DM-CT: CONSISTENCY TRAINING WITH DATA AND MODEL PERTURBATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Consistency training has been widely adopted and shown great promise in deep learning. The common approach of consistency training is performed on the data-level, which typically utilizes the data augmentation strategy (or adversarial training) to make the predictions from the augmented input and the original input to be consistent, so that the model is more robust and attains better generalization ability. Recently, consistency training is also incorporated from the model-level, in which the randomness existed in the model (e.g., dropout) is constrained during the training stage, and the inference model can be more consistent with the training phase. In this work, we investigate these two aspects and propose an integrated framework, DM-CT, that incorporates both the data-level and model-level consistency training. Concretely, the input data is first augmented, and the output distributions of different sub models generated by model variance are forced to be consistent (model-level). Meanwhile, the predictions of the original input and the augmented one are constrained to be consistent (data-level). We study different data augmentation strategies and model variances in the DM-CT framework. Experiments on different tasks, including neural machine translation (4 IWSLT14 translation tasks, multilingual translation task, and WMT16 Romanian→English translation), natural language understand (GLUE benchmark), and image classification (CIFAR-100 dataset), well demonstrate the superiority of DM-CT by obtaining significant and consistent performance improvements.

1 INTRODUCTION

Deep learning has shown its disruptive success in different fields, such as image processing (He et al., 2016; Dosovitskiy et al., 2020), language processing (Bahdanau et al., 2014; Vaswani et al., 2017), and speech processing (Oord et al., 2016; Wang et al., 2017). Along the development, consistency training (CT) (Bachman et al., 2014) contributes a lot to the success. In semi-supervised learning, CT is becoming the dominant framework for leveraging unlabeled data under the cluster assumption (Laine & Aila, 2016; Verma et al., 2019; Xie et al., 2020). It simply regularizes model predictions to be invariant to small noise applied to either input (e.g., data augmentation) (Clark et al., 2018) or hidden states (e.g., adversarial training) (Miyato et al., 2018), so that the decision boundary lies in low density regions (Ouali et al., 2020). CT also becomes increasingly popular in supervised learning with labeled data (Jiang et al., 2020; Aghajanyan et al., 2020; Qu et al., 2021). For example, Shen et al. (2020) introduce a data augmentation strategy *cutoff* and leverage a Jensen-Shannon Divergence consistency loss to incorporate the augmented samples into training. Above-mentioned methods all work on the input data modification to perform CT, hence can be termed as *data-level CT*.

Recently, *model-level CT* is also incorporated in deep model training. In particular, it is motivated from the randomness/variance existed in the model training stage, and the objective is to make the sub models be more consistent and less affected by the randomness during inference. Take dropout (Srivastava et al., 2014) as an example, though it is effective, there is a gap between training and inference for a model with dropout. The training stage is performed on the sub model structure (caused by dropped units) while the inference is conducted on a single full model (without dropout). In view of this, several works (Ma et al., 2016; Zolna et al., 2018; Liang et al., 2021) have investigated this problem and incorporated the principle of CT to reduce the gap by explicitly constraining the sub model and full model to be consistent. Specially, R-Drop Liang et al. (2021) achieves state-of-the-art (SOTA) results in many tasks and datasets, which greatly demonstrates the value of model-level CT.

In this paper, to furthest excavate the power of CT, we propose to integrate the **D**ata-level and **M**odel-level **C**onsistency **T**raining into a same framework named **DM-CT**[1], which is not investigated before. In DM-CT, each input sample is first processed with a data augmentation strategy. Then both the original data and the augmented one will go through the model forward pass twice. Due to the randomness existed in the model training phase (e.g., dropout), each forward pass is actually conducted through one sampled sub model. Therefore, the two outputs from the two forward passes (two sub models) are different. Next, we perform the model-level CT between the two outputs for each data. Furthermore, the outputs of the original input sample and the augmented sample are also constrained by a data-level CT. The CT forces the distributions outputted by the two sub models or the two data samples to be consistent with each other, through minimizing the bidirectional Kullback-Leibler (KL) divergence between the two distributions. In such a way, the single full model can achieve better performance by reducing the training/inference model gap (model-level CT) and be more robust with improved generalization ability (data-level CT).

We evaluate our DM-CT framework in various areas, including neural machine translation, natural language understanding, and image classification, from both natural language processing and computer vision. Specifically, the experiments are conducted on 4 small-scale IWSLT14 translation datasets, the larger WMT16 Romanian→English translation dataset, the low-resource multilingual translation, and we achieve significant BLEU improvement, for example, about 0.76 BLEU score gain over strong R-Drop on IWSLT14 German→English translation. On the GLUE benchmark and the CIFAR-100 image classification dataset, DM-CT also yields consistent performance improvement over the strong baseline models.

The contributions of this paper can be summarized as follows:

- We first incorporate the data-level and model-level consistency training and propose the DM-CT framework to improve the model robustness and generalizations.
- We comprehensively study the different aspects in both data-level and model-level consistency training to show the effects of these variants.
- We demonstrate the effectiveness of our DM-CT with strong performance improvements on both natural language processing and computer vision tasks.

2 APPROACH

In this section, we introduce our DM-CT framework and the training algorithm. To have a clear understanding before presenting the details, we first give the required notations. Given a paired training dataset $D = \{(x_i, y_i)\}_{i=1}^{n}$, n is the number of training data samples, x_i is the input data, and y_i is the corresponding label. For each data sample x_i, the corresponding augmented input will be denoted as x_i'. The training model is $\mathcal{M}$ and the goal is to learn the probability distribution $P_{\mathcal{M}}(y|x)$. Then, for one sub model $\mathcal{M}_s$ sampled from the full model $\mathcal{M}$, the corresponding distribution is denoted as $P_{\mathcal{M}_s}(y|x)$, shortened as $P_s(y|x)$. For two probability distributions $P_1(y|x)$ and $P_2(y|x)$, their averaged distribution is denoted by $\tilde{P}(y|x)$, and the difference of these two distributions is measured by Kullback-Leibler (KL) divergence, which is $D_{KL}(P_1||P_2)$.

