STEMM: Self-learning with Speech-text Manifold Mixup for Speech Translation

Qingkai Fang[1,2†], **Rong Ye**[3], **Lei Li**[4*†], **Yang Feng**[1,2*], **Mingxuan Wang**[3*]
[1] Key Laboratory of Intelligent Information Processing
Institute of Computing Technology, Chinese Academy of Sciences (ICT/CAS)
[2] University of Chinese Academy of Sciences, Beijing, China
[3] ByteDance AI Lab [4] University of California, Santa Barbara
{fangqingkai21b,fengyang}@ict.ac.cn
{yerong,wangmingxuan.89}@bytedance.com, leili@cs.ucsb.edu

Abstract

How to learn a better speech representation for end-to-end speech-to-text translation (ST) with limited labeled data? Existing techniques often attempt to transfer powerful machine translation (MT) capabilities to ST, but neglect the representation discrepancy across modalities. In this paper, we propose the **S**peech-**TE**xt **M**anifold **M**ixup (STEMM) method to calibrate such discrepancy. Specifically, we mix up the representation sequences of different modalities, and take both unimodal speech sequences and multimodal mixed sequences as input to the translation model in parallel, and regularize their output predictions with a self-learning framework. Experiments on MuST-C speech translation benchmark and further analysis show that our method effectively alleviates the cross-modal representation discrepancy, and achieves significant improvements over a strong baseline on eight translation directions.

1 Introduction

Speech-to-text translation (ST) aims at translating acoustic speech signals into text in a foreign language, which has wide applications including voice assistants, translation for multinational video conferences, and so on. Traditional ST methods usually combine automatic speech recognition (ASR) and machine translation (MT) in a cascaded manner (Sperber et al., 2017; Cheng et al., 2018; Sperber et al., 2019; Dong et al., 2019b; Zhang et al., 2019a; Lam et al., 2021b), which might suffer from error propagation and high latency. To break this bottleneck, end-to-end ST systems attracted much attention recently (Wang et al., 2020b,c; Dong et al., 2021a,b; Han et al., 2021; Inaguma et al., 2021a; Tang et al., 2021a), which learn a unified model to generate translations from speech directly. Some recent work has shown great potential for end-to-end speech translation, even surpassing traditional cascaded systems (Ye et al., 2021; Xu et al., 2021).

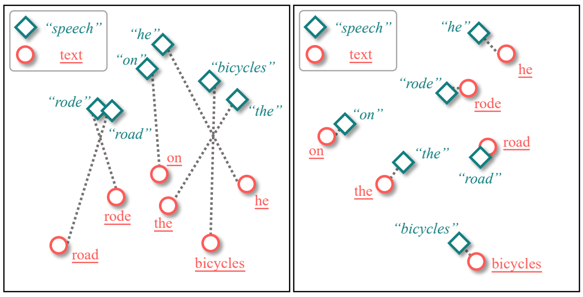

Figure 1: STEMM aims at bridging the modality gap of speech and text. Different modalities with the same meaning are projected to a shared space.

As a cross-modal task, a major challenge in training an end-to-end ST model is the representation discrepancy across modalities, which means there is a modality gap between speech representations and text embeddings, as shown in the left sub-figure of Figure 1. Existing approaches often adopt a sophisticated MT model to help the training of ST, with some techniques like pretraining (Wang et al., 2020c; Ye et al., 2021; Xu et al., 2021), multi-task learning (Ye et al., 2021; Han et al., 2021; Tang et al., 2021a) and knowledge distillation (Liu et al., 2019; Gaido et al., 2020; Inaguma et al., 2021b; Tang et al., 2021a). Although these methods have achieved impressive improvements in ST task, these methods are not necessarily the best way to leverage the MT knowledge. Considering that during training, the input of the translation module only include speech sequences or text sequences, the lack of multimodal contexts makes it difficult for the ST model to learn from the MT model. Inspired by recent studies on some cross-

* indicates corresponding authors.

† Work was done while at ByteDance AI Lab.

Part of joint project between ICT/CAS and ByteDance AI Lab. Work was done when QF was a member of the joint project.

Code and models are publicly available at https://github.com/ictnlp/STEMM.

lingual (Lample and Conneau, 2019; Liu et al., 2020a; Lin et al., 2020) and cross-modal (Li et al., 2021b; Zhou et al., 2020; Dong et al., 2019a) tasks, we suggest that building a shared semantic space between speech and text, as illustrated in the right sub-figure of Figure 1, has the potential to benefit the most from the MT model.

In this paper, we propose the **S**peech-**TE**xt **M**anifold **M**ixup (STEMM) method to bridge the modality gap between text and speech. In order to calibrate the cross-modal representation discrepancy, we mix up the speech and text representation as the input and keep the target sequence unchanged. Specifically, STEMM is a self-learning framework, which takes both the speech representation and the mixed representation as parallel inputs to the translation model, and regularizes their output predictions. Experimental results show that our method achieves promising performance on the benchmark dataset MuST-C (Di Gangi et al., 2019a), and even outperforms a strong cascaded baseline. Furthermore, we found that our STEMM could effectively alleviate the cross-modal representation discrepancy, and project two modalities into a shared space.

2 Method

In this section, we will begin with the basic problem formulation (Section 2.1) and introduce the model architecture (Section 2.2). Then, we introduce our proposed **S**peech-**TE**xt **M**anifold **M**ixup (STEMM) in Section 2.3. Finally, we introduce our proposed self-learning framework with STEMM in Section 2.4 and present two mixup ratio strategies in Section 2.5. Figure 2 illustrates the overview of our proposed method.

2.1 Problem Formulation

The speech translation corpus usually contains *speech-transcription-translation* triples, which can be denoted as $\mathcal{D} = \{(\mathbf{s}, \mathbf{x}, \mathbf{y})\}$. Here $\mathbf{s}$ is the sequence of audio wave, $\mathbf{x}$ is the transcription in the source language, and $\mathbf{y}$ is the translation in the target language. End-to-end speech translation aims to generate translation $\mathbf{y}$ directly from the audio wave $\mathbf{s}$, without generating intermediate transcription $\mathbf{x}$.

2.2 Model Architecture

Inspired by recent works (Dong et al., 2021b; Xu et al., 2021) in end-to-end speech translation, we decompose the ST model into three modules: *acoustic encoder*, *translation encoder*, and *translation decoder*. The *acoustic encoder* first encodes the original audio wave into hidden states, fed into the *translation encoder* to learn further semantic information. Finally, the *translation decoder* generates the translation based on the output of the *translation encoder*.

