

End-to-End Simultaneous Speech Translation with Differentiable Segmentation

Shaolei Zhang^{1,2}, Yang Feng^{1,2*}

¹Key Laboratory of Intelligent Information Processing
Institute of Computing Technology, Chinese Academy of Sciences (ICT/CAS)

²University of Chinese Academy of Sciences, Beijing, China
{zhangshaolei20z, fengyang}@ict.ac.cn

Abstract

End-to-end simultaneous speech translation (SimulST) outputs translation while receiving the streaming speech inputs (a.k.a. streaming speech translation), and hence needs to segment the speech inputs and then translate based on the current received speech. However, segmenting the speech inputs at unfavorable moments can disrupt the acoustic integrity and adversely affect the performance of the translation model. Therefore, learning to segment the speech inputs at those moments that are beneficial for the translation model to produce high-quality translation is the key to SimulST. Existing SimulST methods, either using the fixed-length segmentation or external segmentation model, always separate segmentation from the underlying translation model, where the gap results in segmentation outcomes that are not necessarily beneficial for the translation process. In this paper, we propose *Differentiable Segmentation (DiSeg)* for SimulST to directly learn segmentation from the underlying translation model. DiSeg turns hard segmentation into differentiable through the proposed expectation training, enabling it to be jointly trained with the translation model and thereby learn translation-beneficial segmentation. Experimental results demonstrate that DiSeg achieves state-of-the-art performance and exhibits superior segmentation capability¹.

1 Introduction

End-to-end simultaneous speech translation (SimulST) (Fügen et al., 2007; Oda et al., 2014; Ren et al., 2020; Zeng et al., 2021; Zhang et al., 2022a) outputs translation when receiving the streaming speech inputs, and is widely used in real-time scenarios such as international conferences, live broadcasts and real-time subtitles. Compared with the offline speech translation waiting for

*Corresponding author: Yang Feng.

¹ Code is available at <https://github.com/ictnlp/DiSeg>.

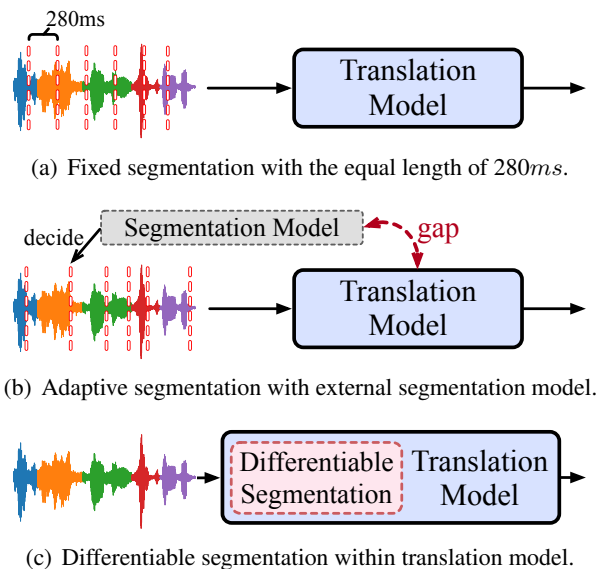


Figure 1: Illustration of differentiable segmentation (DiSeg) compared with the previous methods.

the complete speech inputs (Weiss et al., 2017; Wang et al., 2020), SimulST needs to segment the streaming speech inputs and synchronously translate based on the current received speech, aiming to achieve high translation quality under low latency (Hamon et al., 2009; Cho and Esipova, 2016; Ma et al., 2020b; Zhang and Feng, 2022c).

However, it is non-trivial to segment the streaming speech inputs as the speech always lacks explicit boundary (Zeng et al., 2021), and segmenting at unfavorable moments will break the acoustic integrity and thereby drop the translation performance (Dong et al., 2022). Therefore, the precise segmentation of streaming speech is the core challenge of SimulST task (Zhang et al., 2022a). To ensure that the speech representations derived from the segmentation results can produce high-quality translation, SimulST model should learn a translation-beneficial segmentation from the underlying translation model.

Existing SimulST methods, involving fixed and

adaptive, always fail to learn the segmentation directly from the underlying translation model. The fixed method divides the streaming inputs based on the equal length, e.g., 280ms per segment (Ma et al., 2020b, 2021; Nguyen et al., 2021), as shown in Figure 1(a). Such methods completely ignore the translation model and always break the acoustic integrity (Dong et al., 2022). The adaptive method dynamically decides the segmentation and thereby achieves better SimulST performance, as shown in Figure 1(b). However, previous adaptive methods often use the external segmentation model (Ma et al., 2020b; Zhang et al., 2022a) or heuristic detector (Zeng et al., 2021; Chen et al., 2021; Dong et al., 2022) for segmentation, which leave a gap between the segmentation and translation model. This gap hinders learning segmentation directly from the translation model (Arivazhagan et al., 2019), hence making them difficult to get segmentation results that are most beneficial to translation quality.

Under these grounds, we aim to integrate the segmentation into translation model, and thereby directly learn segmentation from the underlying translation model, as shown in Figure 1(c). To this end, we propose *Differentiable Segmentation* (*DiSeg*) for SimulST, which can be jointly trained with the underlying translation model. DiSeg employs a Bernoulli variable to indicate whether the streaming speech inputs should be segmented or not. Then, to address the issue that hard segmentation precludes back-propagation (i.e., learning) from the underlying translation model, we propose an expectation training to turn the segmentation into differentiable. Owing to powerful segmentation, DiSeg can handle simultaneous and offline speech translation through a unified model. Experiments show that DiSeg achieves state-of-the-art performance on SimulST, while also delivering high-quality offline speech translation.

2 Background

Offline Speech Translation The corpus of speech translation task is always denoted as the triplet $\mathcal{D} = \{(\mathbf{s}, \mathbf{x}, \mathbf{y})\}$, where $\mathbf{s} = (s_1, \dots, s_{|\mathbf{s}|})$ is source speech, $\mathbf{x} = (x_1, \dots, x_{|\mathbf{x}|})$ is source transcription and $\mathbf{y} = (y_1, \dots, y_{|\mathbf{y}|})$ is target translation. The mainstream speech translation architecture often consists of an acoustic feature extractor and a translation model following (Nguyen et al., 2020). Acoustic feature extractor extracts speech features $\mathbf{a} = (a_1, \dots, a_{|\mathbf{a}|})$ from source speech \mathbf{s} , which

is often realized by a pre-trained acoustic model (Baeovski et al., 2020). Then, the translation model, realized by a Transformer model (Vaswani et al., 2017), generates \mathbf{y} based on all speech features \mathbf{a} . During training, existing methods always improve speech translation performance through multi-task learning (Anastasopoulos and Chiang, 2018; Tang et al., 2021a,b), including speech translation, automatic speech recognition (ASR) and machine translation (MT) (add a word embedding layer for MT task), where the learning objective \mathcal{L}_{mtl} is:

$$\mathcal{L}_{mtl} = \mathcal{L}_{st} + \mathcal{L}_{asr} + \mathcal{L}_{mt}, \quad (1)$$

where \mathcal{L}_{st} , \mathcal{L}_{asr} and \mathcal{L}_{mt} are the cross-entropy loss of pairs $\mathbf{s} \rightarrow \mathbf{y}$, $\mathbf{s} \rightarrow \mathbf{x}$ and $\mathbf{x} \rightarrow \mathbf{y}$, respectively.

Simultaneous Translation (SimulST) Unlike offline speech translation, SimulST needs to decide when to segment the inputs and then translate based on the received speech features (Ren et al., 2020; Ma et al., 2020b). Since the decisions are often made based on the speech features after downsampling, we use $g(t)$ to denote the number of speech features when the SimulST model translates y_t , where speech features $\mathbf{a}_{\leq g(t)}$ are extracted from the current received speech $\hat{\mathbf{s}}$. Then, the probability of generating y_t is $p(y_t | \hat{\mathbf{s}}, \mathbf{y}_{<t})$. How to decide $g(t)$ is the key of SimulST, which should be beneficial for the translation model to produce the high-quality translation.

3 Method

We propose differentiable segmentation (DiSeg) to learn segmentation directly from the translation model, aiming to achieve translation-beneficial segmentation. As shown in Figure 2, DiSeg predicts a Bernoulli variable to indicate whether to segment, and then turns the hard segmentation into differentiable through the proposed expectation training, thereby jointly training segmentation with the translation model. We will introduce the segmentation, training and inference of DiSeg following.

3.1 Segmentation

To segment the streaming speech inputs, DiSeg predicts a Bernoulli variable 0/1 for each speech feature, corresponding to waiting or segmenting. Specifically, for the speech feature a_i , DiSeg predicts a Bernoulli segmentation probability p_i , corresponding to the probability of segmenting the speech at a_i . Segmentation probability p_i is calculated through a feed-forward network (FFN) fol-

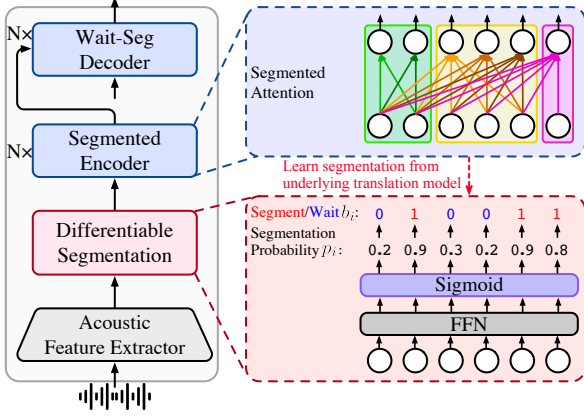


Figure 2: Architecture of the proposed DiSeg.

lowed by a sigmoid activation, and then p_i is used to parameterize the Bernoulli variable b_i :

$$p_i = \text{Sigmoid}(\text{FFN}(a_i)), \quad (2)$$

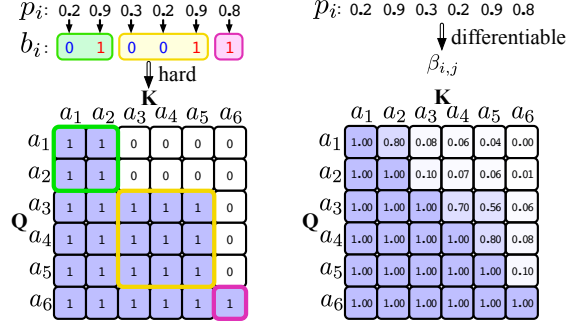
$$b_i \sim \text{Bernoulli}(p_i). \quad (3)$$

If $b_i = 1$, DiSeg segments the streaming speech at a_i ; If $b_i = 0$, DiSeg waits for more inputs. In inference, DiSeg sets $b_i = 1$ if $p_i \geq 0.5$, and sets $b_i = 0$ if $p_i < 0.5$ (Raffel et al., 2017).

