Fluent and Low-latency Simultaneous Speech-to-Speech Translation with Self-adaptive Training *

Renjie Zheng [1 †] **Mingbo Ma** [1 †] **Baigong Zheng** [1 ‡] **Kaibo Liu** [1]
Jiahong Yuan [1] **Kenneth Church** [1] **Liang Huang** [1,2]
[1]Baidu Research, Sunnyvale, CA, USA
[2]Oregon State University, Corvallis, OR, USA
{renjiezheng, mingboma}@baidu.com

Abstract

Simultaneous speech-to-speech translation is widely useful but extremely challenging, since it needs to generate target-language speech *concurrently* with the source-language speech, with only a few seconds delay. In addition, it needs to continuously translate a stream of sentences, but all recent solutions merely focus on the single-sentence scenario. As a result, current approaches accumulate latencies progressively when the speaker talks faster, and introduce unnatural pauses when the speaker talks slower. To overcome these issues, we propose Self-Adaptive Translation (SAT) which flexibly adjusts the length of translations to accommodate different source speech rates. At similar levels of translation quality (as measured by BLEU), our method generates more fluent target speech (as measured by the naturalness metric MOS) with substantially lower latency than the baseline, in both Zh↔En directions.

1 Introduction

Simultaneous speech-to-speech translation, which mimics the human interpreter's practice to translate the source speech into a different language with 3 to 5 seconds delay, has wide usage scenarios such as international conference meetings, traveling and negotiations as it provides more natural communication process than simultaneous speech-to-text translation. This task has been widely considered as one of the most challenging tasks in NLP with (but not limited to) following reasons: on one hand, the simultaneous translation is a hard task due to the word order difference between source and target languages, e.g.,

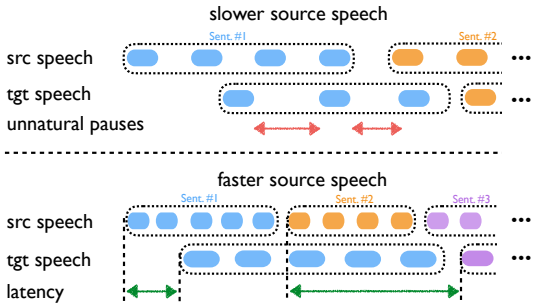


Figure 1: Slower source speech causes unnatural pauses (↔) between words. Faster source speech propagates extra latencies (↔) to the following sentences.

SOV languages (German, Japanese, etc.) and SVO languages (English, Chinese, etc.); on the other hand, simultaneous speech-to-speech translation escalates the challenge by considering the smooth cooperation between the modules of speech recognition, translation and speech synthesis.

In order to achieve simultaneous speech-to-speech translation (SSST), to the best of our knowledge, most recent approaches (Oda et al., 2014; Xiong et al., 2019) dismantle the entire system into a three-step pipelines, streaming Automatic Speech Recognition (ASR) (Sainath et al., 2020; Inaguma et al., 2020; Li et al., 2020), simultaneous Text-to-Text translation (sT2T) (Gu et al., 2017; Ma et al., 2019; Arivazhagan et al., 2019; Ma et al., 2020b), and Text-to-Speech (TTS) synthesis (Wang et al., 2017; Ping et al., 2017; Oord et al., 2017). Most recent efforts mainly focus on sT2T which is considered the key component to further reduce the translation latency and improve the translation quality for the entire pipeline. To achieve better translation quality and lower latency, there has been extensive research efforts which concentrate on the sT2T by introducing more robust models (Ma et al., 2019; Arivazhagan et al., 2019), better policies (Gu et al., 2017; Zheng et al., 2020a, 2019b,a), new decoding algorithms (Zheng et al., 2019c, 2020b), or multimodal

* See our speech-to-speech simultaneous translation demos (including comparison with human interpreters) at https://sat-demo.github.io.

† Equal contribution

‡ Work done at Baidu Research. Current address: Kwai Inc., Seattle, WA, USA.

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 3928–3937
November 16 - 20, 2020.

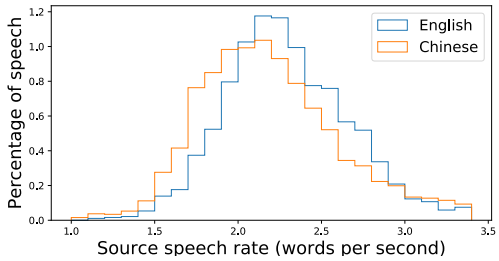


Figure 2: Average Chinese and English speed rate distribution for different speakers.

