

# Recent Advances in End-to-End Simultaneous Speech Translation

Xiaoqian Liu<sup>1</sup>, Guoqiang Hu<sup>2</sup>, Yangfan Du<sup>1</sup>, Erfeng He<sup>1</sup>,  
YingFeng Luo<sup>1</sup>, Chen Xu<sup>3</sup>, Tong Xiao<sup>1,4</sup> and Jingbo Zhu<sup>1,4\*</sup>

<sup>1</sup>School of Computer Science and Engineering, Northeastern University, Shenyang, China

<sup>2</sup>International School, Jinan University, Guangzhou, China

<sup>3</sup>College of Computer Science and Technology, Harbin Engineering University, Harbin, China

<sup>4</sup>NiuTrans Research, Shenyang, China

{liuxiaoqian0319, luoyf98, xuchennlp, giannis-huguogiang}@outlook.com,  
{dduyangfan, heerfeng1023}@gmail.com, {xiaotong, zhujingbo}@mail.neu.edu.cn

## Abstract

Simultaneous speech translation (SimulST) is a demanding task that involves generating translations in real-time while continuously processing speech input. This paper offers a comprehensive overview of the recent developments in SimulST research, focusing on four major challenges. Firstly, the complexities associated with processing lengthy and continuous speech streams pose significant hurdles. Secondly, satisfying real-time requirements presents inherent difficulties due to the need for immediate translation output. Thirdly, striking a balance between translation quality and latency constraints remains a critical challenge. Finally, the scarcity of annotated data adds another layer of complexity to the task. Through our exploration of these challenges and the proposed solutions, we aim to provide valuable insights into the current landscape of SimulST research and suggest promising directions for future exploration.

## 1 Introduction

End-to-end simultaneous speech translation (SimulST) is a task of generating translation text in the target language while receiving speech input in the source language [Ma *et al.*, 2020b]. This process of directly processing input and providing translation is seamless and continuous, giving a faster and more natural translation experience. It is especially beneficial for real-time conversations, voice conferencing, and other scenarios that require fast and smooth communication, and therefore has received widespread attention and progress in recent years [Papi *et al.*, 2023b]. Meanwhile, since speech translation (ST) itself is already a cross-lingual and cross-modal task [Xu *et al.*, 2023], the demand for streaming generation makes it even more complex.

Figure 1 presents an overview of the SimulST model. Based on the encoder-decoder structure, the model also needs an additional segmentation module and a simultaneous read-write module for streaming inference. Giving the training



Figure 1: Overview of the SimulST model.

data as a triple  $\mathcal{D} = (S, X, Y)$ ,  $S$  denotes the acoustic features extracted from the input speech in the source language,  $X$  denotes the corresponding transcription, and  $Y$  denotes the text in the target language. Considering a segmentation strategy to obtain input units and a simul R-W policy  $g$ , we denote the number of speech features as  $g(t)$  at the translation of the  $t$ -th token  $y_t$ . The training objective using cross-entropy loss parameterized by  $\theta$  can be formalized as:

$$\mathcal{L}_\theta = -\mathbb{E}_{(\hat{s}, y_{<t})} \sum_{t=1}^T \log P(y_t | y_{<t}, \hat{s}; \theta) \quad (1)$$

where  $\hat{s}$  denotes the current received features  $[s_1 \dots s_{g(t)}]$ . The determination of  $g(t)$  is crucial, demonstrating the important role of streaming input processing and decision-making methods. Moreover, the latency cannot be directly optimized through the traditional loss  $\mathcal{L}_\theta$ , which reflects the complexity of SimulST. Hence, existing research primarily focuses on addressing the following key challenges and issues:

- **Processing long-form inputs.** SimulST demands models to possess both translation accuracy and low-latency capabilities. However, lengthy and continuous inputs

\*Corresponding author.

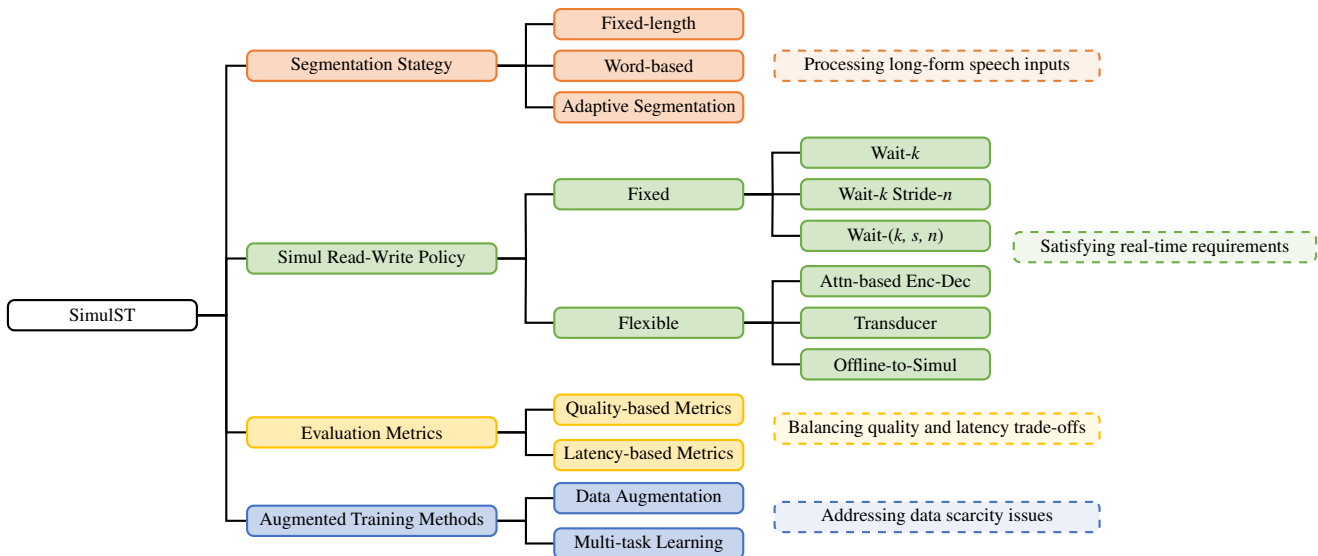


Figure 2: Key challenges to address in the task of SimulST and their corresponding solutions.

fail to meet the low-latency requirements for real-time output [Zhang and Feng, 2023].

