CTC-based Non-autoregressive Speech Translation

Chen Xu[1†], Xiaoqian Liu[1], Xiaowen Liu[1], Qingxuan Sun[1], Yuhao Zhang[1], Murun Yang[1], Qianqian Dong[2], Tom Ko[2], Mingxuan Wang[2*], Tong Xiao[1,3*], Anxiang Ma[1,3], Jingbo Zhu[1,3]
[1]School of Computer Science and Engineering, Northeastern University, Shenyang, China
[2]ByteDance
[3]NiuTrans Research, Shenyang, China
{xuchennlp, liuxiaoqian0319, liuxiaowenneu}@outlook.com
{dongqianqian, tom.ko, wangmingxuan.89}@bytedance.com
{xiaotong, maanxiang, zhujingbo}@mail.neu.edu.cn

Abstract

Combining end-to-end speech translation (ST) and non-autoregressive (NAR) generation is promising in language and speech processing for their advantages of less error propagation and low latency. In this paper, we investigate the potential of connectionist temporal classification (CTC) for non-autoregressive speech translation (NAST). In particular, we develop a model consisting of two encoders that are guided by CTC to predict the source and target texts, respectively. Introducing CTC into NAST on both language sides has obvious challenges: 1) the conditional independent generation somewhat breaks the interdependency among tokens, and 2) the monotonic alignment assumption in standard CTC does not hold in translation tasks. In response, we develop a prediction-aware encoding approach and a cross-layer attention approach to address these issues. We also use curriculum learning to improve convergence of training. Experiments on the MuST-C ST benchmarks show that our NAST model achieves an average BLEU score of 29.5 with a speed-up of 5.67×, which is comparable to the autoregressive counterpart and even outperforms the previous best result of 0.9 BLEU points[1].

1 Introduction

End-to-end speech translation (E2E ST) has attracted unprecedented attention and achieved dramatic development in recent years (Duong et al., 2016; Berard et al., 2016; Weiss et al., 2017; Anastasopoulos and Chiang, 2018; Wang et al., 2020b,c; Xu et al., 2021; Zhang et al., 2022b). Stand-alone modeling reduces the inference latency by almost half compared to cascaded systems, where the automatic speech recognition (ASR) model and the machine translation (MT) model run serially. This helps the application in real scenarios, especially with limited computational resources.

However, this advantage only holds in the context of autoregressive (AR) decoding, where each token is generated depending on the previously predicted results. Non-autoregressive (NAR) generation (Gu et al., 2018), the recently popular decoding method in ASR and MT, makes the inference process fast by predicting the output sequence in parallel, resulting in the E2E ST no longer being superior in terms of inference speed-up.

A natural question arises: can we build a powerful non-autoregressive speech translation (NAST) model? The NAR results in the latest literature are still inferior to the AR counterparts with a large gap of about 2 $\sim$ 3 BLEU points, even with the iterative refinement process (Inaguma et al., 2021a). In this work, we aim to develop a promising NAST model for comparable performance to the AR model without complex decoding.

We resort to the connectionist temporal classification (CTC, Graves et al., 2006) because of its great success in ASR and MT and the convenience of variable length prediction. CTC is well suited for speech-to-text modeling, where the input sequence is longer than the output. Recent studies show that CTC-based NAR models achieve comparable or even better performance than their AR counterparts, providing insight into the design of the powerful CTC-NAST model.

Our CTC-NAST model is decoder-free and consists of two stacked encoders: an acoustic encoder and a textual encoder. They are guided by CTC to

---
*Corresponding author.
†Work was done while at ByteDance AI Lab.
[1]The code is available at https://github.com/xuchennlp/S2T.

predict transcription and translation, respectively (Chuang et al., 2021). Then, we carry out a careful and systematic inspection of the underlying issues and address the challenges of CTC-NAST. In particular,

- The conditional independence assumption allows fast inference but omits interdependency across the whole sequence. We identify the *prediction-aware encoding* (PAE) method underlying the success of a series of studies (Nozaki and Komatsu, 2021; Huang et al., 2022; Higuchi et al., 2021a), which observe preliminary prediction and refine it in the final generation. Following this idea, we predict the CTC result in the intermediate layer and then integrate it into the subsequent encoding.

- Another inherent property of CTC, the monotonic assumption, is valid for ASR but does not hold for translation tasks, where a future word in the target text may be aligned with the earlier part of the source text, especially on distant language pairs (Hannun, 2017). A critical requirement of the decoder-free design is the *reordering augmentation* (Chuang et al., 2021). As a remedy, we introduce an additional cross-layer attention module, which is complementary to the self-attention module.

Even with the above efforts, NAST is still a difficult task that suffers from heavy modeling burdens. A *curriculum learning strategy* that guides the training in an easy-to-hard way is significant for better convergence. We replace part of the incorrect prediction with ground truth in PAE to prompt the generation of the whole sequence. In this way, the model relieves the CTC learning burden by observing almost the whole sequence in the early stages, while only a few tokens are replaced as CTC performance improves, ensuring consistency between training and inference.

Our CTC-NAST model is simple, completely parallel, and works well for both similar and distant language pairs. The proposed methods yield a remarkable gain of 3.0 BLEU points on MuST-C En-De, achieving an average BLEU score of 29.5 with an inference speed-up of 5.67×, and even outperforming the best previous AR results by 0.9 BLEU points. We also report competitive results on the more challenging MuST-C En-Ja and Fisher-Callhome corpus.

2 Background

2.1 Connectionist Temporal Classification

CTC (Graves et al., 2006) was originally proposed for labeling unsegmented sequences. It learns monotonic alignment between acoustic features and transcriptions, which is valid for cross-modal learning like ASR. CTC helps convergence and allows re-scoring decoding through a lightweight output layer, achieving great success in ASR as an auxiliary loss on top of the encoder (Watanabe et al., 2017; Karita et al., 2019). Given the encoder representation h and the corresponding sequence y, the CTC loss is defined as:

$$\mathcal{L}_{\mathrm{CTC}} = -\log \mathrm{P}_{\mathrm{CTC}}(y|h) \tag{1}$$

where the probability is calculated by marginalizing over all possible alignments $\Phi(y)$ between h and y:

$$\mathrm{P}_{\mathrm{CTC}}(y|h) = \sum_{\pi \in \Phi(y)} \mathrm{P}(\pi|h) \tag{2}$$

CTC has the same conditional independence property as NAR generation, where the probability of the path π is the product of the probability $P(\pi_t|h_t)$ at each time step t:

$$\mathrm{P}(Y|X) \approx \prod_{t=1}^{T} \mathrm{P}(\pi_t|h_t) \tag{3}$$

where T is the length of h.

2.2 AR and NAR

Given a source sequence $X = (x_1, \cdots, x_{T'})$, a sequence-to-sequence model predicts the target sequence $Y = (y_1, \cdots, y_T)$ by conditional distribution:

$$\mathrm{P}(Y|X;\theta) = \prod_{t=1}^{T} \mathrm{P}_{\mathrm{AR}}(y_t|y_{<t}, X;\theta) \tag{4}$$

where θ is the model parameters. This autoregressive generation learns sequential dependency but suffers from high inference latency.

Instead, NAR carries out the conditional independent prediction for parallel inference (Gu et al., 2018):

$$\mathrm{P}(Y|X;\theta) = \prod_{t=1}^{T} \mathrm{P}_{\mathrm{NAR}}(y_t|X;\theta) \tag{5}$$

Although the vanilla NAR model speeds up inference by about $15\times$ (Gu et al., 2018), it is still inferior to the AR counterpart by a large gap.

