Divergence-Guided Simultaneous Speech Translation

Xinjie Chen[1]*, Kai Fan[2], Wei Luo[2], Linlin Zhang[1]*, Libo Zhao[3]*
Xinggao Liu[1]†, Zhongqiang Huang[2]†
[1]Zhejiang University
[2]Alibaba DAMO Academy
[3]South China University of Technology
{xinjiechen, zhanglinlinlin, lxg}@zju.edu.cn, {z.huang, k.fan, w.luo}@alibaba-inc.com, wilbzhao@mail.scut.edu.cn

Abstract

To achieve high-quality translation with low latency, a Simultaneous Speech Translation (SimulST) system relies on a policy module to decide whether to translate immediately or wait for additional streaming input, along with a translation model capable of effectively handling partial speech input. Prior research has tackled these components separately, either using "wait-k" policies based on fixed-length segments or detected word boundaries, or dynamic policies based on different strategies (e.g., meaningful units), while employing offline models for prefix-to-prefix translation. In this paper, we propose **Di**vergence-**G**uided **S**imultaneous **S**peech **T**ranslation (DiG-SST), a tightly integrated approach focusing on both translation quality and latency for streaming input. Specifically, we introduce a simple yet effective prefix-based strategy for training translation models with partial speech input, and develop an adaptive policy that makes read-/write decisions for the translation model based on the expected divergence in translation distributions resulting from future input. Our experiments on multiple translation directions of the MuST-C benchmark demonstrate that our approach achieves a better trade-off between translation quality and latency compared to existing methods.

Introduction

Simultaneous Speech Translation (SimulST) aims to achieve real-time, high-quality translation from streaming speech input while maintaining low latency. Early efforts have conventionally employed a cascaded approach involving both a streaming Automatic Speech Recognition (ASR) model and a Simultaneous Text Machine Translation (SimulMT) model (Oda et al. 2014; Dalvi et al. 2018). While this approach has its merits, it nevertheless suffers from issues such as error propagation and latency accumulation (Le, Lecouteux, and Besacier 2017; Xue et al. 2020).

In response to these challenges, recent progress in speech translation (ST) has been primarily focused on end-to-end approaches, leading to significant improvements in both offline and simultaneous ST tasks (Berard et al. 2016; Weiss et al. 2017; Berard et al. 2018; Bansal et al. 2019; Ren et al. 2020; Liu et al. 2021). Following the advancements in SimulMT, researchers have investigated both fixed and adaptive read/write policies for SimulST. The absence of explicit linguistic boundaries in continuous speech signals presents a unique challenge. To adapt the wait-k policy (Ma et al. 2019) for speech input, various approaches have been proposed, including segmenting audio streams into fixed-length chunks (Ma, Pino, and Koehn 2020; Ma et al. 2021), or at the subword/word level (Dong et al. 2022; Zhang and Feng 2023). Adaptive policies have also been studied to leverage contextual information when making read/write decisions. Zhang et al. (2022) propose the detection of meaningful units in speech that can be independently translated without considering future inputs, while Papi, Negri, and Turchi (2023) experiment with the use of attention scores to develop adaptive policies for the inference process.

In spite of their demonstrated improvements, there remains significant discrepancies with respect to the desirable attributes for the translation model and the policy module in SimulST. First, most prior research has employed offline models trained on complete audio utterances to translate partial speech input, which raises questions about their effectiveness given the apparent gap between training and inference. Second, fixed policies, even when grounded in the detection of subword/word boundaries, disregard available context and cannot make informed read or write decisions. While existing adaptive policies can take into account the partial input and translation history to make dynamic decisions, they are based on heuristics and lack any direct measure of the potential impact of such decisions on the quality of translation, which is essential for achieving a delicate balance between translation quality and latency.

Recently, Transducer-based approaches (Liu et al. 2021; Tang et al. 2023) have achieved success in addressing the aforementioned issues using synchronized audio inputs and translation outputs, without explicitly modeling read/write decisions. However, these approaches are computationally intensive and requires training a distinct model for each latency configuration. In this paper, we adhere to the conventional approach of separately modeling and improving translation and read/write decisions, and introduce an integrated approach called Divergence-Guided Simultaneous Speech Translation (DiG-SST). Specifically, we propose:

Prefix-enhanced Translation: We include prefix-to-prefix and prefix-to-full ST samples, in addition to the

*Work done when interning at Alibaba DAMO Academy

†Corresponding authors

conventional offline training data, during translation model training. This strategy reduces the gap between conventional offline training and simultaneous inference, improving the model's effectiveness in low latency settings.

Divergence-based Policy Module: We suggest using the divergence between the translation distributions of the next target word, computed based on the partial input versus the complete input using the model from prefix-enhanced training, as guidance for the read/write decisions of translation model. We develop a new modeling approach to estimate divergence scores using only the partial input for use during inference, achieved by adding a few lightweight layers on top of the translation model.

Our experiments demonstrate that the proposed approach compares favorably against other methods across three dimensions of the MuST-C dataset in both offline and simultaneous translation scenarios. The code is available at https://github.com/cxjfluffy/DiG-SST.

Background and Related Works

Speech translation is categorized into offline and simultaneous scenarios based on inference modes. A standard speech translation training sample, denoted by $\mathcal{D} = (\mathbf{s}, \mathbf{x}, \mathbf{y})$, comprises a speech audio $\mathbf{s} = (s_1, \ldots, s_T)$, its transcription $\mathbf{x} = (x_1, \ldots, x_I)$, and translation sequences $\mathbf{y} = (y_1, \ldots, y_J)$. Subsequent discussions mainly focus on end-to-end techniques.

Offline Speech Translation generates all target tokens based on the complete audio input. The offline ST model first encodes the audio input, represented as $\mathbf{s}$, into a representation, $\mathbf{h}$, which is then decoded to predict $\mathbf{y}$. The decoding process of an offline ST model parameterized by θ is defined as:

$$p(\mathbf{y} \mid \mathbf{s}; \theta) = \prod_{j}^{J} p\left(y_j \mid \mathbf{s}, \mathbf{y}_{<j}; \theta\right). \qquad (1)$$

Since the introduction of the end-to-end neural network model for ST (Berard et al. 2016), much of the research in end-to-end ST has been concentrated on offline scenarios. Pre-training has been shown to improve translation quality in various studies (Weiss et al. 2017; Berard et al. 2018; Bansal et al. 2019; Alinejad and Sarkar 2020; Dong et al. 2021), and has become an integral component of the standard framework in this field. Other prevalent research areas include data augmentation (Pino et al. 2019; Anastasopoulos et al. 2022), knowledge distillation (Gaido et al. 2020), multi-task learning (Liu et al. 2020; Indurthi et al. 2020; Han et al. 2021; Ye, Wang, and Li 2021), curriculum learning (Kano, Sakti, and Nakamura 2018; Wang et al. 2020), and mix-up contrastive learning (Fang et al. 2022).

