electronics

MDPI

Article

Double Consistency Regularization for Transformer Networks

Yuxian Wan, Wenlin Zhang * and Zhen Li

School of Information System Engineering, PLA Strategic Support Force Information Engineering University, Zhengzhou 450001, China
* Correspondence: wenlinzzz@163.com

Abstract: The large-scale and deep-layer deep neural network based on the Transformer model is very powerful in sequence tasks, but it is prone to overfitting for small-scale training data. Moreover, the prediction result of the model with a small disturbance input is significantly lower than that without disturbance. In this work, we propose a double consistency regularization (DOCR) method for the end-to-end model structure, which separately constrains the output of the encoder and decoder during the training process to alleviate the above problems. Specifically, on the basis of the cross-entropy loss function, we build the mean model by integrating the model parameters of the previous rounds and measure the consistency between the models by calculating the KL divergence between the features of the encoder output and the probability distribution of the decoder output of the mean model and the base model so as to impose regularization constraints on the solution space of the model. We conducted extensive experiments on machine translation tasks, and the results show that the BLEU score increased by 2.60 on average, demonstrating the effectiveness of DOCR in improving model performance and its complementary impacts with other regularization techniques.

Keywords: cross-entropy loss; deep neural network; KL divergence; overfitting; transformer; regularization

check for updates

Citation: Wan, Y.; Zhang, W.; Li, Z. Double Consistency Regularization for Transformer Networks. *Electronics* **2023**, *12*, 4357. https://doi.org/10.3390/electronics12204357

Academic Editors: Katia Lida Kermanidis, Phivos Mylonas and Manolis Maragoudakis

Received: 19 September 2023
Revised: 16 October 2023
Accepted: 16 October 2023
Published: 20 October 2023

1. Introduction

End-to-end models [1–3] with different deep neural network topologies have excelled in sequential tasks like neural machine translation (NMT). On the strength of its impressive performance in recent years, Transformer [4], the most popular end-to-end model, has evolved into the most fundamental model in NMT. Even so, the scale of state-of-the-art Transformer neural networks is still linear in the number of training examples. In low-resource task research, the amount of model parameters is much greater than the amount of data. Therefore, a key point for research in low-resource tasks is how to solve Transformer's tendency to overfit small-scale labeled data [5–7].

Some academics employ semi-supervised learning to address the issue of less labeled data and add a significant proportion of unlabeled data to the model training process to address these issues [8]. This approach does not require specific labels and is frequently based on the prediction vector produced by the model. By generating an unsupervised regularization loss term between the disturbed prediction result and the normal prediction result on the unlabeled data, this approach increases the generalization ability of the model [9–11].

On the basis of supervised cross-entropy loss, the π model [9] adds unlabeled samples in each round to different noises, propagates forward twice, and calculates the consistency loss of the two prediction results. The final training target is the sum of the cross-entropy loss and consistency loss. The temporal ensembling [9] model replaces the result of the second forward propagation of the unlabeled data with the prediction results obtained in the past epoch to calculate the consistency loss based on the π model. Mean teacher [10] uses two models, in which the teacher model is updated by the exponential moving average [11] (EMA) of the student model parameters so as to update the model weight

online at each step and obtain the consistency loss by calculating the error between the prediction results of the teacher model and the student model. Many variants were also proposed later, such as dual students [12], Fast-SWA [13], virtual adversarial training [14], interpolation consistency training [15], unsupervised data augmentation [16], etc.

These methods raise the training overhead due to the increase in the amount of data. And the presence of some data in unlabeled data is harmful to the model. In this paper, our strategy to solve the overfitting problem under the condition of scarce labeled data is to introduce more regularization methods. We propose a consistent regularization method for end-to-end model frameworks that can be used for supervised learning under few-shot conditions called double consistency regularization (DOCR). DOCR encourages the model to produce high-confidence and high-consistency feature extraction and prediction outputs for inputs with similar distances in the feature space. In particular, data points with different labels are separated in low-density regions based on the smoothing assumption and the clustering assumption, and comparable data points have similar outputs. Therefore, if the input of the model is slightly perturbed, then the encoder output and decoder prediction results should not change significantly, and the output should be consistent, thereby constraining the model. In contrast to the previous work that only imposed constraints on the decoder's output space, we impose double constraints on the solution space of the model by measuring the consistency between the features extracted by the two model encoders and the predicted probabilities of the decoder, and then we realize the regularization model's internal parameters.

We conduct tests on small-sample machine translation tasks to test the efficacy of our method, and the results demonstrate that DOCR may successfully enhance model performance, decrease overfitting, and provide complementing effects with other regularization techniques. Compared with the base Transformer model, DOCR improves the scores on the IWSLT'14 German-English and IWSLT'15 English-Vietnamese datasets by 2.51 BLEU and 1.87 BLEU points, respectively. When used in combination with other regularizations, model performance improves by about 5.00 BLEU points. In addition, we conducted experiments on the state-of-the-art kNN-KD model [17] in the field of knowledge distillation and verified that our regularization method also has a certain improvement effect on the strong baseline model.

