

# DiMS: Distilling Multiple Steps of Iterative Non-Autoregressive Transformers for Machine Translation

**Sajad Norouzi\***  
Layer6 AI  
sajad@layer6.ai

**Rasa Hosseinzadeh\***  
Layer6 AI  
rasa@layer6.ai

**Felipe Pérez**  
Layer6 AI  
felipe@layer6.ai

**Maksims Volkovs**  
Layer6 AI  
maks@layer6.ai

## Abstract

The computational benefits of iterative non-autoregressive transformers decrease as the number of decoding steps increases. As a remedy, we introduce **Distill Multiple Steps (DiMS)**, a simple yet effective distillation technique to decrease the number of required steps to reach a certain translation quality. The distilled model enjoys the computational benefits of early iterations while preserving the enhancements from several iterative steps. DiMS relies on two models namely student and teacher. The student is optimized to predict the output of the teacher after multiple decoding steps while the teacher follows the student via a slow-moving average. The moving average keeps the teacher’s knowledge updated and enhances the quality of the labels provided by the teacher. During inference, the student is used for translation and no additional computation is added. We verify the effectiveness of DiMS on various models obtaining 7.8 and 12.9 BLEU points improvements in single-step translation accuracy on distilled and raw versions of WMT’14 De-En. Full code for this work is available here: <https://github.com/layer6ai-labs/DiMS>.

## 1 Introduction

Neural machine translation models typically follow an autoregressive decoding strategy, generating the target sentence one token at a time. This sequential nature makes the inference process slow and dependent on the output sequence length. To address this limitation Gu et al. (2018) introduces the Non-Autoregressive Transformer (NAT). NAT generates the entire target sentence in parallel, reducing the latency by an order of magnitude. NAT can be considered as a member of a broader family of iterative non-autoregressive Transformers (iNAT) (Lee et al., 2020; Stern et al., 2019; Ghazvininejad et al., 2019) where the number of decoding steps is fixed

\*Equal contribution.

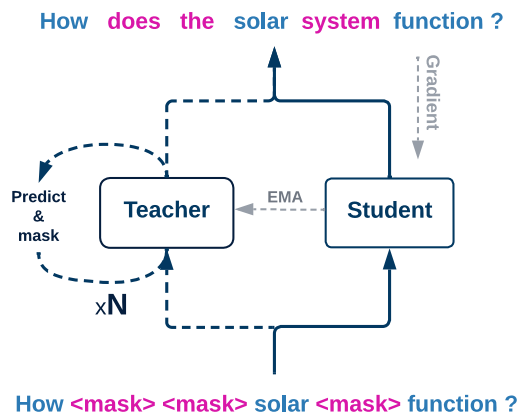


Figure 1: DiMS training. The student is trained to match the predictions of the teacher after several iterative steps. Teacher is updated with an exponential moving average of the student.

and independent of the sequence length. By tuning the number of decoding steps, one can control the trade-off between speed and quality. While iNATs can be considered as efficient alternatives to their autoregressive counterparts, Kasai et al. (2020b) shows that autoregressive models can be sped up without loss in accuracy by combining shallow decoders with deep encoders. This diminishes the computational advantage of iNATs and challenges their motivation. The focus of recent work has thus been shifted to design single-step NAT models (Ghazvininejad et al., 2020a; Qian et al., 2021; Du et al., 2021).

In order to preserve the enhancements obtained by multiple decoding iterations of iNATs, we introduce Distill Multiple Steps (DiMS), a distillation algorithm applicable to a wide range of iterative models. Given a pre-trained iNAT, referred to as teacher, a student aims to replicate the behavior of multiple iterative steps of the teacher with one decoding pass. This process resembles the well-known knowledge distillation framework (Hinton et al., 2015). However, instead of reducing the number of parameters, we aim to decrease the number

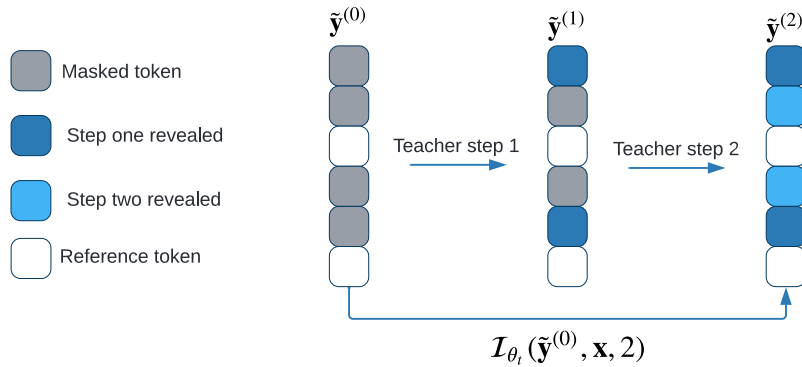


Figure 2: Two iterative steps of the teacher applied to a partially masked sentence.

of decoding passes. The final model then enjoys the translation quality of multi-steps iNAT with the computational efficiency of single-step translation.

The proposed distillation can be repeated iteratively, where at the end of each round the newly optimized student becomes the next teacher. While effective, iterative distillation is slow as it requires multiple rounds of training until convergence. Alternatively, we propose updating the parameters of the teacher with an exponential moving average (EMA) of the student. This gradually transfers the new knowledge learned by the student to the teacher and can be viewed as a continuous variant of iterative distillation. Figure 1 depicts the DiMS algorithm.

We demonstrate the effectiveness of our approach on several public datasets by showing that DiMS obtains substantial improvements on single-step translation with gains of up to 7.8 BLEU points on the distilled training dataset, while the gains on raw datasets are even greater. Notably, we are able to surpass many leading NAT models designed specifically for single-step translation. We further show that EMA considerably speeds up training and converges to a comparable accuracy with iterative distillation in a fraction of epochs.

## 2 Background

In this section, we lay out a formal framework for iNATs. We use the setup of Conditional Masked Language Models (CMLM). CMLM first introduced in Ghazvininejad et al. (2019) and subsequently adopted in many iNAT models (Ghazvininejad et al., 2020b; Kasai et al., 2020a; Huang et al., 2021). The source sentence, target sentence, and target sequence length are denoted by  $\mathbf{x}$ ,  $\mathbf{y}$  and  $N$ , respectively.