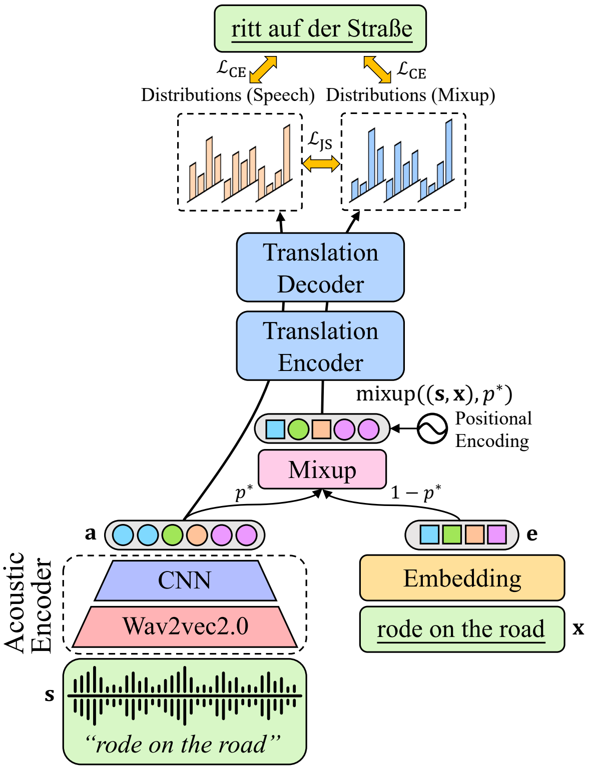

Figure 2: Overview of our proposed self-learning framework with STEMM. We first mix up the sequence of speech representations and word embeddings with STEMM. Then, both the unimodal speech sequence and the multimodal mixed sequence are fed into the shared translation module to predict the translation, and we regularize two output predictions with an additional JS Divergence loss.

Acoustic Encoder As recent works (Ye et al., 2021; Han et al., 2021) show that Wav2vec2.0 (Baevski et al., 2020) can improve the performance of speech translation, we first use a pretrained Wav2vec2.0 to extract speech representations $\mathbf{c}$ from the audio wave $\mathbf{s}$. We add two additional convolutional layers to further shrink the length of speech representations by a factor of 4, denoted as $\mathbf{a} = \text{CNN}(\mathbf{c})$.

Translation Encoder Our *translation encoder* is composed of N_e transformer (Vaswani et al., 2017) encoder layers, which includes a self-attention layer, a feed-forward layer, normalization layers, and residual connections. For MT task, the input of the *translation encoder* is the embedding of transcription $\mathbf{e} = \text{Emb}(\mathbf{x})$. For ST task, it is the output sequence of the *acoustic encoder* $\mathbf{a}$. The input can also be the multimodal mixed sequence with our proposed STEMM (see details in Section 2.3). Generally, for the input sequence χ, we obtain the contextual representations $\mathbf{h}(\chi)$ after N_e transformer (Vaswani et al., 2017) layers, which are fed into the *translation decoder* for predicting the translation.

Translation Decoder Our *translation decoder* is composed of N_t transformer decoder layers, which contain an additional cross-attention layer compared with transformer encoder layers. For the input sequence χ, the cross entropy loss is defined as:

$$\mathcal{L}_{\text{CE}}(\chi, \mathbf{y}) = -\sum_{i=1}^{|\mathbf{y}|} \log p_\theta(\mathbf{y}_i | \mathbf{y}_{<i}, \mathbf{h}(\chi)). \quad (1)$$

Pretrain-finetune We follow the pretrain-finetune paradigm to train our model. First, we pretrain the *translation encoder* and *translation decoder* with parallel *transcription-translation* pairs, derived from both the speech translation corpus and the external MT dataset. Also, the *acoustic encoder* is pretrained on large amounts of unlabeled audio data in a self-supervised manner. We combine those pretrained modules and finetune the whole model for ST.

2.3 Speech-Text Manifold Mixup (STEMM)

As we mentioned in Section 1, to alleviate the representation discrepancy due to the lack of multimodal contexts, we present the **S**peech-**TE**xt **M**anifold **M**ixup (STEMM) method to mix up the sequence of speech representations and word embeddings. We first introduce STEMM in this section and later show how to use it to help the training of ST.

Note the sequence of sub-word embeddings as $\mathbf{e} = [\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_{|\mathbf{e}|}]$ and the sequence of speech representations as $\mathbf{a} = [\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_{|\mathbf{a}|}]$, where the sequence lengths usually follow $|\mathbf{a}| \geq |\mathbf{e}|$. We first perform a word-level forced alignment between speech and text transcriptions to determine when particular words appear in the speech segment. Formally, the aligner recognizes a sequence of word units $\mathbf{w} = [w_1, w_2, ..., w_T]$, and for each word w_i, it returns the start position l_i and end position r_i in the sequence of speech representation $\mathbf{a}$. Meanwhile, we denote the corresponding sub-word span for word w_i as $[x_{m_i} : x_{n_i}]$, with its embeddings matrix $[\mathbf{e}_{m_i} : \mathbf{e}_{n_i}]$, where m_i and n_i are the start position and end position in the sequence of sub-words. To mix up both sequences, for each word unit w_i, we choose either the segment of speech representations $[\mathbf{a}_{l_i} : \mathbf{a}_{r_i}]$ or sub-word embeddings $[\mathbf{e}_{m_i} : \mathbf{e}_{n_i}]$ with a certain probability p^*, referred to *mixup ratio* in this paper.

$$\mathbf{m}_i = \begin{cases} [\mathbf{a}_{l_i} : \mathbf{a}_{r_i}] & p \leq p^* \\ [\mathbf{e}_{m_i} : \mathbf{e}_{n_i}] & p > p^* \end{cases}, \quad (2)$$

where p is sampled from the uniform distribution $\mathcal{U}(0, 1)$.

Finally, we concatenate all $\mathbf{m}_i$ together and obtain the mixup sequence:

$$\mathbf{m} = \text{Concat}(\mathbf{m}_1, \mathbf{m}_2, ..., \mathbf{m}_T). \quad (3)$$

Note that in terms of the mixup representation sequence length, we have $|\mathbf{e}| \leq |\mathbf{m}| \leq |\mathbf{a}|$. Considering the positions of tokens have changed after mixup, we add positional encodings to the token embeddings. We further perform layer normalization to normalize the embeddings:

$$\text{Mixup}((\mathbf{s}, \mathbf{x}), p^*) = \text{LayerNorm}(\mathbf{m} + \text{Pos}(\mathbf{m})), \quad (4)$$

where $\text{Pos}(\cdot)$ is the sinusoid positional embedding (Vaswani et al., 2017). $\text{Mixup}((\mathbf{s}, \mathbf{x}), p^*)$ indicates the *mixup sequence* of speech $\mathbf{s}$ and text $\mathbf{x}$ with probability p^*, which is fed into the *translation encoder* for predicting the translation.