In the field of deep learning, the low-resource problem is an urgent challenge. There exists a large amount of low-sample data in tasks such as image recognition, machine translation, and speech recognition. However, deep models perform poorly in the face of these few-sample tasks due to the need to use a large amount of supervised data for training. One of the difficulties faced is the overfitting problem. From the above experimental results, our method can effectively mitigate the model overfitting phenomenon. Therefore, when using large-scale deep learning models to train small-sample tasks, our method should be used to alleviate the model overfitting phenomenon and thus improve model performance.

Our main contributions are summarized below:

- We propose DOCR, which adopts a two-model framework and utilizes consistent regularization to alleviate the overfitting problem of large-scale models that are prone to overfitting in the face of small-sample tasks.
- Extensive experiments on machine translation tasks verify the effectiveness of our approach, and the model performance improves significantly after using DOCR, which effectively mitigates the overfitting problem by constraining the encoder and decoder.
- We conduct a large number of ablation and analytical experiments to further analyze the reasons for the improvement of the model.

2. Method

Our method introduces the mean model based on cross-entropy and constrains the model solution space by regularizing the consistency loss between the mean model and the base model [18]. The model framework is shown in Figure 1. Algorithm 1 illustrates the

entire algorithm. The following three aspects are described in detail: model construction, the noise model, and model training.

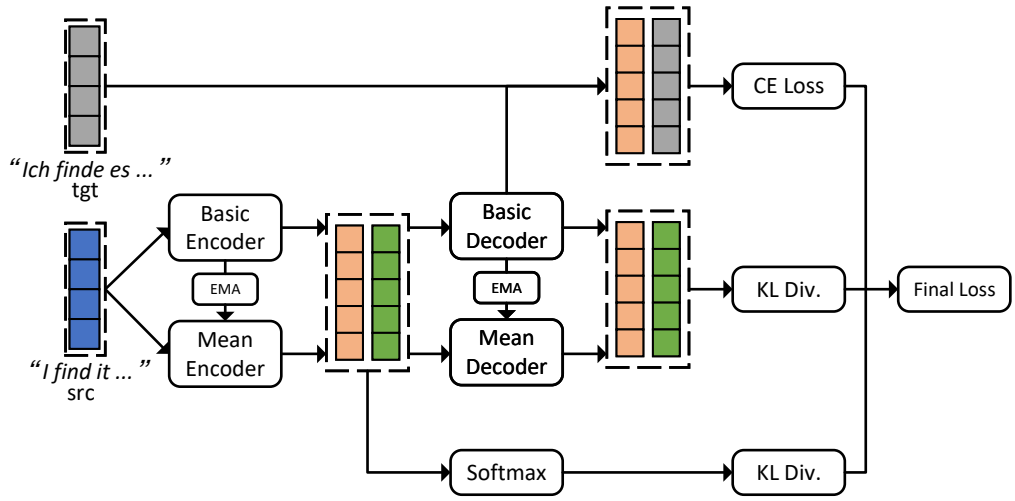


Figure 1. Illustration of the double consistency regularization. KL Div. represents KL divergence. The framework inputs the same vector into the base model and the mean model. The two models have the same structure but different parameters. The mean model is updated using the exponential moving average (EMA) of the base model parameters, while the base model is updated via parameter backpropagation.The KL divergence between the output of two model encoders and decoders is calculated to constrain the model solution space.The final training objective is the sum of the three loss values in the figure.

2.1. Model Construction

We used the basic model of the previous step to build the mean model. The structure of the two models is identical. We used the exponential moving average to set the parameter update rule of the mean model as follows :

$$\theta_m \leftarrow \lambda\theta_m + (1-\lambda)\theta_b, \tag{1}$$

where θ_b is the parameter of the base model, which is updated through backpropagation, and θ_m is the parameter of mean model. If the starting model is given too much weight, then the training effect will be quite weak due to the initial model's low performance. To balance the weight of the model parameters in the various training stages, we will therefore adhere to the cosine schedule from 0.996 to 1 during the training period [19].

2.2. Noise Model

Before the model was trained, the input statements needed to be noised using a noise model $N(x)$. The noise model contained two different types of noise. The first type of noise was words randomly discarded in the input sentences with a probability p_w. The second type of noise was a randomized arrangement of the input sentences following the condition

$$\forall i \in \{1, n\}, |\sigma(i) - i| \leq k,$$

where n is the sentence length, k is an adjustable parameter, and σ is the randomized sentence. It has been proven that $p_w = 0.1, k = 2$ are the parameter that enables the model to achieve the best performance.

