Adaptive Label Smoothing with Self-Knowledge in Natural Language Generation

Dongkyu Lee[1,2*] **Ka Chun Cheung**[2] **Nevin L. Zhang**[1]
[1]Department of Computer Science and Engineering, HKUST
[2]NVIDIA AI Technology Center, NVIDIA
dleear@cse.ust.hk chcheung@nvidia.com lzhang@cse.ust.hk

Abstract

Overconfidence has been shown to impair generalization and calibration of a neural network. Previous studies remedy this issue by adding a regularization term to a loss function, preventing a model from making a peaked distribution. Label smoothing smoothes target labels with a pre-defined prior label distribution; as a result, a model is learned to maximize the likelihood of predicting the soft label. Nonetheless, the amount of smoothing is the same in all samples and remains fixed in training. In other words, label smoothing does not reflect the change in probability distribution mapped by a model over the course of training. To address this issue, we propose a regularization scheme that brings dynamic nature into the smoothing parameter by taking model probability distribution into account, thereby varying the parameter per instance. A model in training self-regulates the extent of smoothing on the fly during forward propagation. Furthermore, inspired by recent work in bridging label smoothing and knowledge distillation, our work utilizes self-knowledge as a prior label distribution in softening target labels, and presents theoretical support for the regularization effect by knowledge distillation and the dynamic smoothing parameter. Our regularizer is validated comprehensively, and the result illustrates marked improvements in model generalization and calibration, enhancing robustness and trustworthiness of a model.

1 Introduction

In common practice, a neural network is trained to maximize the expected likelihood of observed targets, and the gradient with respect to the objective updates the learnable model parameters. With hard targets (one-hot encoded), the maximum objective can be approached when a model assigns a high probability mass to the corresponding target label over the output space. That is, due to the normalizing activation functions (i.e. `softmax`), a model is trained in order for logits to have a marked difference between the target logit and the other classes logits (Müller et al., 2019).

Despite its wide application and use, the maximum likelihood estimation with hard targets has been found to incur an overconfident problem; the predictive score of a model does not reflect the actual accuracy of the prediction. Consequently, this leads to degradation in model calibration (Pereyra et al., 2017), as well as in model performance (Müller et al., 2019). Additionally, this problem stands out more clearly with a limited number of samples, as a model is more prone to overfitting. To remedy such phenomenon, Szegedy et al. (2016) proposed label smoothing, in which one-hot encoded targets are replaced with smoothed targets. Label smoothing has boosted performance in computer vision (Szegedy et al., 2016), and has been highly preferred in other domains, such as Natural Language Processing (Vaswani et al., 2017; Lewis et al., 2020).

However, there are several aspects to be discussed in label smoothing. First, it comes with certain downsides, namely the static smoothing parameter. The smoothing regularizer fails to account for the change in probability mass over the course of training. Despite the fact that a model can benefit from adaptive control of the smoothing extent depending on the signs of overfitting and overconfidence, the smoothing parameter remains fixed throughout training in all instances.

Another aspect of label smoothing to be considered is its connection to knowledge distillation (Hinton et al., 2015). There have been attempts to bridge label smoothing and knowledge distillation, and the findings suggest that the latter is an adaptive form of the former (Tang et al., 2021; Yuan et al., 2020). However, the regularization effect on overconfidence by self-knowledge distillation is

[*] This work was done while Dongkyu was an intern at NVIDIA

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 9781–9792
December 7-11, 2022 ©2022 Association for Computational Linguistics

still poorly understood and explored.

To tackle the issues mentioned above, this work presents adaptive label smoothing with self-knowledge as a prior label distribution. Our regularizer allows a model to self-regulate the extent of smoothing based on the entropic level of model probability distribution, **varying the amount per sample and per time step.** Furthermore, our theoretical analysis suggests that self-knowledge distillation and the adaptive smoothing parameter have a strong regularization effect by rescaling gradients on logit space. To the best of our knowledge, our work is the first attempt in making **both smoothing extent and prior label distribution adaptive**. Our work validates the efficacy of the proposed regularization method on machine translation tasks, achieving superior results in model performance and model calibration compared to other baselines.

2 Preliminaries & Related Work

2.1 Label Smoothing

Label smoothing (Szegedy et al., 2016) was first introduced to prevent a model from making a peaked probability distribution. Since its introduction, it has been in wide application as a means of regularization (Vaswani et al., 2017; Lewis et al., 2020). In label smoothing, one-hot encoded ground-truth label ($\boldsymbol{y}$) and a pre-defined prior label distribution ($\boldsymbol{q}$) are mixed with the weight, the smoothing parameter (α), forming a smoothed ground-truth label. A model with label smoothing is learned to maximize the likelihood of predicting the smoothed label distribution. Specifically,

$$\begin{aligned}\mathcal{L}_{ls} = -\sum_{i=1}^{|C|}(1-\alpha)y_i^{(n)}\log P_\theta(y_i|\boldsymbol{x}^{(n)}) \\ + \alpha q_i \log P_\theta(y_i|\boldsymbol{x}^{(n)})\end{aligned} \tag{1}$$

$|C|$ denotes the number of classes, (n) the index of a sample in a batch, and P_θ the probability distribution mapped by a model. α is commonly set to 0.1, and remains fixed throughout training (Vaswani et al., 2017; Lewis et al., 2020). A popular choice of $\boldsymbol{q}$ is an uniform distribution ($\boldsymbol{q} \sim U(|C|)$), while unigram distribution is another option for dealing with an imbalanced label distribution (Vaswani et al., 2017; Szegedy et al., 2016; Müller et al., 2019; Pereyra et al., 2017). The pre-defined prior label distribution remains unchanged, hence the latter cross-entropy term in Equation 1 is equivalent to minimizing the KL divergence between the model prediction and the pre-defined label distribution. In line with the idea, Pereyra et al. (2017) proposed confidence penalty (ConfPenalty) that adds negative entropy term to the loss function, thereby minimizing the KL divergence between the uniform distribution and model probability distribution. Ghoshal et al. (2021) proposed low-rank adaptive label smoothing (LORAS) that jointly learns a noise distribution for softening targets and model parameters. Li et al. (2020); Krothapalli and Abbott (2020) introduced smoothing schemes that are data-dependent.

