Towards Robust Neural Machine Translation with Iterative Scheduled Data-Switch Training

Zhongjian Miao[1 †*] Xiang Li[2†] Liyan Kang[1] Wen Zhang[2] Chulun Zhou[1]
Yidong Chen[1‡] Bin Wang[2] Min Zhang[3] Jinsong Su[1,4‡]
[1]School of Informatics, Xiamen University, China
[2]Xiaomi AI Lab, China
[3]Harbin Institute of Technology, Shenzhen, China
[4]Pengcheng Lab, China
miaozhongjian@stu.xmu.edu.cn lixiang21@xiaomi.com {ydchen,jssu}@xmu.edu.cn

Abstract

Most existing methods on robust neural machine translation (NMT) construct adversarial examples by injecting noise into authentic examples and indiscriminately exploit two types of examples. They require the model to translate both the authentic source sentence and its adversarial counterpart into the identical target sentence within the same training stage, which may be a suboptimal choice to achieve robust NMT. In this paper, we first conduct a preliminary study to confirm this claim and further propose an *Iterative Scheduled Data-switch Training Framework* to mitigate this problem. Specifically, we introduce two training stages, iteratively switching between authentic and adversarial examples. Compared with previous studies, our model focuses more on just one type of examples at each single stage, which can better exploit authentic and adversarial examples, and thus obtaining a better robust NMT model. Moreover, we introduce an improved curriculum learning method with a sampling strategy to better schedule the process of noise injection. Experimental results show that our model significantly surpasses several competitive baselines on four translation benchmarks. Our source code is available at https://github.com/DeepLearnXMU/RobustNMT-ISDST.

1 Introduction

In recent years, neural machine translation (NMT) has achieved great success (Sutskever et al., 2014; Bahdanau et al., 2015; Vaswani et al., 2017). Usually, the NMT models are trained on clean parallel corpus and thus achieve promising performance under clean inputs. However, small perturbations, such as replacing words in the input sentences, can mislead the trained model to generate incorrect translations (Belinkov and Bisk, 2018). In real-world scenarios, it is often required to deal with such sentences. Thus, it has important academic value and application prospects to design a robust NMT model for both clean and noisy inputs.

To reach this goal, some researchers explore data-oriented approaches focusing on constructing adversarial examples (Cheng et al., 2020; Zou et al., 2020). Generally, adversarial examples are used to augment the authentic dataset or fine-tune an NMT model pre-trained on the authentic dataset to improve robustness. Although data-oriented approaches are simple and efficient, they leverage adversarial examples coarsely, as concluded by Wang et al. (2021a) and Passban et al. (2021), which can not reach the full potential of these examples.

Besides, researchers also study model-oriented approaches. Some design additional model components to correct noisy inputs (Zhou et al., 2019; Qin et al., 2021; Wang et al., 2021a). There are more studies exploring training strategies for robust NMT, including multi-task learning (Zhou et al., 2019; Zhang et al., 2020), contrastive learning (Yang et al., 2019; Lee et al., 2021), and adversarial training (Cheng et al., 2018, 2019).

Despite their success, there still exist two drawbacks: 1) most existing methods indiscriminately exploit authentic and adversarial examples within the same training stage, which is a suboptimal choice confirmed in our preliminary study; 2) previous studies on robust NMT adopt a constant noise ratio to construct adversarial examples during training, while the determination of noise ratio is a subtle process, *i.e.*, too little noise may lead to poor robustness and too much noise may also hurt the model performance (Jiao et al., 2021). Therefore, dealing with both clean and noisy inputs well for NMT remains to be a significant but challenging task.

In this paper, we first conduct a preliminary study, which reveals that indiscriminately exploit-

[*] Work was done when interning at Xiaomi AI Lab.
[†] Equal Contribution.
[‡] Corresponding Author.

Proceedings of the 29th International Conference on Computational Linguistics, pages 5266–5277
October 12–17, 2022.

ing authentic and adversarial examples within the same training stage is suboptimal. Concretely, we find that this training strategy can not significantly reduce the source sentence representation (SSR) discrepancies[1] between authentic examples and the corresponding adversarial examples, resulting in a suboptimal model training which is reflected by lower model confidence[2] on examples. Based on this observation, we further propose an *Iterative Scheduled Data-Switch Training Framework* for robust NMT. Under this framework, we train the model in a two-stage scheme, iteratively switching between authentic and adversarial examples with their individual modified training objectives. During training, we introduce an additional Kullback-Leibler (KL) divergence loss, expecting the model to make similar predictions on authentic and adversarial datasets. By doing so, at each training stage, the model not only focuses on one of authentic and adversarial datasets but also avoids forgetting the knowledge from the other. Therefore, our model is able to handle both clean and noisy inputs well.

Furthermore, we introduce curriculum learning (CL) to better schedule the process of noise injection. Particularly, inspired by the *Baby Step* strategy (Wang et al., 2021b) in CL that gradually exposes more difficult examples to the model while still involving simple examples, we sample the noise ratio from a uniform distribution, where the sampling interval is progressively extended. Compared with the naive CL strategy of continuously increasing the noise ratio, our strategy is re-sampling previous simple adversarial examples which is beneficial to the model generalization.

In summary, our contributions are as follows:

- Through in-depth analyses, we expose the sub-optimum of indiscriminately exploiting authentic and adversarial examples within the same training stage, and further propose an iterative data-switch training framework for robust NMT.
- Instead of using a constant noise ratio, we introduce an improved curriculum learning method with a sampling strategy to better schedule the process of noise injection at each training stage.
- Empirical evaluations on four translation benchmarks validate the superiority of our framework, and in-deep analyses also verify the effectiveness of various factors on our framework.