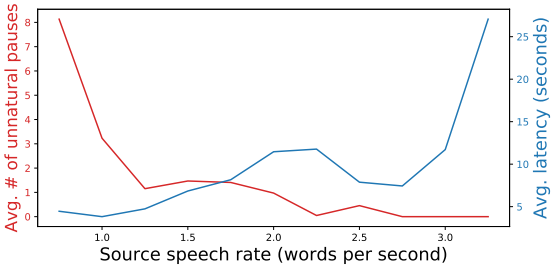


Figure 3: Relationship between source speech rate with latency and number of unnatural pauses by naively using wait-k model in simultaneous Chinese-to-English speech-to-speech translation in our dev set.

information (Imankulova et al., 2019). However, is it sufficient to only consider the effectiveness of sT2T and ignore the interactions between other different components?

Furthermore, in practice, when we need to translate multiple sentences continuously, it is not only important to consider the cooperations between sT2T and other speech-related modules, but also essential to take the effects between current and later sentence into consideration as shown in Fig. 1. Unfortunately, all the aforementioned techniques ignore other speech modules and merely establish their systems and analysis on single-sentence scenario, which is not realistic. To achieve fluent and constant, low-latency SSST, we also need to consider speech speed difference between target and source speech.

Fig. 2 shows the speech rate distributions for both Chinese and English in our speech corpus. The speech rate varies especially for different speakers. As shown in Fig. 1, when we have various source speech speed, the number of unnatural pauses and the latency vary dramatically. More specifically, when speaker talks slowly, TTS often needs to make more pauses to wait for more tokens from sT2T which usually does not output new translations with limited source information. These unnatural pauses lead to semantic and syntactic confusion (Lege, 2012; Bae, 2015). On the contrary, when speaker talks fast, the target speech synthesized from previous sT2T models (e.g. wait-k) always introduce large latency which accumulates through the entire paragraph and causes significant delays.Therefore, in realistic, the latency for the latter sentences are far more than the claimed latency in the original system. Fig. 3 supports the above hypothesis when source side speech rate varies while using one wait-k translation model and iTTS model.

To overcome the above problems, we propose Self-Adaptive Translation (SAT) for simultaneous speech-to-speech translation, which flexibly determines the length of translation based on different source speech rate. As it is shown in Figure 6, within this framework, when the speakers talk very fast, the model is encouraged to generate abbreviate but informative translation. Hence, as a result of shorter translation, the previous translation speech can finish earlier and alleviate their effects to the latter ones. Similarly, when the speakers have slower speech rate, the decoder will generate more meaningful tokens until a natural speech pause. The speech pauses can be understood as a natural boundary between sentences or phrase which does not introduce ambiguity to the translation. In conclusion, we make the following contributions:

- We propose SAT to flexibly adjust translation length to generate fluent and low-latency target speeches for the entire speech (Sec. 3).
- We propose paragraph based Boundary Aware Delay as the first latency metric suitable for simultaneous speech-to-speech translation (Sec. 4).
- We annotate a new simultaneous speech-to-speech translation dataset for Chinese-English translation, together with professional interpreters' interpretation (Sec 5).
- Our system is the first simultaneous speech-to-speech translation system using iTTS (as opposed to full-sentence TTS) to further reduce the latency (Sec. 2).
- Experiments show that our proposed system can achieve higher speech fluency and lower latency with similar or even higher translation quality compared with baselines and even human interpreters (Sec. 5).

2 Preliminaries

In this section, we first introduce each component of three-step pipeline, which are streaming ASR,

Figure 4: Pipeline of Speech-to-speech simultaneous translation.

#	Incremental Transcription	Time (ms)
1	thank you	960
2	thank you miss	1120
3	thank you Mr chair	1600
4	Thank you , Mr chairman .	2040

Table 1: Example of English streaming ASR. Red words are revised in latter steps. Punctuations only appear in the final step.

simultaneous translation models and incremental TTS techniques.

2.1 Streaming Automatic Speech Recognition

We use anonymous real-time speech recognizer as the speech recognition module. As shown in Fig. 4, streaming ASR is first step of the entire pipeline which converts the growing source acoustic signals from speaker into a sequence of tokens $\boldsymbol{x} = (x_1, x_2, ...)$ timely with about 1 second latency. Table 1 demonstrates one example of English streaming ASR which generates the English outputs incrementally. Each row in the table represents the streaming ASR outputs at each step. Note that streaming ASR sometimes revises some tail outputs from previous step (e.g. 3th and 4th steps in Table 1). To get stabler outputs, we exclude the last word in ASR outputs (except the final steps) in our system.

2.2 Simultaneous Machine Translation

As an intermediate step between source speech recognition and target speech synthesis modules, the goal of this step is to translation all the available source language tokens from streaming ASR into another language. There are many Text-to-Text simultaneous translation models (Gu et al., 2017; Ma et al., 2019; Arivazhagan et al., 2019; Ma et al., 2020b) that have been proposed recently.

