Learning When to Translate for Streaming Speech

Qianqian Dong[1]*, **Yaoming Zhu**[1]*, **Mingxuan Wang**[1], **Lei Li**[2] †
[1] ByteDance AI Lab   [2] University of California, Santa Barbara
{dongqianqian,zhuyaoming,wangmingxuan.89}@bytedance.com
leili@cs.ucsb.edu

Abstract

How to find proper moments to generate partial sentence translation given a streaming speech input? Existing approaches waiting-and-translating for a fixed duration often break the acoustic units in speech, since the boundaries between acoustic units in speech are not even. In this paper, we propose MoSST, a simple yet effective method for translating streaming speech content. Given a usually long speech sequence, we develop an efficient **mo**notonic **s**egmentation module inside an encoder-decoder model to accumulate acoustic information incrementally and detect proper speech unit boundaries for the input in **s**peech **t**ranslation task. Experiments on multiple translation directions of the MuST-C dataset show that MoSST outperforms existing methods and achieves the best trade-off between translation quality (BLEU) and latency. Our code is available at https://github.com/dqqcasia/mosst.

1 Introduction

Speech translation (ST) aims at translating from source language speech into target language text, which is widely helpful in various scenarios such as conference speeches, business meetings, cross-border customer service, and overseas travel. There are two kinds of application scenarios, including the non-streaming translation and the streaming one. The non-streaming models can listen to the complete utterances at one time and then generate the translation afterward. While, the streaming models need to balance the latency and quality and generate translations based on the partial utterance, as shown in Figure 1.

Recently, end-to-end approaches have achieved remarkable progress in non-streaming ST. Previous work (Weiss et al., 2017; Berard et al., 2018; Bansal et al., 2019a,b; Alinejad and Sarkar, 2020;

Figure 1: An illustration of streaming speech-to-text translation. ST models listen to the audio in source language, and generate tokens in target language.

Stoian et al., 2020) Ansari et al. (2020) has shown that an end-to-end model achieves even better performance compared to the cascaded competitors. However, attempts at end-to-end streaming ST are still not fully explored. Traditional streaming ST is usually formed by cascading a streaming speech recognition module with a streaming machine translation module (Oda et al., 2014; Dalvi et al., 2018). Most of the previous work focuses on simultaneous text translation (Gu et al., 2017a). Ma et al. (2019) propose a novel wait-k strategy based on the prefix-to-prefix framework, which is one of the popular research methods of simultaneous text translation. For end-to-end streaming ST, Ma et al. (2020b); Ren et al. (2020); Ma et al. (2021a) introduce the methodology of streaming machine translation into streaming ST and formalize the task, which belongs to the first study to propose simultaneous ST in an end-to-end manner.

However, those previous streaming ST systems generally treat a fixed time-span of audio as a acoustic unit and translate new words based on fixed time segmentation, which might be unfavorable for streaming ST translation. Since the speaker's speech speed and the length of the phonemes are distinct, previous methods cannot find the best pol-

---

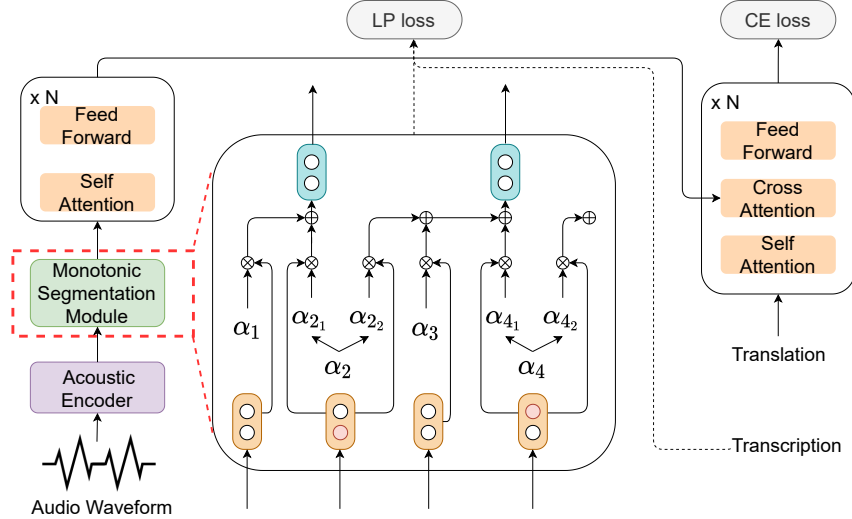*Equal contribution.
† Work is done while at ByteDance.

Figure 2: Overview of the proposed MoSST. MoSST consists of a pre-trained acoustic model, a monotonic segmentation module (MSM) module, and standard Transformer blocks. The acoustic model extracts features from the raw audio waveform. MSM learns a soft and monotonic alignment over the extracted features from the acoustic model and outputs accumulated acoustic vectors as the input for down-streaming Transformer blocks.

icy to tell whether to continue reading source audio or translate new words when the source audio is streaming in. Hence, we expect the model can determine whether the streaming audio information input is enough to translate new words, similar to the manual simultaneous interpretation. This idea inspires Monotonic-segmented Streaming Speech Translation (MoSST) system. Specifically, we design a new module that helps to judge the acoustic boundaries of the input audio. We then propose a translation strategy that enables the model to decide whether to read the audio stream or write new tokens given the audio prefix. With the new module and decoding strategy, the model's performance on streaming speech translation has been significantly improved.

We highlight our innovations and findings as follows:

- We propose a simple but effective framework, MoSST for streaming speech translation.
- We introduce a new monotonic segmentation module to segment audio waveform into acoustic units, based on which we design the adaptive decision strategy which dynamically decides when to translate a new word in streaming scenarios.
- We validate MoSST on the MuST-C dataset. The results show that our model significantly outperforms SOTA baselines. Surprisingly, we also find that MoSST can rival or even

surpass other SOTA systems in non-streaming speech translation. Furthermore, we conduct a comprehensive study to analyze the utility of the proposed module and decoding strategy.

2 Proposed Method: MoSST

This section first formulates the ST task in streaming and non-streaming scenarios. Then, we introduce the detailed architecture of MoSST, as shown in Figure 2. Finally, we give the training and inference strategies of MoSST for streaming and non-streaming cases.

2.1 Problem Formulation

The ST corpus usually contains speech-transcription-translation triples $(\mathbf{x}, \mathbf{z}, \mathbf{y})$. Specially, $\mathbf{x} = (x_1, ..., x_{T_x})$ is a sequence of acoustic features. $\mathbf{z} = (z_1, ..., z_{T_z})$ and $\mathbf{y} = (y_1, ..., y_{T_y})$ represents the corresponding transcription in source language and the translation in target language respectively. Usually, the acoustic feature $\mathbf{x}$ is much longer than text sequences $\mathbf{z}$ and $\mathbf{y}$, as the sampling rate of audio is usually above 16,000 Hz, and each word syllable (about 300 ms) will be recorded by thousands of sampling points.