Researchers have proposed many series of methods to improve the generation quality and investigate a better trade-off between performance and speed in the MT task, such as the iterative decoding method (Lee et al., 2018; Stern et al., 2019; Ghazvininejad et al., 2019; Kasai et al., 2020), latent variable method (Gu et al., 2018; Song et al., 2021; Gu and Kong, 2021), data manipulation method (Zhou and Keung, 2020; Bao et al., 2022; Ding et al., 2021), enhancement based method (Guo et al., 2019; Wang et al., 2019), and semi-autoregressive decoding (Ran et al., 2020). There are also some studies to design the architecture of the NAR models, such as the use of CTC for prediction for its ability of variable length prediction (Libovický and Helcl, 2018; Shu et al., 2020; Saharia et al., 2020).

In addition, the NAR generation also shows promising results in ASR task, especially the CTC-based systems (Higuchi et al., 2020, 2021b; Lee and Watanabe, 2021; Nozaki and Komatsu, 2021; Kim et al., 2022).

2.3 Speech Translation

Recently, E2E ST has received a lot of attention due to its direct modeling (Berard et al., 2016). Unlike the conventional cascaded system that decouples the cross-modal and cross-lingual modeling into ASR and MT models respectively (Ney, 1999; Mathias and Byrne, 2006), the end-to-end manner is more elegant and has the potential for fast inference and error-free propagation.

One promising route to improve ST is to develop more adaptive architectures according to the task characteristics. Based on the idea of modeling decoupling, the stacked encoding method divides cross-modal and cross-lingual learning into acoustic and semantic encoders, respectively (Liu et al., 2020; Xu et al., 2021). In this design, the CTC loss for transcription is usually introduced to guide the learning of the acoustic encoder, which significantly helps convergence. In addition, the latent alignment learned in the CTC is used to bridge the two encoders. Liu et al. (2020) shrink the sequence length based on CTC prediction. Xu et al. (2021) introduce an adapter to bridge two encoders by integrating CTC prediction.

Several studies investigate the NAR generation in ST (Inaguma et al., 2021a,b; Chuang et al., 2021). However, current NAR systems are still inferior to AR counterparts, especially CTC-based systems. Researchers also continue to extend the use of CTC to learn target text as an auxiliary loss of the encoder (Zhang et al., 2022a; Yan et al., 2022). But there is no work to inspect the underlying issues in the CTC modeling of target text in ST. To this end, we study the challenges of building a powerful CTC-based NAST model and then propose corresponding methods. We also extend our method to AR models for a comprehensive exploration.

3 CTC-NAST

Among many well-established NAR designs for ASR or MT models, CTC is particularly suitable for ST modeling because the input length is remarkably longer than its output. In this section, we present CTC-NAST in detail. We first describe the base architecture, then identify and address three underlying challenges. See Figure 1 for an overview of our system.

3.1 Base Architecture

ST aims to translate audio in the source language to text in the target language directly. Let $(x; y^s; y^t)$ be a training sample of ST, where x is the input speech feature sequence, y^s is the corresponding transcription of x, and y^t is the translation in the target language. We assume that transcription is always available in our work.

We drop the decoder network and rely only on the CTC-based encoder. Following the design of SATE (Xu et al., 2021; Chuang et al., 2021), we decouple the encoding into an acoustic encoder and a textual encoder in a stack architecture, as shown in Figure 1(a). They are guided by CTC loss for transcription and translation (denoted CTC and XCTC for distinction), respectively.

Formally, given a representation h^a of the acoustic encoder output, the CTC loss is calculated as:

$$\mathcal{L}_{\text{CTC}} = -\text{log}\text{P}_{\text{CTC}}(y^s|h^a) \tag{6}$$

Similarly, the XCTC loss is calculated as:

$$\mathcal{L}_{\text{XCTC}} = -\text{log}\text{P}_{\text{XCTC}}(y^t|h^t) \tag{7}$$

where h^t is the representation of the textual encoder output.

Then, the training objective is formulated as the interpolation of the two CTC losses:

$$\mathcal{L} = \alpha_A \cdot \mathcal{L}_{\text{CTC}} + \alpha_T \cdot \mathcal{L}_{\text{XCTC}} \tag{8}$$

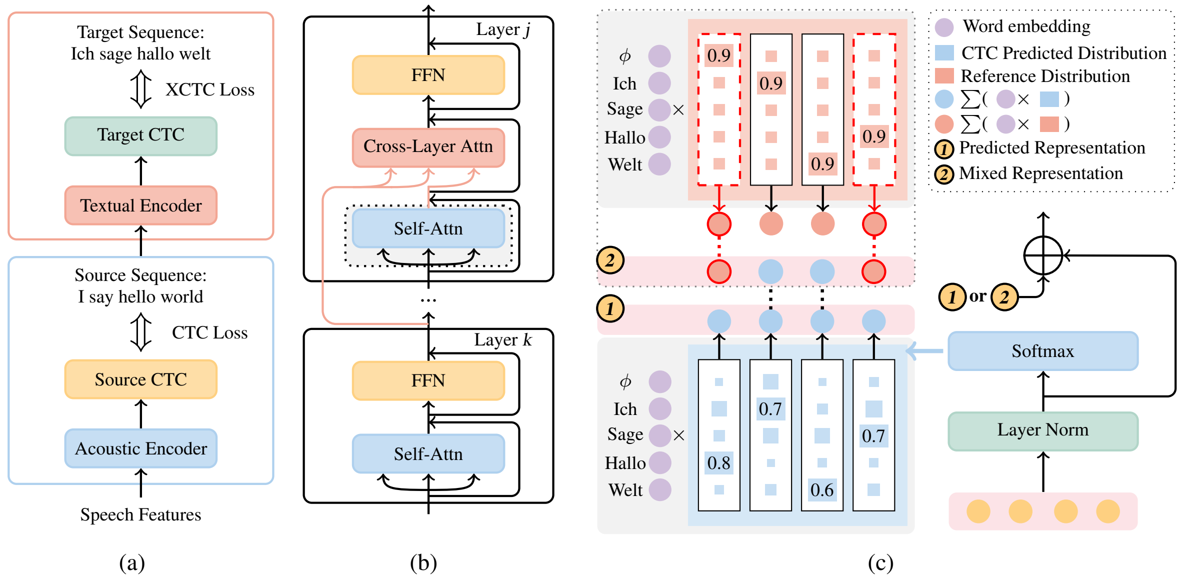

Figure 1: Overview of our CTC-NAST model. (a) The base architecture consisting of two CTC-guided encoders, (b): The cross-layer attention module, where the layer normalization is omitted for simplification, (c) Prediction-aware encoding, and its variant of curriculum learning mixing.

where α_{A} and α_{T} are the coefficients of the CTC and XCTC losses, respectively.

Although CTC works well for the NAR ASR model, extending CTC naively to the more challenging ST task is fragile. We claim that CTC-NAST can be improved by addressing three issues:

- **Conditional independence assumption** is an inherent property of CTC, which ignores interdependency with past or future contexts, leading to poor generation (Chan et al., 2020), like repetition and omission errors.

- Although the self-attention network has the modest reordering capability (Chuang et al., 2021), our encoder-only architecture is hard to handle the **monotonic assumption**, especially for distant language pairs.

- E2E ST already suffers from the heavy burden of cross-modal and cross-lingual mapping, while NAR modeling further aggravates the difficulty and results in **poor convergence**.

3.2 Prediction-aware Encoding

NAR generation enlarges the search space in inference due to conditional independence (Ran et al., 2021), especially with the long speech sequence of hundreds and thousands of units. A commonly-used solution, incorporating latent variables that contain the initial prediction into modeling, has been demonstrated to be effective (Lee et al., 2018). In this way, the NAR generation is decoupled as the multiple-step refinement of the target sequence, enabling the model to be aware of the previous prediction.

