointly optimized towards minimizing the multi-task loss function, as introduced in the next section.

Multi-Task Framework The loss function of the proposed MULTI-TASK NAT $\mathcal{L}$ at iteration step t is defined as the weighted sum of AT loss and NAT loss:

$$\begin{aligned} \mathcal{L} = &\lambda_t \mathcal{L}_{\text{nat}}\left(X, Y; \theta_{\text{enc}}, \theta_{\text{dec}}^{\text{nat}}\right) \\ &+ (1 - \lambda_t)\mathcal{L}_{\text{at}}\left(X, Y; \theta_{\text{enc}}, \theta_{\text{dec}}^{\text{at}}\right) \end{aligned} \quad (1)$$

where $\mathcal{L}_{\text{at}}$ and $\mathcal{L}_{\text{nat}}$ are AT loss and NAT loss. $\theta_{\text{enc}}, \theta_{\text{dec}}^{\text{nat}}$, and $\theta_{\text{dec}}^{\text{at}}$ are parameters of the shared encoder, NAT decoder, and AT decoder respectively. λ_t is the importance factor to balance the preference between the AT and NAT models at time step t as illustrated bellow.

Importance Annealing The term $\mathcal{L}_{\text{at}}$ only serves as an auxiliary and does not directly affect the inference of NAT. Therefore, we intuitively determine to lower the importance of the AT loss when the training process is close to the ending, which we named *importance annealing*. Formally, we set

$$\lambda_t = \frac{t}{T}$$

where T is the total steps of training. Under such a scheme, the weight for $\mathcal{L}_{\text{at}}$ is linearly annealed from 1.0 to 0.0 along the training process, while the weight for $\mathcal{L}_{\text{nat}}$ is increased from 0.0 to 1.0.

Training and Inference During the model training with training pairs (X, Y), we feed the source sentence X to the encoder and the target sentence Y to two decoders separately. The target sentence

Model	WMT14	
	En⇒De	De⇒En
MASK-PREDICT[1]	24.61	–
TRANSFORMER-LEV[2]	25.20	–
MASK-PREDICT[3]	24.70	29.52
MULTI-TASK NAT	25.66	30.09
+ *Importance Annealing*	25.79	30.32

[1] Ghazvininejad et al. (2019); [2] Gu et al. (2019); [3] Our implementation.

Table 2: Evaluation of translation performance on WMT14 En⇒De and WMT14 De⇒En test sets without knowledge distillation.

Y can be either the target sentence in the raw training data (4.1) or the generated target sentence with knowledge distillation (4.2). During the model inference, we only use the NAT decoder to generate the target tokens simultaneously while ignoring the AT decoder. Therefore, the inference overhead is the same as the NAT model before sharing.

4 Experiment

We conducted experiments on two widely used WMT14 English⇔German and WMT16 English⇔Romanian benchmark datasets, which consist of 4.5M and 610K sentence pairs, respectively. We applied BPE (Sennrich et al., 2016) with 32K merge operations for both language pairs. The experimental results are evaluated in case-sensitive BLEU score (Papineni et al., 2002).

We use TRANSFORMER (Vaswani et al., 2017) as our baseline autoregressive translation model and the MASK-PREDICT (Ghazvininejad et al., 2019) as our baseline non-autoregressive model. We integrate the TRANSFORMER decoder into the MASK-PREDICT to implement the proposed MULTI-TASK NAT model. For λ_t, we use the annealing scheme described in Section 3. Since the major NAT architecture of our method is exactly the MASK-PREDICT model, any established decoding latency results (Kasai et al., 2021) for MASK-PREDICT can also be applied to ours. All of the parameters are randomly initialized for a fair comparison with the MASK-PREDICT. More training details are introduced in Appendix A.