2.3. Model Training

We input the given input x with different perturbations to the basic model and the mean model, denoted as x_b and x_m, respectively, the feature outputs of the encoders of the two models are denoted as $f_b(x_b)$ and $f_m(x_m)$, respectively, and the predicted outputs of

the decoder are denoted as $p_b(y_i \mid \mathbf{x}_b, \mathbf{y}_{<i})$ and $p_m(y_i \mid \mathbf{x}_m, \mathbf{y}_{<i})$. Then, $f_b(x_b)$ and $f_m(x_m)$ are transformed into probability distributions $p_b(x_b)$ and $p_m(x_m)$ by the softmax layer, respectively. For the similar input, the model encoder and decoder should have consistent outputs. Algorithm 1 illustrates the entire training process. We used the KL divergence [20] to measure the distance between the encoder output and decoder output distributions of the two models and define the consistency loss:

$$\begin{aligned}\mathcal{L}_{con} = & \sum_i D_{KL}(p_{m,i}(x_m) \| p_{b,i}(x_b)) \\ & + \sum_{y_i \in \mathcal{V}} D_{KL}(p_m(y_i \mid \mathbf{x}_m, \mathbf{y}_{<i}) \| p_b(y_i \mid \mathbf{x}_b, \mathbf{y}_{<i})).\end{aligned} \tag{2}$$

Algorithm 1: DOCR

1. **Input:** Batch data S, parallel corpus D, mean model M, base model B.
2. **for** each S in D **do**
3. Calculate the cross-entropy loss $\mathcal{L}_{ce}$.
4. Calculate the loss of consistency $\mathcal{L}_{con}$.
5. Calculate the total loss $\mathcal{L}$.
6. Update the base model B via gradient backpropagation.
7. Update the mean model $M \leftarrow \lambda M + (1 - \lambda)B$.
8. **end**

After our tests, the KL divergence was the more suitable objective function for our framework. In addition to that, we also tested using the MSE loss, MAE loss, or JS divergence as the objective function for consistency training in our experiments, the results of which are in Section 4.2.1. When the MSE loss and MAE loss were used as the objective functions for consistency training, we had

$$\mathcal{L}_{MSE} = \frac{1}{n} \sum_i^n \left\| p_{m,i}(x_m) - p_{b,i}(x_b) \right\|_2^2 + \frac{1}{n} \sum_i^n \| p_m(y_i \mid \mathbf{x}_m, \mathbf{y}_{<i}) - p_b(y_i \mid \mathbf{x}_b, \mathbf{y}_{<i}) \|_2^2, \tag{3}$$

$$\mathcal{L}_{MAE} = \frac{1}{n} \sum_i^n \left\| p_{m,i}(x_m) - p_{b,i}(x_b) \right\|_2 + \frac{1}{n} \sum_i^n \| p_m(y_i \mid \mathbf{x}_m, \mathbf{y}_{<i}) - p_b(y_i \mid \mathbf{x}_b, \mathbf{y}_{<i}) \|_2, \tag{4}$$

where n is the number of samples in each batch. When the JS divergence was used as the objective function for consistency training, we had

$$\begin{aligned}\mathcal{L}_{JS} = & \sum_i D_{JS}(p_{m,i}(x_m) \| p_{b,i}(x_b)) \\ & + \sum_{y_i \in \mathcal{V}} D_{JS}(p_m(y_i \mid \mathbf{x}_m, \mathbf{y}_{<i}) \| p_b(y_i \mid \mathbf{x}_b, \mathbf{y}_{<i})).\end{aligned} \tag{5}$$

The final model training objective is defined as

$$\mathcal{L} = \mathcal{L}_{ce} + \alpha \mathcal{L}_{con}, \tag{6}$$

where α is a weight to balance the two loss values. In order to better balance the proportion of consistency loss during training, we defined α with the function sigmoid [21] and made a simple transformation of the function to make the previous function as small as possible. Thus, the model pays more attention to the standard supervision training in the early stage of training and pays more attention to the consistency training in the later stage. Here, α is defined as

$$\alpha = \mu \frac{1}{1 + \exp(-e - \gamma/2)}, \tag{7}$$

where μ and γ are hyperparameters and e is the epoch number. Because the teacher model is greatly affected by the performance of the previous training model, the definition of α and the choice of hyperparameters greatly affected the results of the experiment. In addition, we also tested three other functions that met our requirements to define α (Figure 2): cosine [10], linear, and Equation (8). In Section 4.2, we give a comparison of their effects:

$$\alpha = \begin{cases} 0.2\mu, & \text{if } e < \gamma \\ \mu, & \text{otherwise} \end{cases} \tag{8}$$