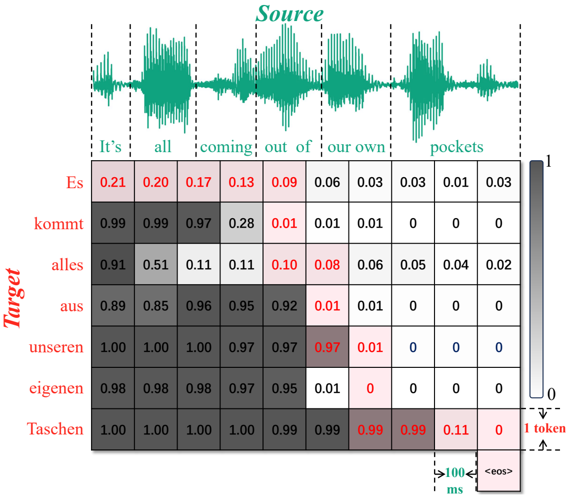
Simultaneous Speech Translation generates target tokens from partial input. Different from Eq. (1), the decoding process of SimulST is formulated as:

$$p(\mathbf{y} \mid \mathbf{s}; \theta) = \prod_{j}^{J} p(y_j \mid \mathbf{s}_{\leq g(j)}, \mathbf{y}_{<j}; \theta), \qquad (2)$$

where $g(j)$ is a monotonically non-decreasing function that indicates the ending timestamp of the audio required to generate the j-th target token.

Similar to the trend in SimulMT and offline ST, recent research in SimulST has shifted towards end-to-end models, especially regarding the read/write policy that decides whether to produce new target tokens or await more audio input. Policies utilizing a fixed-size speech chunk for read-/write actions are presented in (Ma, Pino, and Koehn 2020; Nguyen, Estève, and Besacier 2021; Ma et al. 2021; Liu et al. 2021). These policies may face challenges in maintaining the semantic boundary in audio and leveraging the context information. Recognizing these limitations, adaptive read/write policies have received attention in the SimulST research community following their success in SimulMT (Arivazhagan et al. 2019; Ma et al. 2020; Zhang et al. 2020). Dong et al. (2022) aim to align the audio input to the text through continuous integrate-and-fire for more precise audio boundary. Zhang et al. (2022) segment the source streaming speech into meaningful units by considering both acoustic features and translation history. Transducer-based structures (Liu et al. 2021; Tang et al. 2023) use all path training and multiple models for superior performance but suffer from increased complexity in training and deployment. Attention-based methods (Papi, Negri, and Turchi 2023; Zhang and Feng 2023) leverage the information from attention maps to guide the read/write policy.

Main Method



Es	0.21	0.20	0.17	0.13	0.09	0.06	0.03	0.03	0.01	0.03
kommt	0.99	0.99	0.97	0.28	0.01	0.01	0.01	0	0	0
alles	0.91	0.51	0.11	0.11	0.10	0.08	0.06	0.05	0.04	0.02
aus	0.89	0.85	0.96	0.95	0.92	0.01	0.01	0	0	0
unseren	1.00	1.00	1.00	0.97	0.97	0.97	0.01	0	0	0
eigenen	0.98	0.98	0.98	0.97	0.95	0.01	0	0	0	0
Taschen	1.00	1.00	1.00	1.00	0.99	0.99	0.99	0.99	0.11	0

Figure 1: An example of the divergence-guided read/write process for SimulST. The green part denotes the source audio-transcript and the red text indicates the target translation. The matrix visualizes the divergence scores varying with the lengths of the source and target. The cell's column corresponds to one target token, and its row represents a duration of 100ms in the input audio. Each cell, with values from 0 to 1, shows the divergence score between outputs from partial and complete audio inputs. The path colored in red within the matrix is the path chosen by our divergence-guided read/write policy (with a fixed threshold λ=0.1).

During the process of simultaneous interpretation, human interpreters actively comprehend the perceived speech inputs while also anticipating incoming speech. This allows them to produce accurate translations with minimal delay. As an essential aspect of this process, interpreters need to constantly determine whether they possess sufficient information for their modeling of translations that would not be affected by future input, and then act accordingly to either produce the translation or wait for more input. Drawing inspiration from this process, we present a divergence-based measure that simulates human interpreters' ability to assess how much future input might influence the translation of the next word.

Figure 1 illustrates how this measure guides the read/write process for English-German simultaneous speech translation. The cell at the top-left of the matrix represents the divergence between the translation distribution of the first target word, based on the initial 100ms of audio signal roughly containing the English spoken words "it's", and the translation distribution obtained after processing the entire utterance. A relatively low divergence score of 0.21 indicates that these two distributions are similar. Further waiting for more speech input would progressively reduce the divergence, as evident from the decreasing scores in the first row from left to right. However, this comes at the expense of increased latency. If a decision is made to translate after the initial 100ms, resulting in the first target word "Es (is)", the divergence score for the subsequent target word rises significantly to 0.99. This signifies the lack of sufficient information in the first 100ms of audio to accurately generate the next target word. In fact, according to the second row in the matrix, waiting an additional 400ms of audio is necessary to reduce the divergence to a sufficiently low score of 0.01. Only then can one be confident in accurately producing the next word "kommt (come)" in the translation.

For the divergence-based policy module to be effective, the divergence score needs to accurately reflect the translation model's predictive ability for both partial and complete utterances. Simultaneously, the translation model should be capable of effectively translating both partial and complete utterances. The general framework for Divergence-Guided Simultaneous Speech Translation is depicted in Figure 2. We will next elaborate on the prefix-enhanced training strategy and the divergence-based policy module.