The streaming ST model aims to translate instantly when speech audio streams in, that is, given a valid audio prefix $\mathbf{x}_{<\tau}$, where τ is the time span of the audio piece, we expect the model can translate enough information $\mathbf{y}_{<K}$, where K is the maximum number of tokens that the model can translate

as time τ, *i.e.*:

$$\Pr(\mathbf{y}_{<K}|\mathbf{x}_{<\tau}) = \prod_{t=1}^{K} \Pr(y_t|\mathbf{x}_{<\tau}, \mathbf{y}_{<t}; \theta) \quad (1)$$

where θ is the parameters of the streaming ST model. Our goal is to find the best θ^* that maximizes the $\Pr(\mathbf{y}_{<K}|\mathbf{x}_{<\tau})$ in Eq. 1.

Note that in our research scenario, we require that the translated piece of the sentence shall not be modified once generated, similar to the settings in simultaneous machine translation (Ma et al., 2019).

2.2 Model Structure

MoSST consists of an acoustic encoder, a monotonic segmentation, and a standard Transformer.
Acoustic Encoder The conventional acoustic encoder using FBANK (log-Mel filterbank, FBANK) as feature extractors faces reduced performance with insufficient training data (San et al., 2021), which is especially the case in speech-to-text translation tasks. The FBANK also leads to potential information loss, and may corrupt long-term correlations (Pardede et al., 2019).

To tackle such problems, we apply the recently-proposed pre-trained acoustic models (Chen et al., 2020; Baevski et al., 2020) as the feature extractor for MoSST. Those pre-trained acoustic models learn the speech representation in a self-supervised learning (SSL) way. Since pre-trained acoustic models require only a large amount of unlabeled speech, which also alleviates the corpus shortage of ST tasks. In this paper, we utilize Wav2Vec2 (Baevski et al., 2020) as our instance.
Monotonic Segmentation Module The previous speech translation model generally attends the whole audio sequence to the translation tokens with a sequence-to-sequence (seq2seq) framework, which brings two problems: *1)* the model does not learn the alignment between audio and translation explicitly, which may confuse the streaming translation model on whether it has read enough acoustic information when generating the translated text; *2)* the audio sequences are usually much longer than text sequences, which is computationally demanding for the conventional encoder-decoder speech-to-text model to apply the global attention mechanism. Such high computational cost deviates from the requirements in streaming translation scenarios.

We introduce a Monotonic Segmentation Module (MSM), to relieve drawbacks of existing mod-els. The MSM is inspired by the integrate-and-fire (IF) model (Abbott, 1999; Dong and Xu, 2020; Yi et al., 2021). Specifically, IF neuron has two modes: *integrate* and *firing*. In integrate mode, the IF neuron dynamically receives signals and accumulates information; when the received information exceeds a certain threshold, IF neuron enters firing mode, at which time it outputs a signal (a.k.a. spiking), where the accumulated state contains information received in the previous integrate phase; and finally, the IF neuron will reset itself and re-enter the integrate mode once the firing mode ends.

In the MSM, we utilize the integrate-and-fire cycle to dynamically locate the boundaries of meaningful speech segments. At the integrate mode, the model keeps reading and processing speech frames, while at firing mode the model writes the translated tokens. MSM takes the representation from the Acoustic Encoder and uses one of the dimensions as signals for integrate-and-fire. These signals are passed through a Sigmoid function to produce integration weights. Once the weights are accumulated to a certain threshold (e.g. =1.0), the module marks the boundary of the current segment and enters a firing mode. It then aggregates the rest dimensions of encoder representations according to the weights within this segment. These are passed to further processing blocks for WRITE operation.

The MSM operations are defined as follows:

$$\alpha_t = \text{sigmoid}(\mathbf{h}_{t,d}) \quad (2)$$

$$\boldsymbol{l}_u = \sum_{t}^{\text{S}_u} \alpha_t' \mathbf{h}_{t,1:d-1} \quad (3)$$

$$\hat{n} = \sum_{t}^{T} \alpha_t \quad (4)$$

$$\alpha_t' = \frac{n^*}{\hat{n}} \alpha_t, \quad (5)$$

Where $\mathbf{h}$ is the acoustic vector as an output of the acoustic encoder, and its subscript denotes the scale value of $\mathbf{h}$ at timestamp t and d-th dimension (*i.e.*, we use the last dimension as the input of IF neurons). The Sigmoid value of the scale $\mathbf{h}_{t,d}$ is the current weight, denoted as α_t. We use the current weight to decide mode conversion from integrate to firing: when the accumulated sum of α_t exceeds the threshold value, the model is believed to have READ sufficient speech signals in this integrate stage, and the IF neural fires the accumulated information $\boldsymbol{l} = (\boldsymbol{l}_1, ..., \boldsymbol{l}_u)$ to fulfill one integrate-and-fire cycle. And S_u represents the firing step

corresponding to l_u.

Note that the accumulated information l is calculated as a weighted sum of acoustic vectors $\mathbf{h}_t$ at a single integrate stage t. We call it as information weight α'_t, which helps to scale the amount of information contained in each integrate stage. We calculate the information weight α'_t by normalizing the current weight α_t with the number of tokens in the corresponding transcription n^*, which divides the length of the accumulated acoustic vector $\hat{n}$.

Transformer block The last module of the MoSST is the standard Transformer. The Transformer blocks take the integrated acoustic vector l from the MSM layer as the input, which aims to extract the semantic feature ($\mathbf{h}^{SE}$) of the input audio. Since MSM has significantly compressed the length of acoustic features, the Transformer can attend the input and output directly without the excessive computational overhead. Note that to ensure that MSM learns the correct length of acoustic units, we use the length of the corresponding transcription as a supervised signal and introduce length penalty loss ("LP loss" in Figure 2) to assist MSM's learning.

$$\mathcal{L}_{\text{lp}}(\theta; \mathbf{x}, \mathbf{z}) = ||n^* - \hat{n}||_2, \quad (6)$$

During inference, an extra rounding operation is applied on $\hat{n}$ to simulate n^*. Based on the matched sequence length, the accumulated acoustic vector l is mapped back into the model size by a randomly initialized fully connected layer.

2.3 Training Strategies

Multi-task Joint Training with ASR MoSST jointly fulfills the ST and ASR tasks with the multi-task learning (MTL) strategy as its main model. To distinguish two tasks, we add two special task indicators at the beginning of the text as the *BOS* operator for decoding. For example, if the audio input for "Thank you ." is in English, for ASR, we use [en] as the *BOS* and decode $\mathbf{z}=$ "[en] Thank you .". We add [De] at the start of German translation, thus $\mathbf{y}$ is "[De] Danke ." Both ST and ASR are optimized with cross-entropy ("CE loss" in Figure 2) losses, defined in Equation (7) and (8) respectively.

$$\mathcal{L}_{st}(\theta; \mathbf{x}, \mathbf{y}) = -\sum_{i=1}^{T_y} \log p_\theta(y_i|y_{<i}, \mathbf{h}^{SE}) \quad (7)$$

$$\mathcal{L}_{asr}(\theta; \mathbf{x}, \mathbf{z}) = -\sum_{i=1}^{T_z} \log p_\theta(z_i|z_{<i}, \mathbf{h}^{SE}) \quad (8)$$

where the decoder probability p_θ is calculated from the final softmax layer based on the output of the decoder.