2.1 PRELIMINARY

As DM-CT is based on consistency training (CT), we first introduce the CT methodology, specifically on the data-level and the model-level CT, then our integrated DM-CT framework.

Data-level CT As mentioned before, consistency training (CT), which regularizes the model predictions to be consistent to improve the model robustness and generalization ability, is becoming the dominant regularization in deep learning. For the data-level CT, it is required to make the model predictions to be invariant to the small perturbation applied to the data sample, either from the input representation (Sajjadi et al., 2016; Clark et al., 2018) or the hidden states (Miyato et al., 2018). The common approaches for the perturbation is to add noises, such as data augmentation (Shorten

[1]In this paper, we mainly focus on the supervised learning scenario, but DM-CT can be easily extended to semi-supervised learning.

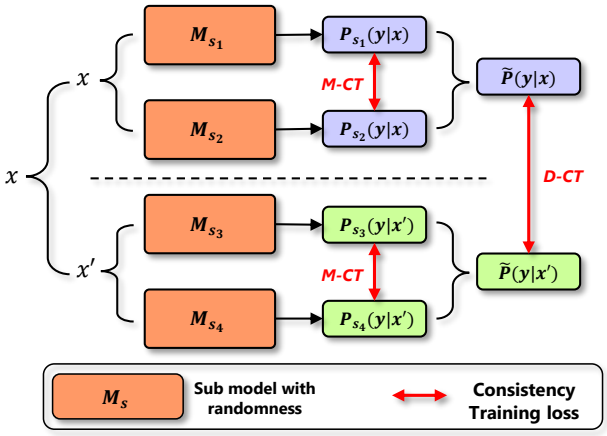

Figure 1: The overall framework of DM-CT. x and x_i are the original data and the noised (augmented) version. Each $\mathcal{M}_{s_i}$ represents one sampled sub-model. 'M-CT' is the model-level consistency training loss, and 'D-CT' is the data-level consistency training loss.

& Khoshgoftaar, 2019), Guassian noise (McHutchon & Rasmussen, 2011), and adversarial training (Ravì et al., 2019). For example, Xie et al. (2020) apply RandAugment (Cubuk et al., 2019b) strategy on the unlabeled data in semi-supervised learning, where the model predictions of augmented sample and the original input is regularized to be consistent, thus improving the generalization ability of the model. Mathematically, for one data sample x and its noised version x', the data-level CT is to constrain the predictions $P(y|x)$ and $P(y|x')$ to be consistent, measured by the bidirectional Kullback-Leibler (KL) divergence, and the training objective is to minimize the following loss[2]:

$$\mathcal{L}_{\mathcal{D}-\mathcal{CT}} = \frac{1}{2}[D_{KL}(P(y|x)||P(y|x')) + D_{KL}(P(y|x')||P(y|x))]. \tag{1}$$

Model-level CT Apart from the data-level CT, the model-level CT, which is inspired from the intrinsic randomness existed in the model training, is also drawing attention recently. Specifically, the model-level CT investigates the training of the sub models generated by some methods, such as Dropout (Srivastava et al., 2014), Stochastic Depth (Huang et al., 2016), LayerDrop (Fan et al., 2019). These regularization methods randomly sample different sub models during each forward pass. However, during inference, the single full model is utilized for prediction, which indeed exists a gap between the training and inference stages. Besides, the output probability distributions for one data sample from these sub models are also distinct. Concretely, for dropout (Srivastava et al., 2014), it randomly drops some parts of the units in each layer of the neural model, then the model parameters connected to these dropped units will not participate in the training (no gradient backward), which leads to a sub model of the full model. For each forward pass, the sub model is different due to the random dropping. When inference, however, there is no dropout applied to the model, it approximates the combination of an exponential number of sub models. Therefore, the model-level CT is to make the sub models to be more consistent with the full model so that the training and inference gap is reduced (Ma et al., 2016), and the model performance will be less affected by the sub model randomness. It is further demonstrated that making the sub models to be consistent with each other is a better choice (Zolna et al., 2018; Liang et al., 2021). In formulation, for two sub models $\mathcal{M}_{s_1}$ and $\mathcal{M}_{s_2}$ sampled from $\mathcal{M}$, the two model predictions are $P_{s_1}(y|x)$ and $P_{s_2}(y|x)$ for data pair (x, y), the model-level CT constrains them to be consistent via minimizing the following bidirectional KL-divergence training objective:

$$\mathcal{L}_{M-CT} = \frac{1}{2}[D_{KL}(P_{s_1}(y|x)||P_{s_2}(y|x)) + D_{KL}(P_{s_2}(y|x)||P_{s_1}(y|x))]. \tag{2}$$

[2]Note that during implementation, the data-level consistency between an original data x and the augmented data x' is usually conducted on NLP tasks (Jiang et al., 2020; Qu et al., 2021), while for CV tasks, data-level CT is between two independently augmented samples x_1 and x_2 both from x (Xie et al., 2020).

2.2 DM-CT

As introduced earlier, the regularization effect of above CT approaches is from different views: the data-level CT regularizes the model to be consistent to data variants so that the model is more robust, and the model-level CT regularizes the training-inference to be consistent from the sub-model level. Therefore, it is interesting to investigate both in the same framework to see whether the potential of CT can be further exploited. Towards this goal, we present a unified framework, DM-CT, that integrates the data-level and model-level CT with specific designs. The overall architecture of DM-CT is shown in Figure 1, the details are introduced below.