Prefix-enhanced Translation

As illustrated in Figure 2 (a), our translation model comprises three components: an audio encoder (wav2vec 2.0), a semantic encoder, and a translation decoder. It is trained with both ST and MT tasks. The semantic encoder consists of a stack of Transformer layers for encoding acoustic tokens in the ST task, an embedding layer for the MT task, and a translation encoder shared between both tasks. To enhance the model's capability of translating partial speech inputs during streaming inference, we introduce prefix-based training samples alongside standard ST training data. The training loss of the prefix-enhanced translation model includes two components: the offline translation loss and the streaming translation loss.

Offline Loss For the ST training sample $\mathcal{D} = (\mathbf{s}, \mathbf{x}, \mathbf{y})$, where $\mathbf{s}$ denotes the source audio, $\mathbf{x}$ the source transcript, and $\mathbf{y}$ the target translation, the objective of both offline ST and MT training tasks is to minimize their respective negative log-likelihoods over the training set as follows:

$$\mathcal{L}_{\text{st}}^{\text{off}} = -\sum_{(\mathbf{s},\mathbf{y})} \sum_{j}^{J} \log p\left(y_j \mid \mathbf{y}_{1:j-1}, \mathbf{s}_{1:T}\right) \tag{3}$$

$$\mathcal{L}_{\text{mt}}^{\text{off}} = -\sum_{(\mathbf{x},\mathbf{y})} \sum_{j}^{J} \log p\left(y_j \mid \mathbf{y}_{1:j-1}, \mathbf{x}_{1:I}\right) \tag{4}$$

where I, J and T represent the corresponding sequence lengths.

Streaming Loss In simultaneous scenarios, the translation model must be able to effectively translate partial audio input, a situation not addressed by the offline training loss. To bridge this gap, we create prefix-to-prefix ST training pairs[1] $\mathcal{D}' = (\mathbf{s}', \mathbf{y}')$. In each pair, $\mathbf{s}' = \mathbf{s}_{1:T'}$ is a random prefix of the original source $\mathbf{s}$ with length $T' \sim \mathcal{U}(1, T)$ (where $\mathcal{U}$ denotes a uniform distribution), and $\mathbf{y}' = \mathbf{y}_{1:J'}$ is a prefix of length J' from the original translation $\mathbf{y}$. The streaming loss is formulated as:

$$\mathcal{L}_{\text{st}}^{\text{prefix}} = -\sum_{(\mathbf{s},\mathbf{y})} \sum_{j}^{J'} \log p\left(y_j \mid \mathbf{y}_{1:j-1}, \mathbf{s}_{1:T'}\right) \tag{5}$$

Determining the optimal length J' for y' is a nontrivial task, as s' might not end at word boundaries, and there is no consensus on the best approach for selecting y' in prefix-to-prefix training for simultaneous machine translation with text input. In this study, we experiment with two simple implementations.

In the first approach, we assume that the target text y is approximately aligned monotonically with the source audio, and we sample J' as follows:

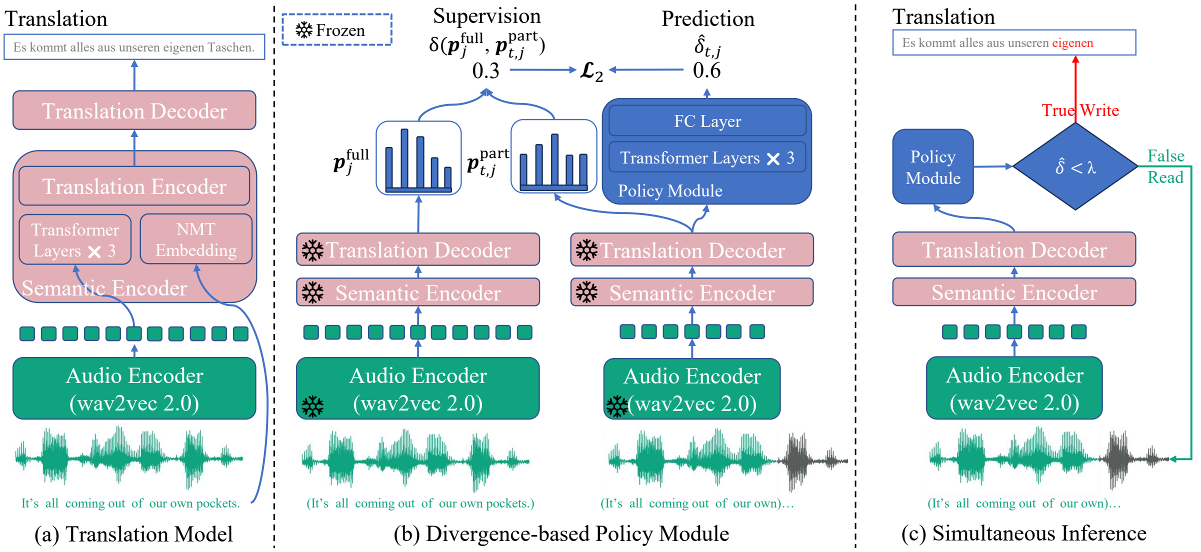
$$J' \sim \mathcal{U}(\max([\frac{J}{T} * T' - k_1], 0), \min([\frac{J}{T} * T' + k_2], J))$$

where k_1 and k_2 are chosen to account for different latency settings[2] similar to the multipath wait-k strategy in SimulMT (Elbayad, Besacier, and Verbeek 2020). We refer to this method as prefix-to-prefix training.

In the second approach, we simply set $J' = J$, using the full translation y as the target for the source prefix s'. This prefix-to-full training strategy addresses all reordering issues, but it could potentially result in translation hallucinations. It's important to note that in simultaneous speech translation, the translation process is guided by the read-/write policy module, which only engages in translation generation when the policy module determines that there's sufficient information in the source audio to generate the next word. As demonstrated in the experiments section, the

[1] We generate two samples of prefix-to-prefix ST training pairs from each training sample $(\mathbf{s}, \mathbf{y})$ in our experiments.

[2] We set $k_1 = 10, k_2 = 2$ in our experiments.