We use the joint training strategy to optimize all modules. The overall objective function is the weighted sum for all aforementioned losses:

$$\mathcal{L}(\theta; \mathbf{x}, \mathbf{y}, \mathbf{z}) = \alpha\mathcal{L}_{\text{lp}}(\theta; \mathbf{x}, \mathbf{z}) + \beta\mathcal{L}_{ce} \quad (9)$$

Where $\mathcal{L}_{ce}$ represents $\mathcal{L}_{asr}$ or $\mathcal{L}_{st}$. In the following experimental sections, α is set to 0.05, and β is set to 1 by default.

2.4 Inference Strategies

Wait-k Policy MoSST adopts wait-k policy for streaming translation, which originates from simultaneous machine translation (Ma et al., 2019). Wait-k policy waits for K source tokens and then translates target tokens concurrently with the source streams in (*i.e.*, output N tokens when given $N+K$ source tokens).

The previous online ST systems adopt *Pre-fix Decision* (Ma et al., 2021b, 2020b) for wait-k policy, where a fixed time span (usually 280ms) of the source waveform is regarded as a new unit. However, the pre-fixed decision is limited on real-world scenarios since the speech speed of speakers and the length of acoustic units are distinct, where a fixed time stride guarantees neither sufficient information if the phonemes are too long, nor a proper translation latency if the phonemes are too short.

Adaptive Decision We propose a new decision strategy for streaming speech translation, namely *Adaptive Decision*. Our new strategy dynamically decides when to write the new token according to the integrated state length of MSM (*i.e.*, $|l_u|$ in Equation (5)).

Since MSM scales up the acoustic information monotonically, the model can estimate the acoustic boundary for each units in the audio. We use such integrate feature as a basis to tell whether the information carried by the waveform segment is sufficient; hence the proposed adaptive decision revises the drawbacks in fixed decision.

We propose our new decoding policy in Algorithm 1. The new policy utilizes wait-k to decide when to write new translation tokens and adaptive decisions to decide how long the input is regarded

---
Algorithm 1: Adaptive Decision Strategy

Input: The waveform sequence $\mathbf{x}$, the MSM model $\mathbf{M}$, wait lagging K
Output: The translated sentence $\hat{\mathbf{y}}$

1 initialization: the read waveform segment $\hat{\mathbf{x}} = []$, the output sentence $\hat{\mathbf{y}} = []$;
2 **while** $\hat{\mathbf{y}}_{i-1}$ *is not EndOfSentence* **do**
3   calculate MSM integrated state l_u ;
4   **if** $\hat{\mathbf{x}} == \mathbf{x}$;
5   **then**
        `/* the waveform is finished */`
        `/* write new token */`
6         $\hat{\mathbf{y}} = \hat{\mathbf{y}} + $ decoder.predict();
7   M.decoder.update($\hat{\mathbf{y}}$) ;
8   **else if** $|l_u| - |\hat{\mathbf{y}}| < K$;
9   **then**
        `/* read waveform */`
10         $\hat{\mathbf{x}} = \hat{\mathbf{x}} + $ new_segment($\mathbf{x}$);
11   M.encoder.update($\hat{\mathbf{x}}$)
12   **else**
        `/* write new token */`
13         $\hat{\mathbf{y}} = \hat{\mathbf{y}} + $ decoder.predict();
14   M.decoder.update($\hat{\mathbf{y}}$) ;
15 **end**
16 return $\hat{\mathbf{y}}$;
---

as a unit. Specifically, during the online ST translation, the model shall decide whether to read new audio frames or translate a new word at any time, called the READ/WRITE decision. We denote $\hat{\mathbf{x}}$ as the audio sub-sequence that the model has READ from the source and $\hat{\mathbf{y}}$ as the sentence prefix that has been already generated. The wait-k policy makes the READ/WRITE decision according to the length difference between the MSM integrated state $|l_u|$ and the generated sentence $|\hat{\mathbf{y}}|$. When the integrated state $|l_u|$ is K word behind the generated $|\hat{\mathbf{y}}|$, the MoSST generates a new token (line 12) and updates decoder states recursively, otherwise, the model waits and reads the audio streaming (line 9), and updates the encoder states.

Train-full Test-k Streaming translation needs to predict the output based on part of the input. If the *train-full test-k* paradigm is applied, the streaming performance will decrease a lot due to the mismatch between training and inferring. The previous streaming work generally uses a prefix-to-prefix training framework (Ma et al., 2019), implemented by a unidirectional encoder and decoder, and equipped with the waik-k policy. In MoSST, the learned monotonic segmentation module allows our model to have streaming decoding capability without performance drop.

3 Experiments

3.1 Dataset

MuST-C[1] (Di Gangi et al., 2019a) is a multilingual ST corpus with triplet data sources: source audio, transcripts, and text translations. To the best of our knowledge, MuST-C is currently the largest ST dataset available. It includes data from English TED talks with auto-aligned transcripts and translations at the sentence level. We mainly conduct experiments on English-German and English-French language pairs. And we use the dev and tst-COMMON sets as our development and test data, respectively.

k	1	3	5	7	9	inf
SimulSpeech [†]	10.73	15.52	16.90	17.46	17.87	18.29
MoSST	**11.76**	**15.57**	**18.23**	**18.71**	**19.37**	**19.95**

Table 1: Comparison with SimulSpeech on MuST-C EN-DE tst-COMMON test set. [†] represents results from Ren et al. (2020).

3.2 Preprocessing

For speech input, the 16-bit raw wave sequences are normalized by a factor of 2^{15} to the range of $[-1, 1)$. For text input, on each translation pair, all texts (including transcript and translation) are preprocessed in the same way. Texts are case-sensitive. We keep and normalize the punctuations, but remove non-print characters. We tokenize sentences with Moses tokenizer [2] and filter out samples longer than 250 words. For subword modeling, we use a unigram sentencepiece (Kudo and Richardson, 2018) with a dictionary size of 10000. On each translation direction, the sentencepiece model is learned on all text data from ST corpora.

3.3 Model and Experimental Configuration

Model Configuration For audio input, the Wav2Vec2 module follows the base[3] configuration in Baevski et al. (2020). It uses parameters self-supervised pre-trained on LibriSpeech audio data only. The subsequently shared Transformer module has a hidden dimension of 768 and 4 attention heads. The encoder is 8 layers, and the decoder

---

[1] https://ict.fbk.eu/must-c/
[2] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/tokenizer.perl
[3] https://dl.fbaipublicfiles.com/fairseq/wav2vec/wav2vec_small.pt

is 6 layers. We use the simplified version of the Continuous IF implementation (Yi et al., 2021) for MSM module, which introduces no additional parameters except for a fully connected layer.