4.1 Ablation Study

Table 2 shows that the performance of our MULTI-TASK NAT model and baseline models on WMT14

Model	WMT14		WMT16	
	En⇒De	De⇒En	En⇒Ro	Ro⇒En
Baseline Models				
Transformer (Ghazvininejad et al., 2019)	27.74	31.09	34.28	33.99
Hint-based NAT (Li et al., 2019)	25.20	29.52	–	–
NAT-REG (Wang et al., 2019b)	24.61	28.90	–	–
FCL-NAT (Guo et al., 2020)	25.75	29.50	–	–
Levenshtein Transformer (Gu et al., 2019)	27.27	–	–	33.26
Mask-Predict (Ghazvininejad et al., 2019)	27.03	30.53	33.08	33.31
Mask-Predict w/ Raw Data Prior (Ding et al., 2021)	27.8	–	–	33.7
Our Experiments				
Mask-Predict	27.18	30.86	33.03	32.71
Multi-Task NAT (w/ IA)	**27.98**$^\Uparrow$	**31.27**$^\uparrow$	**33.80**$^\uparrow$	**33.60**$^\Uparrow$

Table 3: Evaluation of translation performance on WMT14 En⇔De and WMT16 En⇔Ro test sets. All NAT models are trained with AT knowledge distillation. "$\uparrow$ / $\Uparrow$": indicate a statistically significant improvement over the corresponding baseline $p < 0.05/0.01$ respectively. Some baselines do not perform well because they are not built upon the MASK-PREDICT.

En⇔De datasets without using the knowledge distillation. The vanilla MULTI-TASK NAT model with the the λ fixed as 0.5 outperforms the baseline MASK-PREDICT model by 0.96 and 0.57 BLEU score in En⇒De and De⇒En direction respectively and even surpasses the strong baseline TRANSFORMER-LEV by 0.46 BLEU points in En⇒De translation. With the *importance annealing*, the MULTI-TASK NAT model achieves slight but consistent improvements over the vanilla model ("+*Importance Annealing*" in Table 2). The improvements demonstrate the effectiveness of our proposed model using multi-task learning.

4.2 Main Result

We further evaluate the proposed MULTI-TASK NAT model with the standard practice of knowledge distillation. Table 3 depicts the performances of our model as well as strong baseline models. Our proposed MULTI-TASK NAT model achieves a significant improvement of 0.80 and 0.41 BLEU point over the strong baseline MASK-PREDICT model on En⇒De and De⇒En translation. On En⇔Ro translation, our model outperforms the baseline model by 0.77 and 0.89 BLEU scores respectively. We use the `compare-mt` (Neubig et al., 2019)[2] to determine the significance. Details for significance tests are described in Appendix A.4.

Task		OURS	Δ_{AT}	Δ_{NAT}
Surface	SeLen	94.4	2.7	1.0
	WC	79.3	3.3	0.2
Syntactic	TrDep	47.2	1.4	1.2
	ToCo	79.3	1.0	-0.4
	BShif	74.7	-0.1	1.3
Semantic	Tense	88.9	-0.3	-0.3
	SubN	87.1	0.9	-0.4
	ObjN	85.8	0.6	0.5
	SoMo	54.8	0.8	1.8
	CoIn	63.3	-1.6	0.5

Table 4: Performance on the probing tasks of our MULTI-TASK NAT model. Δ_{AT} and Δ_{NAT} denote the relative increase over the AT and NAT probing performance, respectively.

4.3 Analysis

We conduct probing tasks to empirically reconfirm our hypothesis in Section 2 and better understand our MULTI-TASK NAT in terms of linguistic properties. The results are presented in Table 4. In most of the cases, our MULTI-TASK NAT could learn better surface, syntactic, and semantic information than the TRANSFORMER and MASK-PREDICT baseline models, indicating that our multi-task learning framework can indeed take the advantages of two separate tasks and capture better linguistic properties. Notably, on the sen-

Model	WMT19		WMT20	
	En⇒De	De⇒En	En⇒De	De⇒En
MASK-PREDICT	34.79	37.04	25.24	36.36
MULTI-TASK NAT	**35.38**$^\uparrow$	**37.62**$^\uparrow$	**25.72**$^\uparrow$	**36.58**

Table 5: Evaluation of translation performance on WMT19 En⇔De and WMT20 En⇔De test sets with knowledge distillation. The *Importance Annealing* is adopted by default.

tence length (Selen) prediction task and tree depth (TrDep) task, the MULTI-TASK NAT shows significantly better performance. On other tasks, our model demonstrates better or on-par performance compared to the NAT model. Regarding the coordination inversion (CoIn) task, though the MULTI-TASK NAT shows certainly lower performance than the TRANSFORMER, it still outperforms the MASK-PREDICT by 0.5.