2.4 Self-learning with STEMM

With the help of our proposed STEMM, we are now able to access multimodal mixed sequences, in addition to the unimodal speech sequences. We integrate them into a self-learning framework. Specifically, we input both unimodal speech sequences and multimodal mixed sequences into the translation module (*translation encoder* and *translation decoder*). In this way, translation of unimodal speech sequences focuses on the ST task itself, while the translation of multimodal mixed sequences is devoted to capture the connections between representations in different modalities. Besides, we try to regularize above two output predictions by minimizing the Jensen-Shannon Divergence (JSD) between two output distributions, which is

$$\mathcal{L}_{\text{JSD}}(\mathbf{s},\mathbf{x},\mathbf{y},p^*) = \sum_{i=1}^{|\mathbf{y}|} \text{JSD}\{p_\theta(\mathbf{y}_i|\mathbf{y}_{<i},\mathbf{h}(\mathbf{s}))\| \\ p_\theta(\mathbf{y}_i|\mathbf{y}_{<i},\mathbf{h}(\text{Mixup}((\mathbf{s},\mathbf{x}),p^*)))\}, \tag{5}$$

where $\mathbf{h}(\cdot)$ is the contextual representation outputted by the *translation encoder*. $p_\theta(\mathbf{y}_i|\mathbf{y}_{<i},\mathbf{h}(\mathbf{s}))$ is the predicted probability distribution of the i-th target token given the speech sequence $\mathbf{s}$ as input, and $p_\theta(\mathbf{y}_i|\mathbf{y}_{<i},\mathbf{h}(\text{Mixup}((\mathbf{s},\mathbf{x}),p^*)))$ is that given the multimodal mixed sequence as input.

With the cross-entropy losses of two forward passes, the final training objective is as follows:

$$\mathcal{L} = \mathcal{L}_{\text{CE}}(\mathbf{s},\mathbf{y}) + \mathcal{L}_{\text{CE}}(\text{Mixup}((\mathbf{s},\mathbf{x}),p^*),\mathbf{y}) \\ + \lambda\mathcal{L}_{\text{JSD}}(\mathbf{s},\mathbf{x},\mathbf{y},p^*), \tag{6}$$

where λ is the coefficient weight to control $\mathcal{L}_{\text{JSD}}$.

2.5 Mixup Ratio Strategy

When using our proposed STEMM, an important question is how to determine the mixup ratio p^*. Here we try two strategies: *static mixup ratio* and *uncertainty-aware mixup ratio*.

Static Mixup Ratio We use the same mixup ratio p^* for all instances throughout the whole training process. We will show how we determined this important hyper-parameter in Section 4.3.

Uncertainty-aware Mixup Ratio With this strategy, we determine the mixup ratio for each instance according to the prediction uncertainty of the ST task, defined as the average entropy of predicted distributions of all target tokens:

$$u = \frac{1}{|\mathbf{y}|}\sum_{i=1}^{|\mathbf{y}|}\text{Entropy}(p_\theta(\mathbf{y}_i|\mathbf{y}_{<i},\mathbf{h}(\mathbf{s}))), \tag{7}$$

and then we set the mixup ratio p^* as follows:

$$p^* = \sigma\left(\frac{u}{U} - \frac{1}{2}\right), \tag{8}$$

where U is a normalization factor which re-scales u to $[0, 1]$, $\sigma(\cdot)$ is a sigmoid function to prevent p^* from dropping too quickly.

3 Experiments

3.1 Datasets

MuST-C We conduct experiments on MuST-C (Di Gangi et al., 2019a) dataset. MuST-C is a multilingual speech translation dataset, which contains translations from English (En) to 8 languages: German (De), French (Fr), Russian (Ru), Spanish (Es), Italian (It), Romanian (Ro), Portuguese (Pt), and Dutch (Nl). It is one of the largest speech translation datasets currently, which contains at least 385 hours of audio recordings from TED Talks, with their manual transcriptions and translations at the sentence level. We use `dev` set for validation and `tst-COMMON` set for test.

	ST (MuST-C)		**MT**	
En→	hours	#sents	name	#sents
De	408	234K	WMT16	4.6M
Fr	492	280K	WMT14	40.8M
Ru	489	270K	WMT16	2.5M
Es	504	270K	WMT13	15.2M
Ro	432	240K	WMT16	0.6M
It	465	258K	OPUS100	1.0M
Pt	385	211K	OPUS100	1.0M
Nl	442	253K	OPUS100	1.0M

Table 1: Statistics of all datasets

MT Datasets Our model architecture allows us to utilize external parallel sentence pairs in large-scale machine translation datasets. Therefore, we incorporate data from WMT for En-De, En-Fr, En-Ru, En-Es, En-Ro, and OPUS100[1] for En-Pt, En-It, En-Nl, as pretraining corpora. The detailed statistics of all datasets included are shown in Table 1.

3.2 Experimental setups

Pre-processing For speech input, we use the raw 16-bit 16kHz mono-channel audio wave. To perform word-level force alignment, we use Montreal Forced Aligner[2] toolkit, whose acoustic model is trained with LibriSpeech (Panayotov et al., 2015). For text input, we remove the punctuation from the source texts for the ST dataset. Both source and target texts are case-sensitive. For each translation direction, we use a unigram SentencePiece[3] model to learn a vocabulary on the text data from ST dataset, and use it to segment text from both ST and MT corpora into subword units. The vocabulary is shared for source and target with a size of 10k.

[1] http://opus.nlpl.eu/opus-100.php

[2] https://github.com/MontrealCorpusTools/Montreal-Forced-Aligner

[3] https://github.com/google/sentencepiece

Models	External Data			BLEU								
	Speech	ASR	MT	En-De	En-Fr	En-Ru	En-Es	En-It	En-Ro	En-Pt	En-Nl	Avg.
Pretrain w/o external MT data												
Fairseq ST (Wang et al., 2020a)	×	×	×	22.7	32.9	15.3	27.2	22.7	21.9	28.1	27.3	24.8
AFS (Zhang et al., 2020)	×	×	×	22.4	31.6	14.7	26.9	23.0	21.0	26.3	24.9	23.9
DDT (Le et al., 2020)	×	×	×	23.6	33.5	15.2	28.1	24.2	22.9	30.0	27.6	25.6
Self-training (Pino et al., 2020)	✓	✓	×	25.2	34.5	-	-	-	-	-	-	-
BiKD (Inaguma et al., 2021a)	×	×	×	25.3	35.3	-	-	-	-	-	-	-
SATE (Xu et al., 2021)	×	×	×	25.2	-	-	-	-	-	-	-	-
XSTNet (Ye et al., 2021)	✓	×	×	25.5	36.0	16.9	29.6	25.5	**25.1**	**31.3**	30.0	**27.5**
W2V2-Transformer	✓	×	×	24.1	35.0	16.3	29.4	24.8	23.1	30.0	28.9	26.5
STEMM	✓	×	×	**25.6****	**36.1****	**17.1****	**30.3****	**25.6****	24.3**	31.0**	**30.1****	**27.5**
Pretrain w/ external MT data												
MTL (Tang et al., 2021b)	×	×	✓	23.9	33.1	-	28.6	-	-	-	-	-
FAT-ST (Zheng et al., 2021a)	✓	✓	✓	25.5	-	-	30.8	-	-	-	30.1	-
JT-S-MT (Tang et al., 2021a)	×	×	✓	26.8	37.4	-	**31.0**	-	-	-	-	-
SATE (Xu et al., 2021)	×	✓	✓	28.1†	-	-	-	-	-	-	-	-
Chimera (Han et al., 2021)	✓	×	✓	27.1†	35.6	17.4	30.6	25.0	24.0	30.2	29.2	27.4
XSTNet (Ye et al., 2021)	✓	×	✓	27.8	**38.0**	**18.5**	30.8	**26.4**	**25.7**	**32.4**	**31.2**	**28.8**
W2V2-Transformer	✓	×	✓	26.9	36.6	17.3	30.0	25.4	23.9	30.7	29.6	27.6
STEMM	✓	×	✓	**28.7****	37.4**	17.8**	**31.0****	25.8*	24.5**	31.7**	30.5**	28.4