Segmented Attention After segmenting the speech features, we propose segmented attention for the encoder of the translation model, which is an attention mechanism between uni-directional and bi-directional attention. In segmented attention, each speech feature can focus on the features in the same segment and the previous segments (i.e., bi-directional attention within a segment, uni-directional attention between segments), as shown in Figure 3(a). In this way, segmented attention can not only satisfy the requirement of encoding streaming inputs in SimulST task (i.e., the characteristic of uni-directional attention) (Elbayad et al., 2020; Zeng et al., 2021), but also capture more comprehensive context representations of segments (i.e., the advantage of bi-directional attention).

3.2 Expectation Training

Hard segmentation based on Bernoulli variable b_i precludes back-propagation (i.e., learning) from the translation model to the segmentation probability p_i during training. To address this, we propose expectation training to turn segmentation into differentiable. In expectation training, we first constrain the number of segments, and then learns segmentation from the translation model at both acoustic and semantic levels, which are all trained in expectation via the segmentation probability p_i .



(a) Segmented attention. (b) Expected segmented attention.

Figure 3: Schematic diagram of segmented attention in inference and expected segmented attention in training.

Learning Segment Number To avoid too many segments breaking the acoustic integrity or too few segments degenerating the model into offline translation, we need to constrain the total number of segments. Intuitively, the source speech should be divided into K segments, where K is the number of words in the source transcription, so that each speech segment can correspond to a complete word. To this end, we apply \mathcal{L}_{num} to constrain the expected segment number to be K . In particular, in order to prevent excessive segmentation on consecutive silent frames, we also to encourage only one segmentation in several consecutive speech frames. Therefore, \mathcal{L}_{num} is calculated as:

$$\mathcal{L}_{num} = \left\| \sum_{i=1}^{|\mathbf{a}|} p_i - K \right\|_2 + \left\| \sum \text{MaxPool} \left(p_i, \left\lfloor \frac{|\mathbf{a}|}{K} \right\rfloor \right) - K \right\|_2, \quad (4)$$

where $\sum_{i=1}^{|\mathbf{a}|} p_i$ is the expected segment number and $\text{MaxPool}(\cdot)$ is the max polling operation with kernel size of $\lfloor |\mathbf{a}| / K \rfloor$.

To make the effect of p_i in expectation training match b_i in the inference, we hope that $p_i \approx 0$ or $p_i \approx 1$ and thereby make $p_i \approx b_i$. To achieve this, we aim to encourage the discreteness of segmentation probability p_i during training. Following Salakhutdinov and Hinton (2009); Foerster et al. (2016); Raffel et al. (2017), a straightforward and efficient method is adding a Gaussian noise before the sigmoid activation in Eq.(2). Formally, in expectation training, Eq.(2) is rewritten as:

$$p_i = \text{Sigmoid}(\text{FFN}(a_i) + \mathcal{N}(0, n)), \quad (6)$$

where $\mathcal{N}(0, n)$ is a Gaussian noise with 0 as mean and n as variance. Noise is only applied in training.

Learning Segmentation at Acoustic Level A good segmentation should avoid breaking acoustic integrity and benefit the underlying translation model. As mentioned in Sec.3.1, the encoder of the translation model applies the segmented attention to model the correlation between speech features and get the source representations. Correspondingly, we propose *expected segmented attention* to turn the hard segmentation into differentiable during training, thereby directly learning translation-beneficial segmentation from the translation model.

In segmented attention during inference, speech feature a_i can only pay attention to feature a_j that locates in the same segment or previous segments, and mask out the rest features. In expected segmented attention, to enable back-propagation, we introduce the probability that a_i can pay attention to a_j instead of the hard segmentation, denoted as $\beta_{i,j}$. As shown in Figure 3(b), $\beta_{i,j}$ measures the probability that a_j locates in the same segment as a_i or in the segment before a_i , calculated as:

$$\beta_{i,j} = \begin{cases} \prod_{l=i}^{j-1} (1 - p_l), & \text{if } i < j \\ 1, & \text{if } i \geq j \end{cases} \quad (7)$$

If a_j lags behind a_i , the premise that a_i and a_j are in the same segment is that no segmentation is between a_i and a_{j-1} , i.e., $\prod_{l=i}^{j-1} (1 - p_l)$. If a_j is before a_i , a_i can necessarily focus on a_j . Then, $\beta_{i,j}$ is multiplied with the original soft attention $\alpha_{i,j}$, and normalized to get the final attention $\gamma_{i,j}$:

$$\tilde{\gamma}_{i,j} = \alpha_{i,j} \times \beta_{i,j}, \quad (8)$$

$$\gamma_{i,j} = \tilde{\gamma}_{i,j} / \sum_{l=1}^{|\mathbf{a}|} \tilde{\gamma}_{i,l}. \quad (9)$$

Finally, $\gamma_{i,j}$ is used to calculate the context vector. Owing to the expected segmented attention, p_i can be jointly trained with the translation model via cross-entropy loss \mathcal{L}_{mtl} . Specifically, if the underlying translation model prefers to let a_i pay attention to the subsequent a_j for better source representation (i.e., a large $\gamma_{i,j}$), the probability $\beta_{i,j}$ that they are in the same segment will be increased, which teaches DiSeg to prevent segmenting between a_i and a_j . In this way, DiSeg can avoid breaking acoustic integrity and learn the segmentation that is beneficial to the translation model.

Learning Segmentation at Semantic Level Besides encouraging the related speech features to locate in the same segment via expected segmented attention, we aim to further learn segmentation at

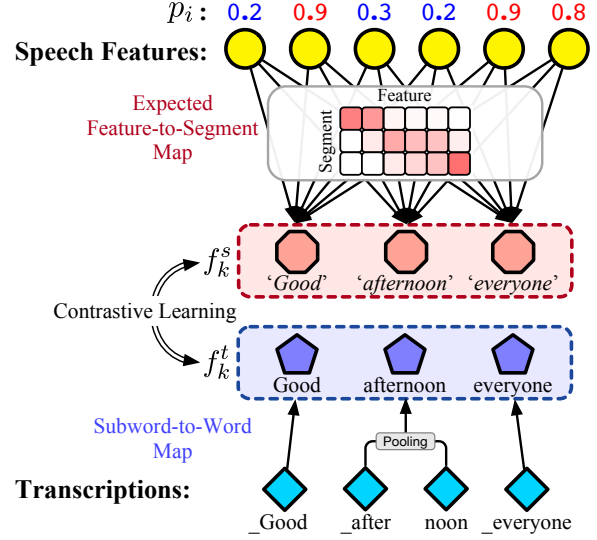


Figure 4: Schematic diagram of learning segmentation at the semantic level, via two mappings.

the semantic level. In the multi-task learning framework, the transcription \mathbf{x} is monotonically corresponding to the source speech, so transcription is a good choice to provide semantic supervision for segmentation. However, there is a significant gap between transcription representations and speech features in sequence length (Liu et al., 2020; Zeng et al., 2021), so how to align them is key challenge for semantic supervision. Fortunately, the proposed differentiable segmentation divides the speech into K segments, where K is also the word number in transcription. Therefore, both speech and transcription sequences can be mapped to the sequence of length K and accordingly reach an agreement on the corresponding representation to achieve semantic supervision, as shown in Figure 4.

For transcription \mathbf{x} , since it is a sequence of subwords after tokenization (Kudo and Richardson, 2018), we introduce a *subword-to-word map* to get the representation of the whole word. Given the text embedding $\mathbf{e} = \text{Emb}(\mathbf{x})$ of subwords, the whole word representation is the average pooling result on the embedding of all subwords it contains. Formally, the representation f_k^t of the k^{th} word that consists of subwords $\mathbf{x}[l_k : r_k]$ is calculated as:

$$f_k^t = \frac{1}{r_k - l_k + 1} \sum_{i=l_k}^{r_k} e_i. \quad (10)$$

For speech, the segment representations also need to be differentiable to segmentation (i.e., p_i), thereby enabling DiSeg to learn segmentation at semantic level. To this end, we propose an *expected*

Algorithm 1: Wait-seg Policy for DiSeg

Input: streaming speech inputs s , lagging segments k **Output:** target outputs \hat{y} **Initialization:** $\hat{y}_0 = \langle \text{BOS} \rangle$, target index $t = 1$,
current received speech $\hat{s} = []$

```
1 while  $\hat{y}_{t-1} \neq \langle \text{EOS} \rangle$  do
2   Extract speech features  $\{a_l\}_{l=1}^i$  from  $\hat{s}$ ;
3   Predict segmentation  $\{b_l\}_{l=1}^i$ ;
4   if  $\sum_{l=1}^i b_l \geq t+k-1$  or  $\hat{s} == s$ 
5     then // Translate
6       Translate  $\hat{y}_t$  based on  $\hat{s}$ ;  $t \leftarrow t + 1$ ;
7     else // Wait
8        $\hat{s} \leftarrow \hat{s} + s.\text{read}()$ ;
9 return  $\hat{y}$ ;
```

feature-to-segment map to get the expected segment representations. Expected feature-to-segment map does not forcibly assign speech feature a_i to a certain segment but calculates the probability that a_i belongs to the k^{th} segment, denoted as $p(a_i \in \text{Seg}_k)$, which can be calculated via dynamic programming (refer to Appendix A for details):

$$p(a_i \in \text{Seg}_k) = p(a_{i-1} \in \text{Seg}_{k-1}) \times p_{i-1} + p(a_{i-1} \in \text{Seg}_k) \times (1 - p_{i-1}). \quad (11)$$

Then, the expected representation f_k^s of the k^{th} segment is calculated by weighting all speech features:

$$f_k^s = \sum_{i=1}^{|a|} p(a_i \in \text{Seg}_k) \times a_i. \quad (12)$$

Owing to the proposed two mappings, transcription and speech are mapped to the representations with the same length, i.e., K segments/words, where f_k^s corresponds to f_k^t . To provide semantic supervision for segmentation, we apply multi-class N-pair contrastive loss \mathcal{L}_{ctr} (Sohn, 2016) between f^s and f^t , where f_k^t is the positive sample of f_k^s and the rest are the negative samples, calculated as:

$$\mathcal{L}_{ctr} = -\sum_{f^s, f^t} \log \frac{\exp(\text{sim}(f_k^s, f_k^t)/\tau)}{\sum_{n=1}^K \exp(\text{sim}(f_k^s, f_n^t)/\tau)}. \quad (13)$$

$\text{sim}(\cdot)$ calculates the cosine similarity between segment and word representations. τ is temperature and we set $\tau = 0.1$ following Wang and Liu (2021).