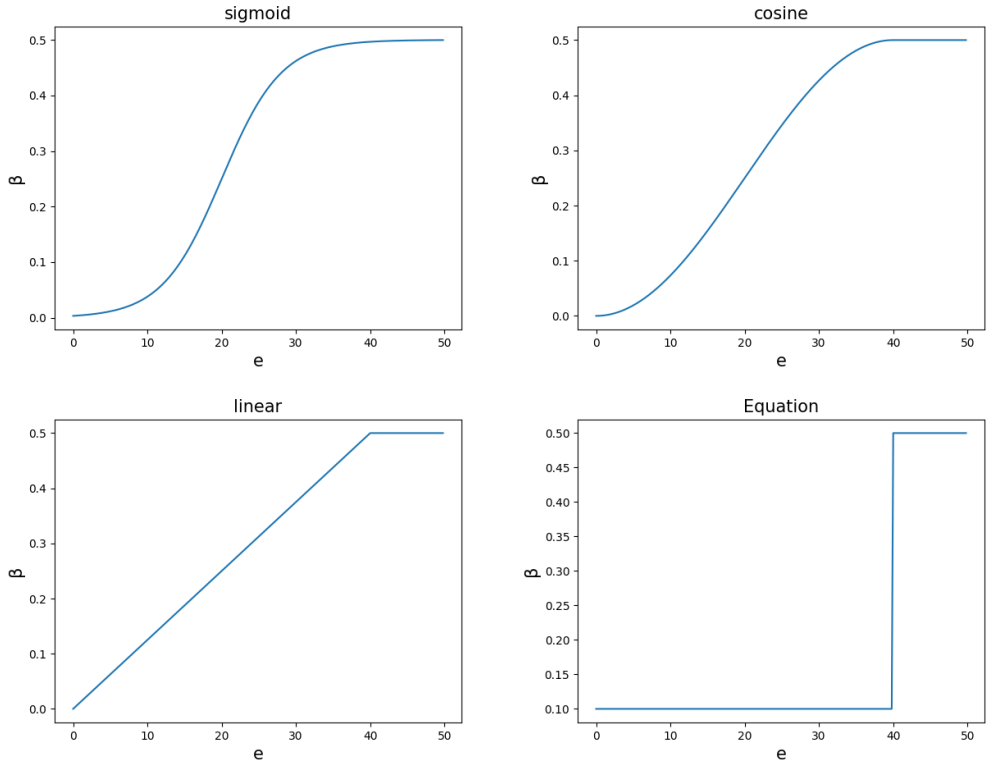


Figure 2. Illustration of the four functions: sigmoid, cosine, linear, and Equation (8) (μ = 0.5 and γ = 40).

3. Experiments

3.1. Datasets

In order to prove the effectiveness and generalization of DOCR, we conducted experiments on the machine translation small-sample task [2]. We conducted experiments on the IWSLT'14 German-English (De-En) and IWSLT'15 English-Vietnamese (En-Vi) datasets. In the experiment on the De-En dataset, the train set, vaild set, and test set sizes were 160,000, 7300, and 6500, respectively, and in the experiment on the En-Vi dataset, the train set, vaild set, and test set sizes were 133,000, 1500, and 1300 respectively.

3.2. Preprocessing and Evaluation

We used the preprocessing steps provided by fairseq [22] to split the data for the IWSLT'14 German-English dataset. The preprocessed dataset supplied by [23] was used for the IWSLT'15 English-Vietnamese dataset, with tst2012 serving as the validation set and tst2013 serving as the test set. Using the scripts that Moses provided [24], sentences were tokenized and truecased. We used BPE [25] to extract shared subword units from the combination of the source and target content. The size of the vocabulary in the IWSLT De-En dataset was 10,000, and the size of the vocabulary in the IWSLT En-Vi dataset was 17,000.

As is customary, the entire work was evaluated using the case-insensitive BLEU score [26] determined by the multi-bleu.perl script [17].

3.3. Competitive Models

Powerful and extensively utilized regularization techniques, including both classical and semi-supervised methods, served as our primary baseline. We compared DOCR to dropout [5], label smoothing [27], mean teacher [10], and unsupervised data augmentation (UDA) [16]. On the dataset used in this research, we re-experimented with these algorithms, and the settings were consistent with DOCR. We also verified the validity of DOCR on a strong baseline: kNN-KD [17].

3.4. Training Settings

Our algorithms were all implemented using the fairseq toolkit [22]. In the kNN-KD task, we followed the settings of [17] and adopted faiss [28] to replicate it. The models in the article all use the Transformer structure. The configuration we used, called transformer_iwslt_de_en, includes six encoder layers, six decoder layers, an embedding size of 512, a feedforward size of 1024, and 4 attention heads. We used a beam search with a beam size of five. The hyperparameters of μ and γ were 0.5 and 40, respectively. On four NVIDIA V100 GPUs, we trained our models using the Adam optimizer [29]. An overview of the experimental set-up is shown in Table 1. Unless otherwise stated, when comparing other algorithms, we similarly followed the hyperparameter settings in Table 1, which are the most basic settings for model training and did not affect the results.

Table 1. Hyperparameter settings.

Hyperparameters	Value
Max tokens	8192
Learning rate	5×10^{-4}
LR scheduler	Inverse sqrt
Minimal LR	1×10^{-9}
Warm-up LR	1×10^{-7}
Warm-up steps	4000
Gradient clipping	0.0
Weight decay	0.0
Optimizer	Adam
-β_1	0.9
-β_2	0.98
μ	0.5
γ	40

4. Results of the Experiments

4.1. Main Results

We begin by confirming the DOCR's efficacy and contrasting it with other widely used regularization methods, and we also show that our method showed complementary effects with other regularization methods. The experimental results are shown in Table 2.

Table 2. Experiments on IWSLT'14 German-English (De-En) and IWSLT'15 English-Vietnamese (En-Vi) datasets. "↑" indicate that higher BLEU values indicate better model performance.