- **Satisfying real-time requirements.** To the current input segment, the model needs to make decisions regarding whether to generate a new translation [Ma *et al.*, 2020b]. Premature output may result in incomplete information, leading to poorer translations. Conversely, delaying output may introduce high latency, thereby impacting user experience.
- **Balancing quality and latency trade-offs.** There is no single evaluation metric that can simultaneously address both quality and latency [Kano *et al.*, 2022]. Achieving a balance between quality and latency is indeed difficult, especially in the context of SimulST.
- **Addressing data scarcity issues.** Unlike related fields such as automatic speech recognition (ASR) and machine translation (MT) which have abundant training data [Ko *et al.*, 2023], SimulST suffers from a scarcity of annotated data, which exacerbates the complexity as the models are hard to be adequately trained.

These factors collectively contribute to the intricate nature of the SimulST task. Existing studies have proposed solutions to these challenges, but there is currently a lack of a comprehensive overview to thoroughly summarize their practices. We find that there are some previous related studies, in which Xu *et al.* [2023] aim at offline ST tasks, Polák [2023] faces long-form inputs, and Sethiya and Maurya [2023] summarize the whole speech-to-text technology. Our work is different from theirs since we give a more complete and comprehensive introduction to SimulST.

As shown in Figure 1 and Figure 2, we structure the paper as follows. Section 2 introduces *segmentation strategies* for (a) processing continuous speech inputs, realizing *where* to segment the input into a suitable acoustic unit. Methods can be divided into fixed-length, word-based, and adap-

tive segmentation methods. Section 3 describes the *simultaneous read-write policies* including fixed read-write methods and flexible ones to judge *when* to output according to the currently obtained units, satisfying (b) real-time requirements. Section 4 introduces the studies related to (c) quality and latency which puts forward two kinds of *metrics* to ensure a comprehensive *evaluation*. Section 5 describes studies of the *augmented training methods* including data enhancement and multi-task learning to tackle the (d) data scarcity issues. Finally, Section 6 anticipates some promising directions for *future SimulST research*, including studies in multilingual SimulST and combining them with large language models.

## 2 Segmentation Strategies

SimulST needs to read long-form streaming inputs and generate partial translation at inference time. Hence, it is crucial to select appropriate segmentation strategies to furnish the model with suitable units during inference. However, speech is a continuous sequence, and the absence of distinct boundaries poses a great challenge in achieving accurate segmentation. The segmentation strategies encompass three specific methods. In Section 2.1, we elucidate the fixed-length strategies, followed by an exposition of the word-based strategies in Section 2.2. Moving to Section 2.3, we present the adaptive segmentation strategies.

### 2.1 Fixed-length Strategies

Fixed-length strategies represent one of the simplest segmentation approaches. As illustrated in Figure 3 (a), it assumes that a certain number of speech frames (e.g. 280ms or 400 frames) equate to a fixed count of words, dividing the speech into equally-sized segments with a consistent frame length [Nguyen *et al.*, 2021a; Chen *et al.*, 2021]. This method bears resemblance to incremental encoding as discussed in [Ma *et al.*, 2019], with each segment akin to a token.

The basic SimulST model is based on vanilla Transformer [Vaswani *et al.*, 2017] in Figure 5 (a), namely a segment-based Transformer, in which the self-attention module attends to the entire sequence and limits the streaming capability of the model. Building upon this issue, Ma *et al.* [2021] introduce an augmented memory encoder to divide the input sequence into sub-utterance level segments where each overlaps with previous and subsequent ones to capture left and right contexts. By computing self-attention only within each segment, the model can handle long input sequences while significantly reducing complexity. To enhance computational efficiency, Raffel and Chen [2023] improve the model with implicit memory, which captures the representation of previous segments during encoding and implicitly preserves based on attention, thus eliminating the need for a memory bank. Due to discrepancies between training and inference in segment-based Transformer models, they subsequently propose a shiftable context method in [Raffel *et al.*, 2023] to produce consistent segment sizes for better alignment.

## 2.2 Word-based Strategies

While studies based on fixed-length segmentation methods have achieved numerous successes, making decisions on every fixed number of frames often leads to suboptimal results. This is because the segment boundaries may not align with the natural endings of pronunciations, thus disrupting acoustic integrity. To alleviate this situation, several unfixed-length strategies have been proposed. Among them, the word-based strategies expect to determine segmentation by aligning with corresponding words [Ma *et al.*, 2020b], as illustrated in Figure 3 (b).

Word-based strategies typically involve introducing additional detectors or similar modules to detect boundaries, representing a hard alignment approach. Meanwhile, Connectionist Temporal Classification (CTC) [Graves *et al.*, 2006] proves effective in detecting word boundaries and is widely used when mapping frame-level classification outputs of speech sequences to text sequences. So in [Ren *et al.*, 2020], a speech segmenter is added after the encoder to detect word boundaries and segment the input streaming speech using CTC loss. Besides, they introduce two knowledge distillation methods to ensure the performance. To relieve the burden of the encoder, Zeng *et al.* [2021] decouple it into three parts. They weight and aggregate the detected frames by a CTC module and introduce a blank penalty for non-blank labels. In addition, Nguyen *et al.* [2021b] leverage an additional toolkit with oracle word boundaries to segment input into corresponding words. In a word, these studies employ external segmentation modules, always leaving a gap between the segmentation and translation model.

## 2.3 Adaptive Segmentation Strategies

Different from previous systems that treat speech with fixed time-span as an acoustic unit or attempt to locate word boundaries, adaptive segmentation strategies depicted in Figure 3 (c) detect boundaries of proper speech units. These strategies consider more meaningful information, such as semantic consistency, or incorporate input segmentation into the model training process.