Overall, the total loss of expectation training is:

$$\mathcal{L}_{DiSeg} = \mathcal{L}_{mtl} + \mathcal{L}_{num} + \mathcal{L}_{ctr}. \quad (14)$$

3.3 Inference Policy

Owing to the proposed differentiable segmentation, the streaming speech inputs are divided into multiple segments, where each segment contains roughly

one word. Accordingly, inspired by the wait- k policy (Ma et al., 2019) in simultaneous machine translation, we propose *wait-seg policy* for DiSeg. Specifically, wait-seg policy first waits for k segments, and then translates a target word whenever deciding to segment the streaming speech inputs, where k is a hyperparameter to control the latency. Formally, given lagging segments k , DiSeg translates y_t when receiving $g(t; k)$ speech features:

$$g(t; k) = \underset{i}{\text{argmin}} \left(\sum_{l=1}^i b_l \geq t + k - 1 \right). \quad (15)$$

The specific inference is shown in Algorithm 1.

To keep the training and inference matching, we also apply the wait-seg policy during training via the proposed wait-seg decoder. When translating y_t , wait-seg decoder will mask out the speech features a_i that $i > g(t; k)$ (Ma et al., 2019). Accordingly, we introduce multi-task training and multi-latency training to enhance DiSeg performance.

Multi-task Training Since we adopt the multi-task learning framework (refer to Sec.2), ASR and MT tasks should also adapt to DiSeg. Specifically, ASR task applies the same segmentation and policy (i.e., decoder) as the SimulST task, as both their inputs are speech. For the MT task, since the segment in the speech corresponds to the word in the transcription, MT task applies a uni-directional encoder and wait- k policy (i.e., decoder). Note that parameters of encoder and decoder are shared among various tasks.

Multi-latency Training To enhance the DiSeg performance under multiple latency, we randomly sample k from $[1, K]$ between batches during training (Elbayad et al., 2020). In inference, DiSeg only needs one model to complete SimulST under multiple arbitrary latency (Zhang and Feng, 2021c), including offline speech translation (the latency is the complete speech duration). In this way, DiSeg develops a unified model that can handle both offline and simultaneous speech translation.

4 Experiments

4.1 Datasets

We conduct experiments on two end-to-end simultaneous translation benchmarks, MuST-C² English \rightarrow German (En \rightarrow De, 234K pairs) and English \rightarrow Spanish (En \rightarrow Es, 270K pairs) (Di Gangi et al., 2019). We use dev as the validation set (1423

²<https://ict.fbk.eu/must-c>

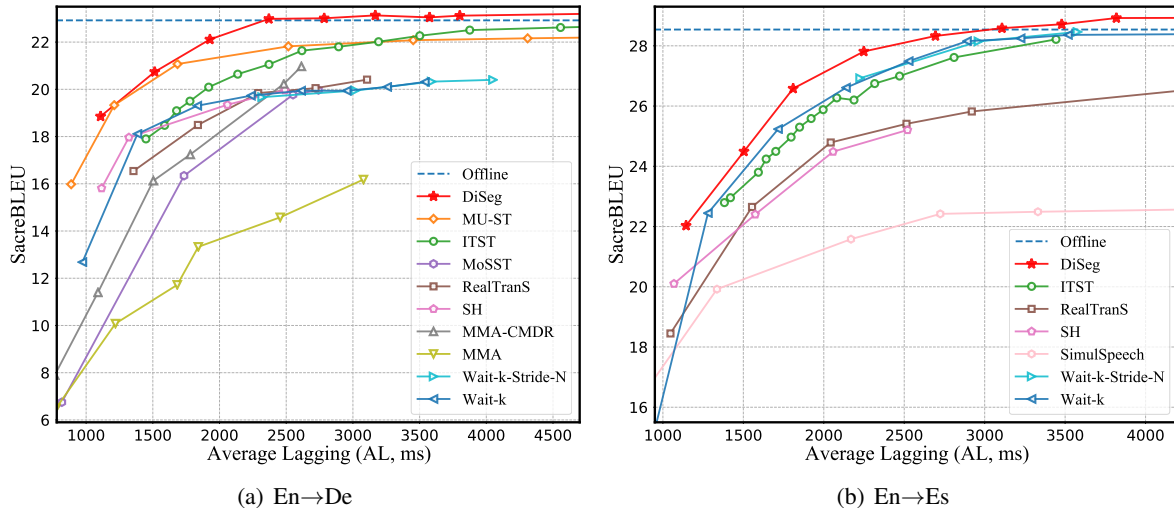


Figure 5: SimulST performance of translation quality against latency (AL, ms) on MuST-C En→De and En→Es.

pairs for En→De, 1316 pairs for En→Es) and use `tst-COMMON` as the test set (2641 pairs for En→De, 2502 pairs for En→Es), respectively. For speech, we use the raw 16-bit 16kHz mono-channel audio wave. For text, we use SentencePiece (Kudo and Richardson, 2018) to generate a unigram vocabulary of size 10000, sharing between languages.

4.2 System Settings

We conduct experiments on the following systems.

Offline Offline speech translation, which waits for the complete speech inputs and then translates (bi-directional attention and greedy search).

Wait-k Wait-k policy (Ma et al., 2019) with fixed segmentation of speech (Ma et al., 2020b), which translates a word every $280ms$.

Wait-k-Stride-n A variation of wait-k policy (Zeng et al., 2021), which translates n word every $n \times 280ms$. We set $n = 2$ following their best result.

MMA³ Monotonic multihead attention (Ma et al., 2020c), which is adapted to SimulST by dividing speech into segments of equal length, i.e., $120ms$, $200ms$ and $280ms \dots$ (Ma et al., 2020b).

MMA-CMDR MMA with cross-modal decision regularization (Zaidi et al., 2022), which leverages the transcription to improve the decision of MMA.

SimulSpeech Segmentation based on word detector (Ren et al., 2020), which also uses two knowledge distillations to improve the performance.

SH Synchronized ASR-assisted SimulST (Chen et al., 2021), which uses the shortest hypothesis in

³https://github.com/pytorch/fairseq/tree/master/examples/simultaneous_translation

ASR results to indicate the word number in speech.

RealTranS A convolutional weighted-shrinking Transformer (Zeng et al., 2021), which detects the word number in the streaming speech and then decodes via the wait-k-stride-n policy.

MoSST⁴ Monotonic-segmented streaming speech translation (Dong et al., 2022), which uses the integrate-and-firing method (Dong and Xu, 2020) to segment the speech based on the cumulative acoustic information.

ITST⁵ Information-transport-based policy for SimulST (Zhang and Feng, 2022b), which quantifies the transported information from source to target, and then decides whether to translate according to the accumulated received information.

MU-ST Segmentation based on the meaning unit (Zhang et al., 2022a), which trains an external segmentation model based on the constructed data, and uses it to decide when to translate.

DiSeg The proposed method in Sec.3.

All implementations are adapted from Fairseq Library (Ott et al., 2019). We use a pre-trained Wav2Vec2.0⁶ (Baevski et al., 2020) as the acoustic feature extractor, and use a standard Transformer-Base (Vaswani et al., 2017) as the translation model. For evaluation, we apply SimulEval⁷ (Ma et al., 2020a) to report SacreBLEU (Post, 2018) for translation quality and Average Lagging (AL) (Ma et al., 2019) for latency. AL measures the average dura-

⁴<https://github.com/dqqcasia/mosst>

⁵<https://github.com/ictnlp/ITST>

⁶dl.fbaipublicfiles.com/fairseq/wav2vec/wav2vec_small.pt

⁷<https://github.com/facebookresearch/SimulEval>

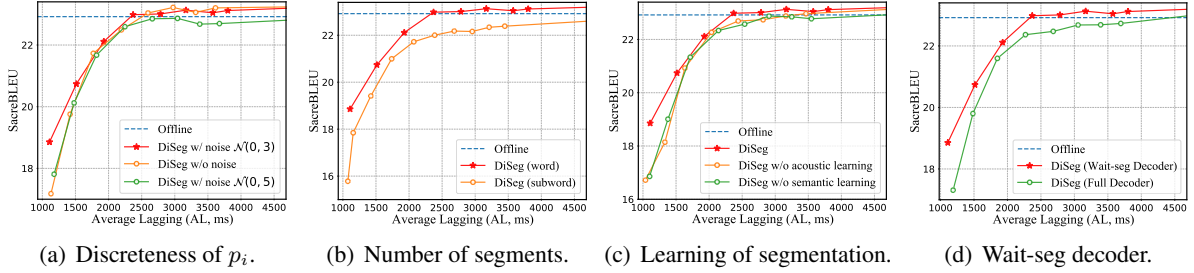


Figure 6: Ablation studies of DiSeg on MuST-C En→De test set.

tion (ms) that target outputs lag behind the speech inputs. The calculation refer to Appendix D.

4.3 Main Results

We compare DiSeg and previous SimulST methods in Figure 5, where we only train a single DiSeg model and adjusted the lagging number k during the inference process to show the translation quality under different latency. Remarkably, DiSeg outperforms strong baselines under all latency and achieves state-of-the-art performance. Compared with fixed methods, such as Wait- k and MMA (Ma et al., 2020b), DiSeg can dynamically decides segmentation according to the inputs instead of equal-length segmentation, which avoids breaking the acoustic integrity and thus achieves notable improvements. Compared with adaptive methods, including the state-of-the-art MU-ST, DiSeg also performs better. In previous adaptive methods, regardless of RealTrans, SH and SimulSpeech detecting the word number (Zeng et al., 2021; Chen et al., 2021; Ren et al., 2020), MoSST and ITST comparing the acoustic information with a threshold (Dong et al., 2022; Zhang and Feng, 2022b), or MU-ST training an external segmentation model (Zhang et al., 2022a), the final translation results is always non-differentiable to the segmentation, which hinders learning segmentation directly from the translation model. The proposed DiSeg turns the segmentation into differentiable, hence can learn translation-beneficial segmentation directly from the translation model, thereby achieving better performance. Furthermore, unlike the previous methods using uni-directional (e.g., RealTrans and ITST) or bi-directional attention (e.g., MU-ST and MoSST), the proposed segmented attention can not only encode streaming inputs but also get comprehensive segment representations (Zhang et al., 2021). In particular, DiSeg achieves comparable performance with the offline model when lag-

ging $2300ms$ on En→De and $3000ms$ on En→Es, which is attributed to the improvements on translation quality brought by the segmented attention.