2 Preliminary Study

Indiscriminately exploiting authentic examples and their adversarial counterparts within the same training stage is an effective way to build a robust NMT model. However, it requires the model to overcome the SSR discrepancy between an authentic example $(\mathbf{x}, \mathbf{y})$ and its adversarial counterpart $(\mathbf{x}', \mathbf{y})$, which increases the training difficulty to maximize $P(\mathbf{y}|\mathbf{x}; \boldsymbol{\theta})$ and $P(\mathbf{y}|\mathbf{x}'; \boldsymbol{\theta})$ simultaneously. We argue it may be a better choice to exploit authentic and adversarial examples at two training stages, iteratively switching between two types of examples. In such a data-switch training manner, the model can better benefit from the knowledge of different stages

To verify our hypothesis, we use Transformer (Vaswani et al., 2017) as our NMT model and conduct a preliminary experiment on the IWSLT14 De⇒En dataset. To be specific, we train the three models: 1) *Transformer*. We follow Vaswani et al. (2017) to train this model on the authentic dataset; 2) *Indisc-Model*. It indiscriminately exploits authentic and adversarial examples for training within the same stage. Besides, following Passban et al. (2021), we introduce a mean square error (MSE) loss to enforce the corresponding encoder outputs to be similar; 3) *Switch-Model*. This model is trained at two training stages, iteratively switching between authentic and adversarial examples. We make an investigation through the two metrics: 1) the Euclidean distances of the SSR between authentic examples and their adversarial counterparts; 2) the model confidence, *i.e.*, log-likelihood values of target ground-truth sentences.

2.1 Source Sentence Representation Discrepancy

Intuitively, to obtain high-quality translations, the SSRs from authentic and adversarial examples are expected to be similar. Therefore, we first calculate the Euclidean distances of the SSRs between two types of examples. As shown in Figure 1, the dis-

[1] We average the word representations from encoder outputs to obtain the SSRs. The SSR discrepancies represent the difference of the source sentence representations between authentic and adversarial examples, and the higher SSR discrepancies correspond to more divergent translations for authentic and adversarial examples.

[2] Model confidence represents the predicted probability for the target ground-truth sentences (Briakou and Carpuat, 2021; Zhou et al., 2022).

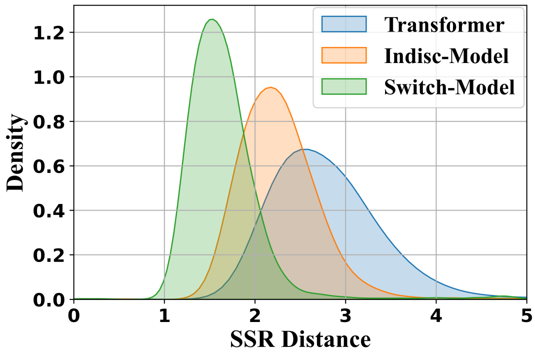


Figure 1: The kernel density estimation visualization of the SSR distances between authentic examples and their adversarial counterparts. Here, we average word representations from encoder outputs to obtain the SSRs. The authentic examples are from the entire test set, and the adversarial examples are constructed from them as mentioned in Section 3.2.

Model	**SSR Distance**	**Model Confidence**	
		Adv.	**Aut.**
Transformer	2.96	-43.0	-39.2
Indisc-Model	2.36	-41.5	-38.7
Switch-Model	**1.69**	**-41.1**	**-38.4**

Table 1: The averaged sentence-level SSR distances between authentic examples (Aut.) and their adversarial counterparts (Adv.), and the model confidence on examples.

tance distribution of *Transformer* is far from the Y-axis, and the distance distribution of *Switch-Model* is closer to the Y-axis, while the distribution of *Indisc-Model* lies between the above distributions. These results indicate *Switch-Model* reduces the SSR discrepancies well. As reported in Table 1, we also calculate the averaged sentence-level SSR distances. *Switch-Model* achieves the lowest score. These results, along with the SSR visualization (See Appendix A.1), further support the above conclusion.

2.2 Model Confidence

Higher model confidence generally leads to high-quality translations (Briakou and Carpuat, 2021; Zhou et al., 2022). Herein, we calculate the averaged sentence-level log-likelihood values for authentic and adversarial examples, respectively. As reported in Table 1, although *Indisc-Model* achieves better model confidence than *Transformer*, especially on adversarial examples, *Switch-Model* still obtains the best scores on both authentic and adversarial examples. These results indicate that *Switch-Model* is trained better on authentic and adversarial examples compared to *Indisc-Model*.

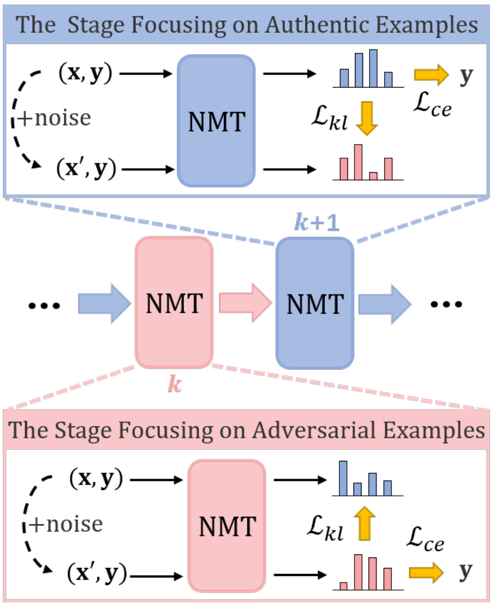


Figure 2: Diagram of the two training stages in our framework, where the pink box and blue box denote the training stages focusing on adversarial examples (the *k*-th iteration) and authentic examples (the (*k*+1)-th iteration), respectively. $(\mathbf{x}', \mathbf{y})$ is an adversarial example constructed from its authentic counterpart $(\mathbf{x}, \mathbf{y})$ with curriculum learning mentioned in Section 3.2. $\mathcal{L}_{kl}$ and $\mathcal{L}_{ce}$ denote KL-divergence loss and cross-entropy loss.

3 Methodology

Based on the observations in Section 2, we further propose an iterative scheduled data-switch training framework for robust NMT.

3.1 Training Framework

In contrast to the previous work, our framework introduces two iterative training stages to handle authentic and adversarial examples, respectively.