Given the training data $D = \{(x_i, y_i)\}_{i=1}^{n}$, for each specific data pair (x_i, y_i) at each training step, we first apply some noising method on x_i to obtain the noised input x_i'. In this paper, we adopt several data augmentation strategies and study the effect of these different choices. For NLP tasks, we study the simple data operations (e.g., word drop, word replacement). For CV tasks, we work on geometric transformations (e.g., rotation, cropping, flipping), RandAugment (Cubuk et al., 2019b) and AugMix (Hendrycks et al., 2020). We also design a simple hybrid strategy where one augmentation operation is randomly sampled from these candidates, referred to as *random-pick*. Then both the original input x_i and the augmented one x_i' will go through the forward pass of the model to output predictions. Due to the randomness existed in the model training, the forward training pass is on a sub model $\mathcal{M}_s$ sampled from the full model $\mathcal{M}$. Hence, we feed the data into the forward pass twice to obtain two different predictions from the two sampled sub models. The model-level CT is then conducted to constrain the two distributions from the sub models to be consistent. As for the model randomness, we here investigate the two most commonly methods used in the deep learning, Dropout (Srivastava et al., 2014) and Stochastic depth (Huang et al., 2016). As introduced, dropout randomly drops several units in each hidden layer, and stochastic depth randomly skips each layer of the model. Finally, we average the above the predictions from two sub models for both original input x_i and augmented x_i'. The data-level CT is then conducted between the two averaged predictions to improve the model generalization.

Concretely, the input x_i is fed to two different sub models $\mathcal{M}_{s_1}$ and $\mathcal{M}_{s_2}$ sampled from $\mathcal{M}$ to obtain two output predictions $P_{s_1}(y_i|x_i)$ and $P_{s_2}(y_i|x_i)$. The augmented input x_i' is executed in the same way, it will go forward two sub models $\mathcal{M}_{s_3}$ and $\mathcal{M}_{s_4}$ to output predictions $P_{s_3}(y_i|x_i')$ and $P_{s_4}(y_i|x_i')$. Then the model-level CT is applied between $P_{s_1}(y_i|x_i)$ and $P_{s_2}(y_i|x_i)$, $P_{s_3}(y_i|x_i')$ and $P_{s_4}(y_i|x_i')$, which is a bidirectional KL-divergence:

$$\begin{aligned}
\mathcal{L}_{M-CT} &= \alpha_1 \cdot \mathcal{L}_{M-CT}(x_i) + \alpha_2 \cdot \mathcal{L}_{M-CT}(x_i'), \\
\mathcal{L}_{M-CT}(x_i) &= \frac{1}{2}[D_{KL}(P_{s_1}(y_i|x_i)||P_{s_2}(y_i|x_i)) + D_{KL}(P_{s_2}(y_i|x_i)||P_{s_1}(y_i|x_i))], \\
\mathcal{L}_{M-CT}(x_i') &= \frac{1}{2}[D_{KL}(P_{s_3}(y_i|x_i')||P_{s_4}(y_i|x_i')) + D_{KL}(P_{s_4}(y_i|x_i')||P_{s_3}(y_i|x_i'))].
\end{aligned} \tag{3}$$

After that, the two predictions for each data sample is averaged, i.e, $\tilde{P}(y_i|x_i) = \frac{1}{2}(P_{s_1}(y_i|x_i) + P_{s_2}(y_i|x_i))$, $\tilde{P}(y_i|x_i') = \frac{1}{2}(P_{s_3}(y_i|x_i') + P_{s_4}(y_i|x_i'))$, and the data-level CT is utilized by the following loss:

$$\mathcal{L}_{D-CT} = \frac{1}{2}[D_{KL}(\tilde{P}(y_i|x_i)||\tilde{P}(y_i|x_i')) + D_{KL}(\tilde{P}(y_i|x_i')||\tilde{P}(y_i|x_i))]. \tag{4}$$

The main learning objective of negative log-likelihood loss on the training data is:

$$\mathcal{L}_{NLL} = -\log P_{s_1}(y_i|x_i) - \log P_{s_2}(y_i|x_i) - \log P_{s_3}(y_i|x_i') - \log P_{s_4}(y_i|x_i'), \tag{5}$$

and the final training objective is to minimize the following loss function:

$$\mathcal{L} = \mathcal{L}_{NLL} + \mathcal{L}_{DM-CT} = \mathcal{L}_{NLL} + \mathcal{L}_{M-CT} + \beta\mathcal{L}_{D-CT}, \tag{6}$$

where β, α_1 and α_2 in Eqn. (3) are the weight hyperparameters to control and balance the consistency losses. We will study the effect of these weights in the experiments.

Alternative Designs Recap the consistency training (CT), there are actually different choices that have been adopted in the previous works. For example, Zolna et al. (2018) conduct the CT on the outputted *hidden logits* before the `softmax` operation, and the loss function is the L_2 distance. Xie et al. (2020) and Shen et al. (2020) apply the CT on the output *probability distributions* (after

Algorithm 1 DM-CT Training Algorithm

Input: Training data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$.
Output: model M after training.

1: Initialize model with random parameters.
2: **while** not converged **do**
3: Randomly sample data pair $(x_i, y_i) \sim \mathcal{D}$,
4: Randomly augment the data x_i to be a noised one x_i',
5: Concatenate and duplicate the data to be $[x_i, x_i', x_i, x_i']$,
6: Forward the data once and obtain the output distributions $[P_{s_1}(y_i|x_i)$, $P_{s_3}(y_i|x_i')$, $P_{s_2}(y_i|x_i)$, $P_{s_4}(y_i|x_i')]$,
7: Calculate the negative log-likelihood loss $\mathcal{L}_{NLL}$ by Eqn. (5),
8: Calculate the model-level CT loss $\mathcal{L}_{M-CT}$ by Eqn. (3),
9: Calculate the data-level CT loss $\mathcal{L}_{D-CT}$ by Eqn. (4),
10: Update the model parameters by minimizing loss $\mathcal{L}$ of Equation (6).
11: **end while**

`softmax`) with either KL-divergence or Jensen-Shannon (JS) divergence. The various designs indeed lead to different optimization and regularization effect with different model performances. Therefore, in this paper, though we take the distribution-guided CT with KL-divergence as the main constraint, we also study the hidden-level L_2 consistency loss function.