5 Analysis

We conduct extensive analyses to study the effectiveness and specific improvements of DiSeg. Unless otherwise specified, all the results are reported on MuST-C En→De test set.

5.1 Ablation Study

Discreteness of Segmentation Probability To make expectation training more suitable for inference, we encourage the discreteness of segment probability via introducing Gaussian noise $\mathcal{N}(0, n)$ in Eq.(6). We compare the effect of discreteness in Figure 6(a), where appropriately encouraging discreteness effectively enhances expectation training, thereby improving DiSeg performance under low latency. However, too much noise will affect translation quality, especially under high latency, which is consistent with Arivazhagan et al. (2019).

Number of Segments In DiSeg, we constrain the number of segments to be the word number K in the transcription rather than subword. To verify the effectiveness of segmentation granularity, we compare different segment numbers in Figure 6(b), noting that \mathcal{L}_{ctr} is also changed to be computed by subword embedding accordingly. Segmentation on word granularity is significantly better than subword granularity, mainly because many subwords are actually continuous and related in the speech, and segmentation at the word granularity can better preserve the acoustic integrity (Dong et al., 2022).

Learning at Acoustic and Semantic Levels DiSeg learns segmentation at the acoustic and semantic levels, so we show the effectiveness of acoustic and semantic learning in Figure 6(c). The results demonstrate that both acoustic and semantic learning play the important role in SimulST per-

Encoder Type	En→De		En→Es	
	Greedy	Beam5	Greedy	Beam5
Uni-directional	22.94	24.26	28.54	28.92
Bi-directional	22.92	24.64	28.47	29.51
DiSeg (segmented)	23.34	24.68	28.96	29.65
− \mathcal{L}_{ctr}	23.10	24.46	28.64	29.48

Table 1: Performance of segmented attention against bi-/uni-directional attention on offline speech translation.

formance. Specifically, acoustic learning encourages related speech features to be in the same segment through expected segment attention, where ensuring acoustic integrity is more important for SimulST under low latency, thereby achieving an improvement of 2 BLEU ($AL \approx 1500ms$). Semantic learning supervises the segment representations through the word representations in the transcription, which helps the segment representations to be more conducive to the translation model (Ye et al., 2022), thereby improving the translation quality.

Effect of Wait-seg Decoder DiSeg introduces wait-seg decoder to learn wait-seg policy. As shown in Figure 6(d), compared with full decoder that can focus on all speech features, wait-seg decoder enhances DiSeg’s ability to translate based on partial speech (Ma et al., 2019) and thus achieves significant improvements during inference.

5.2 Segmented Attention on Offline ST

How to encode streaming inputs is an important concern for SimulST (Zhang et al., 2021), where offline translation uses bi-directional attention to encode the complete source input and existing SimulST methods always apply uni-directional attention (Zeng et al., 2021; Zhang and Feng, 2022b). DiSeg applies segmented attention, which consists of bi-directional attention within a segment and uni-directional attention between segments. To study the modeling capability of segmented attention, we compare the performance of uni-directional attention, bi-directional attention and DiSeg (segmented attention) on offline speech translation in Table 1. DiSeg is a unified model that can handle both simultaneous speech translation and offline speech translation together. Therefore, we employ the same model as SimulST to complete offline speech translation, while only setting $k = \infty$ and applying beam search during inference.

Uni- and bi-directional attention achieve similar performance in greedy search, which is consistent with Wu et al. (2021), while bi-directional

Methods	P(↑)	R(↑)	F1(↑)	OS(0)	R-val(↑)
ES K-Means	30.7	18.0	22.7	-41.2	39.7
BES GMM	31.7	13.8	19.2	-56.6	37.9
VQ-CPC	18.2	54.1	27.3	196.4	-86.5
VQ-VAE	16.4	56.8	25.5	245.2	-126.5
SCPC	35.0	29.6	32.1	-15.4	44.5
DSegKNN	30.9	32.0	31.5	3.5	40.7
Fixed(280ms)	28.1	16.3	20.7	-42.0	38.4
DiSeg	34.9	32.3	33.5	-7.4	44.6
− \mathcal{L}_{ctr}	33.9	31.0	32.4	-8.5	43.9

Table 2: Segmentation quality on Buckeye corpus. Better segmentation has OS close to 0, higher F1 and R-val.

attention performs better in beam search due to more comprehensive encoding. Owing to learning the translation-beneficial segmentation, DiSeg can outperform uni-/bi-directional attention on both greedy and beam search when only applying bi-directional attention within segments. Furthermore, when removing \mathcal{L}_{ctr} , segmented attention also achieves comparable performance to bi-directional attention, providing a new attention mechanism for future streaming models. Appendix C.2 gives visualization and more analyses of segmented attention.

5.3 Segmentation Quality

To explore the segmentation quality of DiSeg, we conduct experiments on speech segmentation task (Demuynck and Laureys, 2002) with the annotated Buckeye corpus⁸ (Pitt et al., 2005). Table 2 shows the segmentation performance of DiSeg and strong baselines (Kamper et al., 2017b,a; Kamper and van Niekerk, 2020; Bhati et al., 2022; Fuchs et al., 2022), and the metrics include precision (P), recall (R), F1 score, over-segmentation (OS) and R-value (refer to Appendix B for the calculation). The results show that DiSeg achieves better segmentation performance and \mathcal{L}_{ctr} can improve the segmentation quality by 1% score. More importantly, DiSeg achieves an OS score close to 0, demonstrating that DiSeg can get the appropriate number of segments, thereby avoiding too many or too few segments affecting SimulST performance (Zhang et al., 2022b). We will analyze the number of segments following.

5.4 Segmentation Quantity

In training, we constrain the number of segments to be the word number in the transcription. To verify its effectiveness, we count the difference between the segment number and the word number during inference (i.e., #Segments−#Words) in Figure 7.

⁸<https://buckeyecorpus.osu.edu>

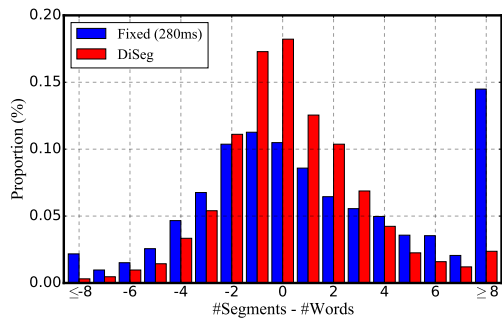


Figure 7: Distribution of the difference between the segment number generated by DiSeg and the word number in transcription, evaluated on MuST-C En→De.

Compared with the previous work considering that 280ms corresponds to a word on average (Ma et al., 2020b; Zaidi et al., 2022), DiSeg can get a more accurate number of segments, where the difference between the segment number and the word number is less than 2 in 70% of cases. Besides, as reported in Table 2, the OS score on the automatic speech segmentation task also demonstrates that DiSeg can achieve an appropriate number of segments. Therefore, constraining the expected segment number in expectation training is effective to control the number of segments.

5.5 Adapting Multi-task Learning to SimulST

During training, we adjust ASR task (segmented encoder + wait-seg decoder) and MT task (uni-encoder + wait-k decoder) in multi-task learning to adapt to DiSeg, so we verify the effectiveness of the adaptation in Table 3. In the proposed adaptation #1, since a speech segment corresponds to a word, both uni-encoder and wait-k decoder in MT task are fully compatible with the segmented attention and wait-seg decoder in SimulST task, thereby performing best. Both using bi-encoder (i.e., setting #5-7) or full decoder (i.e., setting #2-4) in ASR and MT tasks will affect DiSeg performance, and the performance degradation caused by the encoder mismatch is more serious. In general, ASR and MT tasks should be consistent and compatible with the SimulST task when adapting multi-task learning.

6 Related Work

Early SimulST methods segment speech and then use the cascaded model (ASR+MT) to translate each segment (Fügen et al., 2007; Yarmohammadi et al., 2013; Rangarajan Sridhar et al., 2013; Zhang and Feng, 2023; Guo et al., 2023). Recent end-to-end SimulST methods fall into fixed and adaptive.

#	ASR		MT		AL	BLEU
	Enc.	Dec.	Enc.	Dec.		
1	seg	wait-seg	uni	wait-k	1514	20.74
2	seg	wait-seg	uni	full	1428	19.32
3	seg	full	uni	wait-k	1404	19.58
4	seg	full	uni	full	1398	19.39
5	seg	wait-seg	bi	full	1416	18.93
6	bi	full	uni	wait-k	1704	20.23
7	bi	full	bi	full	1374	18.82

Table 3: Performance with various settings of multi-task learning, evaluating with $k=3$ on MuST-C En→De.

For fixed, Ma et al. (2020b) proposed fixed pre-decision to divide speech into equal-length segments, and migrated simultaneous MT method, such as wait-k (Ma et al., 2019; Zhang and Feng, 2021c,a; Guo et al., 2022) and MMA (Ma et al., 2020c), to SimulST. For adaptive, Ren et al. (2020) proposed SimulSpeech to detect the word in speech. Chen et al. (2021) used ASR results to indicate the word number. Zeng et al. (2021) proposed RealTrans, which detects the source word and further shrinks the speech length. Dong et al. (2022) proposed MoSST to translate after the acoustic information exceeding 1. Zhang and Feng (2022b) proposed ITST to judge whether the received information is sufficient for translation. Zhang et al. (2022a) proposed MU-ST, which constructs the segmentation labels based on meaning unit, and uses it to train a segmentation model.

In the previous method, whether using an external segmentation model or the detector, the segmentation cannot receive the gradient (i.e., learning) from the underlying translation model as the hard segmentation is not differentiable. Owing to the differentiable property, DiSeg can be jointly trained with the underlying translation model and directly learn translation-beneficial segmentation.