Translation
Es kommt alles aus unseren eigenen Taschen.
Translation Decoder
Translation Encoder
Transformer Layers × 3
NMT Embedding
Semantic Encoder
Audio Encoder (wav2vec 2.0)
It's all coming out of our own pockets.
Frozen
Supervision
$\delta(\boldsymbol{p}_j^{\text{full}}, \boldsymbol{p}_{t,j}^{\text{part}})$
0.3
$\mathcal{L}_2$
Prediction
$\hat{\delta}_{t,j}$
0.6
FC Layer
Transformer Layers × 3
Policy Module
$\boldsymbol{p}_j^{\text{full}}$
$\boldsymbol{p}_{t,j}^{\text{part}}$
Translation Decoder
Semantic Encoder
Audio Encoder (wav2vec 2.0)
Translation Decoder
Semantic Encoder
Audio Encoder (wav2vec 2.0)
(It's all coming out of our own pockets.)
(It's all coming out of our own)...
Translation
Es kommt alles aus unseren eigenen
True
Write
Policy Module
$\delta < \lambda$
False
Read
Translation Decoder
Semantic Encoder
Audio Encoder (wav2vec 2.0)
(It's all coming out of our own)...

(a) Translation Model

(b) Divergence-based Policy Module

(c) Simultaneous Inference

Figure 2: The overall framework of our proposed DiG-SST: (a) the architecture of the translation model, (b) the architecture and training loss of the divergence-based policy module, based on the frozen translation model from (a), and (c) the simultaneous inference process.

prefix-to-full approach achieves superior translation quality compared to the prefix-to-prefix approach and is used in our main experiments.

Finally, the overall training loss of the translation model combines the offline ST loss, offline MT loss, and the streaming ST loss:

$$\mathcal{L}_1 = \mathcal{L}_{\text{st}}^{\text{off}} + \mathcal{L}_{\text{mt}}^{\text{off}} + \mathcal{L}_{\text{st}}^{\text{prefix}} \tag{6}$$

Divergence-based Policy Module

As previously explained, we utilize the divergence between two translation distributions of the next target word: one computed based on partial audio input and translation history, and the other relying on the complete utterance and history. This is to measure the impact on translation quality using only partial input. However, this approach requires access to the full utterance, which is unavailable during streaming inference. We could design an algorithm to emulate the ability of human interpreters to predict potential completions of the partial input and use them to estimate the translation distribution. In this paper, we opt for a simpler approach, treating it as a supervised learning problem.

We compute the oracle divergence scores from the ST training data and use them to train a divergence prediction module, which is based solely on partial audio input and translation history. Utilizing the probability formulation for SimulST outlined in Eq. (2), we compute two probability distributions for the target word at the j^{th} decoding step: $\mathrm{P}_{t,j}^{\text{part}}$, based on partial input, and $\mathrm{P}_j^{\text{full}}$, computed from complete input:

$$\mathbf{p}_{t,j}^{\text{part}} = \mathrm{P}\left(y_j = \cdot \mid \mathbf{s}_{\leq t}, \mathbf{y}_{<j}\right) \tag{7}$$

$$\mathbf{p}_j^{\text{full}} = \mathrm{P}\left(y_j = \cdot \mid \mathbf{s}, \mathbf{y}_{<j}\right) \tag{8}$$

Here, t is randomly sampled from $\mathcal{U}(1, T)$ to obtain a diverse set of oracle divergence scores to train a robust divergence prediction model.

Given these distributions, different divergence measures $\delta(\mathbf{p}_j^{\text{full}}, \mathbf{p}_{t,j}^{\text{part}})$ can be used. In this study, we adopt cosine similarity:

$$\delta(\mathbf{p}_j^{\text{full}}, \mathbf{p}_{t,j}^{\text{part}}) = 1 - \frac{\mathbf{p}_j^{\text{full}} \cdot \mathbf{p}_{t,j}^{\text{part}}}{\|\mathbf{p}_j^{\text{full}}\| \|\mathbf{p}_{t,j}^{\text{part}}\|} \tag{9}$$

The goal of the divergence prediction module is to predict the oracle divergence scores using only partial audio input and translation history. Since it has access to the same information as the translation model and is tasked to assess translation quality, we designed it to directly access the hidden states of the translation decoder, as shown in Figure 2 (b). Specifically, it predicts the divergence score at each step as:

$$\hat{\delta}_{t,j} = f_c(\text{MHA}(q, k, v = h_{t,j}^{dec})) \tag{10}$$

where $h_{t,j}^{dec}$ is the translation decoder's final hidden state, MHA is a stack of multi-head attention layers, and f_c is a fully-connected layer. The parameters of the translation model are frozen during the training of the divergence prediction module.

The prediction $\hat{\delta}_{t,j}$ is supervised by the following loss[3]:

$$\mathcal{L}_2 = \sum_{(\mathbf{s},\mathbf{y})} \sum_{t \sim \mathcal{U}(1,T)} \sum_{j=1}^{J} (\delta(\mathbf{p}_{t,j}^{\text{full}}, \mathbf{p}_j^{\text{part}}) - \hat{\delta}_{t,j})^2 \tag{11}$$

where T and J are the length of the source audio $\mathbf{s}$ and target translation $\mathbf{y}$ respectively.

[3] We sample $t \sim \mathcal{U}(1, T)$ twice for each decoding step j.

Inference Policy

The inference process of our DiG-SST approach is depicted in Figure 2 (c). The divergence-based policy module dynamically makes read/write decisions by comparing the predicted divergence score $\hat{\delta}_{t,j}$ against a provided threshold λ:

$$\textit{write} \text{ if } \hat{\delta}_{t,j} < \lambda, \text{else } \textit{read} \tag{12}$$

While our approach allows for read/write decisions to be made at any time, given that both the translation model and the policy module are trained on random audio prefixes, we limit these computations to occur every 100ms to reduce unnecessary processing. Ideally, we could dynamically adjust the threshold value at runtime to balance the tradeoff between translation quality and latency. Nonetheless, as noted in our experiments, achieving this would require a very high level of prediction accuracy from the policy module. Given that the divergence-based policy model is trained with the reference translation as history, there is inherent exposure bias during inference, which makes accurate prediction more challenging.

Instead, we present a hybrid read/write policy that combines our divergence-based adaptive policy with a wait-k policy. In this approach, we utilize the wait-k mechanism to define a maximum allowable latency during inference, and permit early translation if the predicted divergence score falls below the threshold, indicating that the current input contains sufficient information for translation. In our experiments, we use a single model with a fixed λ and different k values to generate the AL-BLEU curve.

Experiments

Experimental Setting

Dataset We conduct experiments on the widely used **MuST-C** V1 corpus: English$\rightarrow\{$German, Spanish, French$\}$ (En$\rightarrow\{$De, Es, Fr$\}$) (Gangi et al. 2019), detailed in Table 1. We exclude training audio samples that are longer than 450k frames and use the Montreal Forced Aligner[4] to remove samples that do not contain speech content. Additionally, we use the output of an offline MT model trained on MuST-C as extra training data for ST task.