Models	De-En (BLEU↑)	En-Vi (BLEU↑)
Transformer (base)	31.01	27.06
+Dropout (0.3)	33.37	29.66
+Label smoothing (0.1)	32.89	28.97
+Mean teacher	33.03	29.32
+UDA	33.56	29.65
+DOCR	33.58	29.69
+Dropout (0.3)	36.13	32.22
+Label smoothing (0.1)	35.92	32.04
+Mean teacher	36.12	32.10
+UDA	36.39	32.43

4.1.1. Comparison with Mean Teacher and UDA

We now compare DOCR with two related semi-supervised regularization methods. The training data in the mean teacher and UDA methods are a mixed dataset of bilingual data and monolingual data. The experimental results show that the model performance improved by 2.02 and 2.26 BLEU points in the two translation tasks using the mean teacher algorithm, 2.55 and 2.59 BLEU points using the UDA algorithm, and 2.57 and 2.63 BLEU points using the DOCR algorithm. It can be seen that they all improved the performance of the model, with DOCR having the best results. We believe that this was due to the fact that the data processed by mean teacher and UDA contain additional monolingual data, and some utterances in these monolingual data were harmful to the model. DOCR, on the other hand, deals only with supervised data and produces regularization effects by restricting the model output and thus constraining the model solution space. The DOCR approach did not harm the model as much as the semi-supervised regularization approach.

4.1.2. Comparison with Dropout and Label Smoothing

We also compared DOCR with the dropout and label smoothing methods, which are currently widely used regularization methods applicable to any domain. The dropout methods achieve regularization by temporarily dropping neural network units from the network with a certain probability during the training process of a deep learning network. Dropout methods are also similar to a data augmentation approach. Label smoothing generates soft labels by applying a weighted average between the uniform distribution and hard labels to transform the hard labels into soft labels for smoother network optimization.

We compared DOCR with these two methods. The experimental results show that the model performance improved by 2.36 and 2.60 BLEU points in the two translation tasks using the dropout algorithm, 1.88 and 1.91 BLEU points using the label smoothing algorithm, and 2.57 and 2.63 BLEU points using the DOCR algorithm. It can be seen that all three methods can enhance the model. Compared with dropout and label smoothing, the process of regularizing the solution space can improve the performance of the model better or equally. This illustrates that regularizing the model solution space imposes more constraints on the model than adding noise. Therefore, it is proven that DOCR has some regularization effect.

4.1.3. Integration with Other Regularization Techniques

In addition to this, we further conducted experiments to verify that DOCR is complementary to other regularization techniques. We used DOCR together with other regularization techniques, and the results are shown in Table 2. The experimental results show that the model performance was greatly improved by using the dropout algorithm, label smoothing algorithm, mean teacher algorithm, and UDA algorithm on the basis of the DOCR algorithm. It can be seen that combining DOCR with other regularization techniques can improve the model performance to a greater or lesser extent, which suggests that the method of constraining the model solution space is complementary to the current popular regularization methods. The model performance was stronger when multiple regularization methods were used simultaneously, and no single method could completely solve the overfitting problem.

4.1.4. Results for the Base Model and kNN-KD

DOCR was used to solve the problem that the model is prone to overfitting in the bilingual corpus-constrained situation, and kNN-KD is a nonparametric knowledge distillation method proposed in [17] in 2022 which can significantly improve the model performance without additional data, and it is highly suitable for the bilingual corpus-constrained situation, which is in line with the application scenario of our method. The current state-of-the-art (SOTA) model for supervised training in machine translation is kNN-KD. Therefore, in addition to the basic Transformer model, we further tested the

effectiveness of our method on kNN-KD, which is the best-performing supervised training algorithm at present.

We performed experiments on the basic Transformer model as well as on the strongly baseline kNN-KD model. Following the literature [17], we used dropout (0.3) and label smoothing (0.1) when conducting experiments on the kNN-KD model. The experiment results are shown in Table 3. In the table, kNN-MT [2] is the baseline model of the kNN-KD method (i.e., the kNN-KD method was improved with the kNN-MT method). The most primitive Transformer model was highly improved after using the DOCR method; kNN-KD-supervised training of the SOTA model in the field of machine translation improved the BLEU scores by 0.13 and 0.19 compared with the once SOTA model kNN-MT for both translation tasks. Our method improved the BLEU scores by 0.49 and 0.55 over kNN-KD, which is a relatively successful improvement.

Table 3. Results for the base model and kNN-KD. "↑" indicate that higher BLEU values indicate better model performance.

Models	De-En (BLEU↑)	En-Vi (BLEU↑)
Transformer (base)	31.01	27.06
+DOCR	33.58	29.69
kNN-MT	36.17	32.08
kNN-KD	36.30	32.27
+Mean teacher	36.42	32.41
+UDA	36.76	32.80
+DOCR	36.79	32.82

In addition to this, we conducted experiments to evaluate the impact that other consistency regularization methods have on kNN-KD. We tested the change in model performance after using the mean teacher and UDA consistency regularization methods on top of kNN-KD. The experimental results show that in the two translation tasks, the model performance improved by 0.12 and 0.14 BLEU points using the mean teacher algorithm, 0.46 and 0.53 BLEU points using the UDA algorithm, and 0.49 and 0.54 BLEU points using the DOCR algorithm. It can be seen that our method performed equally as well as the best among the three consistency regularization methods, which suggests that constraining the encoder and decoder can be more effective in improving the model performance and mitigating the overfitting phenomenon.

4.2. Ablation Studies

In this section, dropout and label smoothing are used in the experiments.