### 2.1 Training

Given a partially masked reference sentence  $\tilde{\mathbf{y}}$  and the corresponding source context  $\mathbf{x}$ , the model is trained to reveal all the masked positions simultaneously (Ghazvininejad et al., 2019). From a probabilistic perspective, this imposes a conditional independence assumption on the predicted tokens. Formally, the training loss is:

$$\mathbb{E}_{\tilde{\mathbf{y}} \sim \mathcal{M}(\mathbf{y})} \sum_{i \in \xi(\tilde{\mathbf{y}})} -\log p_{\theta}(y_i | \mathbf{x}, \tilde{\mathbf{y}}),$$

where  $\mathcal{M}$  is a distribution over all partially masked target sentences and  $\xi$  is a function that returns the set of masked indices. The training objective above implicitly assumes access to the target sentence length. To resolve this issue, CMLM trains a parametric model, *length predictor*, to predict the output length.

### 2.2 Inference

The inference begins by creating a template  $\tilde{\mathbf{y}}^{(0)}$  with  $\tilde{N}$  masked tokens, where  $\tilde{N}$  is the output of the length predictor. At iteration  $t$  of the inference, the model predicts the translation  $\mathbf{r}^{(t)}$  given  $\tilde{\mathbf{y}}^{(t-1)}$  and  $\mathbf{x}$  as inputs. Depending on the number of decoding iterations  $S$ , typically a linear unmasking policy is used where at each step  $\tilde{N}/S$  tokens with the highest probability are revealed. This process is repeated  $S$  times, resulting in a fully revealed sentence. In other words,  $\tilde{y}_i^{(t)}$  is set to  $r_i^{(t)}$  when  $i \in \arg\text{-topk}_{k=\frac{\tilde{N}}{S}} \left\{ p_{\theta} \left( r_j^{(t)} \mid \mathbf{x}, \tilde{\mathbf{y}}^{(t-1)} \right) \right\}$ , where  $p_{\theta}$  denotes the output probability of the model. Otherwise  $\tilde{y}_i^{(t)}$  stays equal  $\tilde{y}_i^{(t-1)}$ .

Note that multiple length candidates can be considered (e.g.  $\tilde{N} \pm 1$ ) with the average token probability as a ranking criterion. This is similar to beam

---

**Algorithm 1** DiMS

---

**Require:** Data set  $\mathcal{D}$ , pre-trained model  $\phi$ , Hidden state loss factor  $\lambda$ , teacher steps  $n$ ,  
EMA momentum  $\mu$ , learning rate  $\eta$   
 $\theta_t, \theta_s \leftarrow \phi$  ▷ Initialize teacher and student  
**while** not converged **do**  
     $(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}$  ▷ Sample data  
     $\tilde{\mathbf{y}} \sim M(\mathbf{y})$  ▷ Sample masking  
     $\mathbf{p}_t \leftarrow \mathcal{I}_{\theta_t}(\mathbf{x}, \tilde{\mathbf{y}}, n)$  ▷ Run the teacher for  $n$  iterative steps  
     $\mathbf{p}_s \leftarrow \mathcal{I}_{\theta_s}(\mathbf{x}, \tilde{\mathbf{y}}, 1)$  ▷ Run the student for a single step  
     $\mathcal{L}_{\text{DiMS}} \leftarrow \sum_i \text{KL}(\mathbf{p}_{t,i} | \mathbf{p}_{s,i}) + \lambda \|\mathbf{e}_{t,i} - \mathbf{e}_{s,i}\|^2$  ▷ Compute the DiMS loss  
     $\theta_s \leftarrow \text{Optimizer}(\theta_s, \nabla_{\theta_s} \mathcal{L}_{\text{DiMS}}, \eta)$  ▷ Gradient based optimization of the student  
     $\theta_t \leftarrow (1 - \mu)\theta_s + \mu\theta_t$  ▷ EMA Update of the teacher  
**end while**

---

search in autoregressive models but applied to the output sequence length. It is referred to as *length beam*.

### 3 Distillation of Iterative Non-autoregressive Transformers

Increasing the number of decoding steps typically improves accuracy, but diminishes the computational advantage of iNATs. Our objective is to reduce the number of decoding steps without degrading the performance. More specifically, we want to condense the translation quality of multiple steps of a teacher into one decoding pass of a student. For instance, consider an iterative model (teacher) that uses eight decoding steps. By replicating four steps of the teacher with one decoding pass, two steps of the student would be sufficient to reach a similar performance.

The standard way of knowledge distillation would have the teacher generate soft labels for all intermediate iterations, and optimize the student to track the teacher’s output with fewer steps, but doing such generation on-the-fly greatly increases the training cost. This process can be moved to a pre-processing phase, at the cost of large memory requirement. We propose to use partially masked reference sentences as an approximation to the intermediate predictions of the teacher, which eliminates the need for several decoding passes or large memory capacity.

The distillation process starts by initializing the student and the teacher to the same pre-trained model with parameters  $\phi$  i.e.  $\theta_s = \theta_t = \phi$  where  $\theta_s$  and  $\theta_t$  denote the parameters of the student and teacher. Then, the teacher processes a partially masked sentence  $\tilde{\mathbf{y}}$  through  $n$  iterative steps with

a linear unmasking policy. More precisely,  $i/n$  of the originally masked tokens are revealed up to step  $i$  and after the final pass, no masked token remains. This is similar to the inference procedure outlined in Section 2.2, but instead of starting from a fully masked sentence, it starts from a partially masked one. The student is optimized to match the teacher’s soft labels and a temperature is used to control the smoothness of the labels. With enough capacity, the student is expected to imitate the behavior of  $n$  consecutive steps of the teacher with one decoding pass.