Inspired by the prior efforts in MT (Huang et al., 2022) and ASR (Nozaki and Komatsu, 2021), we introduce prediction-aware encoding (PAE). The detailed illustration is shown in Figure 1(c). Specifically, given one representation h^l outputted by the intermediate encoder layer l, PAE integrates the prediction information (corresponding ① in the Figure) into the following encoding explicitly by weighting the embedding matrix W over the current CTC distribution (called InterCTC) (Xu et al., 2021):

$$\mathrm{PAE}(h^l) = h^l + \mathrm{P}_{\mathrm{InterCTC}}(\pi|h^l) \cdot W \qquad (9)$$

where the weights W are shared in the whole network. Note that we use PAE to augment the learning of both CTC and XCTC.

Since the poor prediction leads to the risk of error propagation, we also optimize the InterCTC loss for guaranteed prediction:

$$\mathcal{L}_{\mathrm{InterCTC}} = -\log \mathrm{P}_{\mathrm{InterCTC}}(y|h) \qquad (10)$$

In this way, we ensure that CTC predicts well. However, the worse result for XCTC limits the benefits of PAE, which may result in negative effects. We alleviate this issue in Section 3.4.

Now, we re-formulate the training loss in Eq. 8 as:

$$\begin{aligned}\mathcal{L} &= \alpha_{\mathrm{A}} \cdot \mathcal{L}_{\mathrm{CTC}} + \alpha_{\mathrm{T}} \cdot \mathcal{L}_{\mathrm{XCTC}} \\ &+ \beta_{\mathrm{A}} \cdot \frac{1}{M} \sum_{m=1}^{m} \mathcal{L}_{\mathrm{InterCTC}}^{m} \\ &+ \beta_{\mathrm{T}} \cdot \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}_{\mathrm{InterXCTC}}^{n} \end{aligned} \tag{11}$$

where M and N are the numbers of the intermediate CTC and XCTC, β_{A} and β_{T} are the corresponding coefficients.

3.3 Reordering Augmentation

Vanilla Transformer generates each token by distributing the weight of the encoder-decoder attention module to the corresponding source part to be translated, which easily handles the order gap between languages. However, CTC modeling faces the intractable issue of reordering the representation into the target language order during encoding. Although previous studies have demonstrated that the MT or ST encoder can capture the global information (Yang et al., 2018; Xu et al., 2021), it is still difficult to rely only on the self-attention module to search the positions that contribute significantly to decoding (Chuang et al., 2021).

To enhance the reordering capability of CTC-NAST, we mimic the design of the decoder and introduce cross-layer attention (CLA) module, which is inserted between the self-attention module and the feed-forward module in the specific layers of the textual encoder, as shown in Figure 1(b). Let SA$(\cdot,\cdot,\cdot)$ and CLA$(\cdot,\cdot,\cdot)$ denote the self-attention and CLA modules, the new Transformer layer j can be formulated as:

$$h^{'} = h^{j-1} + \mathrm{SA}(h^{j-1}, h^{j-1}, h^{j-1}) \tag{12}$$

$$h^{'} = h^{'} + \mathrm{CLA}(h^{'}, h^{k}, h^{k}) \tag{13}$$

$$h^{j} = h^{'} + \mathrm{FFN}(h^{'}) \tag{14}$$

where h^k is the representation output from the layer $k(k < j)$.

In this way, CLA offers a remedy for the lacking attention, that captures the information from the bottom layer directly and is complementary to the self-attention module. Now the textual encoder acts as both a stack of the encoder and the decoder of the vanilla encoder-decoder model.

In order to further enhance the ability of CLA, we introduce the drop-net technique. In each layer containing the CLA module, we drop the self-attention module with a probability $p_{drop} \in [0, 1]$. Note that the self-attention module always keeps during inference.

3.4 Curriculum Learning Strategy

Even with the auxiliary encoding and improved design architecture, the CTC-NAST model still faces the difficulty of a heavy modeling burden, leading to poor convergence. Inspired by Qian et al. (2021), a curriculum learning strategy is remarkably important to reduce the dependency in the early stage and increase the difficulty along the training process.

As illustrated in Figure 1(c), we replace part of the prediction (corresponding ① in the Figure) in Eq. 9 with the ground truth (corresponding ② in the Figure), which mitigates the negative effects of error propagation caused by the poor XCTC performance in PAE and prompts the generation of the whole sequence. Unlike the same lengths between input and output in the decoder, the length of the input acoustic feature is remarkably longer than the corresponding text in CTC. Therefore, we take the best alignment computed by the model as the ground truth (Gu and Kong, 2021; Huang et al., 2022):

$$\hat{\pi} = \arg \max_{\pi \in \Phi(y)} \mathrm{P}(\pi | s; \theta^{'}) \tag{15}$$

where $\theta^{'}$ is the current model parameter. Note that the length of $\hat{\pi}$ is the same as the input.

Denote the replacement ratio as $r \in [0, 1]$, we uniformly sample a random variable U from $[0, 1]$:

$$\hat{P}_t = \mathbb{I}(U >= r) * p_t + \mathbb{I}(U < r) * \hat{\pi}_t \tag{16}$$

where $\mathbb{I}(\cdot)$ is the indicator function.

However, this strategy results in the inconsistency between training and decoding, where the ground truth is unavailable during decoding. To address this issue, Qian et al. (2021) adaptively determine the replacement ratio depending on the current prediction accuracy. But it does not work for CTC-NAST, as shown in Appendix B.3.

Considering the long input sequence in ST, a lower ratio may not provide sufficient prompt, but a higher ratio may result in a severe gap between training and decoding. Therefore, we limit that only the positions where a wrong prediction ($\arg\max p_t \neq \hat{\pi}_t$) occurs are replaced. In this way, we enable the large ratio throughout the whole training process. As the accuracy increases, more and

	Model	De	Es	Fr	It	Nl	Pt	Ro	Ru	Ja	Avg.	Speed-up
MT	Transformer (Ours)	30.8	35.6	43.3	31.6	35.8	37.9	30.1	20.0	16.5	33.1	-
AR	Transformer (Inaguma et al., 2021b)	23.1	-	33.8	-	-	-	-	-	-	-	-
	+ Seq-KD	24.4	-	34.6	-	-	-	-	-	-	-	-
	Transformer (Inaguma et al., 2021a)	22.8	27.8	33.3	23.3	27.3	-	-	-	-	-	-
	+ Seq-KD	24.3	28.9	34.5	24.2	28.4	-	-	-	-	-	-
	Conformer (Inaguma et al., 2021a)	25.0	30.5	35.5	25.4	29.7	-	-	-	-	-	-
	+ Seq-KD	26.3	31.0	36.4	25.9	30.6	-	-	-	-	-	-
	Fairseq ST (Wang et al., 2020a)	22.7	27.2	32.9	22.7	27.3	28.1	21.9	15.3	-	24.8	-
	NeurST (Zhao et al., 2021)	22.8	27.4	33.3	22.9	27.2	28.7	22.2	15.1	-	24.9	-
	XSTNet (Ye et al., 2021)	25.5	29.6	36.0	25.5	30.0	31.3	25.1	16.9	-	27.5	-
	STEMM (Fang et al., 2022)	25.6	30.3	36.1	25.6	30.1	31.0	24.3	17.1	-	27.5	-
	ConST (Ye et al., 2022)	25.7	30.4	36.8	26.3	30.6	32.0	24.8	17.3	-	28.0	-
	$\mathrm{M^3ST}$ (Cheng et al., 2022)	26.4	31.0	37.2	26.6	30.9	32.8	25.4	18.3	-	28.6	-
	CTC-Aug ST (Ours)	26.9	31.5	38.1	27.4	31.9	33.4	25.8	**18.7**	16.1	29.2	1.0×
	+ Seq-KD	**27.7**	**31.6**	**39.5**	**27.5**	**32.3**	**33.7**	**26.6**	**18.7**	**16.4**	**29.7**	1.0×
NAR	CTC (Inaguma et al., 2021b)	19.4	-	27.4	-	-	-	-	-	-	-	20.84×
	Orthros (Inaguma et al., 2021b)	23.9	-	33.1	-	-	-	-	-	-	-	2.39×
	CTC (Inaguma et al., 2021a)	24.1	29.0	34.6	24.3	28.5	-	-	-	-	-	13.83×
	Orthros - CTC (Inaguma et al., 2021a)	25.3	30.4	36.2	25.4	29.9	-	-	-	-	-	1.14×
	Orthros - CMLM (Inaguma et al., 2021a)	24.1	29.2	35.1	24.4	28.6	-	-	-	-	-	2.73×
	CTC-NAST (Ours)	**27.3**	**31.8**	**38.9**	**27.7**	**32.3**	**33.3**	**26.1**	**18.9**	**16.2**	**29.5**	5.67×

Table 1: BLEU scores on MuST-C corpora. The speed-up is calculated on the En-De corpus.

more positions rely on the model's predictions, and the guidance to the fewer positions with errors always remains stable for better convergence. We call this method curriculum learning mixing (CLM).