Table 2: BLEU scores on MuST-C `tst-COMMON` set. "Speech" denotes unlabeled audio data. † use OpenSubtitles (Lison and Tiedemann, 2016) as external MT data. * and ** mean the improvements over W2V2-Transformer baseline is statistically significant ($p < 0.05$ and $p < 0.01$, respectively).

Models	WER↓	MT BLEU↑	ST BLEU↑
Cascaded	9.9	31.7	27.5
W2V2-Transformer	-	31.7	26.9
STEMM	-	31.7	**28.7****

Table 3: Comparison with cascaded baseline on MuST-C En-De `tst-COMMON` set. ** mean the improvements over cascaded baseline is statistically significant ($p < 0.01$).

Model Configuration Our model consists of three modules. For the *acoustic encoder*, we use Wav2vec2.0 (Baevski et al., 2020) following the base configuration, which is pretrained on audio data from LibriSpeech (Panayotov et al., 2015) without finetuning[4]. We add two additional 1-dimensional convolutional layers to further shrink the audio, with kernel size 5, stride size 2, padding 2, and hidden dimension 1024. For the *translation encoder*, we use $N_e = 6$ transformer encoder layers. For the *translation decoder*, we use $N_d = 6$ transformer decoder layers. Each of these transformer layers comprises 512 hidden units, 8 attention heads, and 2048 feed-forward hidden units.

Training and Inference We train our model in a pretrain-finetune manner. During pretraining, we train the MT model i.e., *translation encoder* and *translation decoder*, with *transcription-translation* pairs. The learning rate is 7e-4. We train the model with at most 33k input tokens per batch. During finetuning, the learning rate is set to 1e-4. We finetune the whole model up to 25 epochs to avoid overfitting, with at most 16M source audio frames per batch. The training will early-stop if the loss on *dev* set did not decrease for ten epochs. During both pretraining and finetuning, we use an Adam optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9, \beta_2 = 0.98$, and 4k warm-up updates. The learning rate will decrease proportionally to the inverse square root of the step number after warm-up. The dropout is set to 0.1, and the value of label smoothing is set to 0.1. We use the uncertainty-aware mixup ratio strategy by default, and the mixup ratio p^* is set to 0.4 when using static strategy. The weight λ of JSD loss is set to 1.0.

During inference, We average the checkpoints of the last 10 epochs for evaluation. We use beam search with a beam size of 5. We use sacreBLEU[5] (Post, 2018) to compute case-sensitive detokenized BLEU (Papineni et al., 2002) scores and the statistical significance of translation results with paired bootstrap resampling (Koehn, 2004) for a fair comparison[6]. All models are trained on 8 Nvidia Tesla-V100 GPUs. We implement our models based on fairseq[7] (Ott et al., 2019).

Baseline Systems We compare our method with several strong end-to-end ST systems including:

[4] Model can be downloaded at https://dl.fbaipublicfiles.com/fairseq/wav2vec/wav2vec_small.pt

[5] https://github.com/mjpost/sacrebleu

[6] sacreBLEU signature: nrefs:1 | bs:1000 | seed:12345 | case:mixed | eff:no | tok:13a | smooth:exp | version:2.0.0

[7] https://github.com/pytorch/fairseq

Fairseq ST (Wang et al., 2020a), AFS (Zhang et al., 2020), DDT (Le et al., 2020), MTL (Tang et al., 2021b), Self-training (Pino et al., 2020), BiKD (Inaguma et al., 2021a), FAT-ST (Zheng et al., 2021a), JT-S-MT (Tang et al., 2021a), SATE (Xu et al., 2021), Chimera (Han et al., 2021) and XSTNet (Ye et al., 2021). Besides, we implement a strong baseline W2V2-Transformer based on Wav2vec2.0. It has the same model architecture as our proposed STEMM and is pretrained in the same way. The only difference is that it is only finetuned on the ST task, while we adopt a self-learning framework during finetuning.

4 Results and Analysis

4.1 Results on MuST-C Dataset

Comparison with End-to-end Baselines As shown in Table 2, our implemented W2V2-Transformer is a relatively strong baseline, which proves the effectiveness of Wav2vec2.0 module and MT pretraining. Without external MT data, our method achieves an improvement of 1.0 BLEU (average over 8 directions) over the strong baseline, which proves our proposed self-learning framework could effectively improve the performance of the ST task. It even outperforms baselines with external MT data on En-Es, En-It, En-Ro, En-Pt, and En-Nl. When we introduce additional MT data, our method also yields a 0.8 BLEU improvement compared with baseline. Note that our performance is slightly worse than XSTNet (Ye et al., 2021). However, our method is orthogonal with theirs, which focuses on the training procedure of end-to-end ST model. We will investigate how to combine them together in the future.

Comparison with Cascaded Baseline We also implement a strong cascaded system, whose ASR part is composed of a pretrained Wav2vec2.0 module and 6 transformer decoder layers, and the MT part is the same as our pretrained MT module. Both cascaded systems and end-to-end models are trained with the same data ($\mathcal{D}$ and $\mathcal{D}_{\text{MT}}$). As shown in Table 3, the end-to-end baseline W2V2-Transformer is inferior to the cascaded system, but our method significantly outperforms it, which shows the potential of our STEMM method.

4.2 Ablation Studies

Is Each Learning Objective Effective? As shown in Equation 6, our training objective contains three terms. Besides the cross-entropy objective $\mathcal{L}_{\text{CE}}(\mathbf{s}, \mathbf{y})$ for speech translation, we investigate the effects of the other two auxiliary training objectives. As shown in Table 4, when we input the additional multimodal mixed sequence into the model and optimize the cross-entropy loss (Line 3), it can already outperform the baseline (Line 4) significantly. When we regularize two output predictions with JSD loss (Line 2), the performance can be further boosted.