Experimental Configuration We use an Adam optimizer with $\beta_1 = 0.9, \beta_2 = 0.98$, and 4k warm-up updates. We set the maximum training batch of the waveform audio token to be 3.2 million. We apply an inverse square root schedule algorithm for the learning rate. We average 10 consecutive checkpoints around the one with the best dev loss and adopt a beam size of 5. We implement our models in Fairseq (Ott et al., 2019).

3.4 Evaluation

For offline translation, the model's performance is mainly evaluated with quality metrics. While for streaming translation, ST model is evaluated by the latency-quality trade-off curves.

Quality Metrics We quantify translation accuracy with detokenized BLEU (Papineni et al., 2002) using sacreBLEU [5].

Latency Metrics Existing simultaneous translation work mainly focuses on the latency evaluation of text translation, and has proposed computation unaware metrics, such as Average Proportion (AP) (Cho and Esipova, 2016), Average Latency (AL) (Ma et al., 2019), Continues Wait Length (CW) (Gu et al., 2017b) and Differentiable Average Lagging (DAL) (Cherry and Foster, 2019). Ma et al. (2020a) extends the latency metrics of text translation into ST, including AL, AP, and DAL. The latency metrics for streaming MoSST are evaluated by AL, DAL, and AP based on the SimulEval toolkit [6] (Ma et al., 2020a).

3.5 Experimental Results

3.5.1 Streaming Speech-to-text Translation

We compare the performance of our method with published work on streaming ST tasks. SimulST (Ma et al., 2020b) introduces the wait-k training strategy in simultaneous text translation into simultaneous ST tasks. The comparison result on MuST-C EN-DE tst-COMMON set is shown in Figure 3. It can be seen that MoSST is significantly better than the baseline system in

---

all the three latency metrics and the quality metric. SimulSpeech (Ren et al., 2020) also adopts the wait-k strategy and leverages the connectionist temporal classification (CTC) decoding to split the input streaming speech chunk in real-time. Besides, SimulSpeech introduces attention-level and data-level knowledge distillation (KD) to improve performance. The comparison result on MuST-C EN-DE tst-COMMON set is shown in Table 1. It can be seen that when k ranges from 1 to infinite, our method significantly outperforms SimulSpeech. Existing work all uses the wait-k training strategy implemented with a unidirectional mask, which would damage the performance of offline evaluation in full context. While MoSST can serve well both non-streaming and streaming translation. At the same time, the shrinking mechanism based on the MSM can speed up model convergence, which we give a detailed analysis in Sec A.2.2 of the Appendix.

3.5.2 No-streaming Speech-to-text Translation

We also compare the performance of our method with published work on offline ST tasks under experimental settings without external supervised training data. The result is shown in Table 2. Fairseq (Wang et al., 2020a), ESPnet (Inaguma et al., 2020), and NeurST (Zhao et al., 2021) are recently emerging R&D toolkits for ST. Transformer ST uses a standard SpeechTransformer (Dong et al., 2018) model structure, with a pre-trained ASR model to initialize the encoder and a pre-trained MT model to initialize the decoder. Zhang et al. (2020a) propose adaptive feature selection (AFS) for ST, which applies $\mathcal{L}_0$DROP (Zhang et al., 2021) to dynamically estimate the importance of each encoded speech feature. STAST (Liu et al., 2020b) uses a speech-to-text adaptation method to bridge the modality gap in the semantic space by MTL and representation regulation with MT. Le et al. (2020) adapt the dual-decoder transformer with a dual-attention mechanism to joint ASR and ST for both bilingual (BL) and multilingual (ML) settings. Compared with the best results published so far, MoSST can achieve an improvement of 1.3 BLEU and 0.7 BLEU respectively. It should be noted that the previous methods can be integrated into MoSST to expect better performance. We will leave it for further exploration.
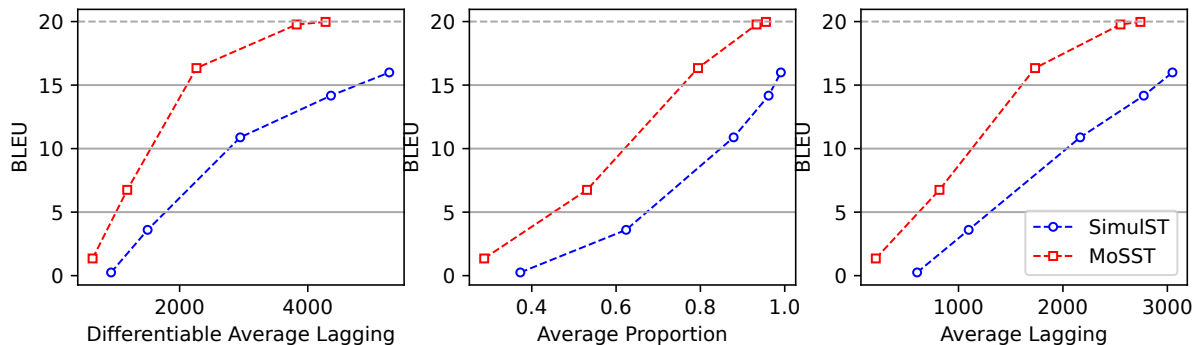

Figure 3: The translation quality against the latency metrics (DAL, AP and AL) on the tst-COMMON set of MuST-C En-De dataset. Decoding strategy here is pre-fixed decision. k in SimulEval is set to 5 as default. The result of SimulST is reproduced by Fairseq [4].

| Model | MuST-C EN-X | |
	EN-DE	EN-FR
Transformer ST Fairseq (Wang et al., 2020a)	22.7	32.9
Transformer ST ESPnet (Inaguma et al., 2020)	22.9	32.8
Transformer ST NeurST (Zhao et al., 2021)	22.8	33.3
AFS ST (Zhang et al., 2020a)	22.4	31.6
STAST (Liu et al., 2020b)	23.1	-
Dual-Decoder Transformer (BL) (Le et al., 2020)	23.6	33.5
Wav2Vec2 + Transformer (Han et al., 2021)	22.3	34.3
W-Transf (Ye et al., 2021)	23.6	34.6
RealTranS (Zeng et al., 2021a)	22.99	-
MoSST	**24.9**[†]	**35.3**[†]

Table 2: Results of non-streaming ST models on MuST-C EN-DE and EN-FR tst-COMMON test set. [†] indicates our improvement is statistically significant.

| Model | EN-DE | | EN-FR | |
	BLEU	$\triangledown$	BLEU	$\triangledown$
MoSST	**24.9**	-	**35.3**	-
w/o MSM	22.7	-2.2	34.4	-0.9
w/o MTL	21.9	-0.8	33.8	-0.6
w/o Pretrain	20.0	-1.9	31.6	-2.2

Table 3: Results of ablation study on MuST-C EN-DE and EN-FR tst-COMMON test set. "w/o MSM" stands for MoSST augmented without the monotonic segmentation module in the acoustic encoder. "w/o MTL" means removing multi-task joint learning with ASR task. "w/o Pretrain" represents using FBANK as input instead of the self-supervised acoustic representation.