2.2 Knowledge Distillation

Knowledge distillation (Hinton et al., 2015) aims to transfer the *dark knowledge* of (commonly) a larger and better performing teacher model to a student model (Buciluundefined et al., 2006). The idea is to mix the ground-truth label with the model probability distribution of a teacher model, resulting in an adaptive version of label smoothing (Tang et al., 2021).

$$\begin{aligned}\mathcal{L}_{kd} = -\sum_{i=1}^{|C|}(1-\alpha)y_i^{(n)}\log P_\theta(y_i|\boldsymbol{x}^{(n)}) \\ + \alpha \bar{P}_\phi(y_i|\boldsymbol{x}^{(n)})\log \bar{P}_\theta(y_i|\boldsymbol{x}^{(n)})\end{aligned} \tag{2}$$

ϕ and θ denote the parameters of a teacher model and a student model respectively. $\bar{P}$ indicates a probability distribution smoothed with a temperature. Similar to label smoothing, ϕ remains unchanged in training; thus a student model is learned to minimize the KL divergence between its probability distribution and that of the teacher model. When $\bar{P}_\phi$ follows a uniform distribution with the temperature set to 1, the loss function of knowledge distillation is identical to that of uniform label smoothing.

Training a large teacher model can be computationally expensive; for this reason, there have been attempts to replace the teacher model with the student model itself, called self-knowledge distillation (Zhang et al., 2019; Yuan et al., 2020; Kim et al., 2021; Zhang and Sabuncu, 2020). TF-KD (Yuan et al., 2020) trains a student with a pre-trained teacher that is identical to the student in terms of structure. SKD-PRT (Kim et al., 2021) utilizes the previous epoch checkpoint as a teacher with linear increase in α. (Zhang and Sabuncu, 2020) incorporates beta distribution sampling (BETA) and self-knowledge distillation (SD), and introduce instance-specific prior label distribution. (Yun et al., 2020) utilizes self-knowledge distillation to minimize the predictive distribution of samples with the same class, encouraging consistent probability distribution within the same class.

3 Approach

The core components of label smoothing are twofold: smoothing parameter (α) and prior label distribution. The components determine how much to smooth the target label using which distribution, a process that requires careful choice of selection. In this section, we illustrate how to make the smoothing parameter adaptive. We also demonstrate how our adaptive smoothing parameter and self-knowledge distillation as a prior distribution act as a form of regularization with theoretical analysis on the gradients.

3.1 Adaptive α

An intuitive and ideal way of softening the hard target is to bring dynamic nature into choosing α; a sample with low entropic level in model prediction, an indication of peaked probability distribution, receives a high smoothing parameter to further smooth the target label. In another scenario, in which high entropy of model prediction (flat distribution) is seen, the smoothing factor is decreased.

With the intuition, our method computes the smoothing parameter on the fly during the forward propagation in training, relying on the entropic level of model probability distribution per sample, and per time step in case of sequential classification.[1]

$$H(P_\theta(\boldsymbol{y}|\boldsymbol{x}^{(n)})) = -\sum_{i=1}^{|C|} P_\theta(y_i|\boldsymbol{x}^{(n)}) \log P_\theta(y_i|\boldsymbol{x}^{(n)}) \tag{3}$$

The entropy quantifies the level of probability mass distributed across the label space; therefore, low entropy is an indication of overfitting and overconfidence (Pereyra et al., 2017; Meister et al., 2020).

Since entropy does not have a fixed range between 0 and 1, one simple scheme is to normalize the entropy with maximum entropy ($\log |C|$). Hence, the normalization is capable of handling variable size of class set among different datasets.

$$\alpha^{(n)} = 1 - \frac{H(P_\theta(\boldsymbol{y}|\boldsymbol{x}^{(n)}))}{\log |C|} \tag{4}$$

With this mechanism, a sample with high entropy is trained with low α, and a sample with low entropy receives high α. The computation for α is excluded from the computation graph for the gradient calculation, hence, the gradient does not flow through adaptive $\alpha^{(n)}$.

There are two essential benefits of adopting the adaptive smoothing parameter. As the smoothing extent is determined by its own probability mass over the output space, the hyperparameter search for α is removed. Furthermore, it is strongly connected to the gradient rescaling effect on self-knowledge distillation, which will be dealt in Section 3.3 in detail.

3.2 Self-Knowledge As A Prior

Similar to (Kim et al., 2021; Liu et al., 2021), our regularizer loads a past student model checkpoint as teacher network parameters in the course of training, though with a core difference in the selection process. The intuition is to utilize past self-knowledge which generalizes well, thereby hindering the model from overfitting to observations in the training set.

$$\phi_t = \underset{\theta_i \in \Theta_t}{\operatorname{argmax}}\, g(f(X'; \theta_i), Y') \tag{5}$$

Θ_t is a set of past model checkpoints up to the current epoch t in training, and function f is a specific task, which in our work is machine translation. X' and Y' are sets of input and ground-truth samples from a validation dataset[2], and the function g could be any proper evaluation metric for model generalization (i.e. accuracy).[3] Our work utilizes the n-gram matching score, BLEU (Papineni et al., 2002) being the function g for finding the suitable prior label distribution.