Mixup Ratio	STEMM Trans.	JSD	BLEU
uncertainty-aware	✓	✓	**28.7****
static	✓	✓	28.5**
static	✓	×	27.9**
static	×	×	26.9

Table 4: BLEU scores on MuST-C En-De `tst-COMMON` set with different auxiliary training objectives. STEMM Trans. indicates the criterion entropy loss of translation of multimodal mixed sequence $\mathcal{L}_{\text{CE}}(\text{Mixup}((\mathbf{s}, \mathbf{x}), p^*), \mathbf{y})$. ** mean the improvements over W2V2-Transformer baseline (last row in the table) is statistically significant ($p < 0.01$).

The uncertainty-aware strategy reduces the cost for searching mixup ratio and has better performance. We present two different mixup ratio strategies in Section 2.5. To evaluate their impacts, we conduct another ablation study on MuST-C En-De. We observe that the BLEU scores on `tst-COMMON` set are 28.5 and 28.7 for *static strategy* and *uncertainty-aware strategy*, respectively. The *uncertainty-aware strategy* can slightly improve the performance, and more importantly, it lowers the manual cost for searching an optimal mixup ratio to get the best performance.

4.3 What is the Optimal Mixup Ratio?

When using *static mixup ratio strategy*, it is important to choose the mixup ratio p^*. We constrain p^* in $[0.0, 0.2, 0.4, 0.6, 0.8]$ for experiments on MuST-C En-De `tst-COMMON` set, as shown in Figure 3. When $p^* = 0.0$, the translation task with the mixed sequence as input degrades to the MT task. We interestingly find that self-learning with MT tasks performed the worst (i.e. lowest BLEU) than self-learning with STEMM at other mixup ratios. This confirms what we mentioned in Section 1, that the representation discrepancy between speech and text makes the MT task an inferior boost to ST.

Our method achieves the best performance at $p^* = 0.4$. To find a reasonable explanation, we do a more in-depth study of the representation of the speech, text, and their mixup sequence (STEMM).

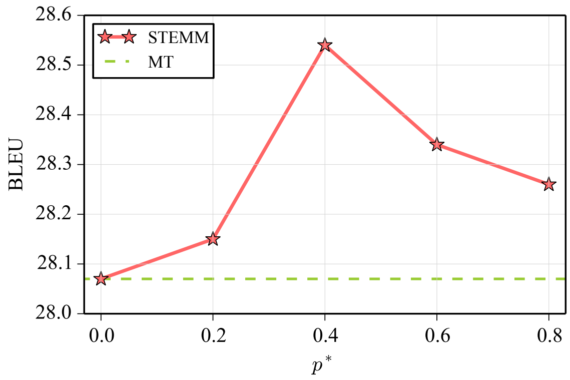

Figure 3: BLEU scores on MuST-C En-De `tst-COMMON` set with different mixup ratio p^*. Our method achieves best performance when $p^* = 0.4$. When $p^* = 0.0$, STEMM will degrade to text-only sequence, which we denote as MT.

In Figure 4, we take out the sequential representation of the speech (output of acoustic encoder), text sequences (output of embedding layer), and the STEMM sequences, average them over the sequence dimension, and apply the T-SNE dimensionality reduction algorithm to reduce the 512 dimensions to two dimensions. We plot the bivariate kernel density estimation based on the reduced 2-dim representation. We find that when $p^* = 0.4$, the mixup representation just lies between the representation of speech and text sequences. That is why it calibrates the cross-modal representation discrepancy more easily and gets the best ST performance.

4.4 Can Our Model Alleviate Cross-modal Representation Discrepancy?

To examine whether our method alleviates the cross-modal representation discrepancy, we conduct some analysis of cross-modal word representations. As described in Section 2.3, for each word unit w_i, we identify the corresponding segment of speech representation $[\mathbf{a}_{l_i} : \mathbf{a}_{r_i}]$ and text embedding $[\mathbf{e}_{m_i} : \mathbf{e}_{n_i}]$. We define the word representation in each modality as follows:

$$\boldsymbol{\alpha}_i = \text{AvgPool}([\mathbf{a}_{l_i} : \mathbf{a}_{r_i}]), \quad (9)$$

$$\boldsymbol{\varepsilon}_i = \text{AvgPool}([\mathbf{e}_{m_i} : \mathbf{e}_{n_i}]), \quad (10)$$

where AvgPool() denotes average-pooling operation across the sequence dimension, $\boldsymbol{\alpha}_i$ and $\boldsymbol{\varepsilon}_i$ denote the representation of word unit w_i in speech and text modalities, respectively.

We calculate the average cosine similarity between $\boldsymbol{\alpha}_i$ and $\boldsymbol{\varepsilon}_i$ over all word units w_i in MuST-C En-De `tst-COMMON` set. As shown in Table 5, our method could significantly improve the similarity of word representations across modalities over baseline. We believe it is because when training with our proposed STEMM, the speech segment and text segment of a word will appear in a similar multimodal context, which leads to similar representations. We also show the visualization of an example in Figure 5, we can observe that our method brings word representations within different modalities closer compared with baseline.

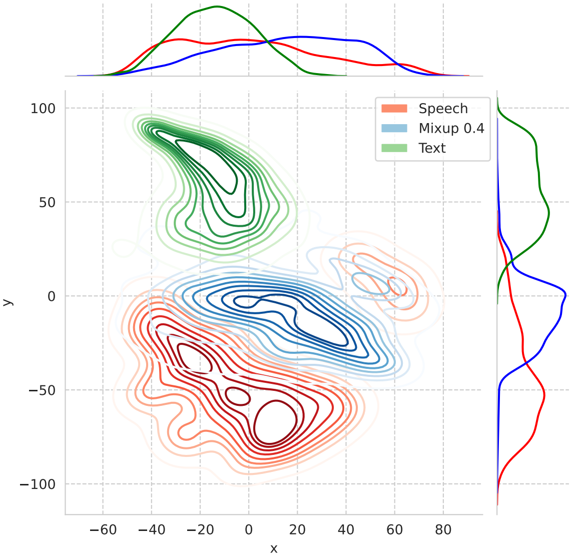

Figure 4: The bivariate kernel density estimation visualization of the averaged sentence representation of the speech, text and STEMM sequences after pretraining. T-SNE algorithm is applied to reduce the 512-dim representations to two dimensions. The green line stands for the averaged sentence embedding. The red line stands for the averaged speech representation. the blue line is the representation for STEMM with mixup ratio $p^* = 0.4$. We observe that the representation of the mixed sequence is in between that of speech and text, which fills the gap between the representation of speech sequences and text sequences. Best view in color.