4 Analysis

4.1 Ablation Studies

We conduct ablation studies to demonstrate the effectiveness of the design of MoSST, including the monotonic segmentation module, multi-task joint training with ASR, and self-supervised acoustic representation. The ablation study results can be seen in Table 3. The translation quality decreases
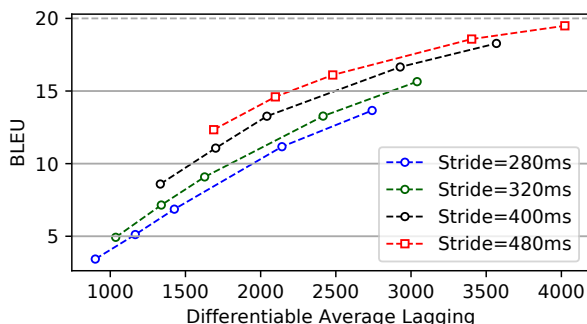

Figure 4: The translation quality against the latency metrics (DAL) on the tst-COMMON set of MuST-C En-De dataset. Decoding strategy here is pre-fixed decision. Points on the curve correspond to k in SimulEval with 5, 7, 9, 15 and 20, respectively.

significantly when each of the modules or strategies is emitted successively. The self-supervised acoustic representation can bring almost 2 BLEU on both EN-DE and EN-FR datasets, which shows that large-scale SSL brings hope to solving the data scarcity problem of end-to-end ST. For EN-DE language pair, joint training with the auxiliary ASR task has a performance gain of 0.8 BLEU. And the monotonic segmentation module has an additional 2.2 performance gain to our method. The results show a consistent performance improvement on EN-FR language pair. This verifies the outstanding advantage of the monotonic soft attention mechanism of MSM in extracting contextual acoustic representations.

4.2 Effects of Decoding Strategy

4.2.1 Pre-fix Decision

For the pre-fixed decision decoding strategy, the parameter setting of stride is very important. In Fig-
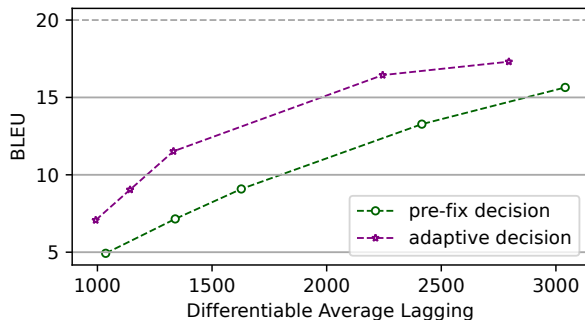
Figure 5: The translation quality against the latency metrics (DAL) on the tst-COMMON set of MuST-C En-De dataset. Pre-fixed decision is tested with the stride size of 320ms.

ure 4, we compare the influence of different strides on the pre-fixed decision strategy. It can be seen that increasing stride within a certain range will have a positive impact on the latency-bleu trade-off. But the model also tends to fall into the field of a larger latency.

4.2.2 Adaptive Decision

We have proposed an adaptive decision in Section 2.4. To better emphasize the latency factor, we compare the performance of the adaptive decision and the pre-fixed decision on the tst-COMMON test subset of MuST-C EN-DE. The results are shown in Figure 5. Compared with the pre-fixed strategy decoding method, the adaptive strategy decoding method has a better balance between delay and quality. Through observation, it is found that the adaptive strategy can ignore the silent frames. For example, after predicting a punctuation, it will read continuously to accumulate enough source acoustic information. In addition, the adaptive strategy can further reduce the delay by setting the number of WRITE operations after the accumulated information is sufficient according to the length ratio of the source sentences and the target sentences between different language pairs, which requires further exploration.

4.2.3 Alignment Visualization

In Figure 6, we show the ground truth alignment and the predicted firing positions learned by MoSST. We can see that what MSM learned is the acoustic boundary, not to mimic wait-k. Therefore, the length of the audio chunk can be adaptively read in during streaming decoding, while ensuring that each chunk includes a complete acoustic unit.

5 Related Work

Speech Translation Bérard et al. (2016) have given the first proof of the potential for end-to-end speech-to-text translation without using the intermediate transcription. The training method based on pre-training (Weiss et al., 2017; Berard et al., 2018; Bansal et al., 2019a,b; Alinejad and Sarkar, 2020; Stoian et al., 2020; Dong et al., 2021a) can effectively use pre-trained models with better performance as initialization to speed up the convergence of the ST model. Multi-task learning (Weiss et al., 2017; Berard et al., 2018; Liu et al., 2020a; Indurthi et al., 2020; Han et al., 2021; Ye et al., 2021) can fully optimize the model parameters and improve the performance with the aid of auxiliary tasks. Knowledge distillation has been proved to be efficient to learn from pre-trained models (Liu et al., 2019, 2020b; Dong et al., 2021b). Le et al. (2021) introduce adapter for multilingual speech translation. Similarly, Kano et al. (2017); Wang et al. (2020b) introduce curriculum learning methods, including different learning courses of increasing difficulty. To overcome data scarcity, Jia et al. (2019); Pino et al. (2019) augment data with pseudo-label generation, and Bahar et al. (2019); Di Gangi et al. (2019b); McCarthy et al. (2020) introduce noise-based spectrum feature enhancement. Zhang et al. (2020a) propose adaptive feature selection to eliminate uninformative features and improve performance.