#### 3.1 Training Loss

We denote the output distribution after  $n$  iterative steps on the partially masked sentence  $\tilde{\mathbf{y}}$  by  $\mathcal{I}_{\theta}(\tilde{\mathbf{y}}, \mathbf{x}, n)$  where  $\theta$  represents the parameters of the model. The distillation loss can be described as:  $\sum_{i \in \xi(\tilde{\mathbf{y}})} \text{KL}(\mathbf{p}_{t,i} | \mathbf{p}_{s,i})$  where  $\mathbf{p}_t = \mathcal{I}_{\theta_t}(\tilde{\mathbf{y}}, \mathbf{x}, n)$ ,  $\mathbf{p}_s = \mathcal{I}_{\theta_s}(\tilde{\mathbf{y}}, \mathbf{x}, 1)$  and  $i$  in subscript denotes the index in the sentence. Note that the teacher’s soft labels do not come from the same decoding iteration i.e. whenever a token is revealed, the corresponding soft labels are fixed in  $\mathbf{p}_t$ . Thus, the student receives labels from various decoding steps of the teacher. Figure 2 depicts the process teacher follows to produce the labels for two iterative steps. From the student’s point of view, the primary difference between DiMS and CMLM training (Section 2.1) is the use of soft labels generated by the teacher instead of the ground truth tokens.

To facilitate the distillation, we combine the KL-divergence with the Euclidean distance of the last layers’ hidden states of the teacher and the student. This transfers the knowledge concealed within the hidden states that might not be discernible in soft labels. We refer to this as *hidden state loss*. Sim-

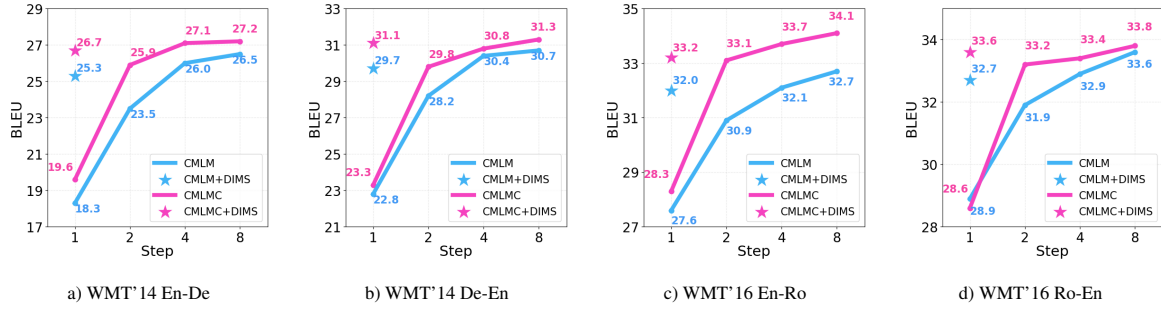


Figure 3: CMLM and CMLMC models distilled with DiMS on four WMT tasks. For each teacher we plot performance across various decoding steps and contrast it with a single-step performance of the student.

ilar to the KL-divergence, the hidden state loss is computed over the masked indices.

To summarize, DiMS training loss has two terms: **i)** KL-divergence between distributions predicted by the teacher and the student. **ii)** The Euclidean distance between the last hidden states of two models. Denoting teacher’s and student’s last hidden state by  $\mathbf{e}_t$  and  $\mathbf{e}_s$ , DiMS loss can be written formally as:

$$\mathcal{L}_{\text{DiMS}} = \sum_i \text{KL}(\mathbf{p}_{t,i} | \mathbf{p}_{s,i}) + \lambda \|\mathbf{e}_{t,i} - \mathbf{e}_{s,i}\|^2,$$

where  $\mathbf{p}_t = \mathcal{I}_{\theta_t}(\tilde{\mathbf{y}}, \mathbf{x}, n)$  and  $\mathbf{p}_s = \mathcal{I}_{\theta_s}(\tilde{\mathbf{y}}, \mathbf{x}, 1)$ .

The hyper-parameter  $\lambda$  controls the contribution of hidden state loss. When the distillation is completed, the student is used for inference.

### 3.2 EMA Update of the Teacher

As the distillation progresses, the performance gap between multiple steps of the teacher and a single-pass of the student shrinks, making the teacher’s labels less informative. Two approaches can be considered to sustain the usefulness of the teacher’s labels: **i)** Increasing the number of teacher’s iterative steps. **ii)** Restarting the distillation where the recently optimized student becomes the new teacher and repeating this process several times, i.e.  $\theta_t^{(n)} \leftarrow \theta_s^{(n-1)}$ . The former makes the training more expensive as the number of sequential steps grows, and the latter requires repeated distillation rounds leading to a longer training time.

Instead, we propose updating the teacher with the student’s recently learned knowledge. As the student’s single-step output approaches the teacher’s multi-step, the student’s multi-step performance would improve as well, and it is beneficial to use the improved student as the new teacher. However, replacing the teacher directly with the student would hurt the training stability, and can

lead to a pathological solution of mapping everything to a constant vector. This degenerate solution shortcuts the  $\mathcal{L}_{\text{DiMS}}$  loss by setting it to a global minimum of zero. To alleviate this, we update the teacher with a slow-exponential-moving average of the student, which transfers the new knowledge learned by the student to the teacher in a controlled manner. The updated teacher now provides a better training target for the student, creating a positive feedback loop between the two models. The teacher also benefits from the ensembling effects of the EMA (Izmailov et al., 2018). Algorithm 1 outlines the steps for DiMS training with EMA.

## 4 Experiments

### 4.1 Experimental Setup

We use Fairseq (Ott et al., 2019) for all the experiments and follow the default data splits. All models are Transformers with encoder-decoder architecture, each having 6 layers and 512-dimensional hidden states. Adam optimizer with inverse squared root learning rate scheduler is used along with mixed precision. EMA and hidden state loss are leveraged with two iterative steps of the teacher unless otherwise stated. We use early stopping based on single-step BLEU score on the validation set. The final model is the average of 5 best checkpoints. Dropout is disabled for the teacher and the student since empirical improvements are observed. We conduct experiments on both the raw and distilled dataset that is obtained from an autoregressive model (Gu et al., 2018). Training is done with 4 Tesla V100 GPUs (32 GB) and we report all the hyper-parameters in Section C of the appendix. The extra computational cost of distillation is a small fraction of original training. We report a detailed comparison in Section E of the appendix.