Discussion A close look into the data-level CT and model-level CT reveals several overlaps and also differences. One may think that the data-level CT includes the model-level CT to some extent. Indeed, when the data-level CT is conducted on a model with randomness, such as dropout, it can be regarded that the data-level CT contains both data-level and model-level CT. However, for a model without randomness, such as a shallow ResNet (He et al., 2016), the data-level and model-level CT are performed on different perspectives at all. Besides, even in the former scenario, explicitly modeling the model-level CT also benefits the model generalization to improve the model performance. This can be demonstrated from our ablation study below.

2.3 Training Algorithm

The overall training algorithm for our DM-CT is shown in Algorithm 1. Note that during implementation, instead of forwarding the sub models forth, we take the same strategy as Liang et al. (2021) that the two sub models CT are conducted in a doubled batch data, and we also put the augmented data in the same batch. That is, the x_i and x_i' are first concatenated in a same batch, $[x_i, x_i']$, then they are duplicated in the same batch to be $[x_i, x_i', x_i, x_i']$ and the concatenated data only forward the model once to save the computational cost. Specifically, the process of the batched data is shown in Line 3-5, and then the data goes forward the model once to obtain the predicted distributions $[P_{s_1}(y_i|x_i), P_{s_3}(y_i|x_i'), P_{s_2}(y_i|x_i), P_{s_4}(y_i|x_i')]$ at Line 6. The negative log-likelihood and the data-level and model-level consistency losses are then calculated in Line 7-9, finally the model is updated in Line 10 according to Eqn. (6).

3 Experiments

3.1 Neural Machine Translation

We first evaluate our proposed method on neural machine translation (NMT) tasks, and we collect different scale datasets to conduct experiments and analysis. In addition, we examine whether DM-CT is useful across different language pairs for low-resource multilingual translation tasks.

Datasets For small-scale datasets, we use 4 language translation pairs from the IWSLT evaluation campaign, which belongs to rich-resource language on low-resource settings. The IWSLT datasets include English↔German (En↔De), English↔Spanish (En↔Es), and IWSLT17 English↔French (En↔Fr), English↔Chinese (En↔Zh) translations. Furthermore, we also choose for multilingual translation, which is low-resource language on low-resource settings following Li et al. (2020).

Model	En→De	De→En	En→Fr	Fr→En	En→Zh	Zh→En	En→Es	Es→En	Avg
Transformer (Vaswani et al., 2017)	28.57	34.64	35.9	36.1	26.3	18.4	39.0	40.6	32.44
R-Drop (Liang et al., 2021)	30.72	37.25	38.0	38.9	28.1	19.5	41.8	43.2	34.68
DM-CT	**30.92**	**38.01**	**38.4**	**39.4**	**28.3**	**20.6**	**41.9**	**43.6**	**35.14**

Table 1: BLEU scores on 8 IWSLT machine translation tasks.

Model	aze	bel	glg	slk	tur	rus	por	ces	Avg
Transformer (Li et al., 2020)	5.5	9.1	22.4	24.6	**15.8**	19.4	38.6	21.9	19.65
DM-CT	**5.5**	**10.1**	**23.0**	**25.4**	15.5	**19.4**	**39.2**	**22.1**	**20.02**

Table 3: BLEU scores for one-to-many multilingual translation on related languages.

That related languages are from same language family, which include four low-resource language (Azerbaijani: aze, Belarusian: bel, Glacian: glg, Slosvak: slk) and high-resource language (Turkish: tur, Russian: rus, Portuguese: por, Czech: ces). For large datasets, we choose WMT16 Romanian→English (Ro→En) translation task, which uses the amount of back-translation data to make improvements. We do not conduct experiment on WMT14 English→German translation since augmentation method utilized in our DM-CT is more preferred for relatively smaller data size.

Settings We implement all NMT models using Transformer (Vaswani et al., 2017) network with fairseq (Ott et al., 2019) toolkit[3]. We use the `transformer_iwslt_de_en` and `transformer_vaswani_wmt_en_de_big` as configurations for small-scale and large-scale datasets respectively. To make a fair comparison, we re-implement standard Transformer and R-Drop (Liang et al., 2021) as a baseline model.More details of experimental settings for each dataset can be found in Appendix A.

Results We first illustrate the BLEU score for IWSLT translation tasks in Table 1. We can see that our DM-CT achieves more than 2.7 BLEU score improvements than Vanilla Transformer and nearly 0.5 BLEU score compared to R-Drop. It is worth noting that DM-CT leverages the data consistency from augmented data pair, which demonstrates the data consistency is complementary to model consistency. Next, we evaluate the DM-CT on WMT translation tasks, as shown in Table 2. The results show that our method surpasses several strong baselines, such as BERT-fused NMT (Zhu et al., 2019) model based on large-scale pretrained model, and Fast Noisy Channel Modeling (FNCM). To the best of our knowledge, we achieve the new state-of-the-art (SOTA) BLEU score on WMT16 Ro→En (40.54). Then, we use the one-to-many multilingual translation task to verify the impact of DM-CT training approach. In Table 3 we show performance on the "Related" language setting, and observe that DM-CT obtains a strong average BLEU score 20.02, and maintains better performance on most language pairs.

Method	Ro→En
Transformer (Vaswani et al., 2017)	37.73
BERT-fused NMT (Zhu et al., 2019)	39.10
FNCM (Bhosale et al., 2020)	40.3
R-Drop (Liang et al., 2021)	39.03
DM-CT	**40.54**

Table 2: BLEU scores on WMT16 Ro→En machine translation tasks.

3.2 Natural Language Understanding

Datasets To further verify our methods with universal impacts, we validate on widely-adopted GLUE benchmark, which consists of 8 Nature Language Understanding tasks: MNLI, MRPC, QNLI, QQP, RTE, SST-2, STS-B, CoLA. The pre-trained model and the backbone implementations are all from Huggingface Transformers[4]. The more details can be found in Appendix A.