As shown in Figure 2, at the training stage focusing on adversarial examples (the *k*-th iteration), we first use the best model at the last training stage (the (*k*-1)-th iteration) as initialization, and then optimize the model on two types of examples using a modified training objective. Specifically, we additionally introduce KL-divergence loss into the conventional training objective, expecting the model predictions on adversarial examples to be close to those on authentic examples. Formally, the modified training objective $\mathcal{L}_{adv}$ at this stage is defined as follows:

$$\mathcal{L}_{adv} = \sum_{\substack{(\mathbf{x},\mathbf{y})\in\mathcal{D}\\(\mathbf{x}',\mathbf{y})\in\mathcal{D}'}} [-\log P(\mathbf{y}|\mathbf{x}';\boldsymbol{\theta}) + \alpha \mathbf{KL}(P(\mathbf{y}|\mathbf{x}';\boldsymbol{\theta})||P(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}))], \quad (1)$$

where α is a weight factor, $\boldsymbol{\theta}$ denotes the model parameters, $(\mathbf{x}, \mathbf{y})$ and $(\mathbf{x}', \mathbf{y})$ denote an authentic example and its adversarial counterpart, respectively.

Likewise, at the training stage focusing on authentic examples, the modified training objective $\mathcal{L}_{aut}$ is given by

$$\begin{aligned}\mathcal{L}_{aut} = \sum_{\substack{(\mathbf{x},\mathbf{y})\in\mathcal{D}\\(\mathbf{x}',\mathbf{y})\in\mathcal{D}'}} & [-\log P(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}) \\ & + \alpha\mathbf{KL}(P(\mathbf{y}|\mathbf{x};\boldsymbol{\theta})||P(\mathbf{y}|\mathbf{x}';\boldsymbol{\theta}))].\end{aligned} \tag{2}$$

We conduct training stages for K iterations. In such an iterative data-switch training manner, the knowledge of different stages can continuously enhance the model in a collaborative way, which has also been verified in previous studies (Zeng et al., 2019; Liu et al., 2020b).

3.2 Generate Adversarial Examples with Curriculum Learning

During training, we generate adversarial examples on the fly by injecting noise into the source sentences of the corresponding authentic examples. Without loss of generality, we inject noise by performing three common operations with equal probability: *delete*, *replace*, and *swap*. Note that our framework is also applicable to other types of noise.

Previous work on robust NMT pays little attention to the noise ratio during training. In this work, we introduce curriculum learning (CL) to schedule the process of noise injection at each training stage. Inspired by the *Baby Step* strategy in CL (Wang et al., 2021b), at each training step, we sample the noise ratio from a uniform distribution, where the sampling interval is progressively extended. By doing so, our sampling strategy re-samples previous simple adversarial examples during training, which is beneficial to the model generalization.

The procedure of generating adversarial examples is presented in Algorithm 1. At the training step t, we first load a batch of examples and sample a noise ratio r_t from a uniform distribution $U(0, R(t))$ (**Lines 3-4**). Intuitively, a sharp increase of $R(t)$ may hurt the model optimization. Therefore, we expect that $R(t)$ increases smoothly. To this end, we define $R(t)$ as follows:

$$R(t) = \sqrt{R^2_{max} \times \frac{t}{T}}, \tag{3}$$

where R_{max} is the maximal noise ratio and T denotes the maximal training step number of each training stage. Note that the derivative of $R(t)$ decreases with the increase of *t*, which satisfies our expectations that $R(t)$ increases smoothly (See Appendix A.2). According to r_t, we traverse each authentic example $(\mathbf{x}, \mathbf{y})$ in the current mini-batch (**Line 5**) and determine the number n_t of perturbed words in **x** (**Line 6**), and then perform three kinds of operations with equal probability on them to generate adversarial examples $(\mathbf{x}', \mathbf{y})$ (**Line 7**). Finally, we train the model with our modified objective based on two types of examples (**Line 8**). For efficiency, we update $R(t)$ every 10K training step (**Lines 10-12**).

Algorithm 1 Generate Adversarial Examples with Curriculum Learning for Each Training Stage

```
Input: Training corpus $\mathcal{D}$, maximal training step
    number $T$, maximal noise ratio $R_{max}$.
 1: $R(t) \leftarrow 0$
 2: for $t$ =1, 2, ..., $T$ do
 3:     Load a mini-batch $\mathcal{B}_t$ from $\mathcal{D}$
 4:     Sample a noise ratio $r_t \sim U(0, R(t))$
 5:     for each example $(\mathbf{x}, \mathbf{y})$ in $\mathcal{B}_t$ do
 6:         $n_t \leftarrow \lceil len(\mathbf{x}) \times r_t \rceil$
 7:         Perturb $n_t$ words in $\mathbf{x}$ to generate its adversarial counterpart $\mathbf{x}'$
 8:         Using $(\mathbf{x},\mathbf{y})$ and $(\mathbf{x}',\mathbf{y})$ to train the model according to the modified training objective defined in Equation 1 or Equation 2
 9:     end for
10:     if $t$ % 10K == 0 then
11:         Update $R(t)$ by Equation 3
12:     end if
13: end for
```

4 Experiments

4.1 Setup

Datasets For the small-scale dataset, we use IWSLT14 German⇒English (De⇒En) corpus, where the training set comprises 160K sentence pairs extracted from TED talks, the original validation set consists of dev2010 and dev2012, and the clean test set consists of tst2010, tst2011 and tst2012. For the middle-scale datasets, we use MTNT[3] French⇒English (Fr⇒En) (Michel and Neubig, 2018) and WMT14 English⇒German (En⇒De) datasets. The former consists of 2.2M sentence pairs for training, newsdiscussdev2015 is

[3] https://pmichel31415.github.io/mtnt/index.html#data

used as the original validation set, newstest2014 (NT14) and newsdicusstest2015 (NT15) are used as the clean test sets. The latter contains 4.5M sentence pairs, and we choose newstest2013 as our original validation set, and newstest2014 as our clean test set. For the large-scale dataset, we use WMT20 Chinese⇒English (Zh⇒En) dataset containing 22M sentence pairs for training and newstest2019 (with 1,997 sentence pairs) for validating and newstest2020 (with 1,418 sentence pairs) for testing.