	Model	WMT'14		WMT'16	
		En-De	De-En	En-Ro	Ro-En
XE-Based	CMLM (Ghazvininejad et al., 2019)	18.1	21.8	27.3	28.2
	SMART (Ghazvininejad et al., 2020b)	18.6	23.8	-	-
	CMLMC (Huang et al., 2021)	19.6	23.6	28.2	29.0
	Aux. Reg. (Wang et al., 2019)	20.7	24.8	-	-
	Bag-of-ngram (Shao et al., 2020)	20.9	24.6	28.3	29.3
	Hint-based Loss (Shao et al., 2020)	21.1	25.2	-	-
	Bigram CRF (Sun et al., 2019)	23.4	27.2	-	-
	EM+ODD (Sun and Yang, 2020)	24.5	27.9	-	-
	ENGINE (Tu et al., 2020)	-	28.1	-	28.2
	GLAT (Song et al., 2021)	25.2	29.8	31.2	32.04
	CMLMC + DiMS	26.7	31.1	33.2	33.6
Alignment-Based	Imputer (Saharia et al., 2020)	25.8	28.4	32.3	31.7
	AXE (Ghazvininejad et al., 2020a)	23.5	27.9	30.8	31.5
	OAXE (Du et al., 2021)	26.1	30.2	32.4	33.3
	AlignNART (Song et al., 2021)	26.4	30.4	32.5	33.1
	FullyNAT + CTC + GLAT (Gu and Kong, 2020)	27.2	31.4	33.7	34.2
	DAT (Huang et al., 2022b)	27.3	31.3	-	-

Table 1: Comparison of the single-step test set BLEU score with previously published works.

## 4.2 Main Results

Our main experiments are conducted on WMT'14 En-De and WMT'16 En-Ro datasets with two models: **i)** CMLM, a pivotal work in iNAT literature showing the effectiveness of conditional masked language models. **ii)** CMLMC, a recent work improving CMLM by incorporating a correction mechanism. The corresponding official repositories are used to train the teachers. Both models exploit a length predictor that is conditioned on the encoder's hidden states. For CMLMC models we use encoder side masking and prediction (Guo et al., 2020) to further boost the performance of the teacher. To make the length predictor compatible with changes in the encoder, we keep the length predictor loss during distillation.

Figure 3 contrasts the single-step BLEU score of students with teachers evaluated for various number of decoding steps. DiMS considerably improves the translation quality of the single-step inference, reducing or eliminating the gap with multi-step inference. For example, on the WMT'14 De-En dataset, the single-step of CMLMC+DiMS surpasses the teacher's 4-step performance. We compared our best single-step model with strong baselines in Table 1 showing the effectiveness of our approach. DiMS outperforms all cross-entropy

based models and makes cross-entropy based models competitive with their alignment based counterparts.

## 4.3 Results on an Alignment Based Model

To show the versatility of DiMS, we conduct experiment on alignment-based models leveraging Connectionist Temporal Classification (CTC) (Graves et al., 2006) objective. Imputer (Saharia et al., 2020) is among a few models that are both alignment based and iterative. There is no official implementation of Imputer available, therefore we implement a version ourselves (denoted with †)<sup>1</sup>.

Table 2 summarizes the results of DiMS applied to Imputer for both directions of the WMT'14 English-German dataset. While DiMS boosts single step translation of Imputer, it still falls behind more recent alignment based models mentioned in Table 1. However, we believe if one incorporates various tricks introduced for alignment based models recently and create a better iterative model, then DiMS can be an effective tool to further enhance the single step translation. Details of Imputer training and distillation are explained in Section F of the appendix.

<sup>1</sup>Based on the following implementation: <https://github.com/rosinality/imputer-pytorch>



Method	WMT' 14 En-De	WMT' 14 De-En
Imputer <sup>†</sup>	25.9	29.0
Imputer <sup>†</sup> + DiMS	<b>26.4</b>	<b>29.8</b>

Table 2: Single-step test set BLEU score for Imputer models trained on WMT' 14 English-German.

#### 4.4 DiMS on Raw Dataset

The performance of the leading iNATs is at best similar to the autoregressive model used for sequence level knowledge distillation. This limits the final performance of iNATs and makes training without distillation desirable (Huang et al., 2021). Table 3 shows that DiMS improves the raw performance by a large margin even more than the corresponding distilled variant. For instance, DiMS gets more than 12 BLEU scores improvements on single-step evaluation of CMLMC.

For one decoding pass, when raw variants of CMLMC are distilled with DiMS the performance is superior to training on the distilled dataset (without DiMS). This makes DiMS preferable to sequence-level knowledge distillation. Nevertheless, the best performance is obtained when the two distillation approaches are combined.

	CMLMC	
	Teacher	Student
En-De	11.7	23.2
De-En	16.4	29.3
En-Ro	21.4	29.3
Ro-En	21.8	32.7

Table 3: Comparison of student and teacher on raw dataset.

#### 4.5 Unsupervised DiMS

In previous sections, we assume access to a parallel dataset and feed a partially masked reference sentence to both student and teacher. One can use the teacher to generate synthetic target sentences during the distillation. This relaxes the dependence on the references and enables using monolingual datasets for distillation. As usual, there is a trade-off between computation and sample quality i.e. using more decoding passes leads to better data while increasing the computational requirements.

We refer to this unsupervised distillation variant as U-DiMS. Note that unsupervised only refers to the distillation, and for training the teacher we still require access to a parallel dataset. The only dis-

Method	WMT' 14 De-En
CMLM	22.77
CMLM + U-DiMS	29.45
CMLM + DiMS	<b>29.74</b>
CMLMC	23.63
CMLMC + U-DiMS	30.52
CMLMC + DiMS	<b>30.81</b>

Table 4: Single-step test set BLEU score for models trained with U-DiMS.

inction between U-DiMS and DiMS is the usage of synthetic data generated by the teacher and the remaining parts are untouched. We run U-DiMS on WMT' 14 De-En for CMLM and CMLMC using two iterative steps to generate the synthetic samples. Table 4 shows the effectiveness of U-DiMS, obtaining a similar performance to DiMS.