Streaming Speech Translation Traditional streaming ST is usually formed by cascading a streaming ASR module and a streaming machine translation module (Oda et al., 2014; Dalvi et al., 2018). The ASR system continuously segments and recognizes the transcription of the audio segment, and then the machine translation system continuously translates the text segment output from the upstream. Most of the previous work focuses on simultaneous text translation (Gu et al., 2017a). Gu et al. (2017a) learn an agent to decide when to read or write. Ma et al. (2019) propose a novel wait-k strategy based on the prefix-to-prefix framework to synchronize output after reading k history tokens. Many following work propose some improvement strategies based on adaptive wait-k (Zheng et al., 2019; Zhang et al., 2020b; Zhang and Zhang, 2020) and efficient decoding (Elbayad et al., 2020; Zheng et al., 2020). Some monotonic attention methods (Arivazhagan et al., 2019; Ma et al., 2020c; Schneider and Waibel, 2020) have been proposed
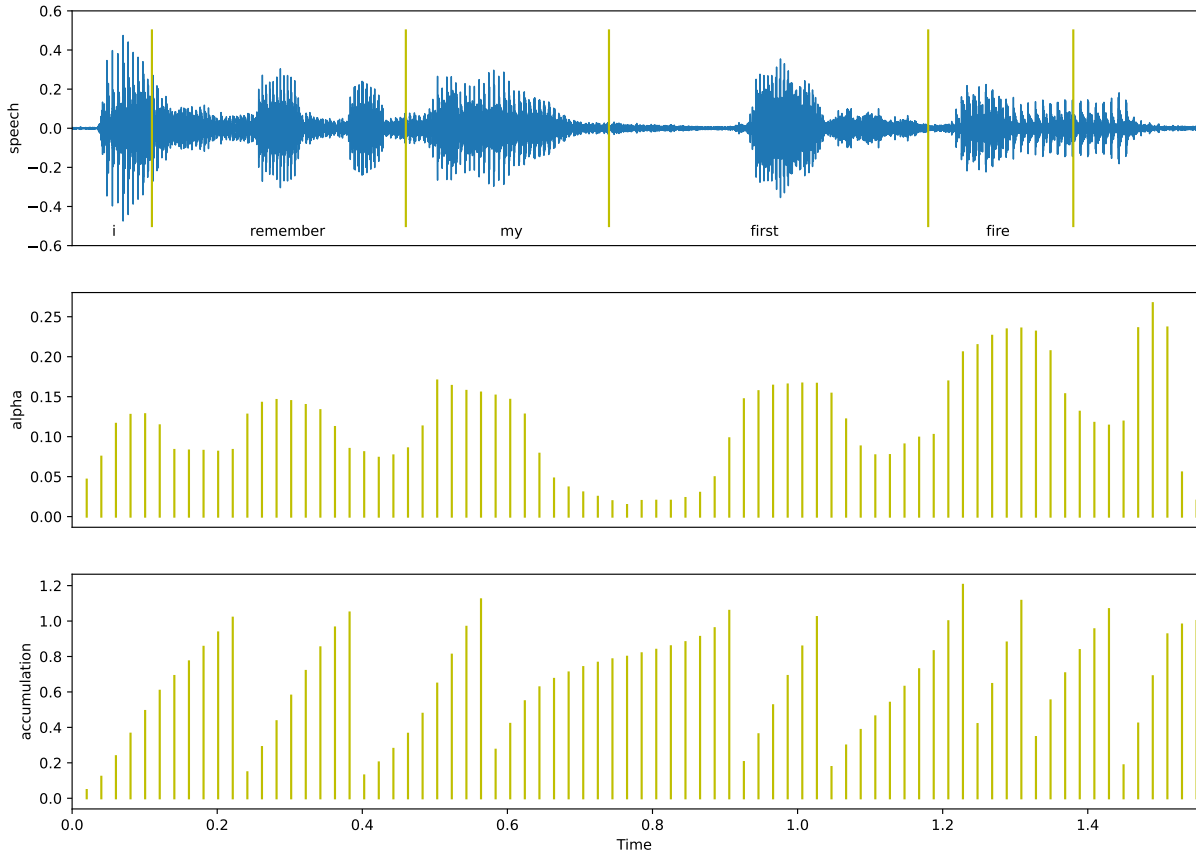
Figure 6: An example speech and its corresponding learned firing positions by MoSST. Top: the waveform of speech sequence and the true segmentation boundary corresponding to each word. Middle: the learned weights (corresponding to α in Eq.(2)) using the last dimension from acoustic encoder states. Bottom: the integrated weights by MSM in MoSST. MSM will fire and start to compute the segment's representation once the accumulated weights exceed a certain threshold (=1.0).

to model the monotonic alignment of input and output. Arivazhagan et al. (2020a,b) propose a re-translation strategy, allowing the model to modify the decoding history to improve the performance of streaming translation. Ma et al. (2020b) propose SimulST, which applies the wait-k method from streaming machine translation (Ma et al., 2019) into streaming ST. Ren et al. (2020) propose Simul-Speech, which uses knowledge distillation to guide the training of the streaming model and the connectionist temporal classification (CTC) decoding to segment the audio stream in real-time. Ma et al. (2021a) enable the streaming model to handle long input by equipping with an augmented memory encoder. Chen et al. (2021) use a separate and synchronized ASR decoder to guide the ST decoding policy. Zeng et al. (2021b) introduce a blank penalty to enhance performance in simultaneous scenarios.

6 Conclusion

We propose MoSST, a simple and effective framework for online speech-to-text translation. MoSST consists of a pretrained acoustic model, a monotonic segmentation module, and a standard Transformer, along with the multitask training strategy and the adaptive decision strategy. The monotonic segmentation module and the adaptive decision strategy tell our method when to translate. Moreover, the pre-trained acoustic encoder and the multitask training strategy boost our method's ability to predict what to generate.

The experiment on MUST-C datasets validates the effectiveness of MoSST over previous work. The results show that MoSST can achieve a better trade-off between quality and latency over prior end-to-end models and cascaded models in diverse latency settings. Besides, we also find MoSST can rival non-streaming speech translation SOTA systems given the complete audio waveform.

References

Larry F Abbott. 1999. Lapicque's introduction of the integrate-and-fire model neuron (1907). *Brain research bulletin*, 50(5-6):303–304.

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

Ebrahim Ansari, Amittai Axelrod, Nguyen Bach, Ondřej Bojar, Roldano Cattoni, Fahim Dalvi, Nadir Durrani, Marcello Federico, Christian Federmann, Jiatao Gu, Fei Huang, Kevin Knight, Xutai Ma, Ajay Nagesh, Matteo Negri, Jan Niehues, Juan Pino, Elizabeth Salesky, Xing Shi, Sebastian Stüker, Marco Turchi, Alexander Waibel, and Changhan Wang. 2020. FINDINGS OF THE IWSLT 2020 EVALUATION CAMPAIGN. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 1–34.

Naveen Arivazhagan, Colin Cherry, Wolfgang Macherey, Chung-Cheng Chiu, Semih Yavuz, Ruoming Pang, Wei Li, and Colin Raffel. 2019. Monotonic infinite lookback attention for simultaneous machine translation. In *Proc. of ACL*, pages 1313–1323.

Naveen Arivazhagan, Colin Cherry, Wolfgang Macherey, and George Foster. 2020a. Re-translation versus streaming for simultaneous translation. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 220–227.

Naveen Arivazhagan, Colin Cherry, I Te, Wolfgang Macherey, Pallavi Baljekar, and George F. Foster. 2020b. Re-translation strategies for long form, simultaneous, spoken language translation. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*, pages 7919–7923.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Parnia Bahar, Albert Zeyer, Ralf Schlüter, and Hermann Ney. 2019. On using SpecAugment for end-to-end speech translation. In *Proceedings of the 16th International Conference on Spoken Language Translation*.

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

Sameer Bansal, Herman Kamper, Karen Livescu, Adam Lopez, and Sharon Goldwater. 2019b. 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 *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*, 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. *ArXiv preprint*, abs/1612.01744.