Settings Following common practice, we apply DM-CT to the Roberta-Large model, which is the strong baseline for GLUE benchmark (Wang et al., 2018). For each task, different random seeds and parameter settings are required, thus we dynamically adjust the coefficient weight for each setting. Other configurations are following the previous works (Liu et al., 2019).

[3] `https://github.com/pytorch/fairseq/tree/master/examples/translation`

[4] `https://github.com/huggingface/transformers`

Model	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	CoLA	Avg
RoBERTa-large	90.2	90.9	94.7	92.2	86.6	96.4	92.4	68.0	88.93
R-Drop	90.9	91.4	95.2	92.5	88.4	96.9	92.5	70.0	89.73
DM-CT	90.9	92.1	95.3	92.5	89.1	96.9	92.6	70.7	**90.01**

Table 4: Fine-tuned model performances on GLUE language understanding benchmark.

Results The empirical results of the GLUE benchmark are presented in Table 4. The evaluation metrics for above 8 tasks are as follows: The result for STS-B is the Pearson correlation; Matthew's correlation is used for CoLA; Other tasks are measured by Accuracy. We can find DM-CT has achieved 1.08 points improvement over the Roberta-large model, and outperforms the strong R-Drop model by nearly 0.3 points, demonstrating the effectiveness of model and data consistency regularization. Besides, we get the consistent conclusion with Qu et al. (2021) that synthetically produced examples are more valuable data-limited tasks (e.g., MRPC, RTE, CoLA, and STS-B). This further validates DM-CT obtains stronger results by leveraging augmented samples more effectively.

3.3 Image Classification

Datasets We chose the well-known CIFAR-100 dataset (Krizhevsky, 2009) to demonstrate the effectiveness of DM-CT in image classification. CIFAR-100 has $50,000/10,000$ images in the train/val split, divided into 100 categories. Following Kolesnikov et al. (2020), we train the model on the official train split and report accuracy on the validation split.

Settings For CIFAR-100, we utilize two backbone models: ViT-B/16 (Dosovitskiy et al., 2020) which is pretrained on ImageNet-21K, and Resnet110 (He et al., 2016) equipped with stochastic depth (Huang et al., 2016). Empirically we have found our ViT model does not benefit from more advanced techniques, so we simply adopt the commonly used random-crop strategy for it. On the contrary, we adopt the random-pick augmentation strategy introduced in Section 2.2 for Resnet110. In Section 4.1, we compare different augmentation strategies in detail.

Results The accuracy on CIFAR-100 is shown in Table 5. For ViTB-B/16 model backbone, DM-CT achieves 0.33 improvement over the vanilla model, and 0.24 over R-Drop. For Resnet110 model backbone, the improvement is more remarkable: 3.26 over the vanilla model and 2.4 over R-Drop. This demonstrates the synergy between imposing data-level consistency and model-level consistency when DM-CT is applied to vision tasks, under both pretraining-finetuing and training-from-scratch scenarios.

Method	CIFAR-100
ViT-B/16 (Dosovitskiy et al., 2020)	93.12
+R-Drop	93.21
+DM-CT	**93.45**
Resnet110 (Huang et al., 2016)	76.07
+R-Drop	76.93
+DM-CT	**79.33**

Table 5: Classification accuracy on CIFAR-100.

3.4 Ablation

In this section, we analyze the effect of different data augmentation methods and two consistency regularization terms (model-level CT and data-level CT) in our proposed DM-CT, and try to understand their contribution to the performance gain.

Regularization Losses We first explore the improvements of regularization loss both NLP and CV, the results are presented in Table 6. For NLP tasks, we choose the IWSLT14 De→En translation tasks and adopt the Transformer (Vaswani et al., 2017) as baseline model. The data augmentation strategy (w/o CT) achieves 0.6 points improvement over the baseline model. Along with the data-level consistency training strategy (w/o model-level CT)

Method	IWSLT14 De→En	CIFAR-100
Baseline	34.64	76.07
DM-CT	**38.01**	**79.33**
w/o data-level CT	37.67	78.37
w/o model-level CT	37.54	78.67
w/o CT	35.26	78.01

Table 6: Comparison among different consistency regularization objectives.

and Model-level consistency training strategy (w/o data-level CT) introduced, our model exhibits significant gains from 35.26 to 37.54 and 37.67 respectively. Furthermore, we can obtain superior performance of 38.01 BLEU scores by integrating two different consistency strategies. For CV tasks, we run the study with Resnet110 as the backbone network on CIFAR-100 image classification tasks. As we can see, augmentation is vital to this task, and the designed random-pick augmentation strategy alone improves the model by nearly two points. Adding the model-level consistency loss term further improves the model by forcing each sub-model sampled by stochastic depth to give consistent predictions for each input. Remarkably, imposing data-level consistency pushes the model to 78.67% accuracy, implying it is essential to constrain the model to be robust to input perturbations explicitly. These observations demonstrate that both the model-level and data-level regularization terms are indispensable for NLP and CV tasks.

4 STUDY

To give a thorough understanding of our methods, we conduct several detailed analyses and discussions in this section. More studies can be found in Appendix B.

4.1 AUGMENTATION METHODS

In this section, we study the impacts of different data augmentation strategies on both NLP and CV tasks. For IWSLT-14 De→En machine translation task, we try out word drop, word replacement, and random-pick based on these two operations. For CIFAR-100 image classification task, we explore flip & crop, RandAugment, and AugMix, as well as randomly picking two operations from these three. As presented in Table 7, with any augmentation strategy listed here, we can observe that DM-CT surpasses R-Drop by a large margin, and random-pick augmentation dominates others on both tasks. We conjecture the reason of the fact that (1) various data augmentation methods can produce more diverse inputs, thus preventing the model from overfitting to fixed of augmentation strategy, and (2) the data-level consistency loss of DM-CT can leverage this more diversified data to make the model more robust to input perturbations.