Split	En-De	En-Es	En-Er
Train	234K	270K	280K
Dev	1423	1316	1412
Tst-COMMON	2641	2502	2632

Table 1: The statistics (sentences) of three language pairs in MuST-C.

Model Configuration Both the translation encoder and decoder employ 6 transformer layers, each with dimensions of 512 and 8 attention heads, and are pre-trained using MuST-C text data. The audio encoder, based on Wav2vec2.0 (Baevski et al. 2020), is trained on Librispeech-960 (Panayotov et al. 2015), aligning with Zhang and Feng (2023); Dong et al. (2022); Zhang et al. (2022). It encodes each 20ms raw audio segment into an acoustic token. The policy module is constructed on top of the decoder's hidden states from the translation model, with an additional 3 transformer layers[5] and 1 fully-connected layer. The text vocabulary comprises 10,000 SentencePiece (Kudo 2018) subwords, shared between source and target languages.

Training and Evaluation Training was conducted on 4 V100 GPUs, each with a batch size of 3.2M audio frames. The translation model was trained for up to 40 epochs with early stopping after 20 non-improving epochs, followed by a 10-epoch policy module training with the translation model frozen. Model selection was based on the development set performance, using detokenized case-sensitive BLEU scores from sacreBLEU[6] and average lagging (AL) (Ma et al. 2019) latency measures calculated with the simuleval toolkit[7].

Main Results

We compare the proposed DiG-STT approach with a variety of models in the literature, all of which are exclusively trained on the MuST-C corpus, including:

- **MU-ST** (Zhang et al. 2022), employing a segmentation model to determine meaningful translation units.
- **MoSST** (Dong et al. 2022), utilizing the integrate-and-firing method for word segmentation in speech.
- **RealTrans** (Zeng, Li, and Liu 2021), employing a convolutional weighted-shrinking Transformer to detect word count in streaming speech.
- **ITST** (Zhang and Feng 2022), quantifying transported information from source to target and translating based on accumulated received information.
- **DiSeg** (Zhang and Feng 2023), utilizing the proposed expectation training to render hard segmentation differentiable, enabling joint training with the translation model.
- **MMA-SLM** (Indurthi et al. 2022), enhancing monotonic attention by an language model to improve its decisions.

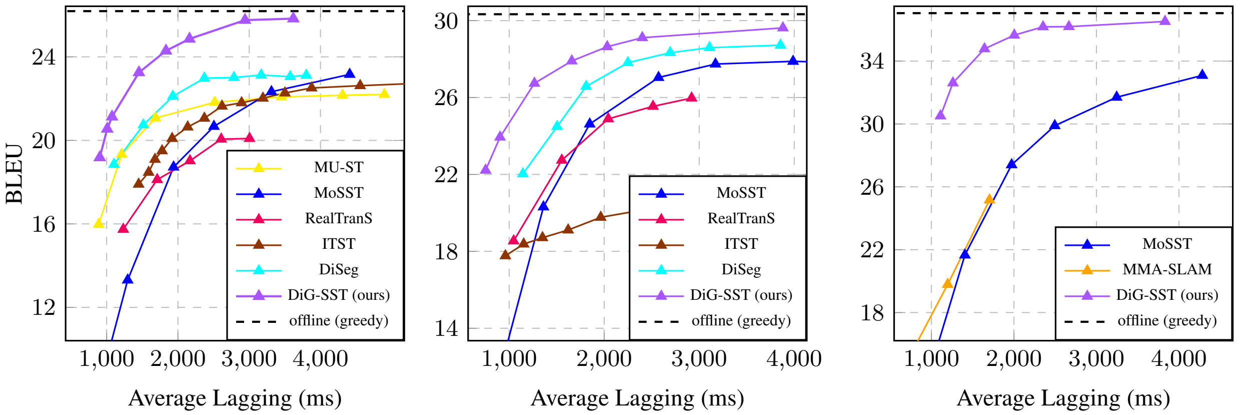
The results are shown in Figure 3, in which "offline" denotes the offline performance of our translation model with a beam size of 1. In the commonly studied En-De direction, our method outperforms all other approaches in translation quality across various latency conditions. In the low latency setting of around 1000ms, DiG-SST surpasses its closest competitors, DiSeg and MU-ST, by 2 BLEU points, and outperforms other methods by over 3 BLEU points. It maintains a margin of 3+ BLEU points over all other methods as latency increases past 2000ms. DiG-SST also performs the best in the En-Es direction, with 4+ BLEU points advantage over other models at around 1000ms latency. In the En-Fr direction, we compare solely with MoSST and MMA-SLAM due to limited studies, and also observe significant improvement.

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

[5] Quality slightly decreases by 0.2 BLEU points with 1 or 2 layers, and no additional improvement is noted beyond 3 layers.

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

[7] https://github.com/facebookresearch/SimulEval



(a) En-De (b) En-Es (c) En-Fr

Figure 3: The translation quality (BLEU) against the latency metrics (AL) on the tst-COMMON sets of MuST-C En-De, En-Es, and En-Fr datasets.

Model	En-De	En-Es	En-Fr	Avg
RealTranS	23.0	-	-	-
MoSST	24.9	-	35.3	-
SpeechT5	25.2	-	35.3	-
DiSeg	24.7	29.7	-	-
XSNET	25.5	29.6	36.0	30.4
STEMM	25.6	30.3	36.1	30.7
ConST	25.7	30.4	36.8	31.0
DiG-SST (Ours)	**26.9**	**30.9**	**37.6**	**31.8**

Table 2: Offline ST performance on MuST-C tst-COMMON using a beam size of 5. All methods use only MuST-C data.

We compare in Table 2 the offline performance of our prefix-enhanced translation model against recent state-of-the-art methods. These methods focus on either offline ST, such as SpeechT5 (Ao et al. 2022), XSNET (Ye, Wang, and Li 2021), STEMM (Fang et al. 2022), and ConST (Ye, Wang, and Li 2022), or on SimulST, including RealTranS (Zeng, Li, and Liu 2021), MoSST (Dong et al. 2022), and DiSeg (Zhang and Feng 2023). Our approach is competitive in all translation directions, outperforming all other methods.