Finally, we smooth the ground truth to obtain a distribution similar to the CTC prediction, where the dominant probability is concentrated on the ground truth position, and the rest is evenly distributed among other tokens.

3.5 Inference

CTC-NAST is a fully parallel decoding model. The inference resembles the training process, except the CLM method is not used. We employ greedy decoding, where CTC picks the tokens with maximum probability in each time-step, then removes the blanks and repeated tokens for final translation.

4 Extension on AR model

Now a natural question arises: can our method proposed for the NAR model be used to improve the AR model? Our method produces better encoder representations for CTC prediction, but there is no evidence to demonstrate that the optimization of the CTC and the cross-entropy in the decoder are completely consistent. Excessive optimization of the encoder may interfere with the learning of the decoder.

To answer it, we adopt these techniques to the encoder-decoder counterpart (called CTC-Aug ST), to investigate the effects of different architectures. And the training loss is formulated as:

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_{\mathrm{S2S}} + \alpha_{\mathrm{A}} \cdot \mathcal{L}_{\mathrm{CTC}} + \alpha_{\mathrm{T}} \cdot \mathcal{L}_{\mathrm{XCTC}} \\ &+ \beta_{\mathrm{A}} \cdot \frac{1}{M} \sum_{m=1}^{m} \mathcal{L}_{\mathrm{InterCTC}}^{m} \\ &+ \beta_{\mathrm{T}} \cdot \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}_{\mathrm{InterXCTC}}^{n} \qquad (17)\end{aligned}$$

where $\mathcal{L}_{\mathrm{S2S}}$ is the cross-entropy loss of the decoder.

5 Experiments

We evaluate our method on the MuST-C and Fisher-Callhome benchmarks. Details about the datasets and model settings are described in Appendix A.

5.1 Main Results

The results on the MuST-C corpora in Table 1 show that our method significantly outperforms previous AR and NAR models. We achieve remarkable gains for all language pairs. Here we highlight several major breakthroughs: i) CTC-Aug ST is shown to be effective for the AR models, which gains an average of 0.6 BLEU points over the previous best work even without the augmentation of sequence-level knowledge distillation (Seq-KD) data. Note that not all proposed methods are used in CTC-Aug ST (see Section 5.2.2). ii) Our CTC-NAST

	Model	Fisher			Callhome		Speed-up
		dev	dev2	test	devtest	evltest	
MT	Transformer (Ours)	64.50	65.20	63.35	32.21	31.58	-
AR	Transformer + Seq-KD (Inaguma et al., 2021b)	-	-	50.32	-	19.81	-
	Transformer + Seq-KD (Inaguma et al., 2021a)	51.10	51.40	50.80	19.60	19.20	-
	Conformer + Seq-KD (Inaguma et al., 2021a)	54.70	55.40	54.10	21.50	21.00	-
	Transformer + MTL + ASR init. (Chuang et al., 2021)	48.27	49.17	48.40	17.26	17.45	-
	CTC-Aug ST (Ours)	53.61	54.07	53.69	22.16	21.33	1.0×
	+ Seq-KD	**55.39**	**55.88**	**55.09**	**23.09**	**22.92**	1.0×
NAR	CTC (Inaguma et al., 2021b)	-	-	45.97	-	15.91	20.84×
	Conformer - CTC (Inaguma et al., 2021a)	51.00	51.60	50.80	18.00	18.70	11.80×
	Orthros - CTC (Inaguma et al., 2021a)	54.00	54.80	54.10	21.00	20.80	1.09×
	Orthros - CMLM (Inaguma et al., 2021a)	51.30	52.20	51.20	20.90	20.40	2.70×
	Transformer - CTC (Chuang et al., 2021)	42.61	43.91	43.50	13.02	13.52	28.9×
	CTC + MTL (Chuang et al., 2021)	44.45	45.23	44.92	14.20	14.19	28.9×
	Mask - CTC (Higuchi et al., 2021a)	51.10	51.70	50.60	17.90	18.30	-
	Intermediate CTC (Higuchi et al., 2021a)	51.30	51.40	51.00	19.00	19.00	-
	Self-conditioned CTC (Higuchi et al., 2021a)	50.70	51.20	50.50	19.10	19.20	-
	CTC-NAST (Ours)	**55.21**	**55.92**	**54.71**	**23.43**	**23.30**	4.10×

Table 2: BLEU scores on Fisher-Callhome corpus.

models achieve comparable or better performance to the powerful AR counterparts on all 9 language pairs, with a high speed-up of 5.67×. Note that CTC-NAST achieves a higher speed-up under large batch sizes (see Section 5.2.4). iii) Referring to Appendix B.1, the En-Ja translation has a strong demand for reordering capability. Our method also works well on this challenging distant language pair, demonstrating the potential of CTC-NAST.

Similar results on Fisher-Callhome are shown in Table 2. Interestingly, the NAST model outperforms the AR counterpart with 0.3 $\sim$ 0.4 BLEU points on the out-of-domain Callhome sets. We find that the AR models miss some segments when translating the long sentences, while the CTC-NAST models still guarantee good performance, as shown in Appendix B.2. It demonstrates the robustness of our CTC-NAST model.

5.2 Analysis

Next, we study several interesting problems on MuST-C En-De and En-Ja datasets to investigate the effects on similar and distant languages. We present further analyses in Appendix B.

5.2.1 Performance over Sentence Lengths

Figure 2 shows the results of the AR and NAR models with and without the proposed methods on the MuST-C En-De corpus with respect to output lengths. The base NAR model performs much

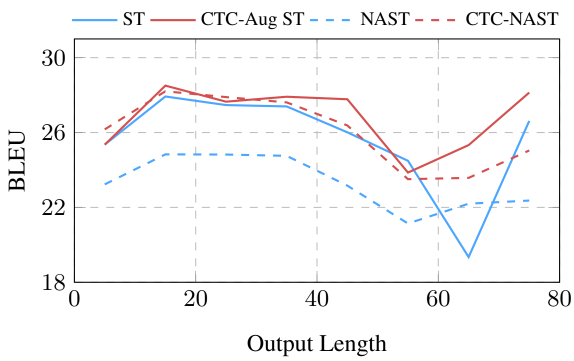

Figure 2: BLEU scores over various output lengths.

worse than AR counterpart. But interestingly, unlike the ST model, which has an outlier as sentence length increases, the NAST model maintains stable performance. This is similar to the results on Fisher-Callhome in Appendix B.2.

Our methods bring remarkable gains over different lengths for both AR and NAR models, leading to comparable translation quality when the length is less than 60. In particular, CTC-NAST performs even better than AR models when the length is less than 30. However, the performance gap increases with sentence length. We speculate that very long input acoustic features make it more difficult to model semantic information. Future work (Xu et al., 2023) can focus on enhancing the ability to handle complex acoustic encoding.