Equation 5 depicts how the selection process of a self-teacher depends on the generalization of each past epoch checkpoint. In other words, a past checkpoint with the least generalization error is utilized as the self-teacher, a source of self-knowledge, to send generalized supervision. Furthermore, at every epoch, with Equation 5, the proposed approach replaces the self-teacher with the one with the best generalization.

Combining the adaptive smoothing parameter and self-knowledge as a prior distribution, our loss

[1] For notational simplicity, time step is not included in the equation hereafter.

[2] Note that validation dataset is used to calculate generalization error, not to train. Therefore, it is similar to early stopping (Prechelt, 2012).

[3] Depending on the objective of the function g, such as loss, $\operatorname{argmin}_\theta$ can also be used.

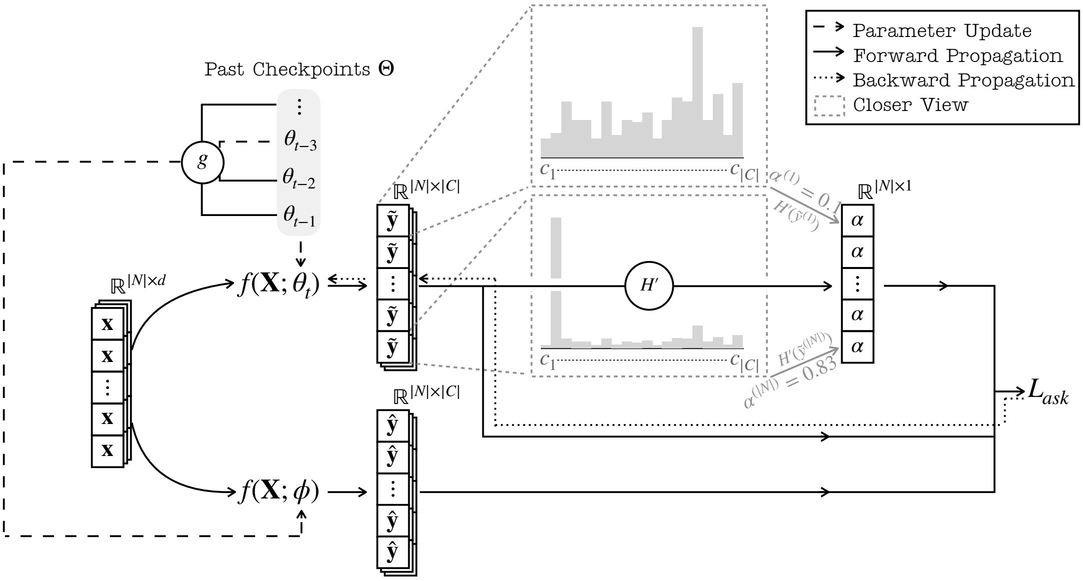


Figure 1: Overview of the proposed regularization. d, $|N|$ and t are input dimension size, batch size, and current epoch respectively. Time step is not described in the figure, yet one can easily extend the above to sequential classification tasks.

function is as follows:

$$\mathcal{L} = -\sum_{i=1}^{|C|}(1-\alpha^{(n)})y_i^{(n)} \log P_\theta(y_i|\boldsymbol{x}^{(n)}) + \alpha^{(n)} P_\phi(y_i|\boldsymbol{x}^{(n)}) \log P_\theta(y_i|\boldsymbol{x}^{(n)}) \tag{6}$$

The core differences to the previous approaches are the introduction of 1) instance-specific α and 2) self-teacher with the least generalization error in training.

3.3 Gradient Analysis

Tang et al. (2021); Kim et al. (2021) theoretically find that the success of knowledge distillation is related to the gradient rescaling in the logit space; the difficulty of a sample determines the rescaling factor, and difficult-to-learn samples receive higher rescaling factors than those of the easy-to-learn samples.[4] We further extend the gradient analysis in the perspective of **regularization effect** and the **direction of the gradient**, and discuss the importance of the adaptive smoothing parameter.

Before dissecting the gradients, we first set a hypothesis: *teacher network makes a less confident prediction than that of the student.* In self-knowledge distillation with a past checkpoint as the teacher, the assumption is valid. The expected predictive score on target label by the teacher model is lower than that of the current checkpoint ***in training*** (Kim et al., 2021).

The gradient with respect to the logit ($\boldsymbol{z}$) by the cross entropy loss ($\mathcal{L}_{ce}$) is as follows[5]:

$$\frac{\partial \mathcal{L}_{ce}}{\partial z_i} = P_\theta(y_i) - y_i \tag{7}$$

With knowledge distillation ($\mathcal{L}_{kd}$), the gradient on the logit is

$$\frac{\partial \mathcal{L}_{kd}}{\partial z_i} = (1-\alpha)(P_\theta(y_i)-y_i)+\alpha(P_\theta(y_i)-P_\phi(y_i)) \tag{8}$$

The following compares the ratio of the gradient from knowledge distillation and with that of the cross entropy.

$$\frac{\partial \mathcal{L}_{kd}/\partial z_i}{\partial \mathcal{L}_{ce}/\partial z_i} = (1-\alpha) + \alpha\frac{P_\theta(y_i) - P_\phi(y_i)}{P_\theta(y_i) - y_i} \tag{9}$$

When $i = j$, with j being the index of the ground truth, it is worth noting that the denominator of the second term in Equation 9 has range $P_\theta(y_i) - 1 \in [-1, 0]$, and the range of the numerator is confined to $P_\theta(y_i) - P_\phi(y_i) \in [0, 1]$. Therefore, the equation can be written as

$$\frac{\partial \mathcal{L}_{kd}/\partial z_i}{\partial \mathcal{L}_{ce}/\partial z_i} = (1-\alpha) - \alpha|\frac{P_\theta(y_i) - P_\phi(y_i)}{P_\theta(y_i) - 1}| \tag{10}$$

[4] For further understanding of gradient rescaling with knowledge distillation, please refer to Proposition 2 in (Tang et al., 2021).