Models	Similarity (%)
W2V2-Transformer	32.31
STEMM	**51.89**

Table 5: Comparison of word-level representation similarity across modalities.

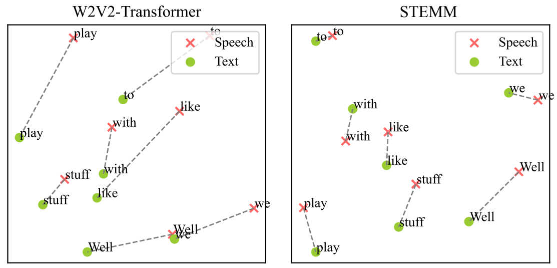

Figure 5: Visualization of word representations in speech and text modalities. We visualize the representations by reducing the dimension with Principal Component Analysis (PCA). Our method brings word representation within different modalities closer.

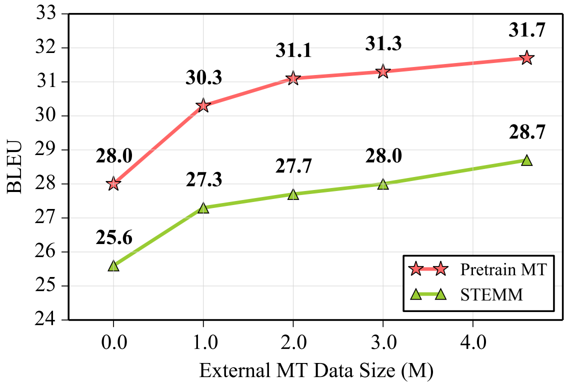

Figure 6: Curve of BLEU scores on MuST-C En-De `tst-COMMON` against the size of external MT data used during pretraining.

4.5 How the Size of MT Data Influences Performance?

One important contributor to our excellent performance is the usage of external MT data. Therefore, how the amount of MT data affects the final performance is an important question. We vary the amount of available external MT data during pretraining on En-De direction. As shown in Figure 6, we observe a continuous improvement of BLEU scores with the increase of MT data, which shows that external MT data is helpful to improve ST.

4.6 Can the Final Model still Perform MT Task?

Our model is first pretrained on the MT task and then finetune for ST. An important question is whether there is a catastrophic forgetting problem during finetuning. We evaluate the model on the MT task and show the result in Table 6. We observe that when we only finetune the model on the ST task (W2V2-Transformer), the ability of text translation will be forgotten a lot. In contrast, when we use our self-learning framework during finetuning, even though there is no MT task, the MT capability can still be preserved.

Models	**BLEU**
Pretrained MT	**31.7**
W2V2-Transformer	19.5
STEMM	31.5

Table 6: BLEU scores of MT task on MuST-C En-De `tst-COMMON` set. Our proposed method almost preserves the text translation capability of pretrained MT model.

5 Related Works

End-to-end ST To overcome the error propagation and high latency in the cascaded ST systems, Bérard et al. (2016); Duong et al. (2016) proved the potential of end-to-end ST without intermediate transcription, which has attracted much attention in recent years (Vila et al., 2018; Salesky et al., 2018, 2019; Di Gangi et al., 2019b,c; Bahar et al., 2019a; Inaguma et al., 2020). Since it is difficult to train an end-to-end ST model directly, some training techniques like pretraining (Weiss et al., 2017; Berard et al., 2018; Bansal et al., 2019; Stoian et al., 2020; Wang et al., 2020b; Pino et al., 2020; Dong et al., 2021a; Alinejad and Sarkar, 2020; Zheng et al., 2021b; Xu et al., 2021), multi-task learning (Le et al., 2020; Vydana et al., 2021; Tang et al., 2021b; Ye et al., 2021; Tang et al., 2021a), curriculum learning (Kano et al., 2017; Wang et al., 2020c), and meta-learning (Indurthi et al., 2020) have been applied. To overcome the scarcity of ST data, Jia et al. (2019); Pino et al. (2019); Bahar et al. (2019b) proposed to generate synthesized data based on ASR and MT corpora. To overcome the modality gap, Han et al. (2021); Huang et al. (2021); Xu et al. (2021) further encode acoustic states which are more adaptive to the decoder. Previous works have mentioned that the modality gap between speech and text is one of the obstacles in the speech translation task, and to overcome such gap, one branch of the works (Liu et al., 2020b; Dong et al., 2021b; Xu et al., 2021) introduced a second encoder based on the conventional encoder-decoder model, to extract semantic information of speech and text. Recently, Han et al. (2021) built a shared semantic projection module that simulates the human brain, while in this work, we explored how to construct an intermediate state of the two modalities via the recent mixup method (*i.e.* **S**peech-**TE**xt **M**anifold **M**ixup) to narrow such gap. Note that our work is orthogonal with Ye et al. (2021)'s study in training procedure of end-to-end ST model.

Mixup Our work is inspired by the mixup strategy. Zhang et al. (2018) first proposed mixup as a data augmentation method to improve the robustness and the generalization of the model, where additional data are constructed as the linear interpolation of two random examples and their labels at the surface level. Verma et al. (2019) extended the surface-level mixup to the hidden representation by constructing *manifold mixup* interpolations. Recent work has introduced mixup on machine translation (Zhang et al., 2019b; Li et al., 2021a; Guo et al., 2022; Fang and Feng, 2022), sentence classification (Chen et al., 2020; Jindal et al., 2020; Sun et al., 2020), multilingual understanding (Yang et al., 2022), and speech recognition (Medennikov et al., 2018; Sun et al., 2021; Lam et al., 2021a; Meng et al., 2021), and obtained enhancements. Our approach is the first to introduce the idea of manifold mixup to the speech translation task with two modalities, speech, and text.

6 Conclusion

In this paper, we propose a **S**peech-**TE**xt **M**anifold **M**ixup (STEMM) method to mix up the speech representation sequences and word embedding sequences. Based on STEMM, we adopt a self-learning framework, which learns the translation of unimodal speech sequences and multimodal mixed sequences in parallel, and regularizes their output predictions. Experiments and analysis demonstrate the effectiveness of our proposed method, which can alleviate the cross-modal representation discrepancy to some extent and improve the performance of ST. In the future, we will explore how to further eliminate this discrepancy and fill the cross-modal transfer gap for ST.

Acknowledgements

We thank all the anonymous reviewers for their insightful and valuable comments. This work was supported by National Key R&D Program of China (NO. 2017YFE0192900).

References

Ashkan Alinejad and Anoop Sarkar. 2020. Effectively pretraining a speech translation decoder with machine translation data. In *Proc. of EMNLP*, pages 8014–8020.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33.

Parnia Bahar, Tobias Bieschke, and Hermann Ney. 2019a. A comparative study on end-to-end speech to text translation. In *Proc. of ASRU*, pages 792–799. IEEE.