Note that in this work, we focus on the performance on clean and noisy test sets. Thus we select the best model according to the hybrid validation sets, each of which contains the original validation set and its disturbed counterpart. In addition to the standard clean test sets, we also evaluate models on noisy test sets. For the De⇒En, En⇒De and Zh⇒En translation tasks, we construct the synthetic noisy test sets by performing operations (See Section 3.2) on a certain ratio of source words in the original test sets. For the Fr⇒En translation task, we evaluate models on two social media test sets with diverse noise: mtnt18 (Michel and Neubig, 2018) and mtnt19 (Li et al., 2019), both of which have been widely used in robust NMT task (Li et al., 2019).

We also employ BPE (Sennrich et al., 2016) to split words into subwords. During this process, the numbers of merge operations are separately set to 10K, 16K, 32K and 32K for De⇒En, Fr⇒En, En⇒De and Zh⇒En datasets. Finally, we report case-sensitive tokenized BLEU (Papineni et al., 2002) for the De⇒En, En⇒De and Zh⇒En translation tasks and sacreBLEU (Post, 2018) for the Fr⇒En translation task.

Training Details We adopt the *fairseq*[4] (Ott et al., 2019) Transformer as our basic model. We use the *transformer_iwslt_de_en* setting for the De⇒En translation task, and the *transformer_wmt_en_de* setting for the En⇒De, Fr⇒En and Zh⇒En translation tasks, respectively.

As for the model optimization, we use the Adam optimizer (Kingma and Ba, 2015) with β_1=0.9, β_2=0.98 and ϵ=10^{-9}. All experiments are done on NVIDIA V100 GPUs with mixed-precision training, where batch sizes are roughly set to 4K, 8K, 32K, and 32K tokens for the De⇒En, Fr⇒En, En⇒De, and Zh⇒En translation tasks, respectively. For all datasets, we set the maximal noise ratio R_{max} as 0.1 and we tune the weight factor $\alpha\in\{0.5, 1.0, 1.5\}$ on our validation sets at the first training stage, then keep it unchanged in subsequent stages for efficiency. We determine the maximal training step number T through an empirical study according to the convergence of the model at each stage. Specifically, we set T for the stages focusing on authentic and adversarial examples to 150K and 200K, respectively.

Baselines In addition to the vanilla Transformer model (Vaswani et al., 2017), we compare our model with the following baselines:

- *Transformer-FT*. It is pre-trained on the authentic dataset and then fine-tuned on the adversarial dataset.
- *Transformer-Mixed*. This model is trained on the dataset mixed with authentic and adversarial examples.
- *Transformer-Indisc*. It indiscriminately exploits authentic and adversarial examples for training. Besides, the model predictions between two types of examples are minimized via a bidirectional KL-divergence loss (Liang et al., 2021).
- *MTNT* (Michel and Neubig, 2018). It is the first benchmark on the MTNT Fr⇒En dataset.
- *AdvST* (Cheng et al., 2018). This model is trained using *adversarial stability training* strategy, which enables the encoder and decoder to generate similar representations for the original inputs and their perturbed counterparts
- *SwitchOut* (Wang et al., 2018). It uses a data augmentation strategy for training, where the augmented data is constructed by randomly replacing words in source and target sentences with other words.
- *DouAdv* (Cheng et al., 2019). It generates discrete adversarial examples with doubly adversarial inputs according to the gradients of word embeddings.
- *MTL* (Zhou et al., 2019). It introduces multitask learning into robust NMT, where two decoders are involved: one learns to denoise the text and the other generates the final translations from the denoised text.

[4] https://github.com/fairseq/fairseq

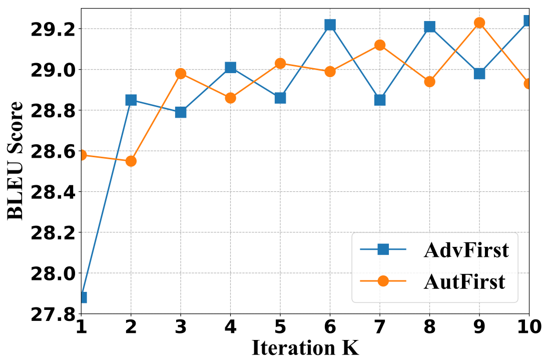


Figure 3: BLUE (%) scores of our model on the Fr⇒En validation set with different K. AdvFirst and AutFirst denote we focus on adversarial and authentic examples first, respectively.

- *ContRec* (Xu et al., 2021). This model reduces the effect of noisy words through a context-enhanced reconstruction component.

4.2 Effects of Data Order and Iteration Number K

Under our framework, it should be determined which type of examples we need to first focus on and what the appropriate iteration number K is. We explore their effects in this subsection. To this end, we train the models focusing on authentic and adversarial examples first with different K, respectively. The results on the validation sets are displayed in Figure 3.

Which Type of Examples to First Focus on? As illustrated in Figure 3, we observe that the model focusing on adversarial examples first reaches a competitive result at the 6th iteration, while the model focusing on authentic examples first needs 9 iterations to obtain a similar result, indicating the former converges faster to a better result.

What Is the Appropriate Iteration Number K? Overall, as iteration number K increases, we find the model performance is improved, whether we focus on authentic or adversarial examples first.

Based on these results on the validation sets, we choose to first focus on adversarial examples and set the iteration number K to 6 for the Fr⇒En dataset. Similarly, we set K to 5 for all other datasets.

Model	**De⇒En**	**En⇒De**	**Zh⇒En**
Transformer	34.82	27.78	26.83
	Data-Oriented		
Transformer-FT	34.90	27.75	25.76
Transformer-Mixed	34.85	27.72	24.07
	Model-Oriented		
AdvST (Cheng et al., 2018)	—	25.26	—
DouAdv (Cheng et al., 2019)	—	28.34	—
Transformer-Indisc	36.59	28.21	26.51
Ours	**37.28***†	**28.93***†	**27.45***†

Table 2: BLEU (%) scores on the clean test sets of four translation tasks. '$*$' and '†' mean the improvements over *Transformer-Indisc* and *Transformer* are significantly with p<0.01 (Koehn, 2004).