[5] For notational simplicity, $\boldsymbol{x}$ is omitted and temperature is assumed to be 1 hereafter, yet the following analysis holds with varying temperature control.

The norm of the gradient drastically diminishes when there is a large difference between the predictions by the models, and when the predictive score of a student model is high, which is a sign of overconfidence. In terms of the direction of the gradient, when the following is seen,

$$(1-\alpha) < \alpha \left| \frac{P_\theta(y_i) - P_\phi(y_i)}{P_\theta(y_i) - 1} \right| \tag{11}$$

the direction of the gradients with respect to knowledge distillation becomes the opposite to that of the cross entropy, pushing parameters to lower the likelihood of the target index.

The same applies when i is the index of an incorrect class ($i \neq j$). From Equation 9, the following can be derived.

$$\frac{\partial \mathcal{L}_{kd}/\partial z_i}{\partial \mathcal{L}_{ce}/\partial z_i} = 1 - \alpha \frac{P_\phi(y_i)}{P_\theta(y_i)} \tag{12}$$

With the generalized teacher, the expected predictive score on the incorrect labels by the teacher model is higher than that of the student model. Therefore, in addition to the shrinking norm effect, the direction of the gradient can be reversed when $1 < \alpha \frac{P_\phi(y_i)}{P_\theta(y_i)}$, similar to Equation 11; as a result, it leads to updating model parameters to increase the likelihood on the *incorrect classes*, an opposite behavior to that of the cross entropy with hard targets. Overall, in either case, the theoretical support depicts strong regularization effects with the generalized supervision by the teacher.

Connection to Label Smoothing As label smoothing is also closely linked to knowledge distillation, the theoretical support can be easily extended to label smoothing if $P_\phi(y_i)$ is replaced with $\frac{\alpha}{|C|}$ in case of uniform label smoothing, and $P(c_i)$ in unigram label smoothing.

Importance of Adaptive α The adaptive α is another factor to be discussed regarding the gradient analysis. As clearly demonstrated in Equation 11 and 12, a high α, an indication of peak probability distribution, not only leads to drastic decrease in the gradient norm, but it is likely to make the gradient go the opposite direction to that of the cross entropy. It enforces a student to distribute the probability mass more evenly on output space, as opposed to the effect of the cross entropy with hard targets. Furthermore, as the parameters updates are performed by aggregating the losses of samples, adaptive smoothing acts as a gradient rescaling mechanism. Hence, the following proposition can be made.

Proposition 1. *Given any two samples* $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}),$ $(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) \in \mathcal{X} \times \mathcal{Y}$ *and* $P_\theta(y_j^{(i)}|\mathbf{x}^{(i)}) = P_\theta(y_j^{(k)}|\mathbf{x}^{(k)})$*, the average gradient rescaling factor* w *for all classes is greater on sample with high probability entropy than that of the one with low probability entropy.*

For details, please refer to Appendix A. The gradient rescaling by adaptive α reweights the gradients in aggregating the losses, hence, the proposed method prioritizes on learning samples with high entropy, less confident instances. The use of adaptive α is not only intuitive in terms of tackling overconfidence, but it also serves as an important aspect in the theoretical support.

4 Experiment

4.1 Dataset & Experiment Setup

We validate the proposed regularizer on three popular translation corpora: IWSLT14 German-English (DE-EN) (Cettolo et al., 2014), IWSLT15 English-Vietnamese (EN-VI) (Cettolo et al., 2015), and Multi30K German-English pair (Elliott et al., 2016). The details can be found in Appendix C.

The core reason for conducting experiments on translation comes from one aspect of natural language: the presence of intrinsic uncertainty (Ott et al., 2018). In a natural language, synonyms can be used interchangeably in a sentence as they denote the same meaning. Such uncertainty is not reflected in one-hot encoded form. Hence, a model in a natural language generation task can benefit from inter-class relations held within a knowledge (Hinton et al., 2015).

All of the experiments are conducted with transformer architecture (Vaswani et al., 2017) on a Telsa V100. For generation, beam size is set to 4 in the inference stage. The training configuration follows the instruction of `fairseq` (Ott et al., 2019).[6]. For the quality of the generated outputs, we report the popular metrics for machine translation: BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), Word Error Rate (WER), ROUGE-L (Lin, 2004), and NIST (Doddington, 2002).

[6] `https://github.com/pytorch/fairseq/tree/main/examples/translation`

Table 1: The scores are reported in percentage and are averaged over three runs with different random seeds. Except for the results from the cross entropy with hard targets, denoted as Base hereinafter, other scores are absolute difference from those of Base. Bold numbers indicate the best performance among the methods.