4.2.1. Effect of Loss Used to Calculate the Consistency

Referring to the work of [10,18,30], we further investigated the performance of the model when using the MAE loss, MSE loss, JS divergence, and KL divergence as the consistency loss. We conducted experiments on the De-En dataset. The results are shown in Table 4, and they show that DOCR performed best when using the KL divergence as the consistency loss while fixing our hyperparameters.

Table 4. BLEU scores of model when using different loss values as consistency loss.

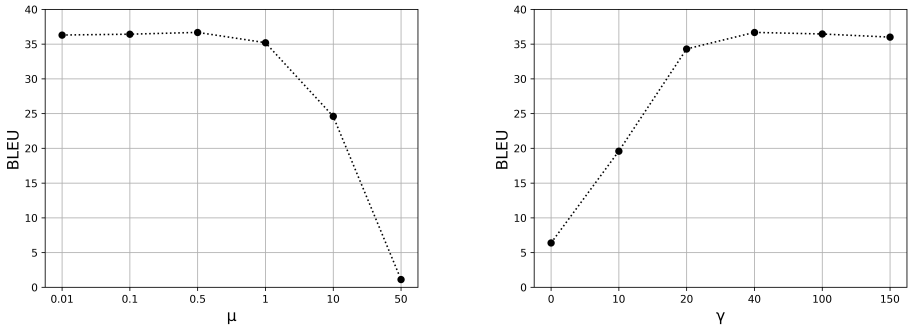
Loss	De-En
MAE Loss	35.65
MSE Loss	35.84
JS Divergence	36.36
KL Divergence	36.41

4.2.2. Effect of the Hyperparameters

The hyperparameters α of DOCR determine the proportion of consistency loss in the total training objective. We analyzed its definition and its hyperparameters μ and γ. In terms of their definition, we fixed the hyperparameters μ and γ to be 0.5 and 40, respectively, and investigated four functions: sigmoid, cosine, linear, and Equation (8). As shown in Table 5, the effect of using the sigmoid function was obviously better than those of the other functions, and it was more suitable for the teacher model update rules in this experiment. In terms of hyperparameter selection, we studied the two parameters separately. We fixed μ to 0.5 to discuss how to choose γ and γ and 40 to discuss how to choose μ. As shown in Figure 3, the results show that the BLEU score first increased with the increase in μ and γ and reached the maximum when μ was 0.5 and γ was 40. And then, with the increase in the parameters, the performance decreased, and the phenomenon of non-convergence occurred. This shows that the model can achieve its best performance when μ is 0.5 and γ is 40.

Table 5. BLEU scores when using different functions as weight β.

Functions	BLEU
Eq	35.90
linear	35.88
cosine	36.29
sigmoid	36.41



(**a**) (**b**)

Figure 3. BLEU scores with different μ and different γ on the validation set of IWSLT'14 De-En dataset. (**a**) BLEU scores with different μ and fixed γ (γ = 40). (**b**) BLEU scores with different γ and fixed μ (μ = 0.5).

4.2.3. Effect on Different Parts

In this section, we examine how consistency regularization affects various model components. We applied consistency regularization to the encoder, decoder, and the entire model. "Encoder" indicates that we only trained the encoder for consistency, and the encoder did not make any changes compared with standard end-to-end training, and "Decoder" and "All" are similarly defined. The results are shown in Table 6, where it is clear that the strong regularization effect made the proposed method more effective when applied to the whole model. In addition to this, there was some improvement in the model performance when consistent regularization was applied to either the encoder or decoder alone. Each component in DOCR had a positive effect on improving the performance of the model, and the removal of any of the components negatively affected the model.

Table 6. BLEU scores of models when using consistency regularization on different parts.

Models	Loss		
	Encoder	**Decoder**	**All**
Transformer (base)	35.52	35.87	36.41
kNN-KD	36.41	36.56	36.79

5. Analysis

5.1. Using Logit Is Better Than Using Features

We considered using the features output by the last layer of the encoder and decoder instead of the probability distribution (logit) as the item for calculating the consistency loss [30], which can regularize the parameters of the model more finely. The consistency loss is defined as

$$\begin{aligned}\mathcal{L}_{con} = & \sum_{i} D_{KL}(f_{m,i}(x_m) \| f_{b,i}(x_b)) \\ & + \sum_{y_i \in \mathcal{V}} D_{KL}(f_m(y_i \mid \mathbf{x}_m, \mathbf{y}_{<i}) \| f_b(y_i \mid \mathbf{x}_b, \mathbf{y}_{<i})),\end{aligned} \tag{9}$$

where $f_b(y_i \mid \mathbf{x}_b, \mathbf{y}_{<i})$ and $f_m(y_i \mid \mathbf{x}_m, \mathbf{y}_{<i})$ are the outputs of the last layer of the decoder.

We conducted the same experiment as in Section 4.1, and the results are shown in Table 7. For both datasets, feature-based consistent regularization gave poorer results compared with probability distribution-based consistent regularization. This is due to the coarse-grained nature of the text, and too much regularization will harm the NMT model, according to our analysis.

Table 7. BLEU scores of models when using features as terms for computing consistency loss.