Parnia Bahar, Albert Zeyer, Ralf Schlüter, and Hermann Ney. 2019b. On using specaugment for end-to-end speech translation. In *Proc. of IWSLT*.

Sameer Bansal, Herman Kamper, Karen Livescu, Adam Lopez, and Sharon Goldwater. 2019. Pre-training on high-resource speech recognition improves low-resource speech-to-text translation. In *Proc. of NAACL-HLT*, pages 58–68.

Alexandre Berard, Laurent Besacier, Ali Can Kocabiyikoglu, and Olivier Pietquin. 2018. End-to-end automatic speech translation of audiobooks. In *Proc. of ICASSP*, pages 6224–6228.

Alexandre Bérard, Olivier Pietquin, Christophe Servan, and Laurent Besacier. 2016. Listen and translate: A proof of concept for end-to-end speech-to-text translation. In *NIPS workshop on End-to-end Learning for Speech and Audio Processing*.

Jiaao Chen, Zichao Yang, and Diyi Yang. 2020. Mixtext: Linguistically-informed interpolation of hidden space for semi-supervised text classification. In *Proc. of ACL*, pages 2147–2157.

Yong Cheng, Zhaopeng Tu, Fandong Meng, Junjie Zhai, and Yang Liu. 2018. Towards robust neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1756–1766, Melbourne, Australia. Association for Computational Linguistics.

Mattia A. Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019a. 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, Volume 1 (Long and Short Papers)*, pages 2012–2017, Minneapolis, Minnesota. Association for Computational Linguistics.

Mattia A Di Gangi, Matteo Negri, and Marco Turchi. 2019b. Adapting transformer to end-to-end spoken language translation. In *Proc. of INTERSPEECH*, pages 1133–1137. International Speech Communication Association (ISCA).

Mattia Antonino Di Gangi, Matteo Negri, Roldano Cattoni, Roberto Dessi, and Marco Turchi. 2019c. Enhancing transformer for end-to-end speech-to-text translation. In *Proceedings of Machine Translation Summit XVII Volume 1: Research Track*, pages 21–31.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019a. Unified language model pre-training for natural language understanding and generation. In *Proc. of NeurIPS*, pages 13063–13075.

Qianqian Dong, Feng Wang, Zhen Yang, Wei Chen, Shuang Xu, and Bo Xu. 2019b. Adapting translation models for transcript disfluency detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6351–6358.

Qianqian Dong, Mingxuan Wang, Hao Zhou, Shuang Xu, Bo Xu, and Lei Li. 2021a. Consecutive decoding for speech-to-text translation. In *The Thirty-fifth AAAI Conference on Artificial Intelligence, AAAI*.

Qianqian Dong, Rong Ye, Mingxuan Wang, Hao Zhou, Shuang Xu, Bo Xu, and Lei Li. 2021b. Listen, understand and translate: Triple supervision decouples end-to-end speech-to-text translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Long Duong, Antonios Anastasopoulos, David Chiang, Steven Bird, and Trevor Cohn. 2016. An attentional model for speech translation without transcription. In *Proc. of NAACL-HLT*, pages 949–959.

Qingkai Fang and Yang Feng. 2022. Neural machine translation with phrase-level universal visual representations. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*.

Marco Gaido, Mattia A Di Gangi, Matteo Negri, and Marco Turchi. 2020. End-to-end speech-translation with knowledge distillation: Fbk@ iwslt2020. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 80–88.

Dengji Guo, Zhengrui Ma, Min Zhang, and Yang Feng. 2022. Prediction difference regularization against perturbation for neural machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*.

Chi Han, Mingxuan Wang, Heng Ji, and Lei Li. 2021. Learning shared semantic space for speech-to-text translation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2214–2225, Online. Association for Computational Linguistics.

Wuwei Huang, Dexin Wang, and Deyi Xiong. 2021. AdaST: Dynamically adapting encoder states in the decoder for end-to-end speech-to-text translation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2539–2545, Online. Association for Computational Linguistics.

Hirofumi Inaguma, Tatsuya Kawahara, and Shinji Watanabe. 2021a. Source and target bidirectional knowledge distillation for end-to-end speech translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1872–1881, Online. Association for Computational Linguistics.

Hirofumi Inaguma, Tatsuya Kawahara, and Shinji Watanabe. 2021b. Source and target bidirectional knowledge distillation for end-to-end speech translation. In *Proceedings of NAACL*, pages 1872–1881.

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 *Proc. of ACL*, pages 302–311.

Sathish Indurthi, Houjeung Han, Nikhil Kumar Lakumarapu, Beomseok Lee, Insoo Chung, Sangha Kim, and Chanwoo Kim. 2020. Data efficient direct speech-to-text translation with modality agnostic meta-learning. In *Proc. of ICASSP*. IEEE.

Ye Jia, Melvin Johnson, Wolfgang Macherey, Ron J. Weiss, Yuan Cao, Chung-Cheng Chiu, Naveen Ari, Stella Laurenzo, and Yonghui Wu. 2019. Leveraging weakly supervised data to improve end-to-end speech-to-text translation. In *Proc. of ICASSP*, pages 7180–7184.

Amit Jindal, Arijit Ghosh Chowdhury, Aniket Didolkar, Di Jin, Ramit Sawhney, and Rajiv Shah. 2020. Augmenting nlp models using latent feature interpolations. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6931–6936.

Takatomo Kano, Sakriani Sakti, and Satoshi Nakamura. 2017. Structured-based curriculum learning for end-to-end english-japanese speech translation. In *Proc. of INTERSPEECH*, pages 2630–2634.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster)*.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Tsz Kin Lam, Mayumi Ohta, Shigehiko Schamoni, and Stefan Riezler. 2021a. On-the-fly aligned data augmentation for sequence-to-sequence ASR. *CoRR*, abs/2104.01393.

Tsz Kin Lam, Shigehiko Schamoni, and Stefan Riezler. 2021b. Cascaded models with cyclic feedback for direct speech translation. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7508–7512. IEEE.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. In *Proc. of NeurIPS*.

Hang Le, Juan Pino, Changhan Wang, Jiatao Gu, Didier Schwab, and Laurent Besacier. 2020. Dual-decoder transformer for joint automatic speech recognition and multilingual speech translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3520–3533, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Jicheng Li, Pengzhi Gao, Xuanfu Wu, Yang Feng, Zhongjun He, Hua Wu, and Haifeng Wang. 2021a. Mixup decoding for diverse machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 312–320, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wei Li, Can Gao, Guocheng Niu, Xinyan Xiao, Hao Liu, Jiachen Liu, Hua Wu, and Haifeng Wang. 2021b. Unimo: Towards unified-modal understanding and generation via cross-modal contrastive learning. In *Proc. of ACL*.