Model	**Clean Test**		**Noisy Test**	
	NT14	**NT15**	**mtnt18**	**mtnt19**
Transformer	31.76	31.14	25.67	29.74
MTNT (Michel and Neubig, 2018)	28.90	30.80	23.30	26.20
		Data-Oriented		
Transformer-FT	32.37	30.71	26.54	29.03
Transformer-Mixed	31.87	30.71	25.44	27.90
SwitchOut (Wang et al., 2018)	29.20	31.10	25.00	28.10
		Model-Oriented		
MTL (Zhou et al., 2019)	—	—	24.50	30.30
ConRec (Xu et al., 2021)	30.70	32.40	26.50	29.10
Transformer-Indisc	32.83	31.37	26.42	28.98
Ours	**34.11***†	**32.67***†	**28.16***†	**30.77***†

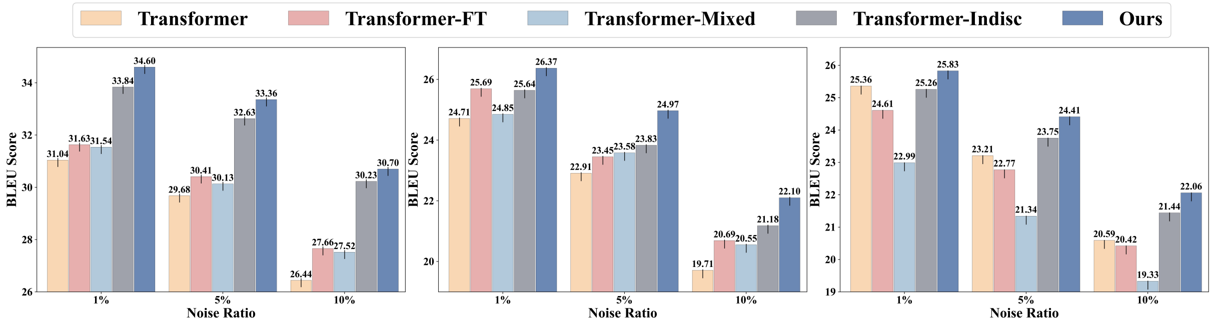
Table 3: BLEU (%) scores on the Fr⇒En translation task. '$*$' and '†' mean the improvements over *Transformer-Indisc* and *Transformer* are significantly with p<0.01 (Koehn, 2004).

4.3 Main Results

Results on Clean Test Sets Table 2 shows the results on clean test sets for the De⇒En, En⇒De, Zh⇒En tasks, and the results for the Fr⇒En task are reported in the second and third columns of Table 3. Data-oriented approaches achieve comparable or worse results compared to *Transformer*, indicating data-oriented approaches may hurt the performance on the standard clean test sets. *Transformer-Indisc* is a strong baseline model. It performs better than *Transformer* and achieves promising performance compared to other baselines, except for the Zh⇒En task. Compared with the data-oriented and model-oriented baselines, our model achieves the best performance across all datasets. Concretely, our model achieves +0.59 BLEU improvement than the most competitive contrast model *DouAdv* on the En⇒De dataset. For the large-scale Zh⇒En dataset, all related approaches fail and do not outperform *Transformer*, while our model achieves +0.62 BLEU improvement over *Transformer*. These results fully demonstrate the superiority of our framework.

Results on Noisy Test Sets To verify the model robustness, we evaluate models on the synthetic noisy test sets and the social media test sets, respectively.

The fourth and fifth columns of Table 3 report the results on the social media test sets. It is worth



(a) IWSLT14 De ⇒ En **(b) WMT14 En ⇒ De** **(c) WMT20 Zh ⇒ En**

Figure 4: BLEU (%) scores on the synthetic noisy test sets with different noise ratios.

noticing that although we inject finite types of noise during training, our model still beats other baselines on both mtnt18 and mtnt19 test sets, which shows the better generalization of our model.

For the synthetic noisy test sets, we compare the performance of all models on the test sets with different noise ratios. As shown in Figure 4, *Transformer* suffers from performance drops under noisy inputs, revealing the vulnerability of the NMT model. By contrast, data-oriented approaches perform slightly better than *Transformer* across different noise ratios, except for the large-scale Zh⇒En dataset. We argue the robustness achieved by data-oriented approaches is restricted because the Zh⇒En training set is large enough to cover diverse noises. Additionally, *Transformer-Indisc* performs better than data-oriented approaches and *Transformer*, showing its strong robustness. Finally, we find our model consistently outperforms other baselines across different noise ratios even under the large-scale data configuration, which confirms again that our framework can significantly enhance the model robustness.

4.4 Source Sentence Representation Discrepancy and Model Confidence

Following the settings of the preliminary study in Section 2, we evaluate models using two metrics: the SSR distances and model confidence. As shown in Figure 5, the distance distribution of our model is significantly closer to the Y-axis compared to *Transformer* and *Transformer-Indisc*, indicating that our model significantly reduces the SSR discrepancies. Analogously, we report the averaged sentence-level SSR distances in Table 4 and visualize the SSRs for clear understanding (See Appendix A.1), all of which demonstrate the effectiveness of our model. Besides, the averaged sentence-level log-likelihood values presented in Table 4 show that our model obtains the highest model confidence on two types of examples. It implies our model is trained better on authentic and adversarial examples. In summary, the results of the two metrics show that our model can deal with clean and noisy inputs well.

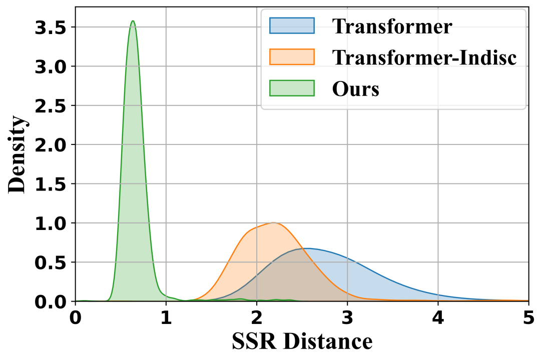


Figure 5: The kernel density estimation visualization of the SSR distances.

Model	SSR Distance	Model Confidence	
		Adv.	Aut.
Transformer	2.96	-43.0	-39.2
Transformer-Indisc	2.25	-38.3	-35.8
Ours	**0.67**	**-38.0**	**-35.6**

Table 4: The averaged sentence-level SSR distances and the model confidence on examples.