Experiment Analysis

Ablation Studies In the ablation studies depicted in Figure 4, we delve into the various aspects of the DiG-STT approach to assess their impact on the quality-latency curve. Notably, the inclusion of the streaming ST loss ($\mathcal{L}_{\text{st}}^{\text{prefix}}$) contributes most significantly to translation quality, yielding an improvement of over 2 BLEU points around the 1000ms mark. The addition of the offline MT loss also proves beneficial in leveraging training data from parallel text.

To assess the influence of the divergence-based read/write policy module, we conducted experiments by removing it (denoted as -RW) from the strongest DiG-STT model, as well as from the weaker version trained without the streaming ST loss ($\mathcal{L}_{\text{st}}^{\text{prefix}}$), relying solely on the wait-k policy.

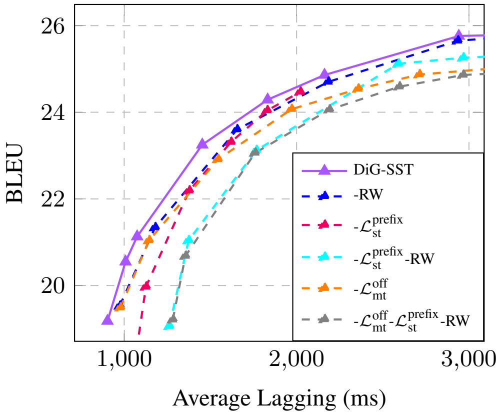


Figure 4: Ablation studies of DiG-SST on En-De tst-COMMON.

The policy module demonstrated consistent performance improvement in both configurations, with a more significant effect observed in the weaker baseline. This could be attributed to the weaker translation model's reduced quality in low latency settings, making it more important to have a robust policy model to determine whether there is sufficient information to perform translation.

Why Necessary to Combine with Wait-k? We begin by considering the upper-bound performance achievable through oracle divergence scores, calculated using complete audio input as a reference. As shown in Figure 5, employing only oracle divergence scores and varying the λ threshold for read/write decisions according to Eq. (12) yields notably superior quality across all latency settings. The inclusion of the wait-k policy has minimal influence on the quality-latency curve. However, relying solely on predicted divergence scores for read/write decisions, without including the wait-k policy, leads to markedly inferior performance. The integration of the divergence prediction module with the

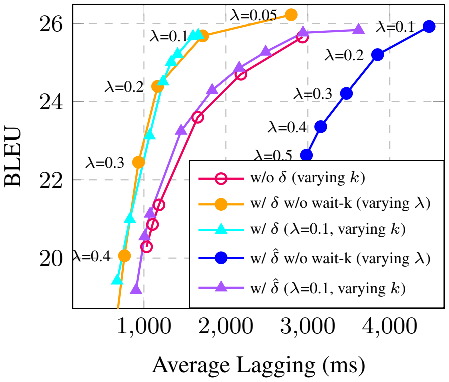


Figure 5: AL-BLEU curves of different configurations of the DiG-SST approach on En-De tst-COMMON, by varying the source of divergence scores (δ: oracle, $\hat{\delta}$: predicted) and how to control the latency (varying k vs λ).

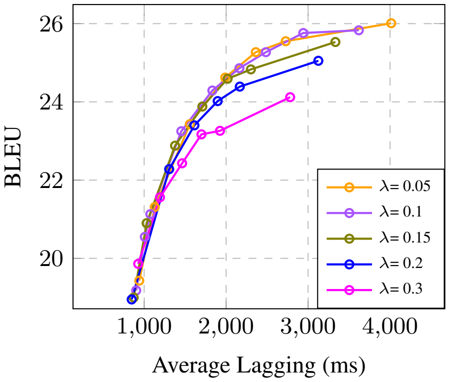


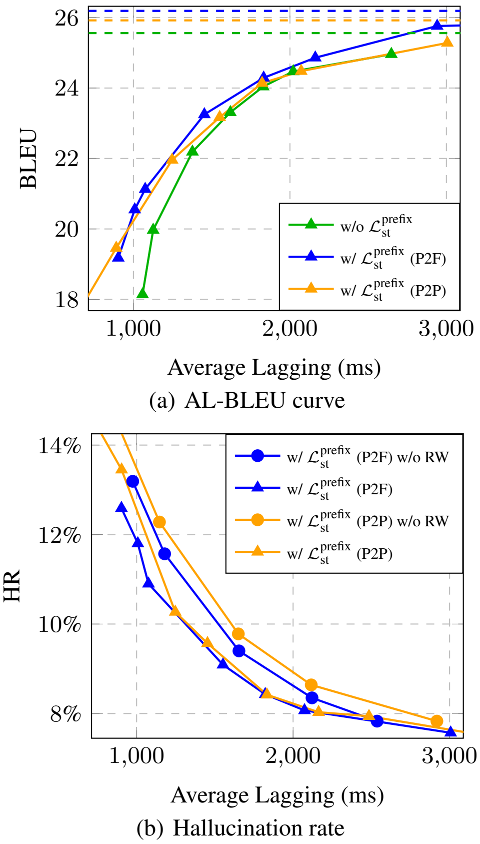
Figure 6: Impact of λ on MuST-C En-De tst-COMMON set when used with the wait-k policy.

wait-k model produces a more favorable curve compared to utilizing any of the policies independently.

Note that the divergence-based policy module, trained on reference translation history but used with a predicted one containing errors, can be significantly impacted by the threshold λ, as shown in Figure 6. Optimal performance is achieved with a small λ, ensuring reliable write actions and quality translations. However, when λ exceeds 0.15, there is a significant decline in translation quality at higher latency levels. This occurs because a higher λ leads to premature write actions, resulting in poorer translations and complicating future read/write decisions due to exposure bias and error accumulation.

Which Sampling Method is Better? Figure 7 (a) compares the impacts of prefix-to-prefix (P2P) and prefix-to-full (P2F) sampling approaches on trained translation model. Both methods improve the quality-latency curve in simultaneous translation and offline quality, with P2F showing overall better performance.

Hallucination Given that neither the prefix-to-prefix nor the prefix-to-full methods used for generating prefix-based training samples guarantee complete audio-target translation alignment, hallucination remains a concern. Figure 7(b) shows the hallucination rate (HR) (Chen et al. 2021) across different settings. The prefix-to-full method exhibits a notably lower HR compared to the prefix-to-prefix method, likely because it ensures that each source word has a corresponding target translation, making it easier to learn word translation. Incorporating the divergence-based policy model further reduces hallucination.