Junkun Chen, Mingbo Ma, Renjie Zheng, and Liang Huang. 2020. Mam: Masked acoustic modeling for end-to-end speech-to-text translation. *ArXiv preprint*, abs/2010.11445.

Junkun Chen, Mingbo Ma, Renjie Zheng, and Liang Huang. 2021. Direct simultaneous speech-to-text translation assisted by synchronized streaming ASR. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4618–4624.

Colin Cherry and George Foster. 2019. Thinking slow about latency evaluation for simultaneous machine translation. *ArXiv preprint*, abs/1906.00048.

Kyunghyun Cho and Masha Esipova. 2016. Can neural machine translation do simultaneous translation? *ArXiv preprint*, abs/1606.02012.

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, and Stephan Vogel. 2018. Incremental decoding and training methods for simultaneous translation in neural machine translation. In *Proc. of NAACL-HLT*, pages 493–499.

Mattia A. Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019a. MuST-C: a Multilingual Speech Translation Corpus. In *Proc. of NAACL-HLT*, pages 2012–2017.

Mattia A. Di Gangi, Matteo Negri, Viet Nhat Nguyen, Amirhossein Tebbifakhr, and Marco Turchi. 2019b. Data augmentation for end-to-end speech translation: FBK@IWSLT '19. In *Proceedings of the 16th International Conference on Spoken Language Translation*.

Linhao Dong and Bo Xu. 2020. CIF: continuous integrate-and-fire for end-to-end speech recognition. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*, pages 6079–6083.

Linhao Dong, Shuang Xu, and Bo Xu. 2018. Speech-transformer: A no-recurrence sequence-to-sequence

model for speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*, pages 5884–5888.

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 *Proc. of AAAI*, volume 35, pages 12749–12759.

Maha Elbayad, Laurent Besacier, and Jakob Verbeek. 2020. Efficient wait-k models for simultaneous machine translation. In *Proc. of INTERSPEECH*, pages 1461–1465.

Jiatao Gu, Graham Neubig, Kyunghyun Cho, and Victor O.K. Li. 2017a. Learning to translate in real-time with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1053–1062.

Jiatao Gu, Graham Neubig, Kyunghyun Cho, and Victor O.K. Li. 2017b. Learning to translate in real-time with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1053–1062.

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.

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 Reddy Indurthi, Houjeung Han, Nikhil Kumar Lakumarapu, Beomseok Lee, Insoo Chung, Sangha Kim, and Chanwoo Kim. 2020. End-end speech-to-text translation with modality agnostic meta-learning. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*, pages 7904–7908.

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 *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019*, pages 7180–7184.

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.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proc. of EMNLP*, pages 66–71.

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.

Hang Le, Juan Pino, Changhan Wang, Jiatao Gu, Didier Schwab, and Laurent Besacier. 2021. Lightweight adapter tuning for multilingual speech translation. In *Proc. of ACL*, pages 817–824.

Yuchen Liu, Hao Xiong, Jiajun Zhang, Zhongjun He, Hua Wu, Haifeng Wang, and Chengqing Zong. 2019. End-to-end speech translation with knowledge distillation. In *Proc. of INTERSPEECH*, pages 1128–1132.

Yuchen Liu, Jiajun Zhang, Hao Xiong, Long Zhou, Zhongjun He, Hua Wu, Haifeng Wang, and Chengqing Zong. 2020a. Synchronous speech recognition and speech-to-text translation with interactive decoding. 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 8417–8424.

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

Mingbo Ma, Liang Huang, Hao Xiong, Kaibo Liu, Chuanqiang Zhang, Zhongjun He, Hairong Liu, Xing Li, and Haifeng Wang. 2019. Stacl: Simultaneous translation with integrated anticipation and controllable latency. pages 3025–3036.

Xutai Ma, Mohammad Javad Dousti, Changhan Wang, Jiatao Gu, and Juan Pino. 2020a. SIMULEVAL: An evaluation toolkit for simultaneous translation. In *Proc. of EMNLP*, pages 144–150.

Xutai Ma, Juan Pino, and Philipp Koehn. 2020b. SimulMT to SimulST: Adapting simultaneous text translation to end-to-end simultaneous speech translation. 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*, pages 582–587.

Xutai Ma, Juan Miguel Pino, James Cross, Liezl Puzon, and Jiatao Gu. 2020c. Monotonic multihead attention. In *Proc. of ICLR*.

Xutai Ma, Yongqiang Wang, Mohammad Javad Dousti, Philipp Koehn, and Juan Pino. 2021a. Streaming simultaneous speech translation with augmented memory transformer. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7523–7527. IEEE.

Xutai Ma, Yongqiang Wang, Mohammad Javad Dousti, Philipp Koehn, and Juan Pino. 2021b. Streaming simultaneous speech translation with augmented memory transformer. pages 7523–7527.

Arya D. McCarthy, Liezl Puzon, and Juan Pino. 2020. Skinaugment: Auto-encoding speaker conversions for automatic speech translation. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*, pages 7924–7928.

Yusuke Oda, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. 2014. Optimizing segmentation strategies for simultaneous speech translation. In *Proc. of ACL*, pages 551–556.

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 (Demonstrations)*, pages 48–53.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proc. of ACL*, pages 311–318.

Hilman F Pardede, Vicky Zilvan, Dikdik Krisnandi, Ana Heryana, and R Budiarianto S Kusumo. 2019. Generalized filter-bank features for robust speech recognition against reverberation. In *2019 International Conference on Computer, Control, Informatics and its Applications (IC3INA)*, pages 19–24. IEEE.

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 *Proceedings of the 16th International Conference on Spoken Language Translation*.

Yi Ren, Jinglin Liu, Xu Tan, Chen Zhang, Tao Qin, Zhou Zhao, and Tie-Yan Liu. 2020. SimulSpeech: End-to-end simultaneous speech to text translation. In *Proc. of ACL*, pages 3787–3796.

Nay San, Martijn Bartelds, Mitchell Browne, Lily Clifford, Fiona Gibson, John Mansfield, David Nash, Jane Simpson, Myfany Turpin, Maria Vollmer, et al. 2021. Leveraging neural representations for facilitating access to untranscribed speech from endangered languages. *ArXiv preprint*, abs/2103.14583.

Felix Schneider and Alexander Waibel. 2020. Towards stream translation: Adaptive computation time for simultaneous machine translation. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 228–236.

Mihaela C. Stoian, Sameer Bansal, and Sharon Goldwater. 2020. Analyzing ASR pretraining for low-resource speech-to-text translation. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*, pages 7909–7913.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

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 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*, pages 33–39.

Chengyi Wang, Yu Wu, Shujie Liu, Ming Zhou, and Zhenglu Yang. 2020b. Curriculum pre-training for end-to-end speech translation. In *Proc. of ACL*, pages 3728–3738.

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.

Rong Ye, Mingxuan Wang, and Lei Li. 2021. End-to-end speech translation via cross-modal progressive training. In *Proc. of INTERSPEECH*, pages 2267–2271.