4.5 Effects of Different Types of Noise

To better understand the effects of different types of noise, we inject only one type of noise into the training data and the test set respectively, and then inspect the performance change of our model. From Table 5, we arrive at the following conclusions:

(1) *The models injecting different types of noise into the training set perform similarly on the clean test set.* From the first column of Table 5, we observe that adopting *delete* and *replace* operations separately during training perform slightly better than our hybrid noise strategy, while adopting *swap* operation obtains the worst performance.

(2) *When only one type of noise is injected, our*

Model	Different Types of Noise				
	Clean	Hybrid	Swap	Replace	Delete
Ours-Hybrid	37.28	**34.60**	**36.56**	**32.98**	33.85
Ours-Swap	36.91	33.54	36.29	31.62	32.63
Ours-Replace	**37.54**	33.83	35.04	32.97	33.48
Ours-Delete	37.51	34.27	35.78	32.11	**34.40**

Table 5: The effects of different types of noise on the IWSLT14 De⇒En dataset. Here, the noise ratios of all noise test sets are set to 1% and **bold** indicates the best result for each noise test set (each column).

model performs better if both training and test sets are injected with the same type of noise. For example, adopting *swap* operation during training obtains 36.29 BLEU on the *swap* noise test set, while adopting *replace* and *delete* operations obtain 35.04 and 35.78 BLEU on the same test set, respectively.

(3) *The performance of the model on the test sets with different types of noise differs greatly*. Comparing each column in Table 5, the model performs worst on the *replace* noise test set, while the *swap* noise has relatively little damage to the model performance.

(4) *The hybrid noise strategy we adopt achieves balanced results*. Comparing each row in Table 5, we find that our model with the hybrid strategy achieves the best results on the *hybrid*, *swap* and *replace* noise test sets and competitive results on the rest test sets.

4.6 Ablation Study

To verify the effectiveness of various factors on our framework, we further compare our framework with the following variants and present the results in Table 6:

(1) *w/ FNR*. In this variant, we directly use a **F**ixed **N**oise **R**atio to schedule the process of noise injection. As reported in Table 6, this variant decreases the performance dramatically on both clean and noisy test sets. It reveals the importance of scheduling the noise injection with CL.

(2) *w/ FSI*. In our improved CL method, the sampling interval is progressively extended. In this variant, we adopt a **F**ixed **S**ampling **I**nterval and the noise ratio is sampled uniformly from it. As shown in Table 6, using a fixed sampling internal also leads to the performance degradation.

(3) *w/o SS*. Inspired by the *Baby Step* (Wang et al., 2021b) in CL, we equip CL with a **S**ampling **S**trategy (See Section 3.2). Note that our CL strategy degenerates into the naive CL strategy (the variant *w/o SS*) if we remove the *SS* component. The results listed in Table 6 demonstrate the effectiveness of our sampling strategy.

Model	Clean Test		Noisy Test	
	NT14	NT15	mtnt18	mtnt19
Ours	34.11	32.67	28.16	30.77
w/ FNR	32.15	29.93	23.70	27.51
w/ FSI	33.77	31.21	26.25	29.44
w/o SS	33.47	31.37	26.22	30.43
w/o KL	33.09	30.28	25.84	29.05
KL⇒MSE	32.52	30.92	26.62	29.16

Table 6: Ablation study on the Fr⇒En translation task.

(4) *w/o KL*. We introduce a KL-divergence loss to ensure that the model focuses more on one type of examples at each stage while preventing forgetting the knowledge from another type. As shown in Table 6, compared with the variant *w/o KL*, this regularization term indeed enhances the model capability to cope with both clean and noisy inputs.

(5) *KL⇒MSE*. In this variant, we replace KL-divergence loss with the MSE loss on decoder output hidden states. From Table 6, we can observe that this variant performs better than the framework without KL-divergence loss (the variant *w/o KL*) in 3 out of 4 test sets, showing the importance of the regularization term. However, compared to the MSE regularization, the KL-divergence regularization is more suitable for our framework.

5 Related Work

To build robust NMT models, researchers have proposed a range of methods, which can be mainly divided into two categories: data-oriented and model-oriented approaches.

In the first category, how to construct adversarial examples is a non-trivial problem (Cheng et al., 2020; Zou et al., 2020). Usually, adversarial examples are used in two ways: one is to directly train a robust model using the dataset mixed with authentic and adversarial examples (Belinkov and Bisk, 2018; Karpukhin et al., 2019), and the other is to use adversarial examples to fine-tune the NMT model pre-trained on authentic examples (Helcl et al., 2019; Dabre and Sumita, 2019; Berard et al., 2019; Alam and Anastasopoulos, 2020).

In the second category, some researchers design additional components for NMT model to correct noisy inputs (Qin et al., 2021; Wang et al., 2021a; Xu et al., 2021) or explore fault-tolerant neural networks(Su et al., 2017; Tan et al., 2018). Mean-while, more researchers resort to exploring training strategies, including multi-task learning (Zhou et al., 2019; Zhang et al., 2020), contrastive learning (Yang et al., 2019; Lee et al., 2021), and adversarial training (Cheng et al., 2018, 2019; Liu et al., 2020a).

In this work, the proposed framework belongs to the second model-oriented category. In this regard, most existing methods indiscriminately exploit authentic and adversarial examples within the same training stage, which are suboptimal confirmed in our preliminary study. To mitigate this problem, we propose an iterative scheduled data-switch training framework for robust NMT, where we introduce two training stages, iteratively switching between authentic and adversarial examples. Besides, inspired by the successful applications of curriculum learning (CL) in NMT (Platanios et al., 2019; Xu et al., 2020; Zhou et al., 2020), we use CL to better schedule the process of noise injection. Particularly, we equip CL with a sampling strategy, which is beneficial to the model generalization.