Corpus	Method	**BLEU**(↑)	**METEOR**(↑)	**WER**(↓)	**ROUGE-L**(↑)	**NIST**(↑)
Multi30K DE→EN	Base	40.64	73.31	38.76	69.21	7.96
	Uniform LS	+1.90	+1.47	-0.76	+0.96	+0.12
	Unigram LS	+1.87	+1.30	-1.22	+1.09	+0.17
	ConfPenalty	+2.50	+1.72	-0.81	+1.33	+0.15
	LORAS	+1.14	+1.00	-0.73	+0.63	+0.16
	TF-KD	+1.13	+1.02	-1.21	+0.73	+0.15
	SKD-PRT	+1.31	+0.95	-0.31	+0.54	+0.09
	BETA	+1.26	+0.94	-0.20	+0.46	+0.07
	SD	+2.76	+2.09	-1.26	+1.55	+0.18
	Ours	**+3.75**	**+2.91**	**-2.19**	**+2.17**	**+0.32**
IWSLT15 EN→VI	Base	30.17	59.09	54.02	63.91	7.18
	Uniform LS	+0.57	+0.62	-0.40	+0.42	+0.05
	Unigram LS	+0.62	+0.48	-0.79	+0.43	+0.08
	ConfPenalty	+0.93	+0.56	-1.02	+0.59	+0.12
	LORAS	-0.04	-0.10	-0.23	-0.19	+0.02
	TF-KD	-0.01	-0.15	-0.01	-0.08	-0.02
	SKD-PRT	+1.03	+0.95	-1.31	+0.80	+0.16
	BETA	+0.30	+0.20	-0.63	+0.17	+0.07
	SD	+0.68	+0.46	-0.63	+0.45	+0.08
	Ours	**+1.37**	**+1.14**	**-1.80**	**+1.05**	**+0.21**
IWSLT14 DE→EN	Base	35.96	64.70	48.17	61.82	8.47
	Uniform LS	+0.86	+0.61	-0.67	+0.59	+0.14
	Unigram LS	+1.01	+0.68	-0.87	+0.76	+0.16
	ConfPenalty	+1.15	+0.86	-1.08	+0.85	+0.19
	LORAS	+0.36	+0.23	+0.61	+0.16	-0.03
	TF-KD	+0.39	+0.19	-0.33	+0.29	+0.06
	SKD-PRT	+1.53	+1.11	-1.69	+1.25	+0.24
	BETA	+0.95	+0.69	-0.30	+0.58	+0.08
	SD	+1.39	+1.06	-0.84	+0.88	+0.16
	Ours	**+1.86**	**+1.55**	**-2.08**	**+1.59**	**+0.32**

4.2 Experimental Result & Analysis

Automatic evaluation results on the three test datasets are shown in Table 1. Though most of the methods achieve meaningful gains, the most noticeable difference is seen with our method. Our regularization scheme shows solid improvements on all of the metrics on the datasets without any additional learnable parameter. For example, the absolute gain in BLEU compared to the base method in Multi30K dataset is around 3.75, which is 9.2% relative improvement. Not only does our method excel in n-gram matching score, but it shows superior performance in having the longest common subsequence with the reference text, as well as in the informativeness of the n-grams. The empirical result demonstrates that our regularizer improves the base method across all the metrics by a large margin.

In Figure 2, the changes in α during training are visualized. As expected, the smoothing parameters start with a very small number, as the entropic level must be high due to the under-fitted models. As training continues, the predictive scores of the models increase, and accordingly, adaptive α increases to prevent overconfidence. One notable aspect is the convergence at a certain level. Each training of the corpora ends up with a different α, and the model in training self-regulates the amount of smoothing and the value converges.

Furthermore, our adaptive α affects the norm of the gradients as depicted in Figure 3. The gradient norm of our regularizer is considerably smaller than that of the other methods. This empirical finding mainly conforms with the gradient analysis in Section 3.3, where the importance of adaptive α and generalized teacher model are discussed.

Table 2: We report Expected Calibration Error (**ECE**) and Maximum Calibration Error (**MCE**), in percentage, on the test sets of the corpora for evaluating the calibration ability.

Method	Multi30K DE→EN		IWSLT15 EN→VI		IWSLT14 DE→EN	
	ECE (↓)	**MCE** (↓)	**ECE** (↓)	**MCE** (↓)	**ECE** (↓)	**MCE** (↓)
Base	14.95	26.01	14.05	20.38	12.98	19.29
Uniform LS	9.17	17.22	8.53	12.13	6.43	9.98
Unigram LS	9.12	17.78	7.89	11.71	6.12	9.46
ConfPenalty	48.21	73.46	43.94	59.28	48.19	57.58
LORAS	20.27	40.86	12.41	19.15	10.54	15.29
TF-KD	21.18	42.87	13.30	19.29	12.20	17.60
SKD-PRT	14.75	26.69	9.34	14.18	5.63	8.88
BETA	11.71	21.90	9.57	14.97	8.63	13.21
SD	6.87	**12.38**	5.01	9.64	7.82	13.71
Ours	**4.76**	12.41	**2.15**	**4.40**	**1.76**	**3.64**

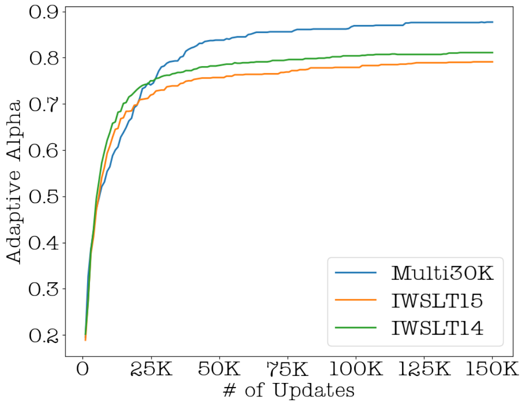


Figure 2: Illustration of the changes in the proposed smoothing parameter α throughout the training on the tested corpora.