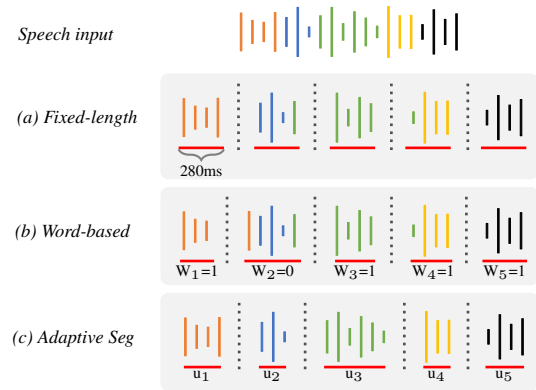


Figure 3: Segmentation strategies.

To realize adaptive segmentation, Dong *et al.* [2022] propose MoSST, introducing a Monotonic Segmentation Module (MSM) to handle streaming speech input. The MSM dynamically reads the acoustic information from the encoder and locates the boundaries of meaningful speech units instead of segmentation. Inspired by the integrate-and-fire (IF) model, it incrementally integrates input when the information is not enough; once sufficient, it enters a firing mode, during which a new token is generated.

Another implementation is based on the concept of Meaningful Units (MUs), defined as the minimum speech segments whose translation will not be altered by subsequent speech [Zhang *et al.*, 2022]. Therefore, a detection module is designed to dynamically detect MUs by comparing the translation of every speech prefix segment with the full-speech translation. Once an MU is detected, it is fed into the model for inference.

Different from adding a heuristic detector, Zhang and Feng [2023] propose Differentiable Segmentation (DiSeg), which predicts a Bernoulli variable 0/1 for each speech feature to indicate when to segment streaming speech inputs. It can be trained jointly with the SimulST model, allowing segmentation to be integrated into the translation process. Since DiSeg learns segmentation directly, it can handle simultaneous and offline speech translation with a unified model.

To summarize, as shown in Figure 1 (a), some studies introduce segmentation strategies before the encoder, while others opt for segmentation after encoding. However, regardless of their placement, these modules transform the original continuous inputs into reasonable acoustic units for SimulST.

## 3 Simultaneous Read-Write Policies

Simul R-W policies aim to identify suitable moments for generating partial sentence translations based on streaming speech units. In section 3.1, we first introduce the fixed R-W policy wait- $k$  and its variants. Subsequently, in Section 3.2, we delve into flexible R-W policies. This section is further divided into attention-based encoder-decoder models (Section 3.2.1) and transducer-based models (Section 3.2.2) based on the model architectures. With offline ST already making significant strides, an increasing number of studies are exploring

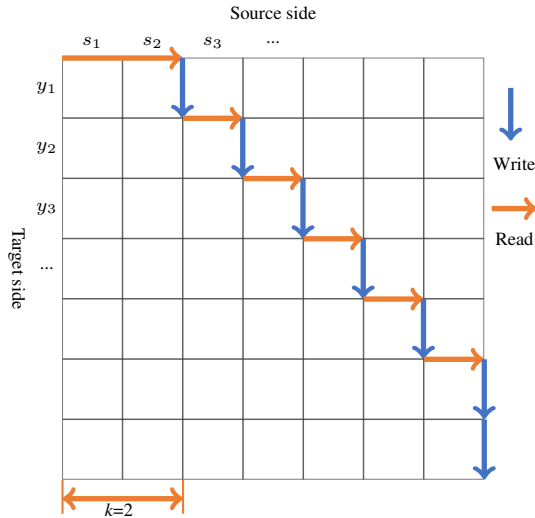


Figure 4: Wait- $k$  policy. The model first waits for  $k$  units (here  $k=2$ ) and then emits target word  $y_t$  given source units  $s_1 \dots s_{t+k-1}$ .

methods to render offline models in real-time, a topic we discuss in Section 3.2.3.

### 3.1 The Wait- $k$ Method and its Variants

In the process of SimulST, simultaneous decoding policies play a pivotal role, with wait- $k$  [Ma *et al.*, 2019] emerging as a fundamental and wide approach in the early stages of simultaneous MT tasks. The underlying concept of wait- $k$  is to momentarily defer the output of translation until the model has received the  $k$  units of the source input, to accumulate more context information and enhance translation accuracy.

As shown in Figure 4, when streaming speech input begins, the model waits for  $k$  source units and alternates back and forth between Wait( $\cdot$ ) and Read( $\cdot$ ). This policy can reduce translation errors to a certain extent, because the model waits for enough contextual information before translation, to have a better understanding [Han *et al.*, 2020]. On this basis, according to the number of read and write units per time, two variants are proposed. Zeng *et al.* [2021] propose wait- $k$  stride- $n$ , in which the model alternates between writing and reading every  $n$  units instead of *one*. Besides, Nguyen *et al.* [2021a] propose wait- $(k, s, n)$ , that is generating  $n$  units after reading  $s$  additional units. They are both wait- $k$  policies with inconsistent step sizes.

As a straightforward approach, this simple way of simul R-W policy is easy to implement with acceptable quality and has been adopted by numerous studies [Ren *et al.*, 2020; Ma *et al.*, 2021; Raffel and Chen, 2023; Raffel *et al.*, 2023]. However, as it fixed alternates in reading and writing without analyzing the currently received input units, it cannot clarify whether the present moment is suitable for output. Therefore, it is often employed in conjunction with segmentation strategies in Section 2.