Task	**Augmentation**	Accuracy/BLEU
CIFAR 100	Flip & Crop	77.84
	RandAugment	78.10
	AugMix	77.95
	Random-pick	**79.33**
IWSLT14 De→En	Word Drop	37.52
	Word Replacement	37.79
	Random-pick	**38.01**

Table 7: Comparison among different augmentation strategies on CIFAR-100 and IWSLT14 De→En.

4.2 LOGITS OR PROBABILITY

We first study the different ways to regularize the model-level or data-level consistency on NLP (IWSLT14 De→En) and CV (CIFAR 100) tasks. With regards to this, we have introduced two different distance functions to guarantee the model output consistency by penalizing the discrepancy, which are (1) Kullback–Leibler (KL) distance regularization for model output probability, (2) Euclidean distance (L2) regularization (Zolna et al., 2018) for model output logits. Since both model-level CT and data-level CT are vital components, we study the impacts of all combinations: *DLML* both data-level and model-level consistency are regularized by minimizing the L2 distance between output logits, *DLMP* and *DPML* replaces one of regularization strategy to optimize the KL distance of output probability, and *DPMP* is our proposed method. As shown in Figure 3, the same conclusion was obtained by two different tasks. The model performance is not improved using logits regularization based on L2 distance. This observation attributed the fact that the hidden state distance is not equality to output probability distributions distance. For this reason, we

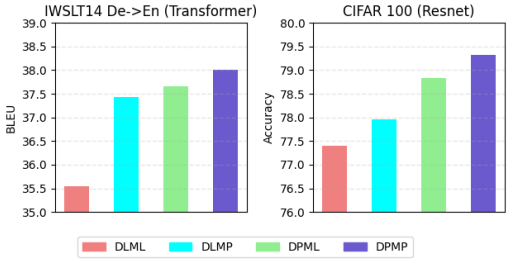

Figure 2: Comparison different regularization loss applied to different model output space (logits and probability)

almost impossible to optimize the negative log-likelihood objective functions by regularizing the L2 loss for output logits. Instead, the KL-divergence between probability distributions is more similar to loss objective functions and more suitable to guide model training. We can achieve the best empirical result by applying probability regularization for both model-level and data-level consistency. These phenomena can demonstrate the fact that probability regularization is the optimal choice.

5 Related Work

Our DM-CT approach is related to several different research fields.

Regularization Methods Our DM-CT is built upon the data (data augmentation) and model (dropout, stochastic depth) perturbations, which are indeed widely adopted regularization methods in deep learning. Here we introduce some of them. Sajjadi et al. (2016) gives a study of data augmentation with stochastic transformations and perturbations. For language tasks, common strategy includes word deletion, shuffle, replacement (Edunov et al., 2018). Advanced strategies like back-translation (Sennrich et al., 2016a), soft contextualized augmentation (Gao et al., 2019), and cutoff (Shen et al., 2020) are also proposed with strong performance. For CV, the common approach is image rotation, cropping, translation and so on (Shorten & Khoshgoftaar, 2019). Zhang et al. (2018) proposes MixUp to mix the two images, Cubuk et al. (2019b) introduces RandAugmentation that combines different methods, Cubuk et al. (2019a) instead giving an automatic augmentation learning approach. As for model randomness, dropout (Srivastava et al., 2014) is the most famous method that randomly drops several units in the model layer. LayerDrop (Fan et al., 2019) drops the entire model layer with some probability. Similarly, stochastic depth (Huang et al., 2016) skips the block in a model with stochastic operation. Instead of proposing new regularization methods, we integrate these methods in our DM-CT framework.

Data-level Consistency Training CT is popular in deep learning and has been the dominant method in semi-supervised learning. It is mostly conducted on data-level, in which the data is first noised and the consistency regularization is performed between the noised input and the original one to improve the model generalization and robustness. There are many works that augment the unlabeled data and constrain the data representation with some specific consistency loss (Laine & Aila, 2016; Tarvainen & Valpola, 2017; Verma et al., 2019; Ouali et al., 2020). For example, Xie et al. (2020) achieves super strong performance on ImageNet classification with unsupervised data augmentation (UDA). It is also extended to the labeled data and the resulted performance is outstanding (Shen et al., 2020; Jiang et al., 2020; Aghajanyan et al., 2020; Qu et al., 2021). Our work is inspired from the data-level CT but we give more studies and strategies in our work.

Model-level Consistency Training Our work also incorporates the model-level CT, which is less investigated. Model-level CT studies the intrinsic randomness in the model and regularizes the resulted sub models to be consistent so that the sub model and full model gap is reduced. Ma et al. (2016); Zolna et al. (2018) and Liang et al. (2021) investigate the inconsistency between the sub model (during training) and the full model (during inference) for models with dropout (Srivastava et al., 2014) strategy. They introduce different consistency losses between sub models (sampled from dropout mask) and full model, e.g., KL-divergence loss on the output distributions, L_2 distance on the hidden states or logits. Our DM-CT extends the model-level CT and integrate with data-level CT to greatly empower the consistency training.

6 Conclusion

In this paper, we propose DM-CT approach, a consistency training approach that incorporates the data-level and model-level CT in a unified framework. Through constraining the data to be invariant to data noises, the data-level CT improves the generalization and robustness of deep learning models. Besides, the model-level CT regularizes the sub models to be consistent so that the training and inference gap is reduced. By conducting experiments on neural machine translation, natural language understanding, and image classification, we demonstrate the effectiveness of our DM-CT approach with strong performances on each task, especially on the low-resource data task. In the future, we will continue reducing the training cost. Besides, we will apply DM-CT in more practical scenarios.

REFERENCES

Armen Aghajanyan, Akshat Shrivastava, Anchit Gupta, Naman Goyal, Luke Zettlemoyer, and Sonal Gupta. Better fine-tuning by reducing representational collapse. In *International Conference on Learning Representations*, 2020.

Philip Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles. *Advances in neural information processing systems*, 27:3365–3373, 2014.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Shruti Bhosale, Kyra Yee, Sergey Edunov, and Michael Auli. Language models not just for pre-training: Fast online neural noisy channel modeling. In *Proceedings of the Fifth Conference on Machine Translation*, pp. 584–593, 2020.