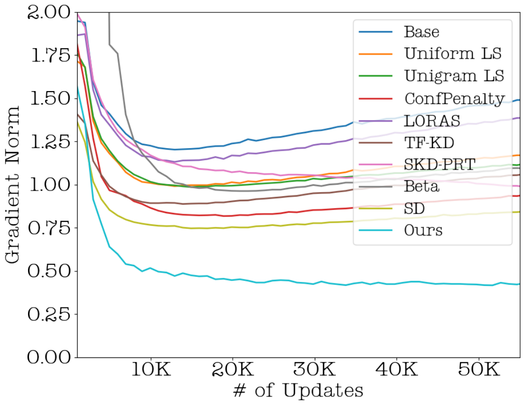


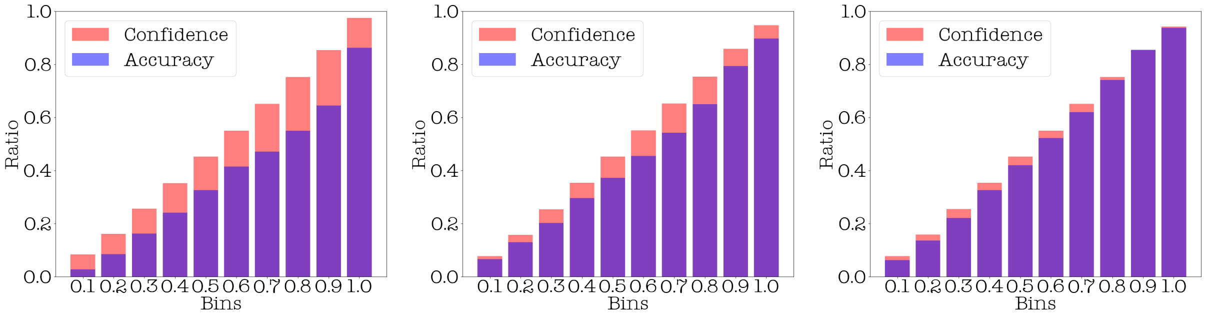
Figure 3: Change in the gradient norm of the baseline methods and the proposed approach on IWSLT15 EN→VI corpus.

4.2.1 Model Calibration

In addition to the automatic evaluation, in which the improved generalization is seen through the performance gains, we look into the calibrations of the models trained with the methods. Figure 4(a) depicts how the cross entropy with hard targets tends to make a model overconfident in prediction. In the reliability diagram, the confidence score in each bin is larger than the corresponding accuracy, the gap of which is fairly noticeable. Label smoothing mitigates the problem to some extent, yet the gap between the accuracy and the confidence score still remains clear. We empirically find that models trained with the baseline methods suffer from either overconfidence or underconfidence. On the other hand, the proposed regularizer significantly reduces the gap, showing the enhanced model calibration. As clearly depicted in Figure 4(c), the confidence level of each bin mainly conforms with the accuracy, demonstrating reliable predictions made by the model trained with the proposed approach.

The improvement in calibration is more clear with expected calibration error (ECE) and maximum calibration error (MCE) reported in Table 2. For instance, on the IWSLT14 dataset, the errors with label smoothing drop significantly in both metrics, which is around 6% absolute decrease in ECE and 10% in MCE. Nonetheless, ECE of the proposed method results in 1.76% which is around 11% absolute decrease and 86% relative improvement. In addition, our method achieves 3.64% in MCE, which is 81% relative improvement over the base method. The improved calibration with our method is seen across the datasets, confirming the effectiveness of our system in enhancing model calibration.

One important finding is the gap between the performance in model generalization and the cal-



(a) Base

(b) Uniform Label Smoothing

(c) Ours

Figure 4: Reliability diagram of Base method, uniform label smoothing, and ours. Predictions on IWSLT14 DE→EN test set are binned to 10 groups based on the predictive scores. Each bar indicates the average confidence score and accuracy of each bin.

Table 3: (+) denotes adding the following components to the base method. $\alpha^{(n)}$ denotes our adaptive α, and $\alpha^{\uparrow}$ indicates a linear increase in α in the course of training. SK and Uniform denote Self-Knolwedge and Uniform distribution as a prior label distribution respectively. g_{NLL} and g_{BLEU} indicates g function set to negative log likelihood and BLEU respectively.

Method	**BLEU** (↑)	**ECE** (↓)
Base	35.96	12.98
(+) Fixed α & SK	36.27	13.56
(+) $\alpha^{(n)}$ & Uniform	37.30	18.76
(+) $\alpha^{\uparrow}$ & SK	37.52	5.58
Ours (g_{NLL})	37.74	1.30
Ours (g_{BLEU})	37.82	1.76

ibration error. Confidence penalty (Pereyra et al., 2017) is highly competitive in n-gram matching scores (BLEU) on all of the dataset tested. Nevertheless, the calibration error is the highest among the methods due to *underconfidence*. Similar to the finding in (Guo et al., 2017), the discrepancy between the performance and model calibration exists, and it calls for caution in training a neural network when considering model calibration.

4.3 Ablation Study

Table 3 shows the change in performance when adding our core components to the base method on the IWSLT14 dataset. When using the fixed smoothing parameter with self-knowledge as a prior, the BLEU score increases by a small margin, and the ECE does not drop significantly. In another case where the smoothing parameter is adaptive, and the prior label distribution is set to uniform distribution, there is a meaningful increase in BLEU score. However, it impairs the ECE score noticeably. We empirically find that the result mainly comes from underconfidence of a model. The confidence score is largely lower than that of the accuracy. In an experiment with linearly increasing smoothing parameter $\alpha^{\uparrow}$ with self-knowledge prior, the BLEU score improves by around 1.6 score, yet the ECE score still shows room for improvement. Since α value is shared among samples in the experiment, there is no gradient rescaling by adaptive α which may explain ECE score being high compared to that of our adaptive α. We also look into a case with a different g function: BLEU and Negative Log Likelihood (NLL). We observe that both g_{BLEU} and g_{NLL} greatly enhance the scores. As g has the purpose of selecting a self-teacher with the least generalization error from the set of past checkpoints, a proper metric would serve the purpose. In conclusion, while the adaptive α plays an important role in regularizing a model, both the adaptive α and the choice of prior label distribution greatly affect model calibration.