### 3.2 Flexible Policies

Intuitively, fixed policies like wait- $k$  may not have sufficient information to generate tokens based on partial inputs. In

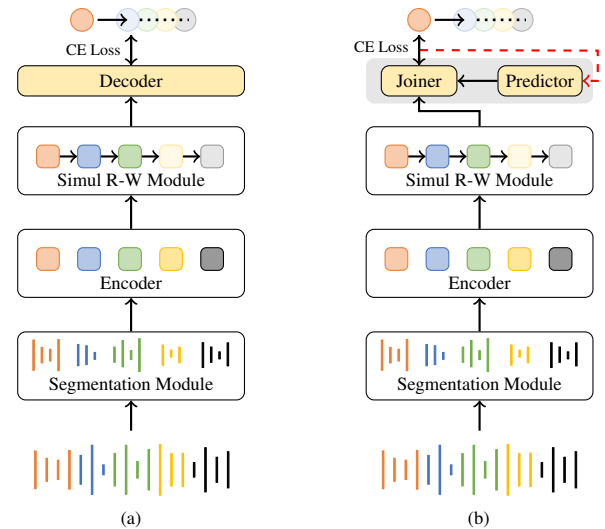


Figure 5: SimulST frameworks. (a) is attention-based encoder-decoder architecture, and (b) is for Transducer.

principle, model-based policies should be capable of adaptation by considering potential alignments between input segments and output tokens during training [Ma *et al.*, 2020b]. Two types of fundamental frameworks are depicted in Figure 5: (a) indicates the Attention-based Encoder-Decoder (AED) structure which allows the decoder to attend to a portion of the source sequence without being constrained by specific modes or sequences, while Transducer in (b) is known for its advantages of monotonic alignment capability.

#### Attention-based Encoder-Decoder Models

Building upon the advantage of being able to flexibly attend to relevant parts of the input, some studies tend to incorporate monotonic capability into AED models. Ma *et al.* [2020b] extend Monotonic Multi-head Attention (MMA) to SimulST for achieving flexible decision-making. MMA achieves flexible decision-making by assigning each head within a layer an independent step probability, which determines when to read or write during the translation process. It is more robust to the granularity of the input, and a pre-decision module is introduced to handle fine-grained input. Functionally, the pre-decision is consistent with the segmentation strategies in Section 2, aiming to achieve suitable segmentation units. Furthermore, Zaidi *et al.* [2021] propose Decision Attentive Regularization (DAR) to improve SimulST by implicitly utilizing the monotonic attention energies seen in SimulMT.

Like CTC or MMA, Continuous Integrate-and-Fire (CIF) is another monotonic alignment method proposed to learn the precise acoustic boundaries [Dong *et al.*, 2022]. Thus, Chang and Lee [2022] leverage CIF to develop a flexible policy. Specifically, they utilize a weight prediction network and establish a threshold. If the accumulated weights fall below the threshold, CIF proceeds to the next encoder step. Otherwise, it triggers the integrate and fire operation, which retains the remaining weight for the next integration and produces an integrated embedding sent to the decoder, a process referred to as firing.

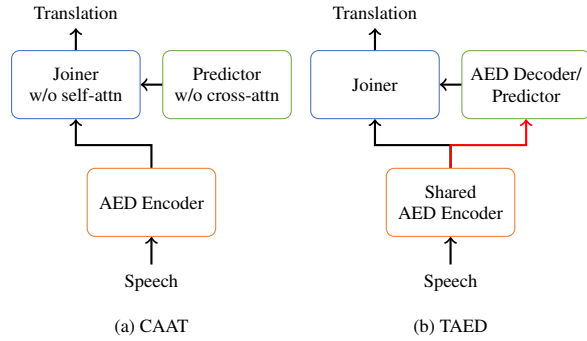


Figure 6: Structures combining AED and Transducer in 3.2. (a) is CAAT, which divides the Transformer decoder like the Transducer. (b) is TAED, providing a hybrid of the two models.

### Transducer Models

Another widely-used framework is RNN Transducer (RNN-T) [Graves, 2012]. As a variant of CTC, RNN-T divides the decoder into a *predictor* and a *joiner*, where the predictor generates historical representations, and the joiner generates output probabilities by combining the representations of the predictor and the encoder. RNN-T naturally handles the monotonic alignment between input and output sequences during streaming decoding. Following RNN-T architecture, Chen *et al.* [2023] propose a revision-controllable method to improve the decoding stability for those utilizing beam search.

Building upon RNN-T’s foundational concepts, the Cross Attention Augmented Transducer (CAAT) [Liu *et al.*, 2021] reconfigures the Transformer decoder into two distinct components: a predictor and a joiner. Both modules retain the original count of Transformer blocks; however, the predictor omits the cross-attention mechanism, and conversely, the joiner lacks a self-attention component as shown in Figure 6 (a). In this way, CAAT realizes the separation of goal and historical representation in different attention mechanisms, realizing simultaneous generation. Further evolving this landscape, Tang *et al.* [2023] introduce the TAED model, a novel hybrid of the Transducer and Attention Encoder-Decoder (AED) frameworks in Figure 6 (b). TAED utilizes a shared encoder across both paradigms while substituting the traditional Transducer predictor with an AED decoder. This strategic integration harnesses the respective advantages of AED and Transducer models, offering a unified and potent approach.

### Offline-to-Simul

Some studies focus on leveraging well-trained offline models in SimulST tasks. Regardless of R-W policies, SimulST models are typically trained only when partial input is available. However, offline models have access to entire speech during training, leading to a discrepancy when employing offline-trained models for simultaneous inference.

To incorporate simultaneous settings to achieve incremental inference with offline models, Liu *et al.* [2020] aim to trade some latency for better output quality. They propose three techniques of partial hypothesis selection, which observes the acoustic information and selectively outputs stable

segment-level hypotheses instead of all predictions. A further study [Polák *et al.*, 2023] propose an incremental block-wise beam-search (IBWBS) algorithm. IBWBS halts only the problematic beam upon detecting an unreliable hypothesis, allowing other beams to continue. This enables incremental SimulST and facilitates latency control without retraining the model.

Compared with adding simultaneous modules, Papi *et al.* [2022] question the need for simultaneous training. They adopt the wait- $k$  policy to an offline ST model only at inference time without any additional training or adaptation. It reduces the computational costs for training a SimulST from scratch without performance degradation.