Method	1-Step BLEU
CMLM	25.77
CMLM + DiMS	30.85
CMLM + DiMS - Hidden.	28.69
CMLM + DiMS (T=4)	31.04
CMLM + DiMS (T=8)	30.97
CMLM + DiMS + EMA	<b>31.63</b>
CMLM + DiMS (T=4) + EMA	31.52
CMLM + DiMS (T=8) + EMA	31.36

Table 5: BLEU score on WMT'16 En-Ro validation set with beam length set to one as its done for early stopping. T stands for the number of teacher decoding steps and is set to two if not specified.

#### 4.6 Ablation Studies

We conduct all the ablation studies on CMLM over WMT' 16 En-Ro as it is smaller than WMT' 14 and validation set is used for evaluation.

##### 4.6.1 Hidden State Loss

To investigate the effects of hidden state loss, we conduct an ablation study in this section. The first block in Table 5 includes BLEU scores for the base DiMS model with and without this term. The single-step performance of the distilled model is improved over 2 BLEU points by leveraging this loss. This supports the fact that the hidden states contain extra information that is not available in soft labels. The exact value of  $\lambda$  is selected based on a grid search reported in Section D of the ap-

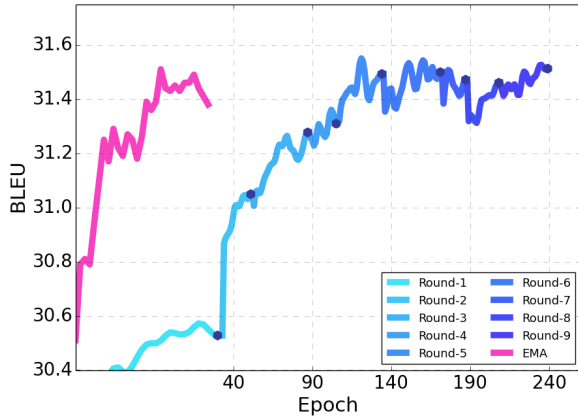


Figure 4: Validation set BLEU score on WMT’16 En-Ro for iterative distillation and a EMA model. Each round of iterative distillation is shown with a unique color and the end of the round is noted by a black dot. The number of steps differs in various rounds as we use early stopping.

pendix.

#### 4.6.2 EMA

In order to establish the computational advantages of the slow-moving average, we compare it with running the base variant for 9 iterative rounds. Figure 4 demonstrates that the EMA variant is able to match the iterative distillation with far fewer updates (almost equal to one round of the distillation).

We observed that it is essential to move the teacher toward the student slowly. For example, when  $\mu \leq 0.9$ , the collapse to a degenerate solution (explained in Section 3.2) occurs before the end of the first epoch. We plot the validation curve for various values of  $\mu$  in Section B of the appendix showing the importance of the slow-moving average.

#### 4.6.3 Teacher Decoding Steps

One hyper-parameter in DiMS algorithm is the number of teacher’s decoding steps. In order to investigate the effect of this hyper-parameter, we set it to 2, 4, and 8 while turning EMA on and off. The two bottom blocks of Table 5 include the results of this ablation. Although running the teacher for 4 decoding steps shows superior performance without EMA, as soon as we turn it on the gap disappears. This shows that EMA can gradually improve the teacher and remove the need for several iterative steps. Thus, we find no reason to set this hyper-parameter larger than 2 as it only increases distillation’s computational cost.

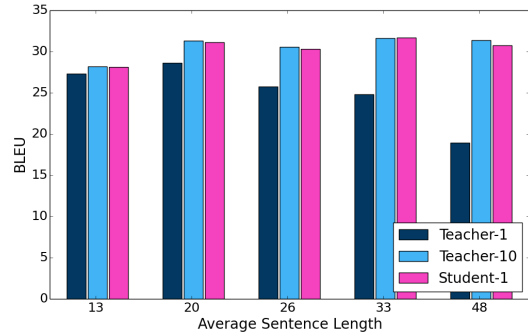


Figure 5: Test set BLEU score on WMT’14 De-En based on the target sentence length for CMLM teacher and student.

#### 4.7 Analysis

We study the effect of target sentence lengths on DiMS performance. The test set is divided into five equally-sized buckets based on the target length. The BLEU scores are reported for each bucket in Figure 5. The main benefit of the iterative model is manifested by large sentences. The reason might be the fact that longer sentences require a context and modeling it becomes challenging with the conditional independence assumption in NAT. It is clear in Figure 5, that the performance is improved in every bucket. This improvement is most visible in the bucket with the highest average sentence length. This is because of the fact that the same bucket has the largest gap between the teacher’s single and multi-step evaluation.

We combine the length predictor objective with ours to account for changes in the encoder’s parameters. Interestingly enough, DiMS improves the performance of the length predictor as depicted in Figure 6. This shows that the encoder benefits from the distillation as well.

Table 6 shows a qualitative example from the WMT’14 De-En dataset. The improvements in samples are evident by comparing the predictions of the teacher and the student with the target sentence. We provide more qualitative examples in the appendix.

#### 5 Related Works

Many techniques have been proposed for iterative non-autoregressive machine translation. Earlier attempts include denoising autoencoder (Lee et al., 2018) and insertion-deletion (Stern et al., 2019; Gu et al., 2019). More recently, Ghazvininejad et al. (2019) introduced the Mask-Predict improving the performance of iNATs by employing a conditional masked language model. CMLMC (Huang

et al., 2021) and SMART (Ghazvininejad et al., 2020b) improve CMLM by incorporating a correction mechanism. DisCo (Kasai et al., 2020b) is another variant conditioning each token on an arbitrary subset of the other tokens. DiMS is entangled with the progress in this domain as it requires a pre-trained iterative teacher.

The position constraint in cross-entropy can make the NAT training challenging, therefore Ghazvininejad et al. (2020a) propose aligned cross-entropy (AXE), an objective that considers the best monotonic alignment between the target and the model’s predictions. Du et al. (2021) relaxes the monotonic assumption and introduces Order Agnostic Cross-Entropy (OAXE). CTC (Libovický and Helcl, 2018) is a similar alignment-based objective that fixes the model output length and considers various alignments leading to the same target. Imputer (Saharia et al., 2020) extends CTC to benefit from iterative refinements.