Models	**De-En**	**En-Vi**
Transformer (base)	31.01	27.06
+double consistency	36.41	32.48
+double feature consistency	35.65	31.83

5.2. Hallucinations

The model's attention mechanism might not accurately reflect the model's actual attention. The authors of [31] proposed the concept of hallucinations to further understand the NMT model. If modest input changes cause rapid changes in the output, then the model is hallucinating and is not really paying attention to the input. In order to verify that the model is more robust, we followed the algorithm in [32], used the 50 and 100 most common subwords as perturbations, and tested the model's performance under input perturbations.

Table 8 shows the number of hallucinations of the model on the De-En test set in the baseline and DOCR. In DOCR training, tests were performed using both the basic and mean models. The number of hallucinations dropped on average by 30% in the basic model and 40% in the mean model compared with the supervised MT. This indicates that the model in our approach is more robust to interference and more focused on the input content. The results show that there were fewer hallucinations in the mean model than in the basic model, proving that it is a more stable model overall. This further confirms the implementability of the internal logic of our method.

Table 8. Number of distinct sentences which caused hallucinations in the baseline and MFSD models.

Models	Hallucinations	
	50 Subwords	**100 Subwords**
Transformer	24	47
Basic Model	16	33
Mean Model	13	29

5.3. Computation Overhead

Our approach employs a teacher-student framework to further strengthen the machine translation model, which increases the training complexity to some extent. On the IWSLT De-En standard dataset, we compared the computational overhead of the DOCR algorithm with the comparison algorithm. The experiments were run on four NVIDIA V100 GPUs. The comparison models included a supervised model without any algorithm, a model obtained by training with the mean teacher algorithm, and a model obtained by training with the UDA algorithm. The experimental results show that it took a total of 20 h to train the model using the simple supervised training method. The training of the model using the mean teacher algorithm took 26 h, which was 30% more than the supervised training took. The model using the UDA algorithm took 27 h to train, which was 35% more than supervised training. Our method increased the training time from 20 h to 24 h, which was 20% more than supervised training. Our method is more computationally expensive than supervised training because DOCR uses two models and needs to generate two generations of models. However, our method takes less time and improves performance more significantly than other consistency regularization methods. This is due to the fact that on top of the teacher-student framework, the mean teacher and UDA algorithms perform consistency regularization by adding monolingual data, which undoubtedly increases the amount of computation by a very large amount. DOCR does not increase the amount of data by consistency regularizing the outputs of the model encoder and decoder.

6. Conclusions

In this paper, we proposed a dual-consistency regularization method to address the problem of model overfitting under small-sample conditions so as to make more effective use of the few labeled data in low-resource situations. Based on the dual-model training framework, consistency regularization constraints were applied to the encoder and decoder parts separately, resulting in a robust improvement in the model performance. By conducting experiments on standard low-sample machine translation tasks and comparisons with classical regularization methods, it was demonstrated that this method can effectively improve the model performance and can complement other regularization methods to effectively alleviate the overfitting phenomenon and improve the model generalization ability. The analysis experiments further validate that our method can make the model more focused on the inputs, and our method can produce better results with less training overhead than other regularization methods.

Author Contributions: Conceptualization, Y.W. and W.Z.; methodology, Y.W. and Z.L.; investigation, Y.W., W.Z. and Z.L.; writing—original draft preparation, Y.W.; writing—review and editing, W.Z.; supervision, W.Z. and Z.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China under Grant No. 62171470, the Natural Science Foundation of Henan Province of China (No. 232300421240), and the Zhongyuan Science and Technology Innovation Leading Talent Project of Henan Province of China (No. 234200510019).

Data Availability Statement: Not applicable.

Acknowledgments: We appreciate the anonymous reviewers for their helpful comments.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to Sequence Learning with Neural Networks. In Proceedings of the NIPS, Montreal, QC, Canada, 12–13 December 2014; pp. 3104–3112.
2. Khandelwal, U.; Fan, A.; Jurafsky, D.; Zettlemoyer, L.; Lewis, M. Nearest Neighbor Machine Translation. *arXiv* **2020**, arXiv:2010.00710.
3. Chen, Y.; Gan, Z.; Cheng, Y.; Liu, J.; Liu, J. Distilling Knowledge Learned in BERT for Text Generation. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, 5–10 July 2020; pp. 7893–7905.

4. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All you Need. In Proceedings of the NIPS, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
5. Srivastava, N.; Hinton, G.E.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958. [CrossRef]
6. Zhang, H.; Qu, D.; Shao, K.; Yang, X. DropDim: A Regularization Method for Transformer Networks. *IEEE Signal Process. Lett.* **2022**, *29*, 474–478. [CrossRef]
7. Loshchilov, I.; Hutter, F. Decoupled Weight Decay Regularization. In Proceedings of the 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, 6–9 May 2019.
8. Pham, V.H.; Pham, T.M.; Nguyen, G.; Nguyen, L.H.B.; Dinh, D. Semi-supervised Neural Machine Translation with Consistency Regularization for Low-Resource Languages. *arXiv* **2023**, arXiv:2304.00557.
9. Laine, S.; Aila, T. Temporal Ensembling for Semi-Supervised Learning. In Proceedings of the 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, 24–26 April 2017.
10. Tarvainen, A.; Valpola, H. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In Proceedings of the 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, 24–26 April 2017.
11. He, K.; Fan, H.; Wu, Y.; Xie, S.; Girshick, R.B. Momentum Contrast for Unsupervised Visual Representation Learning. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, 13–19 June 2020; pp. 9726–9735. [CrossRef]
12. Ke, Z.; Wang, D.; Yan, Q.; Ren, J.S.J.; Lau, R.W.H. Dual Student: Breaking the Limits of the Teacher in Semi-Supervised Learning. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6727–6735. [CrossRef]
13. Athiwaratkun, B.; Finzi, M.; Izmailov, P.; Wilson, A.G. Improving Consistency-Based Semi-Supervised Learning with Weight Averaging. *arXiv* **2018**, arXiv:1806.05594.
14. Miyato, T.; Maeda, S.I.; Koyama, M.; Ishii, S. Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *41*, 1979–1993. [CrossRef] [PubMed]
15. Verma, V.; Lamb, A.; Kannala, J.; Bengio, Y.; Lopez-Paz, D. Interpolation Consistency Training for Semi-Supervised Learning. *Neural Netw.* **2019**, *145*, 90–106. [CrossRef] [PubMed]
16. Xie, Q.; Dai, Z.; Hovy, E.H.; Luong, M.T.; Le, Q.V. Unsupervised Data Augmentation for Consistency Training. *arXiv* **2019**, arXiv:1904.12848.
17. Yang, Z.; Sun, R.; Wan, X. Nearest Neighbor Knowledge Distillation for Neural Machine Translation. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, USA, 10–15 July 2022; pp. 5546–5556. [CrossRef]
18. Caron, M.; Touvron, H.; Misra, I.; Jégou, H.; Mairal, J.; Bojanowski, P.; Joulin, A. Emerging Properties in Self-Supervised Vision Transformers. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, 10–17 October 2021; pp. 9630–9640. [CrossRef]
19. Grill, J.B.; Strub, F.; Altch'e, F.; Tallec, C.; Richemond, P.H.; Buchatskaya, E.; Doersch, C.; Pires, B.Á.; Guo, Z.D.; Azar, M.G.; et al. Bootstrap Your Own Latent: A New Approach to Self-Supervised Learning. *arXiv* **2020**, arXiv:2006.07733.
20. Dedeo, S. "Wrong side of the tracks": Big Data and Protected Categories. *arXiv* **2014**, arXiv:1412.4643.
21. Zhang, W.; Feng, Y.; Meng, F.; You, D.; Liu, Q. Bridging the Gap between Training and Inference for Neural Machine Translation. In Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, 28 July–2 August 2019; pp. 4334–4343. [CrossRef]
22. Ott, M.; Edunov, S.; Baevski, A.; Fan, A.; Gross, S.; Ng, N.; Grangier, D.; Auli, M. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019; pp. 48–53. [CrossRef]
23. Luong, M.; Manning, C.D. Stanford neural machine translation systems for spoken language domains. In Proceedings of the 12th International Workshop on Spoken Language Translation: Evaluation Campaign@IWSLT 2015, Da Nang, Vietnam, 3–4 December 2015.
24. Koehn, P.; Hoang, H.; Birch, A.; Callison-Burch, C.; Federico, M.; Bertoldi, N.; Cowan, B.; Shen, W.; Moran, C.; Zens, R.; et al. Moses: Open Source Toolkit for Statistical Machine Translation. In Proceedings of the ACL 2007, 45th Annual Meeting of the Association for Computational Linguistics, Prague, Czech Republic, 23–30 June 2007.
25. Sennrich, R.; Haddow, B.; Birch, A. Neural Machine Translation of Rare Words with Subword Units. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, Berlin, Germany, 7–12 August 2016. [CrossRef]
26. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W. Bleu: A Method for Automatic Evaluation of Machine Translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, ACL 2002, Philadelphia, PA, USA, 6–12 July 2002; pp. 311–318. [CrossRef]
27. Müller, R.; Kornblith, S.; Hinton, G.E. When Does Label Smoothing Help? In Proceedings of the Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019.
28. Johnson, J.; Douze, M.; Jégou, H. Billion-Scale Similarity Search with GPUs. *IEEE Trans. Big Data* **2021**, *7*, 535–547. [CrossRef]
29. Loshchilov, I.; Hutter, F. Fixing Weight Decay Regularization in Adam. *arXiv* **2017**, arXiv:1711.05101.

30. Wei, Y.; Hu, H.; Xie, Z.; Zhang, Z.; Cao, Y.; Bao, J.; Chen, D.; Guo, B. Contrastive Learning Rivals Masked Image Modeling in Fine-tuning via Feature Distillation. *arXiv* **2022**, arXiv:2205.14141.
31. Lee, K.; Firat, O.; Agarwal, A.; Fannjiang, C.; Sussillo, D. Hallucinations in Neural Machine Translation. 2018. Available online: https://openreview.net/forum?id=SkxJ-309FQ (accessed on 15 October 2023).
32. Raunak, V.; Menezes, A.; Junczys-Dowmunt, M. The Curious Case of Hallucinations in Neural Machine Translation. *arXiv* **2021**, arXiv:2104.06683.