Since  $k$  policy is simple, they then continue to make unremitting efforts in the process of real-time adaptation of offline models. Different from the flexible policies in Section 3.2 which need dedicated simulating training, they propose Encoder-Decoder Attention (EDATT) [Papi *et al.*, 2023a]. It guides offline ST models during simultaneous translation by leveraging the encoder-decoder attention matrix. If the attention is not focused on the most recent frames, the model determines to emit a partial translation since the received information is sufficient. Furthermore, they assume that if a candidate token is aligned with the last input frame, the information might be insufficient for emission. So they present ALIGNATT [Papi *et al.*, 2023b] to guide an offline ST model during simultaneous inference by leveraging speech-to-translation alignments computed from the attention weights.

That is to say, simul R-W policies endeavor to ascertain the timing of token generation during inference, as shown in Figure 1 (b). Consequently, a new trend is to leverage offline ST models during simultaneous inference, which can not only harness superior performance but also mitigate computational resource consumption.

## 4 Evaluation Metrics

Achieving a balance between translation quality and latency is crucial in the SimulST task, ensuring a satisfactory user experience by providing timely translations without compromising accuracy. Introducing multiple evaluation metrics is important for assessing the balance between quality and latency comprehensively. Different metrics offer diverse perspectives on translation performance, enabling a more nuanced understanding of system behavior. By considering both quality-related and latency-related metrics, researchers can make informed decisions about system optimizations.

### 4.1 Quality-based Metrics

The quality evaluation metrics utilized in SimulST are fundamentally aligned with those employed in MT, as both aim to assess the quality of the translated output. BLEU [Papineni *et al.*, 2002] is one of the most commonly used metrics for evaluating the quality of MT based on measuring the closeness of a machine translation to human reference. The score ranges from 0 to 1, with higher scores indicating better translation quality. SacreBLEU [Post, 2018] is an improved version of BLEU that offers additional features like handling tokenization issues, supporting multiple languages, and providing more robust evaluation across different datasets. Another

Domain	Language	Avg. duration(h)	Avg.SacreBLEU	Common Datasets
Audiobook	en → fr	236	19.4	LibriTrans
	de → en	100		
	en → de	53		
Lecture	de → en	37	24.4	BSTC
	zh → en	51		
Common Voice	en → {fr, de, es, ca, it, ru, zh, pt, fa, et, mn, nl, tr, ar, sv, lv, sl, ta, ja, if, cy }	568	21.88	CoVoST2
TED	en → de	272	30.4	MuST-C
	en → zh	542		
	en → {ar, cs, de, es, fa, fr, it, nl, pt, ro, ru, tr, vi, zh }	430		

Table 1: ST datasets across various domains.

recent metric is COMET [Rei *et al.*, 2020], aiming to capture not only surface-level similarities but also deeper semantic and contextual aspects of translations. It incorporates multiple sub-metrics, including fluency, adequacy, and fidelity, to offer a holistic assessment of translation quality.

## 4.2 Latency-based Metrics

The evaluation of latency metrics in SimulST serves to gauge real-time system performance. Average Lagging (AL) [Ma *et al.*, 2019] refers to the measure of average delay. It is typically calculated as the average time taken from the arrival of a speech segment to the completion of translation for that segment. In SimulST, the input features come from the speech  $S = \{s_1, \dots, s_{|S|}\}$ , in which  $s_i$  is a raw audio segment of duration  $T_i$ . Assuming  $\hat{s}$  has been read to generate  $y_t$ , the delay of  $y_t$  can be defined as  $d_t = \sum_{j=1}^{g(t)} T_j$ . For a Simul R-W policy  $g$ , it calculates the average delay from the generation of the first target token to the  $\tau(|S|)$ -th, which can be defined as:

$$AL = \frac{1}{\tau(|S|)} \sum_{t=1}^{\tau(|S|)} d_t - \frac{(t-1)}{r} \quad (2)$$

where  $\tau(|S|) = \min\{t|d_t = |S|\}$  denotes the truncation step of the policy and  $r = |Y|/|S|$  denotes the ratio between the target and source length. It can be inferred that  $(t-1)/r$  term is the ideal policy to compare.

Considering the speech duration  $T$ , Ma *et al.* [2020a] adapt AL as follows:

$$AL = \frac{1}{\tau'(|S|)} \sum_{t=1}^{\tau'(|S|)} d_t - d_t^* \quad (3)$$

where  $\tau'(|S|) = \min\{t|d_t = \sum_{j=1}^{|S|} T_j\}$  and the  $d_t^* = (t-1) \sum_{j=1}^{|S|} T_j / |Y_{ref}|$  are the delays of an ideal policy. It assumed that the ideal policy generates the reference  $Y_{ref}$  rather than the system hypothesis and is a wait-0 policy.

Differentiable Average Lagging (DAL) [Cherry and Foster, 2019] is computed similarly to AL but incorporates differentiable operations, allowing the latency metric to be optimized alongside the training process of the model.

$$DAL = \frac{1}{|Y|} \sum_{t=1}^{|Y|} d'_t - \frac{(t-1)}{r} \quad (4)$$

where

$$d'_t = \begin{cases} d_t & t = 1 \\ \max[d_t, d'_{t-1} + r] & t > 1 \end{cases} \quad (5)$$

$d'_t$  tracks duration before generating  $y_t$ , reflecting the semantics of  $d_t$ . The recursion in  $d'_t$  is differentiable and can be efficiently implemented in computational graph-based programming languages.

Besides, Ma *et al.* [2020b] introduce a computation-aware (CA) and a non-computation-aware (NCA) delay adapted from AL and Kano *et al.* [2022] propose Average Token Delay (ATD) that focuses on the end timings of partial translations in SimulST. These various metrics are all used to evaluate latency, which needs a toolkit to apply. SimulEval [Ma *et al.*, 2020a] uses a server-client scheme to imitate simultaneous scenarios, it automatically performs streaming decoding and collectively reports several popular latency metrics.