GLAT (Qian et al., 2021) shows that the optimization challenges of iNATs can be mitigated by introducing a curriculum learning focusing on sentences with only a few masked tokens in the early stages of the training and gradually increasing the masking ratio. ENGINE (Tu et al., 2020) assumes access to a pre-trained autoregressive model and optimizes a NAT model to maximize the likelihood under the probability distribution defined by the pre-trained model.

Salimans and Ho (2021) applies a distillation technique similar to DiMS on generative models to decrease the number of required steps for generating high-quality images. In contrast to DiMS, the distillation is applied progressively. DiMS eliminates the need for progressive distillation by updating the teacher with EMA. Lastly, the proposed EMA has some resemblance to self-supervised learning techniques (Grill et al., 2020; Caron et al., 2021; He et al., 2020) where two models are updated, one through gradient-based optimization and the other one through EMA. Despite this similarity, the motivations are quite different. In self-supervised learning, EMA is proposed as a technique to remove large negative sets whereas here EMA enhances the quality of the labels generated by the teacher.

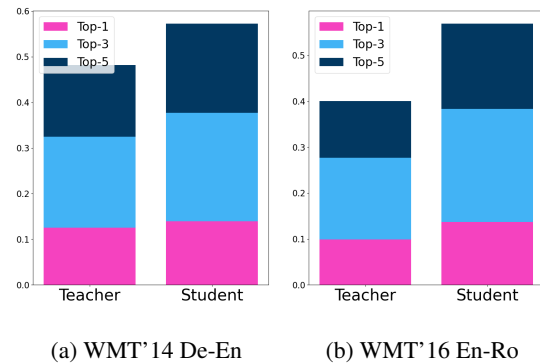


Figure 6: Comparison of CMLM teachers and students in predicting the target length. Top-1 means predicting the target correct and Top-3 and Top-5 means being incorrect by 1 and 2 offsets.

## 6 Discussion

It is not completely clear why knowledge distillation works in general (Zhou et al., 2019; Huang et al., 2022a). But when it comes to DiMS, we hypothesize that the labels generated by the teacher make the task simpler for the student. In other words, it is difficult for the model to close the gap between its single step prediction and ground truth while distillation with teacher-generated labels reduces this gap. The importance of the gap between labels and the model capacity has also been observed before (Mirzadeh et al., 2020).

## 7 Conclusion

We introduce DiMS, an effective distillation algorithm that enhances the single-step translation quality of a pre-trained iterative model. This is done by replicating the model’s multi-step behavior through one decoding pass. The distillation can be repeated to achieve greater gains, but this increases the training time noticeably. We show that the same benefits are obtainable by setting the teacher as a moving average of the student while keeping the training time comparable to one round of the distillation. Experiments over raw and distilled datasets on four translation tasks for supervised and unsupervised variants validate the effectiveness and versatility of DiMS.

Potential directions for future works include: **i)** The same family of iterative models have been applied to automatic speech recognition, thus DiMS is applicable to this domain. **ii)** One can combine a pyramid of techniques introduced for iNATs to obtain a strong iterative model and make it computationally efficient via DiMS. **iii)** Large monolingual



Target	The antibodies hunt down any nicotine molecules in the bloodstream , neutralising them before they reached the brain , preventing a smoker from getting a nicotine hit .
Teacher	The antibodies hunt the nicotine molecules molecblood neutralize them before reach brain a smoker not experience high nicotine .
Student	The antibodies hunt the nicotine molecules in the blood and neutralize them before they reach the brain , so a smoker does not experience a nicotine high .

Table 6: A qualitative example from WMT’14 De-En along with teacher and student’s predictions on CMLMC.

sets can be used to distill models with U-DiMS.

## Limitations

While DiMS makes the cross-entropy based family competitive with alignment based variants, it still falls behind one some cases. Moreover, DiMS can improve the performance of models trained on raw data, but the best performance is still achieved when DiMS is applied on distilled datasets. Therefore, DiMS still depends on an auto-regressive model for the best translation quality.

## Acknowledgments

We are grateful to Xiao Shi Huang for insightful comments and reviewing the initial draft of this paper. We also thank Panteha Naderian for helpful conversations.

## References

- Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. 2021. Emerging properties in self-supervised vision transformers. In *CVPR*.
- Cunxiao Du, Zhaopeng Tu, and Jing Jiang. 2021. Order-agnostic cross entropy for non-autoregressive machine translation. In *ICML*.
- Marjan Ghazvininejad, Vladimir Karpukhin, Luke Zettlemoyer, and Omer Levy. 2020a. Aligned cross entropy for non-autoregressive machine translation. In *ICML*.
- Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. In *EMNLP-IJCNLP*.
- Marjan Ghazvininejad, Omer Levy, and Luke Zettlemoyer. 2020b. Semi-autoregressive training improves mask-predict decoding. *arXiv preprint:2001.08785*.
- Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal

classification: labelling unsegmented sequence data with recurrent neural networks. In *ICML*.

- Jean-Bastien Grill, Florian Strub, Florent Alché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. 2020. Bootstrap your own latent-a new approach to self-supervised learning. *NeurIPS*.
- Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *ICLR*.
- Jiatao Gu and Xiang Kong. 2020. Fully non-autoregressive neural machine translation: Tricks of the trade. *arXiv preprint:2012.15833*.
- Jiatao Gu, Changan Wang, and Junbo Zhao. 2019. Levenshtein transformer. *NeurIPS*.
- Junliang Guo, Linli Xu, and Enhong Chen. 2020. Jointly masked sequence-to-sequence model for non-autoregressive neural machine translation. In *ACL*.
- Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *CVPR*.
- Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. 2015. Distilling the knowledge in a neural network. *arXiv preprint: 1503.02531*.
- Fei Huang, Tianhua Tao, Hao Zhou, Lei Li, and Minlie Huang. 2022a. On the learning of non-autoregressive transformers. In *ICML*.
- Fei Huang, Hao Zhou, Yang Liu, Hang Li, and Minlie Huang. 2022b. Directed acyclic transformer for non-autoregressive machine translation. *arXiv preprint: 2205.07459*.
- Xiao Shi Huang, Felipe Pérez, and Maksims Volkovs. 2021. Improving non-autoregressive translation models without distillation. In *ICLR*.
- Pavel Izmailov, Dmitrii Podoprikin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. 2018. Averaging weights leads to wider optima and better generalization. *arXiv preprint: 1803.05407*.