Kevin Clark, Minh-Thang Luong, Christopher D Manning, and Quoc Le. Semi-supervised sequence modeling with cross-view training. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1914–1925, 2018.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 113–123, 2019a.

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical data augmentation with no separate search. *arXiv preprint arXiv:1909.13719*, 2(4):7, 2019b.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 489–500, 2018.

Angela Fan, Edouard Grave, and Armand Joulin. Reducing transformer depth on demand with structured dropout. In *International Conference on Learning Representations*, 2019.

Fei Gao, Jinhua Zhu, Lijun Wu, Yingce Xia, Tao Qin, Xueqi Cheng, Wengang Zhou, and Tie-Yan Liu. Soft contextual data augmentation for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5539–5544, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Dan Hendrycks, Norman Mu, Ekin D. Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. AugMix: A simple data processing method to improve robustness and uncertainty. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *European conference on computer vision*, pp. 646–661. Springer, 2016.

Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. Smart: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2177–2190, 2020.

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

Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *ECCV*, 2020.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

Xian Li, Asa Cooper Stickland, Yuqing Tang, and Xiang Kong. Deep transformers with latent depth. *Advances in Neural Information Processing Systems*, 33, 2020.

Xiaobo Liang, Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, and Tie-Yan Liu. R-drop: Regularized dropout for neural networks. *arXiv preprint arXiv:2106.14448*, 2021.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Xuezhe Ma, Yingkai Gao, Zhiting Hu, Yaoliang Yu, Yuntian Deng, and Eduard Hovy. Dropout with expectation-linear regularization. *arXiv preprint arXiv:1609.08017*, 2016.

Andrew McHutchon and Carl Rasmussen. Gaussian process training with input noise. *Advances in Neural Information Processing Systems*, 24:1341–1349, 2011.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pp. 1–9, 2018.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 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 (Demonstrations)*, pp. 48–53, 2019.

Yassine Ouali, Céline Hudelot, and Myriam Tami. Semi-supervised semantic segmentation with cross-consistency training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12674–12684, 2020.

Matt Post. A call for clarity in reporting bleu scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*. Association for Computational Linguistics, 2018.

Yanru Qu, Dinghan Shen, Yelong Shen, Sandra Sajeev, Weizhu Chen, and Jiawei Han. Co{da}: Contrast-enhanced and diversity-promoting data augmentation for natural language understanding. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=Ozk9MrX1hvA`.

Daniele Ravì, Agnieszka Barbara Szczotka, Stephen P Pereira, and Tom Vercauteren. Adversarial training with cycle consistency for unsupervised super-resolution in endomicroscopy. *Medical image analysis*, 53:123–131, 2019.

Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *Advances in neural information processing systems*, 29:1163–1171, 2016.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 86–96, 2016a.

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

Dinghan Shen, Mingzhi Zheng, Yelong Shen, Yanru Qu, and Weizhu Chen. A simple but tough-to-beat data augmentation approach for natural language understanding and generation. *arXiv preprint arXiv:2009.13818*, 2020.

Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48, 2019.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in Neural Information Processing Systems*, 30, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Vikas Verma, Alex Lamb, Juho Kannala, Yoshua Bengio, and David Lopez-Paz. Interpolation consistency training for semi-supervised learning. In *International Joint Conference on Artificial Intelligence*, pp. 3635–3641, 2019.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 353–355, 2018.

Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al. Tacotron: Towards end-to-end speech synthesis. *arXiv preprint arXiv:1703.10135*, 2017.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*, 33, 2020.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tieyan Liu. Incorporating bert into neural machine translation. In *International Conference on Learning Representations*, 2019.

Konrad Zolna, Devansh Arpit, Dendi Suhubdy, and Yoshua Bengio. Fraternal dropout. In *International Conference on Learning Representations*, 2018.

	aze	bel	glg	slk	tur	rus	por	ces
train	5.9K	4.5K	10K	61.5K	182K	208K	195K	103K
valid	671	248	682	2271	4045	4814	4035	3462
test	903	664	1007	2445	5029	5483	4855	3831

Table 8: Data statistics for multilingual translation experiments.

A Detailed Experiment Settings

We provide more detailed setting for the experiments of each task in this part.

NMT The IWSLT datasets include English↔German (En↔De), English↔Spanientence pairs, $7k$ valid pairs, and $7k$ test pairs. The the WMT16 Ro→En data contains $0.6M$ bilingual data and $2M$ back translated data, valid and test data are from the corresponding newstest data. For IWSLT[5] tasks, We tokenize all the datasets with byte-pair-encoding (BPE) Sennrich et al. (2016b) approach with the dictionary built jointly upon the source and target sentence pairs. In particular, we build IWSLT17 En↔Zh translation dataset vocabulary separately due to very diverse between two language. For WMT'16 Romanian→English task, we concatenate original data and back translation data[6] to build joint vocabulary. After tokenization, the resulted vocabularies for IWSLT datasets are near $10k$, while for WMT datasets, the vocabulary size is about $32k$.

To train the Transformer based NMT models, we use `transformer_iwslt_de_en` configuration for IWSLT translations, which has 6 layers in both encoder and decoder, embedding size 512, feed-forward size $1,024$, attention heads 4, dropout value 0.3, weight decay 0.0001. For the WMT experiments, the `transformer_vaswani_wmt_en_de_big` setting has 6 layers in encoder and decoder, embedding size $1,024$, feed-forward size $4,096$, attention heads 16, dropout value 0.1, attention dropout 0.1 and relu dropout 0.1. The training is optimized with Adam Kingma & Ba (2014) with $\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 10^{-9}$. The learning rate scheduler is `inverse_sqrt` with default learning rate 0.0005 and warmup steps $4,000$. Label smoothing Szegedy et al. (2016) is adopted with value 0.1. We train the IWSLT translations on 1 GEFORCE RTX 3090 card and the WMT translations on 8 GEFORCE RTX 3090 cards.