Model	En-De				En-Ja				Inference			Params.
	Raw		Seq-KD		Raw		Seq-KD		AR Times	NAR Times	Speed-up	
	AR	NAR	AR	NAR	AR	NAR	AR	NAR				
Base	26.1	-	27.1	-	15.9	-	16.1	-	547.2	-	-	∼ 130M
+ XCTC	26.7	17.3	27.0	24.3	16.3	7.3	16.3	13.7	555.0	79.9	6.95×	∼ 130M
+ PAE	26.9	19.6	**27.7**	25.7	16.1	8.5	16.4	14.9	545.0	84.1	6.48×	∼ 140M
+ CLA	26.8	19.1	27.3	26.2	**16.6**	10.0	16.4	15.3	565.6	91.8	6.16×	∼ 150M
+ CLM	26.6	25.7	27.5	**27.4**	14.4	14.3	**16.6**	16.1	543.1	82.3	6.60×	∼ 140M
+ CLA + CLM	**27.0**	**25.8**	27.6	27.3	13.6	**14.5**	16.2	**16.2**	575.0	96.2	5.98×	∼ 150M

Table 3: The effects of our methods on AR and NAR models.

5.2.2 Effects of Each Method

We compare the results of each method on AR and NAR models in Table 3. More detailed ablation studies of CLA and CLM are presented in Appendix B.3. The base AR model is trained with auxiliary loss, where CTC on top of the acoustic encoder learns to predict the source text. Interestingly, there are different effects on different models, languages, and training data. All methods are lightweight in both computational cost and parameter quantity.

Introducing the XCTC loss and PAE method achieves better performance in nearly all settings. CLA does not work well on the similar En-De language pair due to the less reordering requirement, but stable improvements on the distant En-Ja language pair. The remarkable results of CLM demonstrate that an adaptive training strategy is important for better convergence of NAR models (Qian et al., 2021).

However, CLM leads to slightly worse or better results for AR models trained on Seq-KD data. We conclude that the optimization of XCTC loss in the encoder interferes with the learning of cross-entropy loss in the decoder. Although the XCTC achieves good performance, it does not contribute to the final inference in the encoder-decoder framework. In addition, the performance of the AR model trained on raw En-Ja data drops terribly. Raw data distribution is difficult to learn by CTC, especially for distant En-Ja language pair. In this case, the CLM always provides ground truth in a high ratio to mix, leading to overfitting on the training set and worse performance during inference. Therefore, we only use XCTC and PAE on AR models for stable improvements.

We also notice that the simplified data distribution is crucial for achieving optimal performance with the NAST model. Specifically, the base NAR models, when trained on raw data, significantly underperform models trained on Seq-KD data, with a gap of about 7 BLEU points. By combining proposed methods, we develop a powerful NAR model that narrows the gap to within 2 BLEU points. This result highlights the robustness of CTC-NAST, even in the presence of complex data distributions.

	Model	En-De			En-Ja		
		sub	del	ins	sub	del	ins
AR	Base	31.8	12.2	**12.5**	44.6	**19.3**	16.9
AR	+ XCTC-Aug	**31.4**	**12.0**	**12.5**	**43.9**	19.6	**15.9**
NAR	Base	32.0	14.4	**10.7**	42.8	22.8	**12.8**
NAR	+ PAE	31.6	13.2	11.4	43.2	21.1	14.4
NAR	+ CLA	31.4	12.9	11.7	43.6	**20.3**	14.8
NAR	+ CLM	**30.8**	**12.8**	11.3	**42.1**	21.2	13.7
NAR	+ CLA + CLM	30.9	**12.8**	11.4	**42.1**	21.2	14.0

Table 4: Error analysis based on WERs that are split into substitution (sub), deletion (del), and insertion (ins) error rates.

5.2.3 Error Analysis

To identify the weakness of NAR generation, we measure the word error rates (WERs) of AR and NAR models on the MuST-C En-De and En-Ja datasets[2]. For a token in the target text, the sub error indicates that it is incorrectly translated, and the del error indicates that it is omitted. The ins error indicates that the token not in the target text is translated.

High del error rates show that the dominant disadvantage of the NAST model is missing translation. PAE relaxes the conditional independence assumption, giving better results for En-De but increased sub errors for En-Ja. We speculate that this is because poor CTC prediction introduces excessive errors. CLA is particularly effective at reducing del errors, which is consistent with our motivation to relax the monotonic assumption. And CLM reduces error propagation and improves the

[2]Although WER is the metric for ASR, it helps to understand the error types of the translation results.

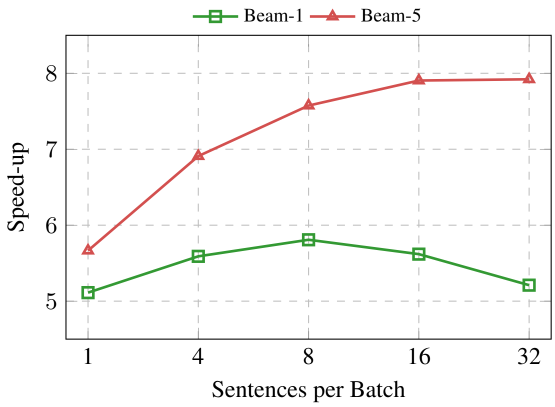

Figure 3: Speed-up under different settings.

robustness of PAE, achieving consistent improvements.

However, the combination of our methods does not lead to a further reduction in del errors. A possible reason is that the inconsistent learning between CLA and CLM limits the effect of the combination. We will explore better methods to alleviate the missing translation in the future.

5.2.4 Speed-up vs. Batch Size

We examine the speed-up compared to AR models under different batch sizes and beam sizes in Figure 3. Our CTC-NAST model consistently maintains a high speed-up, even with a large batch size of 32. The performance of NAR and AR models is comparable when using a beam size of 1, while our NAR model is more than 5× faster. In addition, our encoder-only design simplifies the inference process, eliminating the need for length prediction or iterative refinement. One promising direction is to develop effective encoding methods that can bridge the length gap between acoustic features and text. This has the potential to reduce the computational cost caused by long sequence modeling.

6 Conclusion

Aiming to combine E2E ST and NAR generation, we propose CTC-NAST, which consists of only two CTC-guided encoders for source and target text prediction, respectively. We identify and address several challenges of CTC-NAST: conditional independence assumption, monotonic assumption, and poor convergence. In this way, our CTC-NAST model outperforms the previous best AR models by 0.9 BLEU points. We believe that we are the first to present a NAST model that achieves comparable or better performance than strong AR counterparts.

Limitations

Although our CTC-NAST model achieves excellent performance, there are still some underlying challenges that remain in the follow-up of our work. Here are some limitations that we intend to resolve in the future:

- The better designs of reordering augmentation and training strategy. Although the proposed CLA and CLM approaches achieve good results by alleviating the monotonic assumption and relieving the modeling burden, combing them can not bring remarkable improvement. More importantly, these two methods fail to stable improvements in encode-decoder architecture. This drives us to investigate the interference of the optimizations between CTC and cross-entropy.

- Combination with the pre-training or multi-task learning. Although our methods bring remarkable gains on both AR and NAR models, we do not explore the utilization of external data resources. Although we can use the pre-trained models directly, we expect more effective methods in future work. Theoretically, we need to design NAR ASR and MT models that share the same or similar architectures with the acoustic encoder and textual encoder, respectively. In this way, the NAST model bridges the gap between pre-training and fine-tuning and has more potential for better performance.

- The potential risk for unwritten languages. In our work, we assume that transcription is always available, which is consistent with almost previous studies. Although some datasets have no transcription, we can use a well-trained ASR model to generate pseudo labels. However, it is hard to handle speech translation from unwritten source speech. The supervision of source text is very important for our model. Therefore, we need to develop better methods for stable training.