- Jungo Kasai, James Cross, Marjan Ghazvininejad, and Jiatao Gu. 2020a. Non-autoregressive machine translation with disentangled context transformer. In *ICML*.
- Jungo Kasai, Nikolaos Pappas, Hao Peng, James Cross, and Noah Smith. 2020b. Deep encoder, shallow decoder: Reevaluating non-autoregressive machine translation. In *ICLR*.
- Jason Lee, Elman Mansimov, and Kyunghyun Cho. 2018. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *EMNLP*.
- Jason Lee, Raphael Shu, and Kyunghyun Cho. 2020. Iterative refinement in the continuous space for non-autoregressive neural machine translation. In *EMNLP*.
- Jindřich Libovický and Jindřich Helcl. 2018. End-to-end non-autoregressive neural machine translation with connectionist temporal classification. In *EMNLP*.
- Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. 2020. Improved knowledge distillation via teacher assistant. In *AAAI*.
- 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. *arXiv preprint: 1904.01038*.
- Lihua Qian, Hao Zhou, Yu Bao, Mingxuan Wang, Lin Qiu, Weinan Zhang, Yong Yu, and Lei Li. 2021. Glancing transformer for non-autoregressive neural machine translation. In *ACL*.
- Chitwan Saharia, William Chan, Saurabh Saxena, and Mohammad Norouzi. 2020. Non-autoregressive machine translation with latent alignments. In *EMNLP*.
- Tim Salimans and Jonathan Ho. 2021. Progressive distillation for fast sampling of diffusion models. In *ICLR*.
- Chenze Shao, Jinchao Zhang, Yang Feng, Fandong Meng, and Jie Zhou. 2020. Minimizing the bag-of-ngrams difference for non-autoregressive neural machine translation. In *AAAI*.
- Jongyoon Song, Sungwon Kim, and Sungroh Yoon. 2021. AligNART: Non-autoregressive neural machine translation by jointly learning to estimate alignment and translate. In *EMNLP*.
- Mitchell Stern, William Chan, Jamie Kiros, and Jakob Uszkoreit. 2019. Insertion transformer: Flexible sequence generation via insertion operations. In *ICML*.
- Zhiqing Sun, Zhuohan Li, Haoqing Wang, Di He, Zi Lin, and Zhihong Deng. 2019. Fast structured decoding for sequence models. *NeurIPS*.
- Zhiqing Sun and Yiming Yang. 2020. An em approach to non-autoregressive conditional sequence generation. In *ICML*.
- Lifu Tu, Richard Yuanzhe Pang, Sam Wiseman, and Kevin Gimpel. 2020. Engine: Energy-based inference networks for non-autoregressive machine translation. In *ACL*.
- Yiren Wang, Fei Tian, Di He, Tao Qin, ChengXiang Zhai, and Tie-Yan Liu. 2019. Non-autoregressive machine translation with auxiliary regularization. In *AAAI*.
- Chunting Zhou, Graham Neubig, and Jiatao Gu. 2019. Understanding knowledge distillation in non-autoregressive machine translation. *arXiv preprint: 1911.02727*.

## A Teacher Comparison

Table 7 compares teachers trained by us with the original work proposing the model.

## B EMA Momentum Effect

We showcase the importance of the slow moving average in Figure 7. As we increase the momentum the training becomes more stable and leads to a better validation set BLEU score.

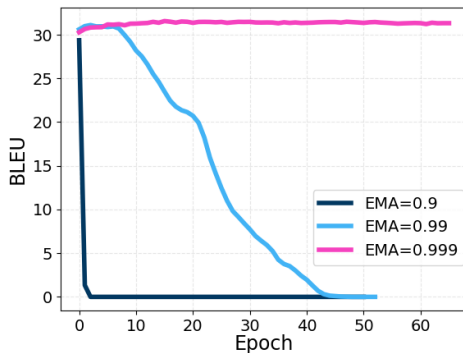


Figure 7: Validation BLEU curve on WMT'16 En-Ro for various EMA momenta with length beam=1.

## C Hyper-parameters for Distillation

We use the same hyper-parameters for all the datasets.

Hyper-parameter	CMLM/CMLMC
Learning rate ( $\eta$ )	1e-3
Adam $\beta$	(0.9, 0.98)
Warm-up updates	0
Max-tokens/GPU	8192
EMA momentum ( $\mu$ )	0.9992
Hidden state loss factor( $\lambda$ )	0.7
Length loss factor	0.1
Mask policy	Uniform
Temperature	0.5

Table 8: Summary of hyper-parameters.

## D Ablation on Hidden State Loss Coefficient

The importance of the hidden state loss is shown in Section 4.6.1 of the main body. We conduct an ablation study in this section to find the optimal value of  $\lambda$  that controls the contribution of the hidden state loss.

## E Computational Cost

During the distillation we have to run teacher for two steps which adds extra computation. More

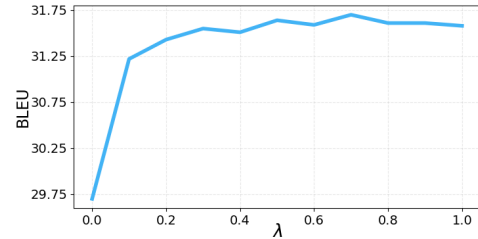


Figure 8: Best validation BLEU on WMT'16 En-Ro for CMLM with various hidden state loss coefficient ( $\lambda$ )

specifically on a machine with 4 Nvidia-V100 32GB GPUs, De-En training takes approximately 11 minutes per epoch compared to 27 minutes for distillation and on En-Ro dataset training and distillation take 2 and 8 minutes per epoch, respectively. However, the number of epochs for distillation is significantly less than teacher training. Precisely, teacher training takes 250 and 200 on De-En and En-Ro datasets respectively while distillations takes 10 epochs for De-En and 30 epochs for En-Ro. Figure 9 compares the overall time for training and distillation on De-En and En-Ro datasets and it shows that the distillation time is one order of magnitude smaller than training time.