Finally, note that Jiao et al. (2021) introduce an alternated training to alleviate the performance drop caused by low-quality back-translation data. Our work differs from theirs in three aspects: 1) we aim at building a robust NMT model dealing with clean and noisy inputs well, while Jiao et al. (2021) try to prevent the model performance on clean test sets from being disturbed by synthetic data; 2) we introduce an improved CL method to better schedule the process of noise injection, which is beneficial to the model performance; 3) in addition to the conventional cross-entropy objective (Jiao et al., 2021), we introduce an additional regularization term to cope with both clean and noisy inputs well.

6 Conclusion

In this paper, we first conduct a preliminary study to reveal that indiscriminately exploiting authentic and adversarial examples for robust NMT is suboptimal. To achieve better robust NMT, we further propose an iterative scheduled data-switch training framework, where we train the model at two training stages, iteratively switching between authentic and adversarial examples. Moreover, we introduce curriculum learning with a sampling strategy to schedule the process of noise injection at each training stage. Extensive experiments show the superiority of our framework.

In the future, we will introduce more types of real noise, such as ASR errors, into our framework. Besides, we plan to apply our framework to other natural language generation tasks, such as dialogue generation, so as to verify the generality of our framework.

Acknowledgements

This research was supported by National Natural Science Foundation of China (Nos. 62036004, 62276219, 62076211, 61573294), Natural Science Foundation of Fujian Province of China (No. 2020J06001), and Youth Innovation Fund of Xiamen (No. 3502Z20206059). We would like to thank Yong Cheng (Google Research), Yang Liu (Tsinghua University) for valuable feedback on this work and the reviewers for their insightful comments.

References

Md Mahfuz Ibn Alam and Antonios Anastasopoulos. 2020. Fine-tuning MT systems for robustness to second-language speaker variations. In *Proc. of W-NUT@EMNLP*.

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

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *Proc. of ICLR*.

Alexandre Berard, Ioan Calapodescu, and Claude Roux. 2019. Naver labs europe's systems for the WMT19 machine translation robustness task. In *Proc. of WMT*.

Eleftheria Briakou and Marine Carpuat. 2021. Beyond noise: Mitigating the impact of fine-grained semantic divergences on neural machine translation. In *Proc. of ACL*.

Yong Cheng, Lu Jiang, and Wolfgang Macherey. 2019. Robust neural machine translation with doubly adversarial inputs. In *Proc. of ACL*.

Yong Cheng, Lu Jiang, Wolfgang Macherey, and Jacob Eisenstein. 2020. Advaug: Robust adversarial augmentation for neural machine translation. In *Proc. of ACL*.

Yong Cheng, Zhaopeng Tu, Fandong Meng, Junjie Zhai, and Yang Liu. 2018. Towards robust neural machine translation. In *Proc. of ACL*.

Raj Dabre and Eiichiro Sumita. 2019. Nict's supervised neural machine translation systems for the WMT19 translation robustness task. In *Proc. of WMT*.

Jindrich Helcl, Jindrich Libovický, and Martin Popel. 2019. CUNI system for the WMT19 robustness task. In *Proc. of WMT*.

Rui Jiao, Zonghan Yang, Maosong Sun, and Yang Liu. 2021. Alternated training with synthetic and authentic data for neural machine translation. In *Proc. of ACL Findings*.

Vladimir Karpukhin, Omer Levy, Jacob Eisenstein, and Marjan Ghazvininejad. 2019. Training on synthetic noise improves robustness to natural noise in machine translation. In *Proc. of W-NUT@EMNLP*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proc. of ICLR*.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proc. of EMNLP*.

Seanie Lee, Dong Bok Lee, and Sung Ju Hwang. 2021. Contrastive learning with adversarial perturbations for conditional text generation. In *Proc. of ICLR*.

Xian Li, Paul Michel, Antonios Anastasopoulos, Yonatan Belinkov, Nadir Durrani, Orhan Firat, Philipp Koehn, Graham Neubig, Juan Miguel Pino, and Hassan Sajjad. 2019. Findings of the first shared task on machine translation robustness. In *Proc. of WMT*.

Xiaobo Liang, Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, and Tie-Yan Liu. 2021. R-drop: Regularized dropout for neural networks. In *Proc. of NIPS*.

Kai Liu, Xin Liu, An Yang, Jing Liu, Jinsong Su, Sujian Li, and Qiaoqiao She. 2020a. A robust adversarial training approach to machine reading comprehension.

Xin Liu, Kai Liu, Xiang Li, Jinsong Su, Yubin Ge, Bin Wang, and Jiebo Luo. 2020b. An iterative multi-source mutual knowledge transfer framework for machine reading comprehension. In *Proc. of IJCAI*.

Paul Michel and Graham Neubig. 2018. MTNT: A testbed for machine translation of noisy text. In *Proc. of EMNLP*.

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 *Proc. of NAACL*.

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

Peyman Passban, Puneeth S. M. Saladi, and Qun Liu. 2021. Revisiting robust neural machine translation: A transformer case study. In *Proc. of EMNLP Findings*.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabás Póczos, and Tom M. Mitchell. 2019. Competence-based curriculum learning for neural machine translation. In *Proc. of NAACL*.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proc. of WMT*.

Wenjie Qin, Xiang Li, Yuhui Sun, Deyi Xiong, Jianwei Cui, and Bin Wang. 2021. Modeling homophone noise for robust neural machine translation. In *Proc. of ICASSP*.

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

Jinsong Su, Zhixing Tan, Deyi Xiong, Rongrong Ji, Xiaodong Shi, and Yang Liu. 2017. Lattice-based recurrent neural network encoders for neural machine translation. In *Proc. of AAAI*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proc. of NIPS*.

Zhixing Tan, Jinsong Su, Boli Wang, Yidong Chen, and Xiaodong Shi. 2018. Lattice-to-sequence attentional neural machine translation models. *Neurocomputing*.

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

Tao Wang, Chengqi Zhao, Mingxuan Wang, Lei Li, Hang Li, and Deyi Xiong. 2021a. Secoco: Self-correcting encoding for neural machine translation. In *Proc. of EMNLP Findings*.

Xin Wang, Yudong Chen, and Wenwu Zhu. 2021b. A comprehensive survey on curriculum learning. *IEEE T-PAMI*.

Xinyi Wang, Hieu Pham, Zihang Dai, and Graham Neubig. 2018. Switchout: an efficient data augmentation algorithm for neural machine translation. In *Proc. of EMNLP*.