Acknowledgement

The authors would like to thank anonymous reviewers for their insightful comments. This work was supported in part by the National Science Foundation of China (No. 62276056), the National Key

R&D Program of China, the China HTRD Center Project (No. 2020AAA0107904), the Natural Science Foundation of Liaoning Province of China (2022-KF-16-01), the Yunnan Provincial Major Science and Technology Special Plan Projects (No. 202103AA080015), the Fundamental Research Funds for the Central Universities (Nos. N2216016, N2216001, and N2216002), and the Program of Introducing Talents of Discipline to Universities, Plan 111 (No. B16009).

References

Antonios Anastasopoulos and David Chiang. 2018. Tied multitask learning for neural speech translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 82–91. Association for Computational Linguistics.

Yu Bao, Hao Zhou, Shujian Huang, Dongqi Wang, Lihua Qian, Xinyu Dai, Jiajun Chen, and Lei Li. 2022. latent-glat: Glancing at latent variables for parallel text generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 8398–8409. Association for Computational Linguistics.

Alexandre Berard, Olivier Pietquin, Christophe Servan, and Laurent Besacier. 2016. Listen and translate: A proof of concept for end-to-end speech-to-text translation. *CoRR*, abs/1612.01744.

William Chan, Chitwan Saharia, Geoffrey E. Hinton, Mohammad Norouzi, and Navdeep Jaitly. 2020. Imputer: Sequence modelling via imputation and dynamic programming. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1403–1413. PMLR.

Xuxin Cheng, Qianqian Dong, Fengpeng Yue, Tom Ko, Mingxuan Wang, and Yuexian Zou. 2022. M3ST: mix at three levels for speech translation. *CoRR*, abs/2212.03657.

Shun-Po Chuang, Yung-Sung Chuang, Chih-Chiang Chang, and Hung-yi Lee. 2021. Investigating the reordering capability in ctc-based non-autoregressive end-to-end speech translation. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 1068–1077. Association for Computational Linguistics.

Liang Ding, Longyue Wang, Xuebo Liu, Derek F. Wong, Dacheng Tao, and Zhaopeng Tu. 2021. Rejuvenating low-frequency words: Making the most of parallel data in non-autoregressive translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 3431–3441. Association for Computational Linguistics.

Long Duong, Antonios Anastasopoulos, David Chiang, Steven Bird, and Trevor Cohn. 2016. An attentional model for speech translation without transcription. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 949–959. The Association for Computational Linguistics.

Qingkai Fang, Rong Ye, Lei Li, Yang Feng, and Mingxuan Wang. 2022. STEMM: self-learning with speech-text manifold mixup for speech translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 7050–7062. Association for Computational Linguistics.

Mattia Antonino Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019. Must-c: a multilingual speech translation corpus. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2012–2017. Association for Computational Linguistics.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 6111–6120. Association for Computational Linguistics.

Alex Graves, Santiago Fernández, Faustino J. Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, volume 148 of *ACM International Conference Proceeding Series*, pages 369–376. ACM.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O. K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Jiatao Gu and Xiang Kong. 2021. Fully non-autoregressive neural machine translation: Tricks of the trade. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 120–133. Association for Computational Linguistics.

Junliang Guo, Xu Tan, Di He, Tao Qin, Linli Xu, and Tie-Yan Liu. 2019. Non-autoregressive neural machine translation with enhanced decoder input. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 3723–3730. AAAI Press.

Awni Hannun. 2017. Sequence modeling with ctc. *Distill*. Https://distill.pub/2017/ctc.

Yosuke Higuchi, Nanxin Chen, Yuya Fujita, Hirofumi Inaguma, Tatsuya Komatsu, Jaesong Lee, Jumon Nozaki, Tianzi Wang, and Shinji Watanabe. 2021a. A comparative study on non-autoregressive modelings for speech-to-text generation. In *IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2021, Cartagena, Colombia, December 13-17, 2021*, pages 47–54. IEEE.

Yosuke Higuchi, Hirofumi Inaguma, Shinji Watanabe, Tetsuji Ogawa, and Tetsunori Kobayashi. 2021b. Improved mask-ctc for non-autoregressive end-to-end asr. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8363–8367.

Yosuke Higuchi, Shinji Watanabe, Nanxin Chen, Tetsuji Ogawa, and Tetsunori Kobayashi. 2020. Mask CTC: Non-Autoregressive End-to-End ASR with CTC and Mask Predict. In *Proc. Interspeech 2020*, pages 3655–3659.

Chenyang Huang, Hao Zhou, Osmar R. Zaïane, Lili Mou, and Lei Li. 2022. Non-autoregressive translation with layer-wise prediction and deep supervision. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 10776–10784. AAAI Press.

Hirofumi Inaguma, Yosuke Higuchi, Kevin Duh, Tatsuya Kawahara, and Shinji Watanabe. 2021a. Non-autoregressive end-to-end speech translation with parallel autoregressive rescoring. *CoRR*, abs/2109.04411.

Hirofumi Inaguma, Yosuke Higuchi, Kevin Duh, Tatsuya Kawahara, and Shinji Watanabe. 2021b. ORTHROS: non-autoregressive end-to-end speech translation with dual-decoder. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2021, Toronto, ON, Canada, June 6-11, 2021*, pages 7503–7507. IEEE.

Hirofumi Inaguma, Shun Kiyono, Kevin Duh, Shigeki Karita, Nelson Yalta, Tomoki Hayashi, and Shinji Watanabe. 2020. Espnet-st: All-in-one speech translation toolkit. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL 2020, Online, July 5-10, 2020*, pages 302–311. Association for Computational Linguistics.

Shigeki Karita, Nelson Enrique Yalta Soplin, Shinji Watanabe, Marc Delcroix, Atsunori Ogawa, and Tomohiro Nakatani. 2019. Improving transformer-based end-to-end speech recognition with connectionist temporal classification and language model integration. In *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, pages 1408–1412. ISCA.

Jungo Kasai, James Cross, Marjan Ghazvininejad, and Jiatao Gu. 2020. Non-autoregressive machine translation with disentangled context transformer. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 5144–5155. PMLR.

Sehoon Kim, Amir Gholami, Albert E. Shaw, Nicholas Lee, Karttikeya Mangalam, Jitendra Malik, Michael W. Mahoney, and Kurt Keutzer. 2022. Squeezeformer: An efficient transformer for automatic speech recognition. *CoRR*, abs/2206.00888.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1317–1327. The Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*. The Association for Computational Linguistics.

Jaesong Lee and Shinji Watanabe. 2021. Intermediate loss regularization for ctc-based speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2021, Toronto, ON, Canada, June 6-11, 2021*, pages 6224–6228. IEEE.

Jason Lee, Elman Mansimov, and Kyunghyun Cho. 2018. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1173–1182. Association for Computational Linguistics.

Jindrich Libovický and Jindrich Helcl. 2018. End-to-end non-autoregressive neural machine translation with connectionist temporal classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 3016–3021. Association for Computational Linguistics.

Yuchen Liu, Junnan Zhu, Jiajun Zhang, and Chengqing Zong. 2020. Bridging the modality gap for speech-to-text translation. *CoRR*, abs/2010.14920.

Lambert Mathias and William Byrne. 2006. Statistical phrase-based speech translation. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing, ICASSP 2006, Toulouse, France, May 14-19, 2006*, pages 561–564. IEEE.

Hermann Ney. 1999. Speech translation: coupling of recognition and translation. In *Proceedings of the 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '99, Phoenix, Arizona, USA, March 15-19, 1999*, pages 517–520. IEEE Computer Society.