4.4 Large-scale Experiments

We conduct the larger-scale experiments on the WMT English⇔German. We adopt newstest2019 and newstest2020 as the test sets. The parallel data consists of about 36.8M sentence pairs. We average the last 5 checkpoints as the final model. The results are listed in Table 5. The improvements suggest that our model are consistently effective on various scale of data.

5 Conclusion and Future Work

In this paper, we have presented a novel multi-task learning approach for NAT model with a hard parameter sharing mechanism. Experimental results confirm the significant effect of the proposed MULTI-TASK NAT model, which shows the complementary effects of multi-task learning to the knowledge distillation method.

Based on our MULTI-TASK NAT, there are many promising directions for future research. For example, 1) decoder interaction: knowledge distillation in an online fashion between AT and NAT decoders; 2) share-all framework: shared-encoder and shared-decoder with two decoding strategies, and the model can dynamically choose the optimal decoding strategy during model inference. 3) data manipulation strategies: such as data rejuvenation (Jiao et al., 2020), lexical frequency discrepancy (Ding et al., 2021).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

R. Cipolla, Y. Gal, and A. Kendall. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*.

Alexis Conneau and Douwe Kiela. 2018. Senteval: An evaluation toolkit for universal sentence representations. *arXiv preprint arXiv:1803.05449*.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loic Barrault, and Marco Baroni. 2018. What you can cram into a single vector: Probing sentence embeddings for linguistic properties. In *ACL*.

Liang Ding, Longyue Wang, Xuebo Liu, Derek F. Wong, Dacheng Tao, and Zhaopeng Tu. 2021. Understanding and improving lexical choice in non-autoregressive translation. In *ICLR*.

Liang Ding, Longyue Wang, Di Wu, Dacheng Tao, and Zhaopeng Tu. 2020. Context-aware cross-attention for non-autoregressive translation. In *COLING*.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. In *EMNLP-IJCNLP*.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *ICLR*.

Jiatao Gu, Changhan Wang, and Junbo Zhao. 2019. Levenshtein transformer. In *NeurIPS*.

Junliang Guo, Xu Tan, Linli Xu, Tao Qin, Enhong Chen, and Tie-Yan Liu. 2020. Fine-tuning by curriculum learning for non-autoregressive neural machine translation. In *AAAI*.

Wenxiang Jiao, Xing Wang, Shilin He, Irwin King, Michael Lyu, and Zhaopeng Tu. 2020. Data rejuvenation: Exploiting inactive training examples for neural machine translation. In *EMNLP*.

Jungo Kasai, Nikolaos Pappas, Hao Peng, James Cross, and Noah A Smith. 2021. Deep encoder, shallow decoder: Reevaluating the speed-quality tradeoff in machine translation. *ICLR*.

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

Zhuohan Li, Zi Lin, Di He, Fei Tian, QIN Tao, WANG Liwei, and Tie-Yan Liu. 2019. Hint-based training for non-autoregressive machine translation. In *EMNLP-IJCNLP*.

Shikun Liu, Edward Johns, and Andrew J. Davison. 2019. End-to-end multi-task learning with attention. In *CVPR*.

Dabiao Ma, Zhiba Su, Wenxuan Wang, and Yu-Hao Lu. 2020. Fpets: Fully parallel end-to-end text-to-speech system. In *AAAI*.

Ari Morcos, Maithra Raghu, and Samy Bengio. 2018. Insights on representational similarity in neural networks with canonical correlation. In *NeurIPS*.