Under reasonable experimental settings, subjective evaluation often has more accurate performance. Unlike the evaluation indicators mentioned above, Continuous Rating (CR) [Macháček *et al.*, 2023] is a method for human assessment of SimulST quality. It aims to provide a comprehensive assessment of the system’s real-time responsiveness and translation quality. CR measures user satisfaction with the system’s outputs through continuous ratings, reflecting users’ real-time experiences and satisfaction levels with each translation result. This evaluation method helps assess the performance in dynamic and real-time interactions and provides intuitive feedback to guide system improvements and optimizations.

As shown in Table 1, we present several commonly used speech datasets (like LibriTrans<sup>1</sup>, BSTC<sup>2</sup>, CoVoST<sup>3</sup>, MuST-C<sup>4</sup>) across different domains, along with their average SacreBLEU scores. Specifically, while models are often trained on multiple datasets, evaluations using the AL metric are typically conducted on the MuST-C corpus, which is the most commonly used in ST, ensuring comparability across different studies.

<sup>1</sup><https://www.openslr.org/12/>

<sup>2</sup><https://aistudio.baidu.com/competition/detail/44>

<sup>3</sup><https://huggingface.co/datasets/covost2>

<sup>4</sup><https://mt.fbk.eu/must-c/>

Task	Modeling	Data Scale
ASR	Cross-modal	100K hours
MT	Cross-lingual	1B sentences
SimuST	Cross-modal and cross-lingual	Ks hours

Table 2: The scale of annotated data for ASR, MT, and ST tasks for a specific language pair (like en  $\rightarrow$  de).

## 5 Augmented Training Methods

The data scale varies significantly across ASR, MT, and ST tasks, with ST datasets notably smaller due to high annotation costs. As shown in Table 2, for a specific language pair, the training data for speech translation tasks typically consists of only a few hundred hours, while the training data for related ASR and MT tasks exceeds it by nearly a hundredfold. Section 5.1 discusses data augmentation methods, while Section 5.2 explores studies based on the multi-task learning framework.

### 5.1 Data Augmentation

Training SimulST models, which rely on ST data, presents a significant challenge due to data scarcity. Despite ST data is not specifically tailored for streaming tasks like SimulST, segmentation tools can be employed to adapt the data, they also pose challenges such as model convergence difficulties and lack of robustness. Data augmentation is an effective means to expand training data. For effective training, Huang *et al.* [2023b] utilize a well-trained MT model to translate the transcriptions from ASR data and synthesize a large amount of pseudo-data. While expanding data, they also used pre-training model weights to initialize SimulST, to make use of the pre-trained models. Rather than a simple data mixture, Ko *et al.* [2023] propose a method to address the scarcity of simultaneous interpretation (SI) data by using a larger-scale offline translation corpus for training. They also introduce a tag-based approach to control the style of translations and handle the differences.

### 5.2 Multi-task Learning

Multi-task learning is an enhancing approach to model training because it allows for learning multiple related tasks simultaneously, providing additional information and constraints to improve model performance. Niehues *et al.* [2018] investigate two methods to select reasonable sub-strings from the reference to build partial parallel corpora for model-training and they opt to use multi-task learning to take advantage of a pre-trained MT model. A similar idea brings Modality Agnostic Meta-Learning (MAML) in SimulST involving meta-learning and fine-tuning steps [Han *et al.*, 2020]. In the former step, a set of high-resource tasks are trained as source tasks to capture general aspects; then they fine-tune the model further to learn the specific SimulST task in the latter step. Except for DAR, Zaidi *et al.* [2021] employ multi-task learning by training the SimulMT model along with SimulST and Chen *et al.* [2021] use multi-task learning to jointly learn tasks with a shared encoder, leveraging streaming ASR to guide SimulST decoding via beam search. Based on AED, Deng *et al.* [2022] achieve a joint CTC/attention by injecting

a CTC objective between encoder outputs and target translations. They additionally calculate a CTC loss from the CTC branch and apply an ASR-based intermediate CTC loss for multi-task learning.

By jointly training multiple tasks within the same model, it enables the sharing of underlying representations, thereby improving data efficiency. As augmented training methods, when combining data enhancement and multi-task learning methods with the original, the model can benefit more as well as be fully trained.

## 6 Future Work

Based on the recent advancements in SimulST tasks and the demands of real-time scenarios, we believe that two promising directions are multilingual SimulST and integration with Large Language Models.

### 6.1 Multilingual SimulST

With the advancement of globalization and increased cross-cultural communication, multilingual SimulST holds significant potential. This approach enables real-time translation of speech inputs into multiple languages, facilitating communication and collaboration in multilingual environments.

Following the simultaneous adaptation procedure in the previous work [Liu *et al.*, 2020], Subramanya and Niehues [2022] explore whether it can be utilized to build multilingual SimulST and they conduct experiments on both cascaded and end-to-end offline models. With the focus on multilingual SimulST, Huang *et al.* [2023a] propose a separate decoder model and a unified encoder-decoder model for joint training and decoding. They also introduce an asynchronous training strategy to enhance knowledge transfer among different languages. Based on the neural transducer, Wang *et al.* [2022] introduce LAMASSU for language-agnostic multilingual speech recognition and translation in a streaming fashion. By incorporating multilingual capabilities, the system becomes more versatile, accommodating diverse language settings and catering to the needs of various demographics.

### 6.2 Integration with LLMs

In recent years, the development of large-scale language models has made significant strides [Zhao *et al.*, 2023]. Large language models (LLMs) leverage extensive pre-existing linguistic knowledge [Radford *et al.*, 2023], thereby improving translation quality and accuracy [Le *et al.*, 2023]. Among this, AudioPaLM combines text-based and speech-based language models into a multi-modal generative model, covering most of the offline text and speech processing and generation tasks [Rubenstein *et al.*, 2023]. A brand-new work Seamless releases a set of full-process large-scale speech translation systems, introducing a family of Seamless models that enable multilingual and expressive SimulST [Communication *et al.*, 2023]. Integrating LLMs into SimulST systems enhances their ability to accurately understand speech inputs, handle contextual dependencies, and generate fluent translations. In a word, we anticipate that we can further enhance the performance and applicability of streaming speech translation systems, meeting the diverse needs of users in real-time scenarios by combining these two directions.