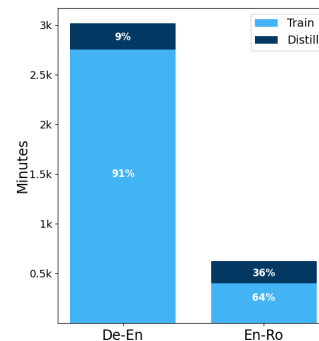


Figure 9: Comparison between teacher training and distillation time.

Note that teacher is being run in the evaluation mode, thus the activations maps are not kept in the memory. Therefore, the teacher can be run with a larger batch-size which further reduces the computational costs. We leave this as future works as it adds implementation complexity.

## F Imputer Details

As mentioned in the main body, there is no official implementation of Imputer available online. Here, we explain the differences between our implementation and the original paper. Imputer proposes a

Method	Iteration	WMT' 14		WMT' 16	
		En-De	De-En	En-Ro	Ro-En
CMLM(Ghazvininejad et al., 2019)	10	27.0	30.5	33.1	33.3
CMLM	10	26.9	31.2	33.1	33.6
CMLMC(Huang et al., 2021)	10	28.4	31.4	34.6	34.1
CMLMC	10	27.3	31.2	34.1	34.0
Imputer(Saharia et al., 2020)	8	28.2	31.8	34.4	34.1
Imputer <sup>†</sup>	8	28.5	31.3	-	-

Table 7: Comparison of our teachers with the numbers reported in the original papers.

pre-training phase where the model is optimized merely with the CTC objective. We find it unnecessary as the model reaches a better or competitive performance without it. Imputer leverages a unified decoder rather than an encoder-decoder architecture incorporated here. For Imputer training, computing the alignment with the highest probability is necessary. This increases the training cost and (Saharia et al., 2020) proposes either a pre-processing stage or using a stale copy of the active model to manage the extra computation. We compute the best alignment on the fly as it is still computationally feasible. Similar to Imputer inference, extra care is taken to make sure consecutive tokens are not unmasked in the same step. Instead of a Bernoulli masking policy during training, we used a block masking policy.

For the distillation, Imputer mainly benefits from two iterative steps and the gains are not as significant after that. Therefore, there is no incentive to use EMA.

### **CMLM WMT'14 De-En**

---

**Target** The rate of 3.1 per cent is indeed better than the previous year and is also better than in September , " however , we had hoped for more , " said Monika Felder - Bauer , acting branch manager of the Employment Agency in Sonthofen .

---

**Teacher** Although the quota was better 3.1 better than last year and better than September , we would hoped more , " Monika - Bauer Deputy of the Labour Agency in Sonthofen .

---

**Student** Although the quota at 3.1 % is better than last year and is also better than in September , " we would have hoped for more , " says Monika Felder - Bauer , deputy head of the Labour Agency in Sonthofen .

---

### **CMLM WMT'16 Ro-En**

---

**Target** we must ask these people to learn the language , try to appropriate our values , to stop having one foot in europe and one in their home country , bringing the rest of the family including through marriages of convenience .

---

**Teacher** let us ask these people to learn their language , try to take values , stop longer stand in europe and with one their country home origin , bringing the rest of family , including through convenience marriages .

---

**Student** let us ask these people to learn the language , try to take over values , no longer stand in europe and with one in their home country , bringing the rest of the family , including through convenience marriages .

---

### **CMLMC WMT'14 De-En**

---

**Target** Edward Snowden , the US intelligence whistleblower , has declared that he is willing to travel to Berlin to give evidence to the German parliament if the US National Security Agency and its director Keith Alexander fail to provide answers about its activities .

---

**Teacher** Edward Snowden , the whistleblower of the US intelligence , has that he is to travel to Berlin and testify the German destag if the National Security Agency its director Keith Alexander not provide answers about their activities .

---

**Student** Edward Snowden , the whistleblower of the US intelligence , has said that he is prepared to travel to Berlin and testify to the German destag if the American National Security Agency and its director Keith Alexander do not provide answers to their activities .

---



**CMLMC WMT'16 Ro-En**

---

**Target** during the routine control , the border policemen observed in the truck 's cab , a large travel bag , which contained personal things of many people , which is why they conducted a thorough check on the means of transport .

---

**Teacher** at specific control , border police officers a large travel of travel the cabvan , where there were things for several people , which is why carried out thorough control over the vehicle of transport .

---

**Student** at specific control , border police observed , in the cabin , a large travel getravel , where there were personal things for several people , which is why carried out thorough control over the vehicle of transport .

## ACL 2023 Responsible NLP Checklist

---

### A For every submission:

- A1. Did you describe the limitations of your work?  
*In the Limitation section after conclusion.*
- A2. Did you discuss any potential risks of your work?  
*The proposed method is to speed up translation (inference time) which does not seem to have a risk. Anything that the method makes possible was already possible with more computation at inference time.*
- A3. Do the abstract and introduction summarize the paper's main claims?  
*Left blank.*
- A4. Have you used AI writing assistants when working on this paper?  
*Left blank.*

### B Did you use or create scientific artifacts?

*Section 4 (experiments).*

- B1. Did you cite the creators of artifacts you used?  
*Section 4 (experiments).*
- B2. Did you discuss the license or terms for use and / or distribution of any artifacts?  
*Fairseq is under MIT license which is one of the permissive licenses.*
- B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?  
*The code will be released under MIT license.*
- B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?  
*The used data were public datasets. (Namely WMT14 and WMT17)*
- B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?  
*Not applicable. Left blank.*
- B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.  
*Not applicable. Left blank.*

### C Did you run computational experiments?

*Section 4 (experiments)*

- C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?  
*Appendix E contains computational cost. Since we did not introduce new model and distilled existing ones the details of those models can be found in the original papers.*

---

*The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.*

- C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?

*Section 4 (experiments) and Appendix C.*

- C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?

*It is transparent that a single run is reported.*

- C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?

*Section 4 (experiments). We used Fairseq.*

**D  Did you use human annotators (e.g., crowdworkers) or research with human participants?**

*Left blank.*

- D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?

*Not applicable. Left blank.*

- D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?

*Not applicable. Left blank.*

- D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?

*Not applicable. Left blank.*

- D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?

*Not applicable. Left blank.*

- D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?

*Not applicable. Left blank.*