7 Conclusion

In this study, we propose differentiable segmentation (DiSeg) for simultaneous speech translation to directly learn segmentation from the underlying translation model. Experiments show the superiority of DiSeg in terms of SimulST performance, attention mechanism and segmentation quality.

Future researches will delve into the untapped potential of differentiable segmentation in such streaming models and long sequence modeling, thereby reducing feedback latency or computational cost without compromising performance.

Acknowledgements

We thank all the anonymous reviewers for their insightful and valuable comments. This work was supported by National Key R&D Program of China (NO. 2018AAA0102502).

Limitations

In this study, we propose differentiable segmentation to learn how to segment speech from the underlying translation model, and verify its effectiveness on simultaneous speech translation. However, since it can be jointly trained with the underlying task (sequence-to-sequence task), differentiable segmentation is not limited to the SimulST task, but can be generalized to more streaming/online tasks, such as streaming automatic speech recognition (streaming ASR), simultaneous machine translation (SiMT), real-time text-to-speech synthesis (real-time TTS), online tagging and streaming parsing. Given that there may be some task-specific differences between various tasks, this work only focuses on the differentiable segmentation in the SimulST task, and we leave the study of how to apply differentiable segmentation to other streaming tasks into our future work.

References

- Antonios Anastasopoulos and David Chiang. 2018. [Tied multitask learning for neural speech translation](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 82–91, New Orleans, Louisiana. Association for Computational Linguistics.
- 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](#). pages 1313–1323.
- 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*, volume 33, pages 12449–12460. Curran Associates, Inc.
- Saurabhchand Bhati, Jesús Villalba, Piotr Żelasko, Laureano Moro-Velazquez, and Najim Dehak. 2022. [Unsupervised speech segmentation and variable rate representation learning using segmental contrastive predictive coding](#). *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:2002–2014.
- 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. Online. Association for Computational Linguistics.
- Kyunghyun Cho and Masha Esipova. 2016. [Can neural machine translation do simultaneous translation?](#)
- Kris Demuyne and Tom Laureys. 2002. [A comparison of different approaches to automatic speech segmentation](#). In *Text, Speech and Dialogue*, pages 277–284. Berlin, Heidelberg. Springer Berlin Heidelberg.
- Mattia A. Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019. [MuST-C: a Multilingual Speech Translation Corpus](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2012–2017. Minneapolis, Minnesota. Association for Computational Linguistics.
- Shuoyang Ding, Hainan Xu, and Philipp Koehn. 2019. [Saliency-driven word alignment interpretation for neural machine translation](#). In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 1–12. Florence, Italy. Association for Computational Linguistics.
- Linhao Dong and Bo Xu. 2020. [Cif: Continuous integrate-and-fire for end-to-end speech recognition](#). In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6079–6083.
- Qian Dong, Yaoming Zhu, Mingxuan Wang, and Lei Li. 2022. [Learning when to translate for streaming speech](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 680–694. Dublin, Ireland. Association for Computational Linguistics.
- Maha Elbayad, Laurent Besacier, and Jakob Verbeek. 2020. [Efficient Wait-k Models for Simultaneous Machine Translation](#). In *Proc. Interspeech 2020*, pages 1461–1465.
- Jakob Foerster, Ioannis Alexandros Assael, Nando de Freitas, and Shimon Whiteson. 2016. [Learning to communicate with deep multi-agent reinforcement learning](#). In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.
- Tzeviya Sylvia Fuchs, Yedid Hoshen, and Joseph Keshet. 2022. [Unsupervised word segmentation using k nearest neighbors](#). *arXiv preprint arXiv:2204.13094*.
- Christian Fügen, Alex Waibel, and Muntsin Kolss. 2007. [Simultaneous translation of lectures and speeches](#). *Machine Translation*, 21(4):209–252.

- Jiatao Gu, Graham Neubig, Kyunghyun Cho, and Victor O.K. Li. 2017. [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, Valencia, Spain. Association for Computational Linguistics.
- Shoutao Guo, Shaolei Zhang, and Yang Feng. 2022. [Turning fixed to adaptive: Integrating post-evaluation into simultaneous machine translation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2264–2278, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Shoutao Guo, Shaolei Zhang, and Yang Feng. 2023. [Learning optimal policy for simultaneous machine translation via binary search](#). In *Proceedings of the 61th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.
- Olivier Hamon, Christian Fügen, Djamel Mostefa, Victoria Arranz, Muntin Kolss, Alex Waibel, and Khalid Choukri. 2009. [End-to-end evaluation in simultaneous translation](#). In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 345–353, Athens, Greece. Association for Computational Linguistics.
- Javier Iranzo Sanchez, Jorge Civera, and Alfons Juan-Císcar. 2022. [From simultaneous to streaming machine translation by leveraging streaming history](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6972–6985, Dublin, Ireland. Association for Computational Linguistics.
- Herman Kamper, Aren Jansen, and Sharon Goldwater. 2017a. [A segmental framework for fully-supervised large-vocabulary speech recognition](#). *Computer Speech & Language*, 46:154–174.
- Herman Kamper, Karen Livescu, and Sharon Goldwater. 2017b. [An embedded segmental k-means model for unsupervised segmentation and clustering of speech](#). In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 719–726.
- Herman Kamper and Benjamin van Niekerk. 2020. [Towards unsupervised phone and word segmentation using self-supervised vector-quantized neural networks](#). *arXiv e-prints*, page arXiv:2012.07551.
- Taku Kudo and John Richardson. 2018. [SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.
- Ludwig Kürzinger, Dominik Winkelbauer, Lujun Li, Tobias Watzel, and Gerhard Rigoll. 2020. [Ctc-segmentation of large corpora for german end-to-end speech recognition](#). In *Speech and Computer*, pages 267–278, Cham. Springer International Publishing.
- Phuc H. Le-Khac, Graham Healy, and Alan F. Smeaton. 2020. [Contrastive representation learning: A framework and review](#). *IEEE Access*, 8:193907–193934.
- Chengdong Liang, Menglong Xu, and Xiao-Lei Zhang. 2021. [Transformer-based end-to-end speech recognition with residual gaussian-based self-attention](#). *arXiv preprint arXiv:2103.15722*.
- Yuchen Liu, Junnan Zhu, Jiajun Zhang, and Chengqing Zong. 2020. [Bridging the modality gap for speech-to-text translation](#). *arXiv preprint arXiv:2010.14920*.
- Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. [Effective approaches to attention-based neural machine translation](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.
- Mingbo Ma, Liang Huang, Hao Xiong, Renjie Zheng, Kaibo Liu, Baigong Zheng, Chuanqiang Zhang, Zhongjun He, Hairong Liu, Xing Li, Hua Wu, and Haifeng Wang. 2019. [STACL: Simultaneous translation with implicit anticipation and controllable latency using prefix-to-prefix framework](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3025–3036, Florence, Italy. Association for Computational Linguistics.
- Xutai Ma, Mohammad Javad Dousti, Changhan Wang, Jiatao Gu, and Juan Pino. 2020a. [SIMULEVAL: An evaluation toolkit for simultaneous translation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 144–150, Online. Association for Computational Linguistics.
- 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, Suzhou, China. Association for Computational Linguistics.
- Xutai Ma, Juan Miguel Pino, James Cross, Liezl Puzon, and Jiatao Gu. 2020c. [Monotonic multihead attention](#). In *International Conference on Learning Representations*.
- Xutai Ma, Yongqiang Wang, Mohammad Javad Dousti, Philipp Koehn, and Juan Pino. 2021. [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.

- Ha Nguyen, Fethi Bougares, Natalia Tomashenko, Yannick Estève, and Laurent Besacier. 2020. [Investigating self-supervised pre-training for end-to-end speech translation](#). In *ICML 2020 Workshop on Self-supervision in Audio and Speech*.
- Ha Nguyen, Yannick Estève, and Laurent Besacier. 2021. [An empirical study of end-to-end simultaneous speech translation decoding strategies](#). In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7528–7532.
- Yusuke Oda, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. 2014. [Optimizing segmentation strategies for simultaneous speech translation](#). In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 551–556, Baltimore, Maryland. Association for Computational Linguistics.
- 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, Minneapolis, Minnesota. Association for Computational Linguistics.
- B. Petek, O. Andersen, and P. Dalgaard. 1996. [On the robust automatic segmentation of spontaneous speech](#). In *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP '96*, volume 2, pages 913–916 vol.2.
- Mark A. Pitt, Keith Johnson, Elizabeth Hume, Scott Kiesling, and William Raymond. 2005. [The buckeye corpus of conversational speech: labeling conventions and a test of transcriber reliability](#). *Speech Communication*, 45(1):89–95.
- Maja Popović. 2015. [chrF: character n-gram F-score for automatic MT evaluation](#). In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.
- Maja Popović. 2017. [chrF++: words helping character n-grams](#). In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.
- Matt Post. 2018. [A call for clarity in reporting BLEU scores](#). In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.
- Colin Raffel, Minh-Thang Luong, Peter J. Liu, Ron J. Weiss, and Douglas Eck. 2017. [Online and linear-time attention by enforcing monotonic alignments](#). In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2837–2846. PMLR.
- Vivek Kumar Rangarajan Sridhar, John Chen, Srinivas Bangalore, Andrej Ljolje, and Rathinavelu Chengalvarayan. 2013. [Segmentation strategies for streaming speech translation](#). In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 230–238, Atlanta, Georgia. Association for Computational Linguistics.
- Okko Räsänen, Unto Laine, and Toomas Altoosaar. 2009. [An improved speech segmentation quality measure: the r-value](#). In *10th Interspeech Conference, Brighton, UK, September 6-10, 2009*.
- 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 *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3787–3796, Online. Association for Computational Linguistics.
- Alaa Ehab Sakran, Sherif Mahdy Abdou, Salah Eldeen Hamid, and Mohsen Rashwan. 2017. [A review: Automatic speech segmentation](#). *International Journal of Computer Science and Mobile Computing*, 6(4):308–315.
- Ruslan Salakhutdinov and Geoffrey Hinton. 2009. [Semantic hashing](#). *International Journal of Approximate Reasoning*, 50(7):969–978. Special Section on Graphical Models and Information Retrieval.
- Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. [A study of translation edit rate with targeted human annotation](#). In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.
- Kihyuk Sohn. 2016. [Improved deep metric learning with multi-class n-pair loss objective](#). In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.
- Yun Tang, Juan Pino, Xian Li, Chaghan Wang, and Dmitriy Genzel. 2021a. [Improving speech translation by understanding and learning from the auxiliary text translation task](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4252–4261, Online. Association for Computational Linguistics.
- Yun Tang, Juan Pino, Chaghan Wang, Xutai Ma, and Dmitriy Genzel. 2021b. [A general multi-task learning framework to leverage text data for speech to](#)