## Acknowledgements

This work was supported in part by the National Science Foundation of China (No.62276056), the Natural Science Foundation of Liaoning Province of China (2022-KF-16-01), the Fundamental Research Funds for the Central Universities (Nos. N2216016 and N2316002), the Yunnan Fundamental Research Projects (No. 202401BC070021), and the Program of Introducing Talents of Discipline to Universities, Plan 111 (No.B16009).

## Contribution Statement

**Xiaoqian Liu** and **Guoqiang Hu** conducted the literature search. They collected and categorized the relevant papers, and drafted the main sections of the manuscript. They also integrated the feedback from other contributors to refine and enhance the overall quality of the manuscript.

**Yangfan Du**, **Erfeng He**, and **Yingfeng Luo** were responsible for the creation of formulas, tables, and figures, and they also provided suggestions to ensure the accuracy and coherence of the paper.

**Chen Xu**, **Tong Xiao**, and **Jingbo Zhu** planned and guided the overall direction of the survey. They provided substantial feedback at various stages of the manuscript and offered insights on future trends in SimulST.

## References

- [Chang and Lee, 2022] Chih-Chiang Chang and Hung-yi Lee. Exploring continuous integrate-and-fire for adaptive simultaneous speech translation. In *Interspeech*, 2022.
- [Chen *et al.*, 2021] Junkun Chen, Mingbo Ma, Renjie Zheng, and Liang Huang. Direct simultaneous speech-to-text translation assisted by synchronized streaming ASR. In *ACL Findings*, 2021.
- [Chen *et al.*, 2023] Junkun Chen, Jian Xue, Peidong Wang, Jing Pan, and Jinyu Li. Improving stability in simultaneous speech translation: A revision-controllable decoding approach. *CoRR*, 2023.
- [Cherry and Foster, 2019] Colin Cherry and George F. Foster. Thinking slow about latency evaluation for simultaneous machine translation. *CoRR*, 2019.
- [Communication *et al.*, 2023] Seamless Communication, Loïc Barrault, Yu-An Chung, Mariano Coria Meglioli, David Dale, Ning Dong, Mark Duppenthaler, Paul-Ambroise Duquenne, Brian Ellis, Hady Elsahar, et al. Seamless: Multilingual expressive and streaming speech translation. *CoRR*, 2023.
- [Deng *et al.*, 2022] Keqi Deng, Shinji Watanabe, Jiatong Shi, and Siddhant Arora. Blockwise streaming transformer for spoken language understanding and simultaneous speech translation. In *Interspeech*, 2022.
- [Dong *et al.*, 2022] Qian Dong, Yaoming Zhu, Mingxuan Wang, and Lei Li. Learning when to translate for streaming speech. In *ACL*, 2022.
- [Graves *et al.*, 2006] Alex Graves, Santiago Fernández, Faustino J. Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *ICML*, 2006.
- [Graves, 2012] Alex Graves. Sequence transduction with recurrent neural networks. *CoRR*, 2012.
- [Han *et al.*, 2020] Hou Jeung Han, Mohd Abbas Zaidi, Sathish Reddy Indurthi, Nikhil Kumar Lakumarapu, Beomseok Lee, and Sangha Kim. End-to-end simultaneous translation system for IWSLT2020 using modality agnostic meta-learning. In *IWSLT*, 2020.
- [Huang *et al.*, 2023a] Wuwei Huang, Renren Jin, Wen Zhang, Jian Luan, Bin Wang, and Deyi Xiong. Joint training and decoding for multilingual end-to-end simultaneous speech translation. In *ICASSP*, 2023.
- [Huang *et al.*, 2023b] Wuwei Huang, Mengge Liu, Xiang Li, Yanzhi Tian, Fengyu Yang, Wen Zhang, Jian Luan, Bin Wang, Yuhang Guo, and Jinsong Su. The xiaomi AI lab’s speech translation systems for IWSLT 2023 offline task, simultaneous task and speech-to-speech task. In *IWSLT*, 2023.
- [Kano *et al.*, 2022] Yasumasa Kano, Katsuhito Sudoh, and Satoshi Nakamura. Average token delay: A latency metric for simultaneous translation. *CoRR*, 2022.
- [Ko *et al.*, 2023] Yuka Ko, Ryo Fukuda, Yuta Nishikawa, Yasumasa Kano, Katsuhito Sudoh, and Satoshi Nakamura. Tagged end-to-end simultaneous speech translation training using simultaneous interpretation data. In *IWSLT*, 2023.
- [Le *et al.*, 2023] Chenyang Le, Yao Qian, Long Zhou, Shujie Liu, Michael Zeng, and Xuedong Huang. Comsl: A composite speech-language model for end-to-end speech-to-text translation. In *NeurIPS*, 2023.
- [Liu *et al.*, 2020] Danni Liu, Gerasimos Spanakis, and Jan Niehues. Low-latency sequence-to-sequence speech recognition and translation by partial hypothesis selection. In *Interspeech*, 2020.
- [Liu *et al.*, 2021] Dan Liu, Mengge Du, Xiaoxi Li, Ya Li, and Enhong Chen. Cross attention augmented transducer networks for simultaneous translation. In *EMNLP*, 2021.
- [Ma *et al.*, 2019] Mingbo Ma, Liang Huang, Hao Xiong, Renjie Zheng, Kaibo Liu, Baigong Zheng, Chuanqiang Zhang, Zhongjun He, Hairong Liu, et al. STACL: Simultaneous translation with implicit anticipation and controllable latency using prefix-to-prefix framework. In *ACL*, 2019.
- [Ma *et al.*, 2020a] Xutai Ma, Mohammad Javad Dousti, Changhan Wang, Jiatao Gu, and Juan Pino. SIMULEVAL: An evaluation toolkit for simultaneous translation. In *EMNLP*, 2020.
- [Ma *et al.*, 2020b] Xutai Ma, Juan Pino, and Philipp Koehn. SimulMT to SimulST: Adapting simultaneous text translation to end-to-end simultaneous speech translation. In *ACL*, 2020.