5 Conclusion & Future Work

In this work, we propose a regularization scheme that dynamically smooths the target label with self-knowledge. Our regularizer self-regulates the amount of smoothing with respect to the entropic level of the model probability distribution, making the smoothing parameter dynamic per sample, and per time step. The given idea is theoretically supported by gradient rescaling and direction, and the finding is backed up by the empirical results, both in model performance and calibration.

Limitation

The proposed regularization method is model driven, and hence it adds additional computation cost when self-knowledge is computed. This limitation, however, does not pertain to our work but is shared in KD training. This issue can be mitigated to some extent when self-knowledge is obtained and saved before training a model. In addition, the smoothing technique requires additional computation for computing the instance-specific smoothing term (normalized entropy).

Acknowledgement

Research on this paper was supported by Hong Kong Research Grants Council (Grant No. 16204920).

References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Cristian Buciluundefined, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 535–541, New York, NY, USA. Association for Computing Machinery.

Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, Roldano Cattoni, and Marcello Federico. 2015. The iwslt 2015 evaluation campaign. In *IWSLT*.

Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico. 2014. Report on the 11th iwslt evaluation campaign. In *IWSLT*.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the Second International Conference on Human Language Technology Research*, HLT '02, pages 138–145, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. 2016. Multi30k: Multilingual english-german image descriptions. In *Proceedings of the 5th Workshop on Vision and Language*, pages 70–74. Association for Computational Linguistics.

Asish Ghoshal, Xilun Chen, Sonal Gupta, Luke Zettlemoyer, and Yashar Mehdad. 2021. Learning better structured representations using low-rank adaptive label smoothing. In *International Conference on Learning Representations*.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*.

Kyungyul Kim, ByeongMoon Ji, Doyoung Yoon, and Sangheum Hwang. 2021. Self-knowledge distillation with progressive refinement of targets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6567–6576.

Diederick P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Ujwal Krothapalli and A. Lynn Abbott. 2020. Adaptive label smoothing. *CoRR*, abs/2009.06432.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Weizhi Li, Gautam Dasarathy, and Visar Berisha. 2020. Regularization via structural label smoothing. In *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pages 1453–1463. PMLR.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yang Liu, Sheng Shen, and Mirella Lapata. 2021. Noisy self-knowledge distillation for text summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 692–703, Online. Association for Computational Linguistics.

Clara Meister, Elizabeth Salesky, and Ryan Cotterell. 2020. Generalized entropy regularization or: There's nothing special about label smoothing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Seattle, USA. Association for Computational Linguistics.

Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. 2019. When does label smoothing help? In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Analyzing uncertainty in neural machine translation. In *International Conference on Machine Learning*.

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

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey E. Hinton. 2017. Regularizing neural networks by penalizing confident output distributions. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*.

Lutz Prechelt. 2012. Early stopping - but when? In *Neural Networks: Tricks of the Trade*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. 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)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826.

Jiaxi Tang, Rakesh Shivanna, Zhe Zhao, Dong Lin, Anima Singh, Ed H. Chi, and Sagar Jain. 2021. Understanding and improving knowledge distillation.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Li Yuan, Francis EH Tay, Guilin Li, Tao Wang, and Jiashi Feng. 2020. Revisiting knowledge distillation via label smoothing regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3903–3911.

Sukmin Yun, Jongjin Park, Kimin Lee, and Jinwoo Shin. 2020. Regularizing class-wise predictions via self-knowledge distillation. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. 2019. Be your own teacher: Improve the performance of convolutional neural networks via self distillation.

Zhilu Zhang and Mert R. Sabuncu. 2020. Self-distillation as instance-specific label smoothing. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

A Gradient Rescaling by α

Proposition 1. *Given any two samples* $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$, $(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) \in \mathcal{X} \times \mathcal{Y}$ *and* $P_\theta(y_j^{(i)}|\mathbf{x}^{(i)}) = P_\theta(y_j^{(k)}|\mathbf{x}^{(k)})$, *the average gradient rescaling factor w for all classes is greater on sample with high probability entropy than that of the one with low probability entropy.*

This proposition is built on the basis of Proposition 2 in (Tang et al., 2021), where the work discusses the gradient rescaling effect on logit space by KD. However, there are a few differences, of which the first is the assumption made. In (Tang et al., 2021), the paper assumes that a teacher makes more confident prediction than a student. This is in opposition to our setting, where a teacher makes less confident prediction than a student; this assumption is valid as a teacher is set to be the previous checkpoint of a student. In addition, our work further extends the proposition in terms of α. The proposition only holds as the proposed work employs **instance-specific** α.

Proof. We rewrite the Equation 9 for readers' better understanding/comprehension.

$$w_i = \frac{\partial \mathcal{L}_{kd}/\partial z_i}{\partial \mathcal{L}_{ce}/\partial z_i} = (1-\alpha) + \alpha \frac{P_\theta(y_i) - P_\phi(y_i)}{P_\theta(y_i) - y_i}$$

The gradient rescaling factor on target index is as follows:

$$w_j = (1-\alpha) + \alpha \frac{P_\theta(y_i) - P_\phi(y_i)}{P_\theta(y_i) - 1}$$

Now, the gradient rescaling factor on the remaining classes is computed as the following.