- text tasks. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6209–6213.
- Cassia Valentini-Botinhao and Simon King. 2021. **Detection and analysis of attention errors in sequence-to-sequence text-to-speech**. In *Interspeech 2021: The 22nd Annual Conference of the International Speech Communication Association*, pages 2746–2750. ISCA.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. **Attention is all you need**. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.
- Jesse Vig and Yonatan Belinkov. 2019. **Analyzing the structure of attention in a transformer language model**. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 63–76, Florence, Italy. Association for Computational Linguistics.
- Changhan Wang, Yun Tang, Xutai Ma, Anne Wu, Dmytro Okhonko, and Juan Pino. 2020. **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, Suzhou, China. Association for Computational Linguistics.
- Feng Wang and Huaping Liu. 2021. **Understanding the behaviour of contrastive loss**. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2495–2504.
- Ron J. Weiss, Jan Chorowski, Navdeep Jaitly, Yonghui Wu, and Zhifeng Chen. 2017. **Sequence-to-sequence models can directly translate foreign speech**. In *Proc. Interspeech 2017*, pages 2625–2629.
- Xueqing Wu, Lewen Wang, Yingce Xia, Weiqing Liu, Lijun Wu, Shufang Xie, Tao Qin, and Tie-Yan Liu. 2021. **Temporally correlated task scheduling for sequence learning**. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11274–11284. PMLR.
- Baosong Yang, Zhaopeng Tu, Derek F. Wong, Fandong Meng, Lidia S. Chao, and Tong Zhang. 2018. **Modeling localness for self-attention networks**. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4449–4458, Brussels, Belgium. Association for Computational Linguistics.
- Shan Yang, Heng Lu, Shiyin Kang, Liumeng Xue, Jinba Xiao, Dan Su, Lei Xie, and Dong Yu. 2020. **On the localness modeling for the self-attention based end-to-end speech synthesis**. *Neural Networks*, 125:121–130.
- Mahsa Yarmohammadi, Vivek Kumar Rangarajan Sridhar, Srinivas Bangalore, and Baskaran Sankaran. 2013. **Incremental segmentation and decoding strategies for simultaneous translation**. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 1032–1036, Nagoya, Japan. Asian Federation of Natural Language Processing.
- Rong Ye, Mingxuan Wang, and Lei Li. 2022. **Cross-modal contrastive learning for speech translation**. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5099–5113, Seattle, United States. Association for Computational Linguistics.
- Mohd Abbas Zaidi, Beomseok Lee, Sangha Kim, and Chanwoo Kim. 2022. **Cross-Modal Decision Regularization for Simultaneous Speech Translation**. In *Proc. Interspeech 2022*, pages 116–120.
- Xingshan Zeng, Liangyou Li, and Qun Liu. 2021. **Real-TranS: 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, Online. Association for Computational Linguistics.
- Ruiqing Zhang, Zhongjun He, Hua Wu, and Haifeng Wang. 2022a. **Learning adaptive segmentation policy for end-to-end simultaneous translation**. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7862–7874, Dublin, Ireland. Association for Computational Linguistics.
- Shaolei Zhang and Yang Feng. 2021a. **ICT’s system for AutoSimTrans 2021: Robust char-level simultaneous translation**. In *Proceedings of the Second Workshop on Automatic Simultaneous Translation*, pages 1–11, Online. Association for Computational Linguistics.
- Shaolei Zhang and Yang Feng. 2021b. **Modeling concentrated cross-attention for neural machine translation with Gaussian mixture model**. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1401–1411, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Shaolei Zhang and Yang Feng. 2021c. **Universal simultaneous machine translation with mixture-of-experts wait-k policy**. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7306–7317, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Shaolei Zhang and Yang Feng. 2022a. **Gaussian multi-head attention for simultaneous machine translation**. In *Findings of the Association for Computational*

- Linguistics: ACL 2022*, pages 3019–3030, Dublin, Ireland. Association for Computational Linguistics.
- Shaolei Zhang and Yang Feng. 2022b. [Information-transport-based policy for simultaneous translation](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 992–1013, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Shaolei Zhang and Yang Feng. 2022c. [Modeling dual read/write paths for simultaneous machine translation](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2461–2477, Dublin, Ireland. Association for Computational Linguistics.
- Shaolei Zhang and Yang Feng. 2022d. [Reducing position bias in simultaneous machine translation with length-aware framework](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6775–6788, Dublin, Ireland. Association for Computational Linguistics.
- Shaolei Zhang and Yang Feng. 2023. [Hidden markov transformer for simultaneous machine translation](#). In *The Eleventh International Conference on Learning Representations*.
- Shaolei Zhang, Yang Feng, and Liangyou Li. 2021. [Future-guided incremental transformer for simultaneous translation](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14428–14436.
- Shaolei Zhang, Shoutao Guo, and Yang Feng. 2022b. [Wait-info policy: Balancing source and target at information level for simultaneous machine translation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2249–2263, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Yibin Zheng, Xinhui Li, Fenglong Xie, and Li Lu. 2020. [Improving end-to-end speech synthesis with local recurrent neural network enhanced transformer](#). In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6734–6738.

A Dynamic Programming for Expected Feature-to-Segment Mapping

In Sec.3.2, we propose the expected feature-to-segment map to get the expected segment representations while keeping segment representations differentiable to the segmentation. Expected feature-to-segment map calculates the probability $p(a_i \in \text{Seg}_k)$ that the speech feature a_i belongs to the k^{th} segment Seg_k , and then gets the expected segment representation by weighting all speech features a_i with $p(a_i \in \text{Seg}_k)$.

Given segmentation probability p_i , we calculate $p(a_i \in \text{Seg}_k)$ via dynamic programming. Whether a_i belongs to the k^{th} segment depends on which segment that speech feature a_{i-1} is located in, consisting of 3 situations:

- $a_{i-1} \in \text{Seg}_{k-1}$: If DiSeg segments at feature a_{i-1} (with probability p_{i-1}), then a_i belongs to Seg_k ;
- $a_{i-1} \in \text{Seg}_k$: If DiSeg does not segment at feature a_{i-1} (with probability $1 - p_{i-1}$), then a_i belongs to Seg_k ;
- Others: a_i can not belong to Seg_k anyway, because the feature-to-segment mapping must be monotonic.

By combining these situations, $p(a_i \in \text{Seg}_k)$ is calculated as:

$$p(a_i \in \text{Seg}_k) = p(a_{i-1} \in \text{Seg}_{k-1}) \times p_{i-1} + p(a_{i-1} \in \text{Seg}_k) \times (1 - p_{i-1}). \quad (16)$$

For the initialization, $p(a_1 \in \text{Seg}_k)$ is calculated as:

$$p(a_1 \in \text{Seg}_k) = \begin{cases} 1 & \text{if } k = 1 \\ 0 & \text{if } k \neq 1 \end{cases}, \quad (17)$$

where the first feature inevitably belongs to the first segment. Since we constrained the number of segments to be K during training, we truncate $p(a_i \in \text{Seg}_k)$ at $p(a_i \in \text{Seg}_K)$.

B Metrics of Word Segmentation Task

In Sec.5.3, we evaluate the segmentation quality of DiSeg on the automatic speech segmentation task (Demuynck and Laureys, 2002; Sakran et al., 2017), and here we give the specific calculation of metrics for the speech segmentation task.

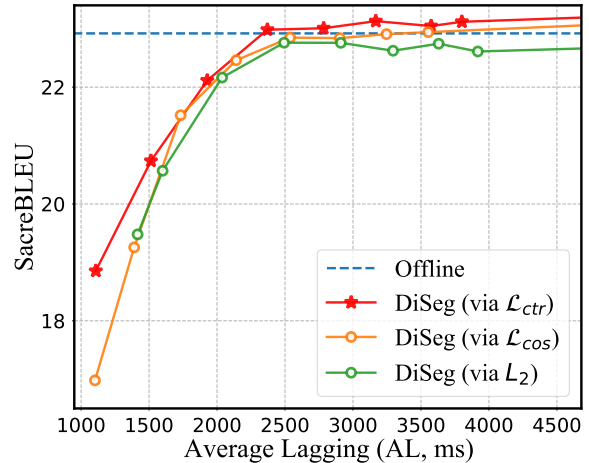


Figure 8: Comparison of semantic supervision with contrastive learning loss \mathcal{L}_{ctr} and L_2 loss.

Precision (P), recall (R) and the corresponding F1 score are used to measure whether the segmentation position is correct compared with the ground-truth segmentation. Over-segmentation (OS) (Petek et al., 1996) measures whether the number of segments generated by the model is accurate, calculated as:

$$\text{OS} = \frac{R}{P} - 1, \quad (18)$$

where $\text{OS} = 0$ means that the number of segments is completely accurate, a larger OS means more segments, and a smaller OS means fewer segments. Since a large number of segments is easy to obtain a high recall score while a poor OS score, a more robust metric R-value (Räsänen et al., 2009) is proposed to measure recall score and OS score together. R-value is calculated as:

$$\text{R-value} = 1 - \frac{|r_1| + |r_2|}{2}, \quad (19)$$

$$\text{where } r_1 = \sqrt{(1 - R)^2 + \text{OS}^2}, \quad (20)$$

$$r_2 = \frac{-\text{OS} + R - 1}{\sqrt{2}}. \quad (21)$$

A larger R-value indicates better segmentation quality, and the only way to achieve a perfect R-value is to get a perfect recall score (i.e., $R = 1$) and a perfect OS score (i.e., $\text{OS} = 0$).

C Extended Analyses

C.1 Effectiveness of Contrastive Learning

During expectation training, we use the word representations to supervise the segment representations via contrastive learning \mathcal{L}_{ctr} (Sohn, 2016). To

verify its effect, we compared the performance of applying contrastive learning loss \mathcal{L}_{ctr} and some other loss functions for semantic supervision in Figure 8, including

- \mathcal{L}_{ctr} : reduce the cosine similarity between the expected segment representation f_k^s and the corresponding word representation f_k^t and meanwhile separates f_k^s with the rest of the word representations;
- \mathcal{L}_{cos} : reduce the cosine similarity between the expected segment representation f_k^s and the corresponding word representation f_k^t .
- L_2 : reduce the L_2 distance between the expected segment representation f_k^s and the corresponding word representation f_k^t .