Jumon Nozaki and Tatsuya Komatsu. 2021. Relaxing the conditional independence assumption of ctc-based ASR by conditioning on intermediate predictions. In *Interspeech 2021, 22nd Annual Conference of the International Speech Communication Association, Brno, Czechia, 30 August - 3 September 2021*, pages 3735–3739. ISCA.

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 the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Demonstrations*, pages 48–53. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers, WMT 2018, Belgium, Brussels, October 31 - November 1, 2018*, pages 186–191. Association for Computational Linguistics.

Matt Post, Gaurav Kumar, Adam Lopez, Damianos G. Karakos, Chris Callison-Burch, and Sanjeev Khudanpur. 2013. Improved speech-to-text translation with the fisher and callhome spanish-english speech translation corpus. In *Proceedings of the 10th International Workshop on Spoken Language Translation: Papers, Heidelberg, Germany, December 5-6, 2013*.

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 *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 1993–2003. Association for Computational Linguistics.

Qiu Ran, Yankai Lin, Peng Li, and Jie Zhou. 2020. Learning to recover from multi-modality errors for non-autoregressive neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 3059–3069. Association for Computational Linguistics.

Qiu Ran, Yankai Lin, Peng Li, and Jie Zhou. 2021. Guiding non-autoregressive neural machine translation decoding with reordering information. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 13727–13735. AAAI Press.

Chitwan Saharia, William Chan, Saurabh Saxena, and Mohammad Norouzi. 2020. Non-autoregressive machine translation with latent alignments. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 1098–1108. Association for Computational Linguistics.

Raphael Shu, Jason Lee, Hideki Nakayama, and Kyunghyun Cho. 2020. Latent-variable non-autoregressive neural machine translation with deterministic inference using a delta posterior. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8846–8853. AAAI Press.

Jongyoon Song, Sungwon Kim, and Sungroh Yoon. 2021. Alignart: Non-autoregressive neural machine translation by jointly learning to estimate alignment and translate. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 1–14. Association for Computational Linguistics.

Mitchell Stern, William Chan, Jamie Kiros, and Jakob Uszkoreit. 2019. Insertion transformer: Flexible sequence generation via insertion operations. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5976–5985. PMLR.

Changhan Wang, Yun Tang, Xutai Ma, Anne Wu, Dmytro Okhonko, and Juan Miguel Pino. 2020a. Fairseq S2T: fast speech-to-text modeling with fairseq. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: System Demonstrations, AACL/IJCNLP 2020, Suzhou, China, December*

4-7, 2020, pages 33–39. Association for Computational Linguistics.

Chengyi Wang, Yu Wu, Shujie Liu, Zhenglu Yang, and Ming Zhou. 2020b. Bridging the gap between pre-training and fine-tuning for end-to-end speech translation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9161–9168. AAAI Press.

Chengyi Wang, Yu Wu, Shujie Liu, Ming Zhou, and Zhenglu Yang. 2020c. Curriculum pre-training for end-to-end speech translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 3728–3738. Association for Computational Linguistics.

Yiren Wang, Fei Tian, Di He, Tao Qin, ChengXiang Zhai, and Tie-Yan Liu. 2019. Non-autoregressive machine translation with auxiliary regularization. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 5377–5384. AAAI Press.

Shinji Watanabe, Takaaki Hori, Suyoun Kim, John R. Hershey, and Tomoki Hayashi. 2017. Hybrid ctc/attention architecture for end-to-end speech recognition. *IEEE J. Sel. Top. Signal Process.*, 11(8):1240–1253.

Ron J. Weiss, Jan Chorowski, Navdeep Jaitly, Yonghui Wu, and Zhifeng Chen. 2017. Sequence-to-sequence models can directly translate foreign speech. In *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*, pages 2625–2629. ISCA.

Chen Xu, Bojie Hu, Yanyang Li, Yuhao Zhang, Shen Huang, Qi Ju, Tong Xiao, and Jingbo Zhu. 2021. Stacked acoustic-and-textual encoding: Integrating the pre-trained models into speech translation encoders. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 2619–2630. Association for Computational Linguistics.

Chen Xu, Yuhao Zhang, Chengbo Jiao, Xiaoqian Liu, Chi Hu, Xin Zeng, Tong Xiao, Anxiang Ma, Huizhen Wang, and Jingbo Zhu. 2023. Bridging the granularity gap for acoustic modeling. In *Findings of the Association for Computational Linguistics: ACL 2023*. Association for Computational Linguistics.

Brian Yan, Siddharth Dalmia, Yosuke Higuchi, Graham Neubig, Florian Metze, Alan W. Black, and Shinji Watanabe. 2022. CTC alignments improve autoregressive translation. *CoRR*, abs/2210.05200.

Baosong Yang, Zhaopeng Tu, Derek F. Wong, Fandong Meng, Lidia S. Chao, and Tong Zhang. 2018. Modeling localness for self-attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4449–4458. Association for Computational Linguistics.

Rong Ye, Mingxuan Wang, and Lei Li. 2021. End-to-end speech translation via cross-modal progressive training. In *Interspeech 2021, 22nd Annual Conference of the International Speech Communication Association, Brno, Czechia, 30 August - 3 September 2021*, pages 2267–2271. ISCA.

Rong Ye, Mingxuan Wang, and Lei Li. 2022. Cross-modal contrastive learning for speech translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 5099–5113. Association for Computational Linguistics.

Biao Zhang, Barry Haddow, and Rico Sennrich. 2022a. Revisiting end-to-end speech-to-text translation from scratch. *CoRR*, abs/2206.04571.

Yuhao Zhang, Chen Xu, Bojie Hu, Chunliang Zhang, Tong Xiao, and Jingbo Zhu. 2022b. Improving end-to-end speech translation by leveraging auxiliary speech and text data. *CoRR*, abs/2212.01778.

Chengqi Zhao, Mingxuan Wang, Qianqian Dong, Rong Ye, and Lei Li. 2021. Neurst: Neural speech translation toolkit. In *Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL 2021 - System Demonstrations, Online, August 1-6, 2021*, pages 55–62. Association for Computational Linguistics.

Jiawei Zhou and Phillip Keung. 2020. Improving non-autoregressive neural machine translation with monolingual data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1893–1898. Association for Computational Linguistics.

A Experimental Settings

A.1 Datasets and Preprocessing

We conduct experiments on the MuST-C (Gangi et al., 2019) and Fisher-Callhome ST (Post et al., 2013) datasets. MuST-C is a multilingual speech translation corpus extracted from TED lectures. We test our method on all MuST-C v1 corpora: English (En) to German (De), Spanish (Es), French (Fr), Italian (It), Dutch (Nl), Portuguese (Pt), Romanian (Ro) and Russian (Ru). In addition, we also investigate the results of the distant language pair English-Japanese (En-Ja) corpus in the MuST-C v2 dataset. We select (and tune) the model on the dev set (Dev) and report the results on the tst-COMMON set (Test).

Fisher-Callhome is a Spanish-English speech-to-text translation dataset with 138k text pairs. This corpus contains 170 hours of Spanish conversational telephone speech, as well as Spanish transcripts and English translations. Following the recipe of ESPnet (Inaguma et al., 2020), we lowercase all texts, and remove all punctuation marks except apostrophes. We select (and tune) the model on the Fisher-dev set, and report the results on the Fisher-{dev, dev2, test} and Callhome-{devtest, evltest} sets.

Following the preprocessing recipes in the fairseq toolkit[3], we remove utterances with more than 3,000 frames or less than 5 frames. We extract the 80-channel Mel filter bank features by a window size of 25ms with a stride of 10ms. The text is tokenized using the scripts of Moses (Koehn et al., 2007) except that the Japanese text uses MeCab[4]. We learn SentencePiece[5] segmentation with a size of 10,000 for MuST-C datasets. We use a shared vocabulary for the source and target languages for MuST-C v1 corpora, the independent vocabulary for the En-Ja corpus. And we use a shared vocabulary with a size of 1, 000 for Fisher-Callhome datasets.