(a) AL-BLEU curve

(b) Hallucination rate

Figure 7: A comparison of different sampling methods on MuST-C En-De tst-COMMON set. In the absence of RW, the wait-k policy is adopted.

Conclusion

This paper presents the Divergence-Guided Simultaneous Speech Translation (DiG-SST) approach to improve quality and reduce latency in SimulST. The prefix-enhanced translation model is able to effectively translate partial audio input encountered during simultaneous inference. Integrating the divergence-based policy module with a wait-k policy enables flexible read/write decisions, ensuring translation is grounded in sufficient context in the streaming input. Experiments on the MuST-C benchmark demonstrate that our approach outperforms all existing methods in achieving high-quality translation at minimal latency.

Acknowledgements

We would like to thank all the anonymous reviewers for their insightful and helpful comments. This work was supported by Alibaba Group, National Natural Science Foundation of China (62073288, 12075212, 12105246, 11975207) and Zhejiang University NGICS Platform, and their supports are thereby acknowledged.

References

Alinejad, A.; and Sarkar, A. 2020. Effectively pretraining a speech translation decoder with Machine Translation data. In *EMNLP*, 8014–8020. Association for Computational Linguistics.

Anastasopoulos, A.; Barrault, L.; Bentivogli, L.; Zanon Boito, M.; Bojar, O.; Cattoni, R.; Currey, A.; Dinu, G.; Duh, K.; Elbayad, M.; Emmanuel, C.; Estève, Y.; Federico, M.; Federmann, C.; Gahbiche, S.; Gong, H.; Grundkiewicz, R.; Haddow, B.; Hsu, B.; Javorský, D.; Kloudová, V.; Lakew, S.; Ma, X.; Mathur, P.; McNamee, P.; Murray, K.; Nǎdejde, M.; Nakamura, S.; Negri, M.; Niehues, J.; Niu, X.; Ortega, J.; Pino, J.; Salesky, E.; Shi, J.; Sperber, M.; Stüker, S.; Sudoh, K.; Turchi, M.; Virkar, Y.; Waibel, A.; Wang, C.; and Watanabe, S. 2022. Findings of the IWSLT 2022 Evaluation Campaign. In *IWSLT*, 98–157. Dublin, Ireland (in-person and online): Association for Computational Linguistics.

Ao, J.; Wang, R.; Zhou, L.; Wang, C.; Ren, S.; Wu, Y.; Liu, S.; Ko, T.; Li, Q.; Zhang, Y.; Wei, Z.; Qian, Y.; Li, J.; and Wei, F. 2022. SpeechT5: Unified-Modal Encoder-Decoder Pre-Training for Spoken Language Processing. In *ACL*, 5723–5738. Association for Computational Linguistics.

Arivazhagan, N.; Cherry, C.; Macherey, W.; Chiu, C.-C.; Yavuz, S.; Pang, R.; Li, W.; and Raffel, C. 2019. Monotonic Infinite Lookback Attention for Simultaneous Machine Translation. In *ACL*, 1313–1323. Florence, Italy: Association for Computational Linguistics.

Baevski, A.; Zhou, Y.; Mohamed, A.; and Auli, M. 2020. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. In *NeurIPS*.

Bansal, S.; Kamper, H.; Livescu, K.; Lopez, A.; and Goldwater, S. 2019. Pre-training on high-resource speech recognition improves low-resource speech-to-text translation. In *NAACL*, 58–68. Association for Computational Linguistics.

Berard, A.; Besacier, L.; Kocabiyikoglu, A. C.; and Pietquin, O. 2018. End-to-End Automatic Speech Translation of Audiobooks. In *ICASSP*, 6224–6228. IEEE.

Berard, A.; Pietquin, O.; Servan, C.; and Besacier, L. 2016. Listen and Translate: A Proof of Concept for End-to-End Speech-to-Text Translation. *CoRR*, abs/1612.01744.

Chen, J.; Zheng, R.; Kita, A.; Ma, M.; and Huang, L. 2021. Improving Simultaneous Translation by Incorporating Pseudo-References with Fewer Reorderings. In *EMNLP*, 5857–5864. Association for Computational Linguistics.

Dalvi, F.; Durrani, N.; Sajjad, H.; and Vogel, S. 2018. Incremental Decoding and Training Methods for Simultaneous Translation in Neural Machine Translation. In *NAACL*, 493–499. Association for Computational Linguistics.

Dong, Q.; Wang, M.; Zhou, H.; Xu, S.; Xu, B.; and Li, L. 2021. Consecutive Decoding for Speech-to-text Translation. In *IAAI*, 12738–12748. AAAI Press.

Dong, Q.; Zhu, Y.; Wang, M.; and Li, L. 2022. Learning When to Translate for Streaming Speech. In *ACL*, 680–694. Dublin, Ireland: Association for Computational Linguistics.

Elbayad, M.; Besacier, L.; and Verbeek, J. 2020. Efficient Wait-k Models for Simultaneous Machine Translation. In *Interspeech*, 1461–1465. ISCA.

Fang, Q.; Ye, R.; Li, L.; Feng, Y.; and Wang, M. 2022. STEMM: Self-learning with Speech-text Manifold Mixup for Speech Translation. In *ACL*, 7050–7062. Dublin, Ireland: Association for Computational Linguistics.

Gaido, M.; Gangi, M. A. D.; Negri, M.; and Turchi, M. 2020. End-to-End Speech-Translation with Knowledge Distillation: FBK@IWSLT2020. In *IWSLT*, 80–88. Association for Computational Linguistics.

Gangi, M. A. D.; Cattoni, R.; Bentivogli, L.; Negri, M.; and Turchi, M. 2019. MuST-C: a Multilingual Speech Translation Corpus. In *NAACL*, 2012–2017. Association for Computational Linguistics.

Han, C.; Wang, M.; Ji, H.; and Li, L. 2021. Learning Shared Semantic Space for Speech-to-Text Translation. In *Findings of ACL*, 2214–2225. Association for Computational Linguistics.

Indurthi, S. R.; Han, H.; Lakumarapu, N. K.; Lee, B.; Chung, I.; Kim, S.; and Kim, C. 2020. End-end Speech-to-Text Translation with Modality Agnostic Meta-Learning. In *ICASSP*, 7904–7908. IEEE.