Graham Neubig, Zi-Yi Dou, Junjie Hu, Paul Michel, Danish Pruthi, Xinyi Wang, and John Wieting. 2019. compare-mt: A tool for holistic comparison of language generation systems. In *NAACL (Demonstrations)*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. Fairseq: A fast, extensible toolkit for sequence modeling. In *NAACL (Demonstrations)*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*.

Alessandro Raganato and Jörg Tiedemann. 2018. An analysis of encoder representations in transformer-based machine translation. In *EMNLP 2018 workshop BlackboxNLP*.

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.

Naomi Saphra and Adam Lopez. 2019. Understanding learning dynamics of language models with SVCCA. In *NAACL*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *ACL*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019. The bottom-up evolution of representations in the transformer: A study with machine translation and language modeling objectives. In *EMNLP-IJCNLP*.

Xing Wang, Zhaopeng Tu, Longyue Wang, and Shuming Shi. 2019a. Self-attention with structural position representations. In *EMNLP-IJCNL*.

Yiren Wang, Fei Tian, Di He, Tao Qin, ChengXiang Zhai, and Tie-Yan Liu. 2019b. Non-autoregressive machine translation with auxiliary regularization. In *AAAI*.

Bingzhen Wei, Mingxuan Wang, Hao Zhou, Junyang Lin, and Xu Sun. 2019. Imitation learning for non-autoregressive neural machine translation. In *ACL*.

Long Zhou, Jiajun Zhang, and Chengqing Zong. 2020. Improving autoregressive NMT with non-autoregressive model. In *Proceedings of the First Workshop on Automatic Simultaneous Translation*. ACL.

A Implementation

We implemented the proposed MULTI-TASK NAT model based on Fairseq (Ott et al., 2019), an open-source framework for the sequence to sequence learning. More specifically, we added the Transformer decoder (AT) to the standard Mask-Predict (NAT) model, and the encoder output is fed into the Transformer decoder with the encoder-decoder attention. We implemented a new loss function by combining the AT loss and NAT loss, and we jointly optimized all parameters. We will make all the code publicly available for future use.

A.1 Hyperparameters

For the NAT baseline model, we followed the hyperparameter settings as described in Mask-Predict (Ghazvininejad et al., 2019). More specifically, we trained all models for up to 300K steps with 128K (16000×8) tokens per batch using Adam (Kingma and Ba, 2014) with $\beta = (0.9, 0.98)$ and $\epsilon = 10^{-6}$. We adopted the warm-up learning rate scheduler, which linearly increases from 10^{-7} to a peak of $5 \cdot 10^{-4}$ within 10,000 steps, and then decays with the inverse square root schedule.

For the AT baseline model in our preliminary experiments, to make a fair comparison, we reused most parameter settings (e.g., training steps, initialization method, warm-up schedule) of the NAT model to train a strong AT baseline. Since half of the training tokens are randomly masked in the NAT baseline model, we set the batch size in the AT model as 64K (8000 × 8) tokens.

For our MULTI-TASK NAT model, we followed the same parameter setting as the NAT baseline model. We saved checkpoints every 2,000 steps and average the last 5 checkpoints as the final model. Due to the limited training data in the En⇔Ro translation task, we adopted the early stopping to prevent over-fitting in both the baseline NAT model and our model.

In the inference phase, all NAT models and our model were using the iterative decoding strategy to perform non-autoregressive translation, with the max decoding steps of 10 and length beams of 5.

A.2 Model Training

We applied the mixed precision training to all models to accelerate the training speed. The baseline model took around 15 hours to finish training on 8 Nvidia V100 GPU, while ours took around 30 hours. During the training phase, our model had an
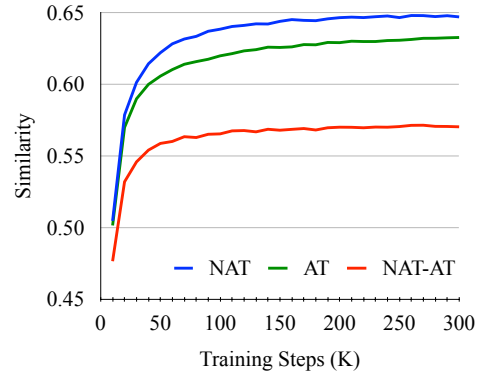

Figure 2: Representation similarity evolvement during the learning course, NAT (or AT) denotes the baseline of representation similarity between two baselines NAT (or AT) models with different initialization seeds. NAT-AT denotes the representation similarity between NAT and AT encoder. Best viewed in color.

extra decoder of 25M parameters, but the parameter size (around 64M) was the same as the baseline model in inference.