To evaluate the performance, we use `multi-bleu.perl`[7] to evaluate IWSLT14 En↔De and all WMT tasks for a fair comparison with previous works Zhu et al. (2019); Ott et al. (2018). For other NMT tasks, we use `sacre-bleu`[8] Post (2018) for evaluation. When inference, we to use beam size 4 and length penalty 0.6 for WMT'16 En→De, beam size 5 and penalty 1.0 for other tasks. We set the hyper-parameter $\alpha_1 = \alpha_2 = \beta = 4/3$.

NLU For language understanding tasks, the Roberta-large model is employed as the testbed for our experiments, and the fine-tuned sets are the GLUE Wang et al. (2018) benchmark. That contains singe-sentence classification tasks (CoLA, SST-2), sentence-pair classification tasks (MNLI, QNLI, RTE, QQP, MRPC), and sentence-pair regression task (STS-B). The detailed data statistics can be found from the original paper Wang et al. (2018).

The pre-trained Roberta-large model contains 24 layers with embedding size $1,024$, feed-forward size $4,096$ and attention heads 16. During fine-tuning, we use Adam Kingma & Ba (2014) as our optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 10^{-6}$, and L_2 weight decay of 0.01. We select the learning rate in range $\{5 \times 10^{-6}, 10^{-5}\}$ and batch size in $\{8, 16, 32\}$. Other hyper-parameter settings are mostly same as previous works Liu et al. (2019). For each task, different random seeds and parameter settings are required, thus we dynamically adjust the coefficient α_1, α_2, and β from 0.01 to 1.0. The fine-tuning experiments are conducted on 1 GEFORCE RTX 3090 GPU card.

CIFAR-100 The weights of our ViT-B/16 model are pretrained on ImageNet-21K dataset and publicly released. We finetune the model with input resolution 224×224, while keeping the other

[5] `https://iwslt.org/`

[6] `http://data.statmt.org/rsennrich/wmt16_backtranslations/ro-en/`

[7] `https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl`

[8] `https://github.com/mjpost/sacrebleu`

hyperparameters the same as the original paper: initial learning rate 0.01 with 500 warmup steps and cosine decay; batch size 512; no weight decay; 10000 steps in total. We set $\alpha_1 = \alpha_2 = 1$ and $\beta = 0.1$.

We train the Resnet110 model from scratch. We set batch size to 1024 to fully utilize the GPU memory, and set initial learning rate to a larger value 0.8 accordingly, with 250 warmup steps and cosine decay. Weight decay is set to 1e-4. We train the model for 25000 steps in total. As to stochastic depth, we linearly increase the probability of being skipped from 0 (bottom block) to 0.5 (top block). We set $\alpha_1 = \alpha_2 = 0.3$ and $\beta = 0.5$.

B More Studies

B.1 Effect of Weight

The important hyper-parameter with DM-CT is the weight α_1, α_2 and β, which is very sensitive for different tasks, and the Proportion of them control how degrees of model and data consistency. To explore the process of hyper-parameter searching, we investigate the impact of three weight for MRPC and RTE task, which is sub task of GLUE benchmarks. The smaller tasks are more easier to benefit from data augmentation algorithms and sensitive with slight hyper-parameter changed. Specifically, we first find the weight scale for selected tasks, then assign weight by product of weight scale and the fixed proportion pattern (e.g., 1:1:1 or 1:2:7). For example, we get the weight scale 0.1 and proportion $1:3:1$ for specific task, Then, we can calculate the specific coefficient $\alpha_1 = 0.2$, $\alpha_2 = 0.6$, and $\beta = 0.4$. As shown in Figure 3, we get the best hyper-parameter that the weight scale 0.1 and the different proportion $5:3:2$ and $7:2:1$ for MRPC and RTE respectively.

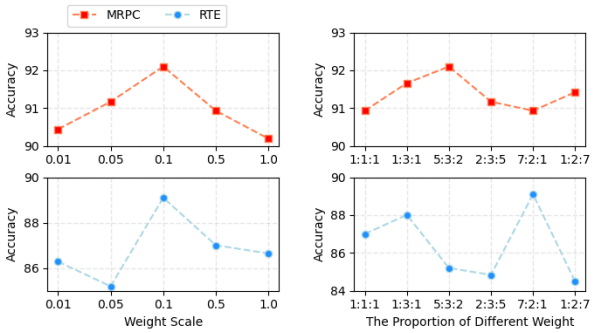

Figure 3: The Accuracy with different weight on MRPC and RTE tasks

B.2 Training Analysis & Visualization

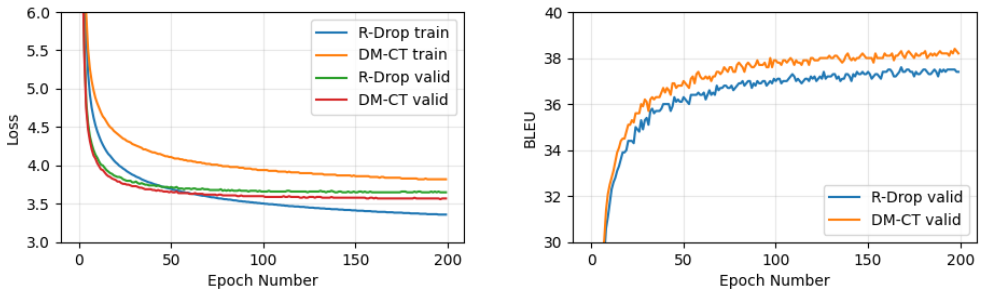

Figure 4: Loss/BLEU curves along with model training.

To better show the advantage of DM-CT model, we plot the curve of training/valid loss and valid BLEU along with training epoch number for R-Drop and our method, as shown in Figure 4. we can see that the DM-CT train loss is large than R-Drop, but the valid loss is better than it. This observation demonstrates that DM-CT provide strong persistent regularization than R-Drop during training. From the BLEU curves, we can clearly find that DM-CT performs better along with each training epoch.