Zehui Lin, Xiao Pan, Mingxuan Wang, Xipeng Qiu, Jiangtao Feng, Hao Zhou, and Lei Li. 2020. Pre-training multilingual neural machine translation by leveraging alignment information. In *Proc. of EMNLP*, pages 2649–2663.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020a. Multilingual denoising pre-training for neural machine translation. *TACL*, 8:726–742.

Yuchen Liu, Hao Xiong, Zhongjun He, Jiajun Zhang, Hua Wu, Haifeng Wang, and Chengqing Zong. 2019. End-to-end speech translation with knowledge distillation. *arXiv preprint arXiv:1904.08075*.

Yuchen Liu, Junnan Zhu, Jiajun Zhang, and Chengqing Zong. 2020b. Bridging the modality gap for speech-to-text translation. *arXiv preprint arXiv:2010.14920*.

Ivan Medennikov, Yuri Y Khokhlov, Aleksei Romanenko, Dmitry Popov, Natalia A Tomashenko, Ivan Sorokin, and Alexander Zatvornitskiy. 2018. An investigation of mixup training strategies for acoustic models in asr. In *Proc. of Interspeech*, pages 2903–2907.

Linghui Meng, Jin Xu, Xu Tan, Jindong Wang, Tao Qin, and Bo Xu. 2021. Mixspeech: Data augmentation for low-resource automatic speech recognition. In *Proc. of ICASSP*, pages 7008–7012. IEEE.

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

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210.

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

Juan Pino, Liezl Puzon, Jiatao Gu, Xutai Ma, Arya D McCarthy, and Deepak Gopinath. 2019. Harnessing indirect training data for end-to-end automatic speech translation: Tricks of the trade. In *Proc. of IWSLT*.

Juan Pino, Qiantong Xu, Xutai Ma, Mohammad Javad Dousti, and Yun Tang. 2020. Self-training for end-to-end speech translation. In *Proc. of INTERSPEECH*.

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

Elizabeth Salesky, Susanne Burger, Jan Niehues, and Alex Waibel. 2018. Towards fluent translations from disfluent speech. In *Proc. of SLT*, pages 921–926. IEEE.

Elizabeth Salesky, Matthias Sperber, and Alexander Waibel. 2019. Fluent translations from disfluent speech in end-to-end speech translation. In *Proc. of NAACL-HLT*, pages 2786–2792.

Matthias Sperber, Graham Neubig, Jan Niehues, and Alex Waibel. 2017. Neural lattice-to-sequence models for uncertain inputs. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1380–1389, Copenhagen, Denmark. Association for Computational Linguistics.

Matthias Sperber, Graham Neubig, Ngoc-Quan Pham, and Alex Waibel. 2019. Self-attentional models for lattice inputs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1185–1197, Florence, Italy. Association for Computational Linguistics.

Mihaela C Stoian, Sameer Bansal, and Sharon Goldwater. 2020. Analyzing asr pretraining for low-resource speech-to-text translation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7909–7913. IEEE.

Jianwei Sun, Zhiyuan Tang, Heng Yin, Wei Wang, Xi Zhao, Shuaijiang Zhao, Xiaoning Lei, Wei Zou, and Xiangang Li. 2021. Semantic data augmentation for end-to-end mandarin speech recognition. *Interspeech 2021*.

Lichao Sun, Congying Xia, Wenpeng Yin, Tingting Liang, S Yu Philip, and Lifang He. 2020. Mixup-transformer: Dynamic data augmentation for nlp tasks. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3436–3440.

Yun Tang, Juan Pino, Xian Li, Changhan Wang, and Dmitriy Genzel. 2021a. Improving speech translation by understanding and learning from the auxiliary text translation task. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4252–4261, Online. Association for Computational Linguistics.

Yun Tang, Juan Pino, Changhan Wang, Xutai Ma, and Dmitriy Genzel. 2021b. A general multi-task learning framework to leverage text data for speech to text tasks. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6209–6213.

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

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. 2019. Manifold mixup: Better representations by interpolating hidden states. In *Proc. of ICML*, pages 6438–6447. PMLR.

Laura Cross Vila, Carlos Escolano, José AR Fonollosa, and Marta R Costa-jussà. 2018. End-to-end speech translation with the transformer. In *IberSPEECH*, pages 60–63.

Hari Krishna Vydana, Martin Karafiát, Katerina Zmolikova, Lukáš Burget, and Honza Černockỳ. 2021. Jointly trained transformers models for spoken language translation. In *Proc. of ICASSP*, pages 7513–7517. IEEE.

Changhan Wang, Yun Tang, Xutai Ma, Anne Wu, Dmytro Okhonko, and Juan Pino. 2020a. fairseq s2t: Fast speech-to-text modeling with fairseq. In *Proceedings of the 2020 Conference of the Asian Chapter of the Association for Computational Linguistics (AACL): System Demonstrations*.

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 *Proc. of AAAI*, volume 34, pages 9161–9168.

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*, pages 3728–3738, Online. Association for Computational Linguistics.

Ron J. Weiss, Jan Chorowski, Navdeep Jaitly, Yonghui Wu, and Zhifeng Chen. 2017. Sequence-to-sequence models can directly translate foreign speech. In *Proc. of INTERSPEECH*, pages 2625–2629.

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 (Volume 1: Long Papers)*, pages 2619–2630, Online. Association for Computational Linguistics.

Huiyun Yang, Huadong Chen, Hao Zhou, and Lei Li. 2022. Enhancing cross-lingual transfer by manifold mixup. In *ICLR*.

Rong Ye, Mingxuan Wang, and Lei Li. 2021. End-to-end speech translation via cross-modal progressive training. In *Proc. of INTERSPEECH*.

Biao Zhang, Ivan Titov, Barry Haddow, and Rico Sennrich. 2020. Adaptive feature selection for end-to-end speech translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2533–2544, Online. Association for Computational Linguistics.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization. In *Proc. of ICLR*.

Pei Zhang, Niyu Ge, Boxing Chen, and Kai Fan. 2019a. Lattice transformer for speech translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6475–6484, Florence, Italy. Association for Computational Linguistics.

Wen Zhang, Yang Feng, Fandong Meng, Di You, and Qun Liu. 2019b. Bridging the gap between training and inference for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4334–4343, Florence, Italy. Association for Computational Linguistics.

Renjie Zheng, Junkun Chen, Mingbo Ma, and Liang Huang. 2021a. Fused acoustic and text encoding for multimodal bilingual pretraining and speech translation. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 12736–12746. PMLR.

Renjie Zheng, Junkun Chen, Mingbo Ma, and Liang Huang. 2021b. Fused acoustic and text encoding for multimodal bilingual pretraining and speech translation. In *Proc. of ICML*.

Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason Corso, and Jianfeng Gao. 2020. Unified vision-language pre-training for image captioning and vqa. In *Proc. of AAAI*, volume 34, pages 13041–13049.