A.2 Model Settings

We implement our method based on the fairseq toolkit (Ott et al., 2019). We use the Adam optimizer with $\beta_1 = 0.9, \beta_2 = 0.98$, and adopt the default learning schedule in fairseq. We apply dropout with a rate of 0.15 and label smoothing of 0.1 for regularization.

Following previous studies on NAR models, our model is trained by sequence-level knowledge distillation (Seq-KD) (Kim and Rush, 2016) data generated by a small MT model with a beam size of 5. Our NAST model consists of an acoustic encoder with 12 Conformer layers and a textual encoder with 12 Transformer layers. Each layer comprises 512 hidden units, 8 attention heads, and 2048 feed-forward sizes. We use PAE in layers 6 and 9 in both the acoustic encoder and the textual encoder. In multitask learning, the weights of α_A, α_T, β_A and β_T are all set to 1. We start the cross-layer attention from layer 4 in the textual encoder and take the representation output from layer 3 as the key and value. The ratio for curriculum learning mixing is set to 0.8.

We extend our method to the encoder-decoder model with similar settings, where the textual encoder has 6 Transformer layers and the decoder has 6 layers. In this way, we control the model parameters to about 150M for fair comparisons. The weights of α_A and α_T are set to 0.2, and the weights of β_A and β_T are to 0.1. We use PAE in layer 4 in the textual encoder. We start the cross-layer attention from layer 3 and take the representation output from layer 2 as the key and value.

[3]https://github.com/pytorch/fairseq
[4]https://github.com/taku910/mecab
[5]https://github.com/google/sentencepiece

En-xx	Raw	Seq-KD
De	7.23	5.10
Es	4.42	2.72
Fr	5.51	2.80
It	5.79	2.94
Nl	6.18	4.16
Pt	5.56	3.26
Ro	5.22	2.95
Ru	6.99	2.94
Ja	14.01	15.21

Table 5: Reordering difficulty of MuST-C datasets.

During inference, we average the model parameters on the best 10 checkpoints based on the performance of the development set. We use beam search with a beam size of 5 for the AR model. The decoding speed is measured on the test set with a batch size of 1 on an Nvidia A100 80GB GPU. We run 5 times to calculate the average time. We report case-sensitive SacreBLEU (Post, 2018) on the MuST-C datasets and case-insensitive SacreBLEU on the Fisher-Callhome dataset for standardization comparison across papers.

B More Analysis

B.1 Reordering Difficulty

Following the metric in Chuang et al. (2021), we measure the reordering difficulties R_π on 9 language pairs of MuST-C datasets in Table 5. The higher the value of R_π, the higher the reordering difficulty between texts from two languages, indicating the high demand for improved reordering capability. The Seq-KD technique reduces the reordering difficulty by simplifying the data distribution, except for En-Ja. The reason is that noisy data leads to poor MT performance on the En-Ja dataset. On this distant language pair, our CTC-NAST model still achieves a high BLEU score of 16.2, which is comparable to the AR model with a small gap of only 0.2 BLEU points.

B.2 Results on Out-of-domain Data

We also measure the BLEU scores of AR and NAR models under different output lengths on the Callhome sets in Figure 4. Note that Callhome sets are out-of-domain because we only use the Fisher set for training. Here, BLEU scores of the NAR model are better than those of the AR model in most cases of output length. In particular, when the output length is greater than 50, the performance of the AR model drops sharply, while the performance of

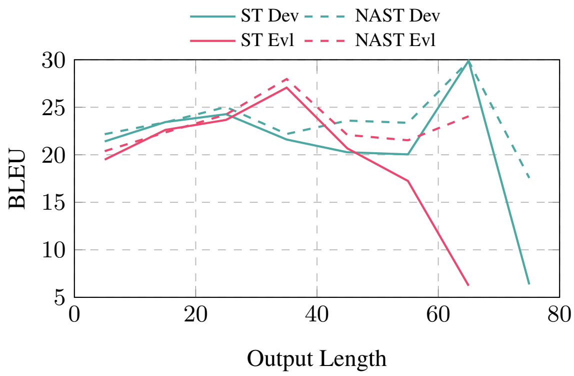

Figure 4: BLEU scores over various output lengths of Callhome sets.

Model	En-De		En-Ja	
	dev	test	dev	test
Base	23.7	24.3	10.5	13.7
+ PAE	24.8	25.7	12.4	14.9
+ CLA	25.1	25.8	12.1	15.3
+ drop 0.1	**25.4**	**26.2**	**12.7**	15.3
+ drop 0.2	25.2	25.5	12.3	**15.6**

Table 6: Ablation study of the CLA module.

Model	En-De		En-Ja	
	0.5	0.8	0.5	0.8
Base	24.3	24.3	13.7	13.7
+ PAE	25.7	25.7	14.9	14.9
+ Mixing	26.7	26.6	15.6	15.7
+ Adaptive	26.2	26.3	15.2	15.4
+ Only error	26.7	27.1	15.8	15.9
+ Smooth	26.7	26.6	15.8	15.6
+ Only error + Smooth	**26.8**	**27.4**	**16.0**	**16.1**

Table 7: Ablation study of the CLM method under different mixing ratios and strategies.

the NAR model keeps stable. This demonstrates that our CTC-NAST has better robustness.

B.3 Ablation Studies

To further verify the effectiveness of our proposed methods, we construct a series of ablation studies on MuST-C En-De and En-Ja datasets.

Effects of CLA Table 6 shows the results of the CLA module. CLA improves the reordering capability and complements the self-attention module. However, using the CLA module naively brings only modest improvements. We randomly drop the self-attention module with a probability of 0.1, which provides better regularization and robust improvements. Note that the high drop probability may lead to insufficient training of the self-attention module. These results demonstrate the

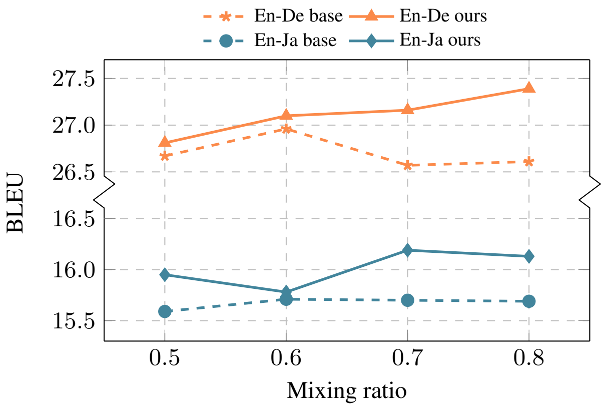

Figure 5: BLEU scores of base mixing and our CLM method.

effectiveness of the CLA module and drop-net technique.

Effects of CLM As shown in Table 7, the straightforward mixed training has produces remarkable gains with a ratio of 0.5 or 0.8 on both En-De and En-Ja datasets. The adaptive strategy in NAR MT does not work in CTC-NAST. This is because the sequence length of the input acoustic feature is very lengthy, and the decreased mixing ratio cannot provide enough cues to facilitate training. For stable training, we only replace positions where wrong predictions arise. In this manner, accurate positions solely rely on self-prediction, guaranteeing consistency between training and decoding. Furthermore, we generate a smooth distribution akin to CTC prediction, in which the ground truth token has a high probability of 0.9, and the probabilities of other tokens sum to 0.1. The combination of these two approaches results in additional and stable improvements.

We also calculate BLEU scores with various mixing ratios in Figure 5. Our CLM approach is superior to the naive mixing method, particularly at a high ratio. In this case, our approach incorporates more revisions solely for incorrect predictions, which facilitates the training process and guarantees consistency.