Cheng Yi, Shiyu Zhou, and Bo Xu. 2021. Efficiently fusing pretrained acoustic and linguistic encoders for low-resource speech recognition. *IEEE Signal Processing Letters*, 28:788–792.

Xingshan Zeng, Liangyou Li, and Qun Liu. 2021a. RealTranS: End-to-end simultaneous speech translation with convolutional weighted-shrinking transformer. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2461–2474.

Xingshan Zeng, Liangyou Li, and Qun Liu. 2021b. RealTranS: End-to-end simultaneous speech translation with convolutional weighted-shrinking transformer. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2461–2474.

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

Biao Zhang, Ivan Titov, and Rico Sennrich. 2021. On sparsifying encoder outputs in sequence-to-sequence models. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2888–2900.

Ruiqing Zhang and Chuanqiang Zhang. 2020. Dynamic sentence boundary detection for simultaneous translation. In *Proceedings of the First Workshop on Automatic Simultaneous Translation*, pages 1–9.

Ruiqing Zhang, Chuanqiang Zhang, Zhongjun He, Hua Wu, and Haifeng Wang. 2020b. Learning adaptive segmentation policy for simultaneous translation. In *Proc. of EMNLP*, pages 2280–2289.

Chengqi Zhao, Mingxuan Wang, Qianqian Dong, Rong Ye, and Lei Li. 2021. NeurST: Neural speech translation toolkit. In *Proc. of ACL*, pages 55–62.

Baigong Zheng, Renjie Zheng, Mingbo Ma, and Liang Huang. 2019. Simpler and faster learning of adaptive policies for simultaneous translation. In *Proc. of EMNLP*, pages 1349–1354.

Renjie Zheng, Mingbo Ma, Baigong Zheng, Kaibo Liu, and Liang Huang. 2020. Opportunistic decoding with timely correction for simultaneous translation. In *Proc. of ACL*, pages 437–442.

A Appendix

A.1 Case Study

A.1.1 Streaming Translation

In Table 9, we show an example of simultaneous decoding for cascaded systems and end-to-end systems. The cascade system has the drawbacks of error accumulation and delay accumulation. While the end-to-end model has inherent advantages in this respect. For example, in this example, our method can attend to the speaker's prosody information from the original audio input, such as pauses, so it can accurately predict the punctuation in the target language text.

A.2 Effects of MSM

A.2.1 Performance on ASR

We also validate the effects of MSM with FBANK as input on the ASR task on MuST-C EN-DE tst-COMMON set. The results are shown in Table 4. There is also a performance improvement of 1.5 points, indicating that the integrate-and-fire model indeed plays an important role in learning encoded shrunk acoustic representation.

Model	ASR (WER ↓)
SpeechTransformer	15.95
SpeechTransformer w/ MSM	**14.48**

Table 4: Results of ASR models on MuST-C EN-DE tst-COMMON set.

A.2.2 Training Time

Our method can be trained in parallel without the help of wait-k strategy, which observably improves training efficiency. And the integration mechanism of MSM module can effectively reduce the output length of the encoder, which can reduce memory usage and increase training batch size. We experiment with FBANK feature as input on MuST-C EN-DE data set. The training time (4 Tesla-V100) for different structures is shown in Table 5.

Model	Training Time ↓
SpeechTransformer w/ wait-k	18 hours
SpeechTransformer w/ MSM	**13 hours**

Table 5: Training time of ST models on MuST-C EN-DE data set.

A.3 Numeric Results for Figures

SimulST					
BLEU	0.25	3.60	10.88	14.17	15.99
DAL	930	1500	2946	4361	5271
AP	0.37	0.62	0.88	0.96	0.99
AL	604	1097	2165	2774	3049
MoSST					
BLEU	1.35	6.75	16.34	19.77	19.97
DAL	642	1182	2263	3827	4278
AP	0.29	0.53	0.79	0.93	0.96
AL	208	818	1734	2551	2742

Table 6: Numeric results for Figure 3.

Stride=280ms					
BLEU	3.43	5.12	6.87	11.17	13.66
DAL	900	1166	1426	2142	2741
Stride=320ms					
BLEU	4.93	7.15	9.08	13.27	15.65
DAL	1036	1339	1628	2415	3041
Stride=400ms					
BLEU	8.59	11.07	13.27	16.65	18.28
DAL	1333	1701	2042	2928	3568
Stride=480ms					
BLEU	12.34	14.60	16.11	18.58	19.49
DAL	1688	2098	2480	3403	4023

Table 7: Numeric results for Figure 4.

pre-fix decision					
BLEU	4.93	7.15	9.08	13.27	15.65
DAL	1036	1339	1628	2415	3041
adaptive decision					
BLEU	7.07	9.04	11.52	16.44	17.31
DAL	992	1142	1332	2244	2795

Table 8: Numeric results for Figure 5.

A.4 Compared with Cascaded System

We build a streaming cascaded system as a baseline system by cascading a streaming speech recognition model and a text translation model. Note that the transcription generated by ASR system in the cascade streaming system is also uncorrectable. The results are shown in Table 10. The error accumulation problem of the cascade system still exists in the streaming model. Compared with the results of the cascaded system, MoSST also has obvious performance advantages in terms of quality metrics.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
En (Source)	If	you	have	something	to	give	,	give	it	now	.			
De (target)	Wenn	Sie	etwas	zu		geben	haben	,	geben	Sie	es	jetzt	.	
ASR	If	you	have	something	to	give	and	give	it	now	.			
Cascades				Wenn	Sie	etwas	zu	geben	haben	und	es	jetzt	geben	.
MoSST				Wenn	Sie	etwas	geben	,	geben	Sie	es	jetzt	.	

Table 9: An example from the test set of MuST-C En-De dataset. "ASR" means a streaming system with 440ms' waiting latency. "Cascades" means a streaming pipeline contains ASR (wait-440ms) and NMT (wait-3). MoSST represents MoSST with the pre-fixed decision and wait-3 strategy.

System	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$	$k = \inf$
ASR (WER↓)	51.9	43.9	42.1	40.6	39.6	39.0	38.7	16.25
MT (BLEU↑)	17.11	19.68	22.80	24.93	26.44	27.20	27.83	31.28
Cascaded ST (BLEU↑)	9.72	11.24	12.74	13.92	14.59	15.22	15.51	17.60

Table 10: Results of Cascaded Systems on MuST-C EN-DE tst-COMMON test set. For $k = \inf$, the streaming model degrades into an offline model without beam search decoding strategy. Here the ASR model is based on SpeechTransformer (Dong et al., 2018), and the MT model is based on Transformer (Vaswani et al., 2017). Since the delays of cascaded system involving speech recognition and text translation modules are more complicated, latency metrics are not reported here. Note that cascaded ST is consisted of streaming ASR (wait-440ms, segment-40ms) and streaming MT system.