The results show that the cosine similarity \mathcal{L}_{cos} is better than the L_2 distance to measure the difference between the segment representation and the word representation (Le-Khac et al., 2020; Ye et al., 2022), and the contrastive learning \mathcal{L}_{ctr} further improves DiSeg performance by introducing negative examples. In particular, since \mathcal{L}_{cos} and L_2 loss fails to separate the representation of the segment and non-corresponding words, it is easy to cause the segment corresponds to more words or fewer words, which can still reduce \mathcal{L}_{cos} or L_2 loss but is not conducive to learning the precise segmentation. By making positive pairs (i.e., segment and the corresponding word) attracted and negative pairs (i.e., segment and those non-corresponding words) separated (Sohn, 2016), contrastive learning \mathcal{L}_{ctr} can learn more precise segmentation boundaries and thereby achieve better performance.

C.2 Visualization of Segmented Attention

We visualize the proposed segmented attention, the previous uni-directional attention and bi-directional attention in Figure 9, 10 and 11, including the speech with various lengths.

Comprehensive Compared with bi-directional attention, uni-directional attention obviously loses some information from the subsequent speech features (Zhang et al., 2021; Zhang and Feng, 2022d; Iranzo Sanchez et al., 2022). Segmented attention applies bidirectional attention within a segment and thereby can get a more comprehensive representation. In Figure 9, compared with uni-directional attention, segmented attention is more similar with bi-directional attention in attention distribution.

Precise DiSeg learns the translation-beneficial segmentation through the proposed expected segmented attention (refer to Eq.(7)), which encourages the model to segment the inputs at the feature a_i if a_i does not need to pay attention to subsequent features. As shown in Figure 9(b), 10(b) and 11(b), DiSeg can learn precise segmentation and ensure the acoustic integrity in each segment. In particular, the segmentation shown in Figure 9(b) and 10(b) almost guarantees that each speech segment corresponds to a word in the transcription. Note that the last segment in Figure 9(b) and 10(b) often corresponds to silence at the end of speech. Since DiSeg learns segmentation without labeled segmentation/alignment data, the proposed segmented attention can be applied to more streaming tasks.

Concentrate The issue of attention dispersion caused by long speech is one of the major challenges for speech modeling (Yang et al., 2020; Liang et al., 2021; Valentini-Botinhao and King, 2021; Zheng et al., 2020). As shown in Figure 11(a) and 11(c), both uni-directional and bi-directional attention tends to become scattered when dealing with long speech, and each feature can only get a very small amount of attention weight (e.g., the maximum attention weight in Figure 11(c) is 0.01.), which affects the modeling capability of the attention mechanism (Vig and Belinkov, 2019; Ding et al., 2019; Valentini-Botinhao and King, 2021). Segmented attention applies bi-directional attention within a segment and uni-directional attention between segments, which naturally introduces locality to attention, thereby effectively mitigating the issue of attention dispersion (Luong et al., 2015; Yang et al., 2018; Liang et al., 2021; Zhang and Feng, 2021b; Zheng et al., 2020). Specifically, as shown in Figure 10(b), segmented attention can be concentrated in the segment and pay more attention to the surrounding features. As shown in Figure 11(b), although the sequence of speech features is extremely long, segmented attention also can focus on the features in each segment (e.g., a clear attention distribution can be found inside each segment in Figure 11(b), and the maximum attention weight is 0.47.). Therefore, segmented attention provides a solution to enhance locality in long speech modeling.

C.3 Case Study

We visualize the simultaneous translation process of DiSeg on simple and hard cases in Figure 12.

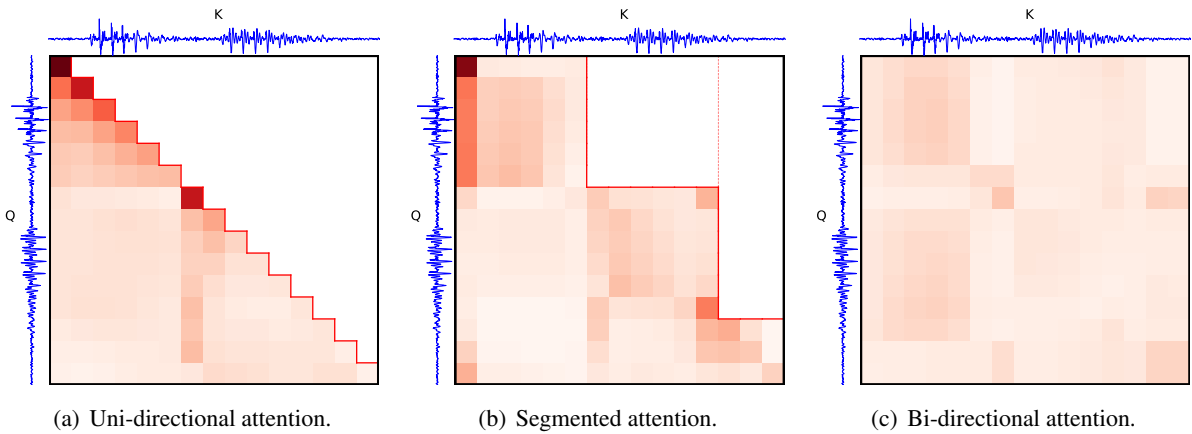


Figure 9: Visualization of segmented attention and uni-/bi-directional attention on short speech Ted_1160_70. Duration: 0.31s; Transcription: ‘Thank you.’. Three speech segments in the segmented attention correspond to ‘Thank’, ‘you’ and silence in the transcription respectively. The color shade indicates the attention weight, and the blank area indicates that the attention is masked out.

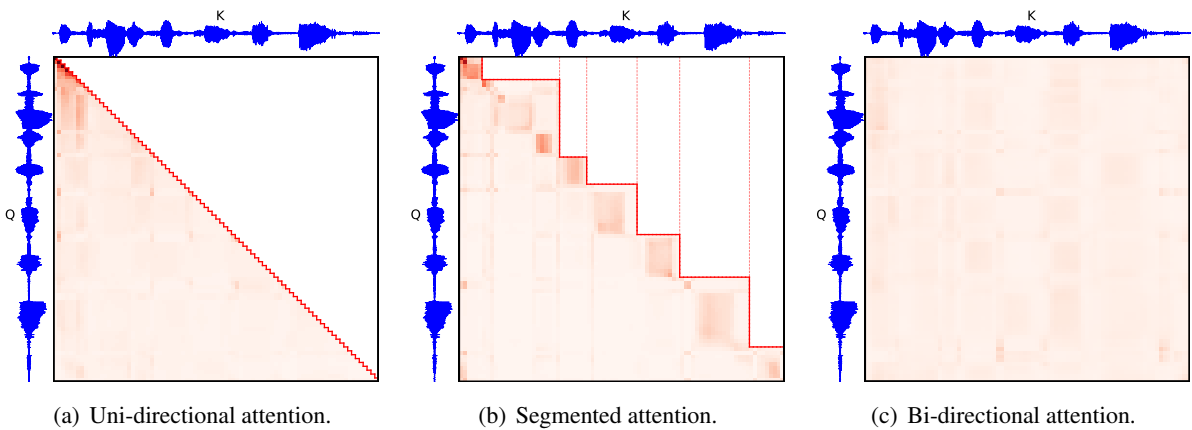


Figure 10: Visualization of segmented attention and uni-/bi-directional attention on medium length speech Ted_1171_11. Duration: 1.69s; Transcription: ‘That’s about a 15-foot boat.’. Seven speech segments in the segmented attention correspond to ‘That’s’, ‘about’, ‘a’, ‘15’, ‘foot’, ‘boat’, and silence in the transcription respectively. The color shade indicates the attention weight, and the blank area indicates that the attention is masked out.

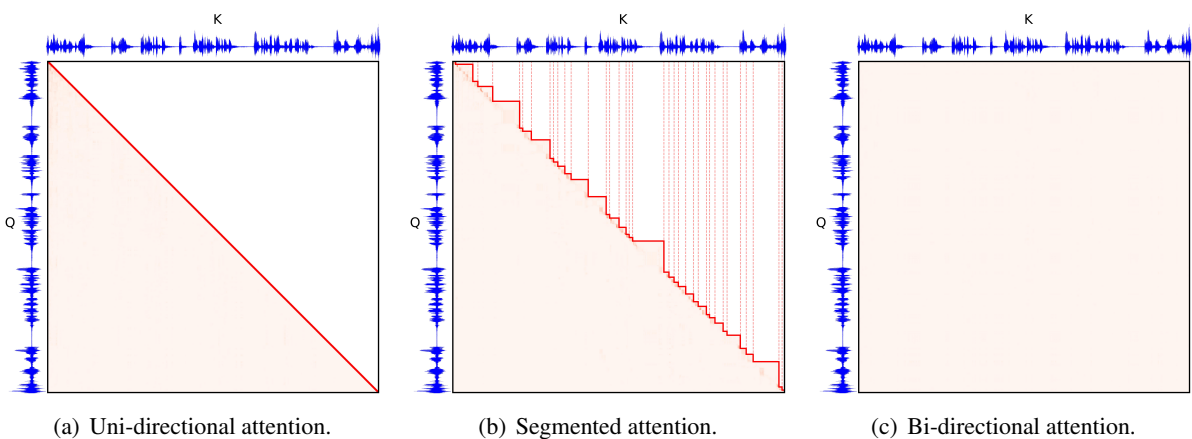
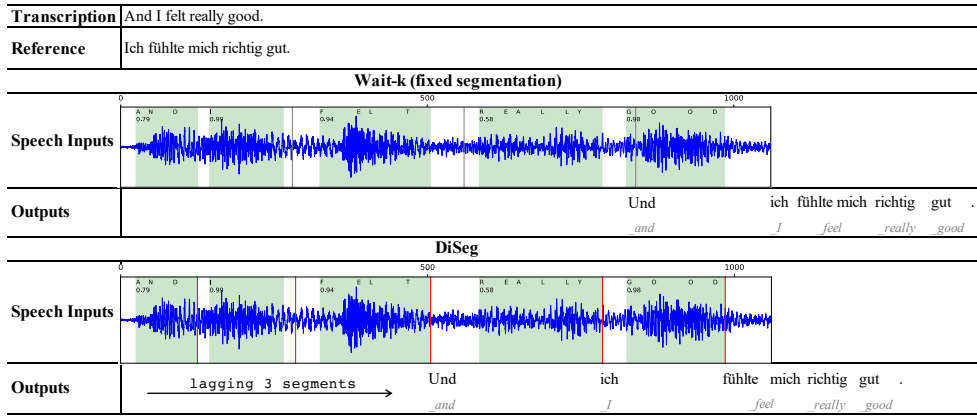
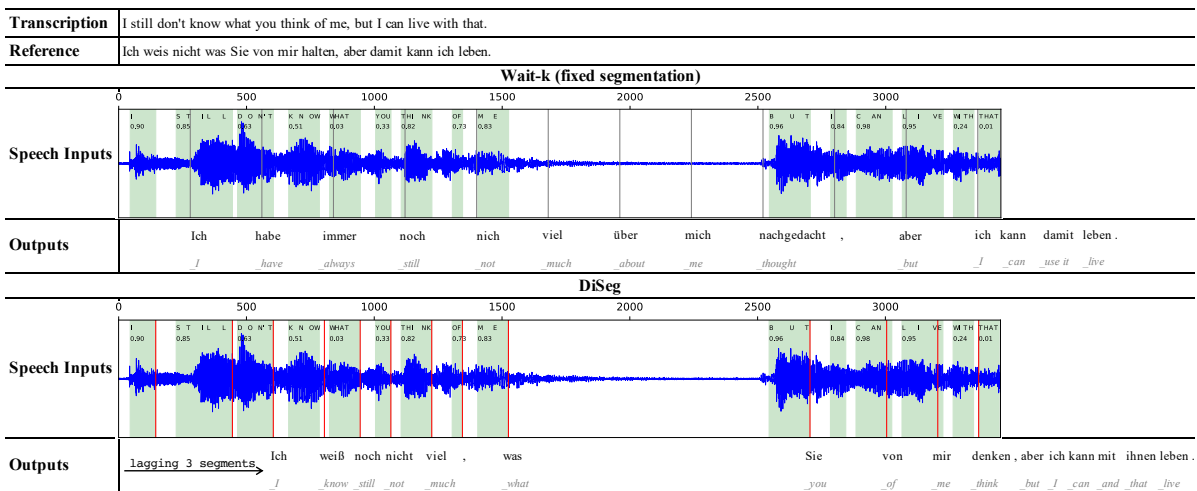