$$\begin{aligned}\sum_{i \neq j} \partial \mathcal{L}_{kd}/\partial z_i = &\sum_{i \neq j}[(1-\alpha)P_\theta(y_i) \\ &+ \alpha P_\theta(y_i) - P_\phi(y_i)] \\ &= (1-\alpha)(1-P_\theta(y_j)) \\ &+ \alpha(P_\phi(y_j) - P_\theta(y_j))\end{aligned}$$

$$\begin{aligned}\sum_{i \neq j} \partial \mathcal{L}_{ce}/\partial z_i &= \sum_{i \neq j} P_\theta(y_i) \\ &= (1-P_\theta(y_j))\end{aligned}$$

$$\frac{\sum_{i \neq j} \partial \mathcal{L}_{kd}/\partial z_i}{\sum_{i \neq j} \partial \mathcal{L}_{ce}/\partial z_i} = (1-\alpha) + \alpha \frac{P_\theta(y_i) - P_\phi(y_i)}{P_\theta(y_i) - 1}$$

$$w_i = \frac{\partial \mathcal{L}_{kd}/\partial z_i}{\partial \mathcal{L}_{ce}/\partial z_i} = \frac{\sum_{i \neq j} \partial \mathcal{L}_{kd}/\partial z_i}{\sum_{i \neq j} \partial \mathcal{L}_{ce}/\partial z_i} \quad (13)$$

Assuming we are given two samples $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$, $(\mathbf{x}^{(k)}, \mathbf{y}^{(k)})$ and $P_\theta(y_j^{(i)}|\mathbf{x}^{(i)}) = P_\theta(y_j^{(k)}|\mathbf{x}^{(k)})$, when the conditional entropy of the probability distributions differ such that $H(P_\theta(Y|\mathbf{x}^{(i)})) > H(P_\theta(Y|\mathbf{x}^{(k)}))$, the average gradient rescaling factor over all classes is smaller on a sample with low entropy than that with a high entropy.

$$\mathbb{E}w^{(i)} > \mathbb{E}w^{(k)} \quad (14)$$

$$\begin{aligned}(1-\alpha^{(i)}) + \alpha^{(i)} \frac{P_\theta(y_i) - P_\phi(y_i)}{P_\theta(y_i) - 1} > \\ (1-\alpha^{(k)}) + \alpha^{(k)} \frac{P_\theta(y_i) - P_\phi(y_i)}{P_\theta(y_i) - 1}\end{aligned} \quad (15)$$

as $\frac{P_\theta(y_i) - P_\phi(y_i)}{P_\theta(y_i) - 1}$ is a negative value and $\alpha^{(i)} < \alpha^{(k)}$; hence the proof. □

B Baselines

B.1 Confidence Penalty

Confidence penalty (Pereyra et al., 2017) adds a negative entropy term to the loss function, hence the model is encouraged to maintain entropy at certain level.

$$\begin{aligned}\mathcal{L}_{cf} = &-\sum_{i=1}^{|C|} y_i^{(n)} \log P_\theta(y_i|\boldsymbol{x}^{(n)}) \\ &- \beta H(P_\theta(\boldsymbol{y}|\boldsymbol{x}^{(n)}))\end{aligned} \quad (16)$$

For the regularization-specific hyperparameter, following (Meister et al., 2020), β was set to 0.78.

B.2 TF-KD

TF-KD (Yuan et al., 2020), similar to conventional knowledge distillation, trains a teacher model prior to training a student; but it is different in that the model architecture is same with that of the student. For the hyperparameters used in this paper, we empirically find that high smoothing parameter leads to better performance. Thus, we set the smoothing parameter to 0.9 and temperature scaling to 20 as reported in the original paper.

B.3 SKD-PRT

SKD-PRT (Kim et al., 2021) is a self-knowledge distillation method, where a student model (epoch t) is trained with its own last epoch checkpoint (epoch $t-1$) in the course of training. Though the idea is similar to ours, yet there are two core differences. The first is that we find the teacher model that generalizes well with a function g. Another difference is that SKD-PRT linearly increases α throughout the training, and this practice inevitably adds two hyperparameters (`max` α and `max` epoch). Following the original work (Kim et al., 2021), we set that maximum α to 0.7 and maximum epoch to 150 in our experiments.

B.4 LORAS

LORAS (Ghoshal et al., 2021) jointly learns a soft target and model parameters in training in the aim of increasing model performance and model calibration, with low rank assumption. For hyperparameters, η, α, rank and dropout probability are set to 0.1, 0.2, 25 and 0.5 respectively.

B.5 BETA & SD

Zhang and Sabuncu (2020) propose amortized MAP interpretation of teacher-student training, and introduce Beta smoothing which is an instance-specific smoothing technique that is based on the prediction by a teacher network. For SD-specific hyperparameters, this work sets α to 0.3 and temperature to 4.0. For BETA-specific hyperparameters, α and a are set to 0.4, 4.0 respectively.

C Dataset Details

IWSLT14 `DE-EN` contains 160K sentence pairs in training, 7K in validation, and 7K in testing. IWSLT15 `EN-VI` has 133K, 1.5K and 1.3K in training, validation, and testing dataset respectively. Lastly, 28K training, 1K validation, and 1K testing sentences are used in Multi30K dataset. Byte pair encoding (Sennrich et al., 2016) is used to process words into sub-word units.

D Reproducibility Statement

For reproducibility, we report the three random seeds tested: {`0000, 3333, 5555`}. For all of the experiments, this work utilizes the transformer architecture (Vaswani et al., 2017). Both the encoder and the decoder are composed of 6 transformer layers with 4 attention heads. The hidden dimension size of the both is 512. For training configuration, the maximum tokens in a batch is set to 4,096. For optimization, Adam (Kingma and Ba, 2015) is used with beta 1 and beta 2 set to 0.9 and 0.98 respectively. We slowly increase the learning rate up to 0.005 throughout the first 4,000 steps, and the learning rate decreases from then on. The source code and training script are included in the supplementary materials.