Indurthi, S. R.; Zaidi, M. A.; Lee, B.; Lakumarapu, N. K.; and Kim, S. 2022. Language Model Augmented Monotonic Attention for Simultaneous Translation. In *NAACL*, 38–45. Seattle, United States: Association for Computational Linguistics.

Kano, T.; Sakti, S.; and Nakamura, S. 2018. Structured-based Curriculum Learning for End-to-end English-Japanese Speech Translation. *CoRR*, abs/1802.06003.

Kudo, T. 2018. Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates. In *ACL*, 66–75. Melbourne, Australia: Association for Computational Linguistics.

Le, N.; Lecouteux, B.; and Besacier, L. 2017. Disentangling ASR and MT Errors in Speech Translation. In *MTSummit*, 312–323.

Liu, D.; Du, M.; Li, X.; Li, Y.; and Chen, E. 2021. Cross Attention Augmented Transducer Networks for Simultaneous Translation. In *EMNLP*, 39–55. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Liu, Y.; Zhang, J.; Xiong, H.; Zhou, L.; He, Z.; Wu, H.; Wang, H.; and Zong, C. 2020. Synchronous Speech Recognition and Speech-to-Text Translation with Interactive Decoding. In *IAAI*, 8417–8424. AAAI Press.

Ma, M.; Huang, L.; Xiong, H.; Zheng, R.; Liu, K.; Zheng, B.; Zhang, C.; He, Z.; Liu, H.; Li, X.; Wu, H.; and Wang, H.

2019. STACL: Simultaneous Translation with Implicit Anticipation and Controllable Latency using Prefix-to-Prefix Framework. In *ACL*, 3025–3036. Association for Computational Linguistics.

Ma, X.; Pino, J. M.; Cross, J.; Puzon, L.; and Gu, J. 2020. Monotonic Multihead Attention. In *ICLR*. OpenReview.net.

Ma, X.; Pino, J. M.; and Koehn, P. 2020. SimulMT to SimulST: Adapting Simultaneous Text Translation to End-to-End Simultaneous Speech Translation. In *AACL*, 582–587. Association for Computational Linguistics.

Ma, X.; Wang, Y.; Dousti, M. J.; Koehn, P.; and Pino, J. M. 2021. Streaming Simultaneous Speech Translation with Augmented Memory Transformer. In *ICASSP*, 7523–7527. IEEE.

Nguyen, H.; Estève, Y.; and Besacier, L. 2021. An Empirical Study of End-To-End Simultaneous Speech Translation Decoding Strategies. In *ICASSP*, 7528–7532. IEEE.

Oda, Y.; Neubig, G.; Sakti, S.; Toda, T.; and Nakamura, S. 2014. Optimizing Segmentation Strategies for Simultaneous Speech Translation. In *ACL*, 551–556. The Association for Computer Linguistics.

Panayotov, V.; Chen, G.; Povey, D.; and Khudanpur, S. 2015. Librispeech: An ASR corpus based on public domain audio books. In *ICASSP*, 5206–5210. IEEE.

Papi, S.; Negri, M.; and Turchi, M. 2023. Attention as a Guide for Simultaneous Speech Translation. In *ACL*, 13340–13356. Toronto, Canada: Association for Computational Linguistics.

Pino, J.; Puzon, L.; Gu, J.; Ma, X.; McCarthy, A. D.; and Gopinath, D. 2019. Harnessing Indirect Training Data for End-to-End Automatic Speech Translation: Tricks of the Trade. In *IWSLT*. Hong Kong: Association for Computational Linguistics.

Ren, Y.; Liu, J.; Tan, X.; Zhang, C.; Qin, T.; Zhao, Z.; and Liu, T.-Y. 2020. SimulSpeech: End-to-End Simultaneous Speech to Text Translation. In *ACL*, 3787–3796. Online: Association for Computational Linguistics.

Tang, Y.; Sun, A. Y.; Inaguma, H.; Chen, X.; Dong, N.; Ma, X.; Tomasello, P.; and Pino, J. 2023. Hybrid Transducer and Attention based Encoder-Decoder Modeling for Speech-to-Text Tasks. In *ACL*, 12441–12455. Association for Computational Linguistics.

Wang, C.; Wu, Y.; Liu, S.; Zhou, M.; and Yang, Z. 2020. Curriculum Pre-training for End-to-End Speech Translation. In *ACL*, 3728–3738. Association for Computational Linguistics.

Weiss, R. J.; Chorowski, J.; Jaitly, N.; Wu, Y.; and Chen, Z. 2017. Sequence-to-Sequence Models Can Directly Translate Foreign Speech. In *Interspeech*, 2625–2629. ISCA.

Xue, H.; Feng, Y.; Gu, S.; and Chen, W. 2020. Robust Neural Machine Translation with ASR Errors. In *Proceedings of the First Workshop on Automatic Simultaneous Translation*, 15–23. Seattle, Washington: Association for Computational Linguistics.

Ye, R.; Wang, M.; and Li, L. 2021. End-to-End Speech Translation via Cross-Modal Progressive Training. In *Interspeech*, 2267–2271. ISCA.

Ye, R.; Wang, M.; and Li, L. 2022. Cross-modal Contrastive Learning for Speech Translation. In *NAACL*, 5099–5113. Association for Computational Linguistics.

Zeng, X.; Li, L.; and Liu, Q. 2021. RealTranS: End-to-End Simultaneous Speech Translation with Convolutional Weighted-Shrinking Transformer. In *Findings of ACL*, 2461–2474. Association for Computational Linguistics.

Zhang, R.; He, Z.; Wu, H.; and Wang, H. 2022. Learning Adaptive Segmentation Policy for End-to-End Simultaneous Translation. In *ACL*, 7862–7874. Association for Computational Linguistics.

Zhang, R.; Zhang, C.; He, Z.; Wu, H.; and Wang, H. 2020. Learning Adaptive Segmentation Policy for Simultaneous Translation. In *EMNLP*, 2280–2289. Online: Association for Computational Linguistics.

Zhang, S.; and Feng, Y. 2022. Information-Transport-based Policy for Simultaneous Translation. In *EMNLP*, 992–1013. Association for Computational Linguistics.

Zhang, S.; and Feng, Y. 2023. End-to-End Simultaneous Speech Translation with Differentiable Segmentation. In *Findings of ACL*, 7659–7680. Association for Computational Linguistics.