Figure 11: Visualization of segmented attention and uni-/bi-directional attention on extremely long speech Ted_1104_16. Duration: 17.09s; Transcription: ‘Let me now introduce you to eLEGS that is worn by Amanda Boxtel that 19 years ago was spinal cord injured, and as a result of that she has not been able to walk for 19 years until now.’. The color shade indicates the attention weight, and blank area indicates that the attention is masked out.



(a) Case Ted_1337_63 on MuST-C En→De. We show the results of DiSeg and Wait-k when both lagging 3 segments, where the latency (AL) of DiSeg and Wait-k are 733ms and 1060ms respectively.



(b) Case Ted_1337_18 on MuST-C En→De. We show the results of DiSeg and Wait-k under the same latency, i.e., $AL \approx 750ms$.

Figure 12: Case study of DiSeg. The horizontal direction indicates when the model outputs the target word with the streaming speech inputs. Red lines indicate where DiSeg decides to segment the speech inputs, and gray lines indicate the equal-length segmentation of 280ms. The green areas indicate the alignments between the transcription and the speech generated by an offline alignment tool.

Horizontally, the position of the word in outputs is the moment when it is translated, corresponding to the speech inputs. Red lines indicate where DiSeg decides to segment the speech inputs, and gray lines indicate the fixed segmentation of 280ms. For clarity, we use an external alignment tool Forced-Alignment⁹ (Kürzinger et al., 2020) to align the transcription with the speech, where the green area is the speech interval corresponding to the transcription marked by the tool. Note that the alignment provided by external tools is only a rough reference, not necessarily absolutely accurate, and the value in the marked interval represents

⁹A CTC-based alignment tool based on the full speech and ground-truth transcription, the tutorial of which can be found at https://pytorch.org/audio/main/tutorials/forced_alignment_tutorial.html

the probability of alignment.

For the simple case in Figure 12(a), where the speech is short and the correspondence between reference and speech (transcription) is much monotonic, DiSeg can basically accurately segment the speech inputs and achieve high-quality translation. In particular, when both lagging 3 segments, DiSeg achieves much lower latency than Wait-k due to more precise segmentation.

For the hard case in Figure 12(b), where the speech is much longer and contains a long silence, DiSeg can also precisely segment the speech inputs. Besides, there is an obvious word order difference (Zhang and Feng, 2022a) between reference and speech (transcription) in this case, which is more challenging for SimulST (Ma et al., 2019). Since

the fixed segmentation cannot adjust, Wait-k misses translating ‘know’. DiSeg can dynamically adjust the segmentation, and thereby decides to segment and translate ‘weiß’ after receiving ‘know’ in the speech. Owing to precise segmentation, DiSeg can achieve better translation quality under the same latency.

D Numerical Results

D.1 Metrics

For latency, besides Average Lagging (AL) (Ma et al., 2019), we additionally use Consecutive Wait (CW) (Gu et al., 2017), Average Proportion (AP) (Cho and Esipova, 2016) and Differentiable Average Lagging (DAL) (Arivazhagan et al., 2019) to evaluate the latency of DiSeg. Assuming that DiSeg translates y_t at the moment $\mathcal{T}(y_t)$, the calculations of latency metrics are as follows.

Consecutive Wait (Gu et al., 2017) CW evaluates the average waiting duration between two adjacent outputs, calculated as:

$$CW = \frac{\sum_{t=1}^{|\mathbf{y}|} (\mathcal{T}(y_t) - \mathcal{T}(y_{t-1}))}{\sum_{t=1}^{|\mathbf{y}|} \mathbb{1}_{\mathcal{T}(y_t) - \mathcal{T}(y_{t-1}) > 0}}, \quad (22)$$

where $\mathbb{1}_{\mathcal{T}(y_t) - \mathcal{T}(y_{t-1}) > 0}$ counts the number of $\mathcal{T}(y_t) - \mathcal{T}(y_{t-1}) > 0$.

Average Proportion (Cho and Esipova, 2016) AP evaluates the average proportion between $\mathcal{T}(y_t)$ and the total duration T of the complete source speech, calculated as:

$$AP = \frac{1}{|\mathbf{y}|} \sum_{t=1}^{|\mathbf{y}|} \frac{\mathcal{T}(y_t)}{T}. \quad (23)$$

Average Lagging (Ma et al., 2019, 2020b) AL evaluates the average duration that target outputs lag behind the speech inputs, is calculated as:

$$AL = \frac{1}{\tau} \sum_{t=1}^{\tau} \mathcal{T}(y_t) - \frac{t-1}{|\mathbf{y}|/T}, \quad (24)$$

$$\text{where } \tau = \underset{t}{\operatorname{argmin}} (\mathcal{T}(y_t) = T). \quad (25)$$

Differentiable Average Lagging (Arivazhagan et al., 2019) DAL is a differentiable version of average lagging, calculated as:

$$DAL = \frac{1}{|\mathbf{y}|} \sum_{t=1}^{|\mathbf{y}|} \mathcal{T}'(y_t) - \frac{t-1}{|\mathbf{y}|/T}, \quad (26)$$

where

$$\mathcal{T}'(y_t) = \begin{cases} \mathcal{T}(y_t) & t = 1 \\ \max\left(\mathcal{T}(y_t), \mathcal{T}(y_{t-1}) + \frac{T}{|\mathbf{y}|}\right) & t > 1 \end{cases}. \quad (27)$$

For translation quality, in addition to SacreBLEU (Post, 2018), we also provide TER (Snover et al., 2006), chrF (Popović, 2015) and chrF++ (Popović, 2017) score of DiSeg.

D.2 Numerical Results

The numerical results of DiSeg with more metrics are reported in Table 4 and Table 5.

En→De								
<i>k</i>	CW	AP	AL	DAL	BLEU	TER	chrF	chrF++
1	462	0.67	1102	1518	18.85	73.13	44.29	42.31
3	553	0.76	1514	1967	20.74	69.95	49.34	47.09
5	666	0.82	1928	2338	22.11	66.90	50.13	47.94
7	850	0.86	2370	2732	22.98	65.42	50.36	48.23
9	1084	0.90	2785	3115	23.01	65.48	50.24	48.13
11	1354	0.92	3168	3464	23.13	65.04	50.42	48.31
13	1632	0.94	3575	3846	23.05	64.85	50.53	48.41
15	1935	0.96	3801	4040	23.12	64.92	50.47	48.36

Table 4: Numerical results of DiSeg on MuST-C En→De.

En→Es								
<i>k</i>	CW	AP	AL	DAL	BLEU	TER	chrF	chrF++
1	530	0.67	1144	1625	22.03	71.34	45.69	43.82
3	563	0.76	1504	2107	24.49	66.63	53.09	50.85
5	632	0.81	1810	2364	26.58	63.35	54.55	52.39
7	788	0.85	2249	2764	27.81	61.87	55.28	53.16
9	1010	0.89	2694	3164	28.33	60.98	55.51	53.40
11	1257	0.92	3108	3530	28.59	60.63	55.64	53.55
13	1534	0.94	3479	3855	28.72	60.49	55.61	53.53
15	1835	0.95	3819	4160	28.92	60.22	55.80	53.71

Table 5: Numerical results of DiSeg on MuST-C En→Es.

ACL 2023 Responsible NLP Checklist

A For every submission:

- A1. Did you describe the limitations of your work?
In Limitations Section.
- A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.
- A3. Do the abstract and introduction summarize the paper's main claims?
In Section 1.
- A4. Have you used AI writing assistants when working on this paper?
Left blank.

B Did you use or create scientific artifacts?

Left blank.

- B1. Did you cite the creators of artifacts you used?
In Section 4 and 5.
- B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.
- B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.
- B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.
- B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.
- B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
In Section 4.

C Did you run computational experiments?

In Section 4 and 5.

- C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
In Section 4.2.

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

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

In Appendix D.

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

Not applicable. Left blank.

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

In Section 4.2.

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

Left blank.

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

No response.

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

No response.

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

No response.

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

No response.

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

No response.