A.3 Model Inference

In our model, the auxiliary AT decoder is employed to better capture the source sentence representation during the training phase. However, since our focus is on the NAT translation performance, the auxiliary AT decoder is not necessary during the inference. We simply ignored the AT decoder in our approach to keep the same translation speed as the baseline NAT models, leaving the space for future works to leverage the ignored AT decoder and benefit the translation.

A.4 Significance Tests

We use the `compare-mt` (Neubig et al., 2019) with 1000 re-samples to perform statistical significance tests for the MASK-PREDICT model output and our model output. The results show that our model significantly outperforms the baseline model in all language directions. All these experimental results confirm the effectiveness of our proposed method even with knowledge distillation. Besides, considering the results in Table 2, as a new knowledge transfer approach, our proposed multi-task learning method MULTI-TASK NAT can complement the classic knowledge distillation method, which is promising for future exploration.

B Probing Tasks

We conducted 10 probing tasks[3] to study what linguistic properties are captured by the encoder (Conneau and Kiela, 2018). A probing task is a classification problem that focuses on simple linguistic properties of sentences. 'SeLen' predicts the number of words in a sentence. 'WC' predicts which of the target words appear on the given sentence given its sentence embedding. 'TrDep' checks whether the encoder representation infers the hierarchical structure of sentences. In 'ToCo' task, it measures the sequence of top constituents immediately below the sentence node. 'BShif' predicts whether two consecutive tokens within the sentence have been inverted or not. 'Tense' asks for the tense of the main-clause verb. 'SubN' focuses on the number of the main clause's subject. 'ObjN' studies the number of the direct object of the main clause. In 'SoMo', some sentences are modified by replacing a random noun or verb with another one and the classifier should identify whether a sentence has been modified. 'CoIn' contains sentences made of two coordinate clauses. Half of the sentences have inverted the order of the clauses and the task is to tell whether a sentence is intact or modified.

We first extracted the sentence representations of input from the AT and NAT encoder, which were used to carry out our probing tasks. For both the AT model and NAT model, the mean of the encoder top layer representations was used as the sentence representation. The classifier we used in this work was a Multi-Layer Perceptron (MLP) with a hidden dimension of 200. We optimized the model using the Adam optimizer with a learning rate of 0.001. The batch size was set to 64 and we trained the model for 4 epochs. The 10-fold cross-validation was also employed to get the final performance.

C Representation Similarity

In this section, we adopt CCA (Morcos et al., 2018), a widely-utilized representation comparison metric (Saphra and Lopez, 2019; Voita et al., 2019), to calculate the encoder representation similarity throughout the training course. The CCA score ranges from 0 to 1, and a higher CCA score indicates both encoder representations contain more similar information.

In our experiments, we first extracted the sentence encoder representations for 100,000 training examples (which is sufficient enough for CCA calculation) for each model. Then, we calculated the CCA score between the encoder representations of two models, where the representation was the mean of the top-layer encoder representations. In addition, to demonstrate the evolevment of the representation similarity during the learning course, we calculated the CCA score between two models for every 10,000 training steps.

Figure 2 shows the evolevment of representation similarity during the model training. We compute the CCA similarity between two NAT (or AT) models under different initialization seeds and take the similarity as the baseline (green and blue lines in Figure 2). Comparing to the baseline similarity of NAT (or AT), a lower similarity between AT and NAT representations (red line in Figure 2) indicates that AT and NAT encoders capture different source information.

---

[3]https://github.com/facebookresearch/SentEval/tree/master/data/probing