Chen Xu, Bojie Hu, Yufan Jiang, Kai Feng, Zeyang Wang, Shen Huang, Qi Ju, Tong Xiao, and Jingbo Zhu. 2020. Dynamic curriculum learning for low-resource neural machine translation. In *Proc. of COLING*.

Weiwen Xu, Ai Ti Aw, Yang Ding, Kui Wu, and Shafiq R. Joty. 2021. Addressing the vulnerability of NMT in input perturbations. In *Proc. of NAACL*.

Zonghan Yang, Yong Cheng, Yang Liu, and Maosong Sun. 2019. Reducing word omission errors in neural machine translation: A contrastive learning approach. In *Proc. of ACL*.

Jiali Zeng, Yang Liu, Jinsong Su, Yubin Ge, Yaojie Lu, Yongjing Yin, and Jiebo Luo. 2019. Iterative dual domain adaptation for neural machine translation. In *Proc. of EMNLP*.

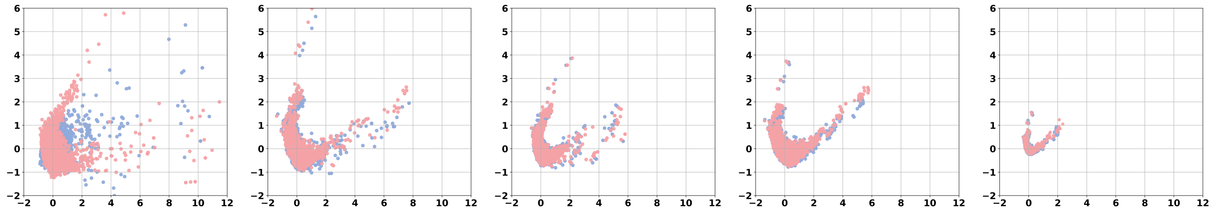
Huaao Zhang, Shigui Qiu, Xiangyu Duan, and Min Zhang. 2020. Token drop mechanism for neural machine translation. In *Proc. of COLING*.

Chulun Zhou, Fandong Meng, Jie Zhou, Min Zhang, Hongji Wang, and Jinsong Su. 2022. Confidence based bidirectional global context aware training framework for neural machine translation. *In Proc. of ACL*.

Shuyan Zhou, Xiangkai Zeng, Yingqi Zhou, Antonios Anastasopoulos, and Graham Neubig. 2019. Improving robustness of neural machine translation with multi-task learning. In *Proc. of WMT*.

Yikai Zhou, Baosong Yang, Derek F. Wong, Yu Wan, and Lidia S. Chao. 2020. Uncertainty-aware curriculum learning for neural machine translation. In *Proc. of ACL*.

Wei Zou, Shujian Huang, Jun Xie, Xinyu Dai, and Jiajun Chen. 2020. A reinforced generation of adversarial examples for neural machine translation. In *Proc. of ACL*.



(a) Transformer (b) Indisc-Model (c) Switch-Model (d) Transformer-Indisc (e) Ours

Figure 6: Visualization of the SSRs for authentic examples and the corresponding adversarial examples. Here, we apply the PCA algorithm to reduce the source sentence representations to the 2-dim ones and visualize them. The pink and blue dots denote adversarial and authentic examples, respectively.

A Appendix

A.1 Visualization of the Source Sentence Representations

To understand the source sentence representations (SSRs) clearly, we apply the PCA algorithm to the SSRs of authentic and adversarial examples and visualize the SSRs. Herein, following the settings of the preliminary study (See Section 2), we average word representations to obtain the SSRs. The authentic examples are obtained from the entire test set of the IWSLT14 De⇒En dataset, and the adversarial counterparts are constructed from them as mentioned in Section 3.2.

From Figure 6(a), we observe the SSRs of authentic and adversarial examples extracted from *Transformer* scatter differently. The reason behind this phenomenon is that *Transformer* is trained only on the authentic dataset and thus fits bad to adversarial examples, leading to huge SSR discrepancies between two types of examples.

According to Figure 6(b) and Figure 6(c), which correspond to the preliminary study in Section 2, although *Indisc-Model* reduces the SSR discrepancies between authentic and adversarial examples well compared to *Transformer*, *Switch-Model*, can further reduce the SSR discrepancies, bringing closer source sentence representations for two types of examples.

Figure 6(d) and Figure 6(e) are correspond to the analysis in Section 4.4. *Transformer-Indisc* reduces the SSR discrepancies well compared to *Transformer* and it achieves competitive results (See Section 4.3). By contrast, our model can further reduce the SSR discrepancies and achieve the best performances across all datasets (See Section 4.3), which confirms the effectiveness of our framework.

A.2 Definition of the function $R(t)$

Intuitively, a sharp increase of $R(t)$ may hurt the model optimization. We expect that $R(t)$ increases smoothly, hence we define the derivative of $R(t)$ as

$$\frac{\mathrm{d}R(t)}{\mathrm{d}t} = \frac{c_1}{R(t)}, \tag{4}$$

for some constant $c_1 \geq 0$, and $R(t)$ is a non-decreasing function. The right side of Equation 4 decreases as the training processes, which indicates the derivative of $R(t)$ gradually decreases, *i.e.*, $R(t)$ increases smoothly. Along with the constraint that $R(t) \geq 0$ for all $t \geq 0$, solving this simple differential equation, we obtain:

$$\begin{aligned} &\int R(t)\mathrm{d}R(t) = \int c_1 \mathrm{d}t \\ &\Rightarrow R(t) = \sqrt{c_1 t + c_2}, \end{aligned} \tag{5}$$

for some constants $c_1 \geq 0$ and $c_2 \geq 0$. Then, we consider the following constraints:

$$\begin{cases} R(0) = 0 \\ R(T) = R_{max}, \end{cases} \tag{6}$$

where T denotes the maximal training step number at each training stage, and R_{max} denotes the maximal noise ratio. Combining Equation 5 and 6, the final formula of $R(t)$ is rewritten as:

$$R(t) = \sqrt{R_{max}^2 \times \frac{t}{T}}. \tag{7}$$