- [Ma *et al.*, 2021] Xutai Ma, Yongqiang Wang, Mohammad Javad Dousti, Philipp Koehn, and Juan Miguel Pino. Streaming simultaneous speech translation with augmented memory transformer. In *ICASSP*, 2021.
- [Macháček *et al.*, 2023] Dominik Macháček, Ondrej Bojar, and Raj Dabre. MT metrics correlate with human ratings of simultaneous speech translation. In *IWSLT*, 2023.
- [Nguyen *et al.*, 2021a] Ha Nguyen, Yannick Estève, and Laurent Besacier. An empirical study of end-to-end simultaneous speech translation decoding strategies. In *ICASSP*, 2021.
- [Nguyen *et al.*, 2021b] Ha Nguyen, Yannick Estève, and Laurent Besacier. Impact of encoding and segmentation strategies on end-to-end simultaneous speech translation. In *Interspeech*, 2021.
- [Niehues *et al.*, 2018] Jan Niehues, Ngoc-Quan Pham, Thanh-Le Ha, Matthias Sperber, and Alex Waibel. Low-latency neural speech translation. In *Interspeech*, 2018.
- [Papi *et al.*, 2022] Sara Papi, Marco Gaido, Matteo Negri, and Marco Turchi. Does simultaneous speech translation need simultaneous models? In *EMNLP Findings*, 2022.
- [Papi *et al.*, 2023a] Sara Papi, Matteo Negri, and Marco Turchi. Attention as a guide for simultaneous speech translation. In *ACL*, 2023.
- [Papi *et al.*, 2023b] Sara Papi, Marco Turchi, and Matteo Negri. Alignatt: Using attention-based audio-translation alignments as a guide for simultaneous speech translation. *CoRR*, 2023.
- [Papineni *et al.*, 2002] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002.
- [Polák *et al.*, 2023] Peter Polák, Brian Yan, Shinji Watanabe, Alex Waibel, and Ondrej Bojar. Incremental blockwise beam search for simultaneous speech translation with controllable quality-latency tradeoff. *CoRR*, 2023.
- [Polák, 2023] Peter Polák. Long-form simultaneous speech translation: Thesis proposal. *CoRR*, 2023.
- [Post, 2018] Matt Post. A call for clarity in reporting BLEU scores. In *WMT 2018*, 2018.
- [Radford *et al.*, 2023] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *ICML*, 2023.
- [Raffel and Chen, 2023] Matthew Raffel and Lizhong Chen. Implicit memory transformer for computationally efficient simultaneous speech translation. In *ACL Findings*, 2023.
- [Raffel *et al.*, 2023] Matthew Raffel, Drew Penney, and Lizhong Chen. Shiftable context: Addressing training-inference context mismatch in simultaneous speech translation. In *ICML*, 2023.
- [Rei *et al.*, 2020] Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. Unbabel’s participation in the WMT20 metrics shared task. In *WMT*, 2020.
- [Ren *et al.*, 2020] Yi Ren, Jinglin Liu, Xu Tan, Chen Zhang, Tao Qin, Zhou Zhao, and Tie-Yan Liu. SimulSpeech: End-to-end simultaneous speech to text translation. In *ACL*, 2020.
- [Rubenstein *et al.*, 2023] Paul K. Rubenstein, Chulayuth Asawaroengchai, Duc Dung Nguyen, Ankur Bapna, Zalán Borsos, Félix de Chaumont Quitry, Peter Chen, Dalia El Badawy, Wei Han, Eugene Kharitonov, et al. Audiopalm: A large language model that can speak and listen. *CoRR*, 2023.
- [Sethiya and Maurya, 2023] Nivedita Sethiya and Chandresh Kumar Maurya. End-to-end speech-to-text translation: A survey. *CoRR*, 2023.
- [Subramanya and Niehues, 2022] Shashank Subramanya and Jan Niehues. Multilingual simultaneous speech translation. *CoRR*, 2022.
- [Tang *et al.*, 2023] Yun Tang, AnnaY. Sun, Hirofumi Inaguma, Xinyue Chen, Ning Dong, Xutai Ma, PadenD. Tomasello, and Juan Pino. Hybrid transducer and attention based encoder-decoder modeling for speech-to-text tasks. *CoRR*, 2023.
- [Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.
- [Wang *et al.*, 2022] Peidong Wang, Eric Sun, Jian Xue, Yu Wu, Long Zhou, Yashesh Gaur, Shujie Liu, and Jinyu Li. Lamassu: Streaming language-agnostic multilingual speech recognition and translation using neural transducers. *CoRR*, 2022.
- [Xu *et al.*, 2023] Chen Xu, Rong Ye, Qianqian Dong, Chengqi Zhao, Tom Ko, Mingxuan Wang, Tong Xiao, and Jingbo Zhu. Recent advances in direct speech-to-text translation. In *IJCAI*, 2023.
- [Zaidi *et al.*, 2021] Mohd Abbas Zaidi, Beomseok Lee, Nikhil Kumar Lakumarapu, Sangha Kim, and Chanwoo Kim. Decision attentive regularization to improve simultaneous speech translation systems. *CoRR*, 2021.
- [Zeng *et al.*, 2021] Xingshan Zeng, Liangyou Li, and Qun Liu. RealTrans: End-to-end simultaneous speech translation with convolutional weighted-shrinking transformer. In *ACL Findings*, 2021.
- [Zhang and Feng, 2023] Shaolei Zhang and Yang Feng. End-to-end simultaneous speech translation with differentiable segmentation. In *ACL Findings*, 2023.
- [Zhang *et al.*, 2022] Ruiqing Zhang, Zhongjun He, Hua Wu, and Haifeng Wang. Learning adaptive segmentation policy for end-to-end simultaneous translation. In *ACL*, 2022.
- [Zhao *et al.*, 2023] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *CoRR*, 2023.