Different from conventional full-sentence translation model, which encodes the entire source sentence $\boldsymbol{x} = (x_1, ...x_m)$ into a sequence of hidden states, and decodes sequentially conditioned on those hidden states and previous predictions as $p(\boldsymbol{y} \mid \boldsymbol{x}) = \prod_{t=1}^{|\boldsymbol{y}|} p(y_t \mid \boldsymbol{x}, \boldsymbol{y}_{<t})$ to form the final hypothesis $\boldsymbol{y} = (y_1, ..., y_t)$, simultaneous translation makes predictions with partial, growing inputs before the source sentence finishes.

Without loss of generality, regardless the actual design of translation policy, simultaneous translation can be represented with prefix-to-prefix fashion as follows:

$$p_g(\boldsymbol{y} \mid \boldsymbol{x}) = \prod_{t=1}^{|\boldsymbol{y}|} p(y_t \mid \boldsymbol{x}_{\leqslant g(t)}, \boldsymbol{y}_{<t}) \quad (1)$$

where $g(t)$ can be used to represent any arbitrary fixed or adaptive policy, denoting the number of processed source tokens at time step t. We choose the wait-k policy (Ma et al., 2019) as our baseline for its simplicity and great performance.

More specifically, in this paper, our wait-k policy is defined as follows:

$$g_{\text{wait-}k}(t) = \min\{k + t - 1, |\boldsymbol{x}|\} \quad (2)$$

This policy starts to decode after the first k source words, and then translates one token every time when one more source token is received.

2.3 Incremental Text-to-Speech

As the last step of the entire pipeline the goal of iTTS is to incrementally generate the target speech audio and play it to the audience instantly with available translated words. Different from conventional full-sentence TTS, which requires the availability of the entire sentence, iTTS usually has 1-2 words delay but with similar audio quality compared with the full-sentence TTS. Compared with previous source sentence segment-based SSST systems (Oda et al., 2014; Xiong et al., 2019), our system can achieve word-level latency. We adapt the iTTS framework from Ma et al. (2019) to our pipeline to generate target speech audio with translated tokens $\boldsymbol{y}_t$ at t time step.

3 Self-Adaptive Translation

To overcome the above practical problems, we propose Self-Adaptive Translation (SAT) technique to enable the ability of adjusting the length of the translation based on the demand of latency and fluency. We first demonstrate the problem of one naive solution. Then, we introduce our training framework and talk about how to apply this technique in practice during inference time.

3.1 Naive Solution is Problematic

To alleviate the various speech rate problem, one naive solution is to adjust the target side speech speed based on the source speaker's speed. However, as shown in Table 2, this solution is problematic as it usually requires the audience to be more

Speech Rate	MOS
0.5×	2.00 ± 0.08
0.6×	2.32 ± 0.08
0.75×	2.95 ± 0.07
Original	4.01 ± 0.08
1.33×	3.34 ± 0.08
1.66×	2.40 ± 0.09
2.0×	2.06 ± 0.04

Table 2: Mean Opinion Score (MOS) evaluations of naturalness for different speech speed changed by ffmpeg. Original English speeches are synthesized by our incremental Text-to-speech system.

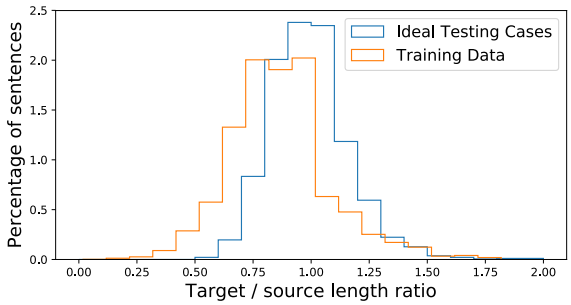


Figure 5: Tgt/src length ratio for English-to-Chinese task in training data (red) and ideal testing cases (blue).

focus on the translated speech when we speed up the speech rate on target side, and sometimes it will disrupt the audiences' comprehension of the translation (Gordon-Salant et al., 2014). Similarly, slowing down the speech only creates overlong phoneme pronunciation which is unnatural and leads to confusion.

Inspired by human interpreters(He et al., 2016; Al-Khanji et al., 2000) who often summarize the contexts in order to catch up the speaker, or make wordy translation to wait the speaker, the optimal translation model should be enable to adjust the length of translated sentence to change the speech duration on target side to avoid further delays or unnatural pauses fundamentally.

3.2 Self-Adaptive Training

Translation between different language pairs have various tgt/src length ratios, e.g., English-to-Chinese translation ratio is roughly around 0.85 (small variations between different datasets). However, this length ratio merely reflects the average statistics for the entire dataset, and as it is shown with red line in Fig. 5, the ratio distribution for individual sentence is quite wide around the average length ratio.

As shown in Fig. 6 and discussed in earlier sections, over short and long translations are not preferred in simultaneous speech-to-speech translation. Ideally, we prefer the system to have a similar

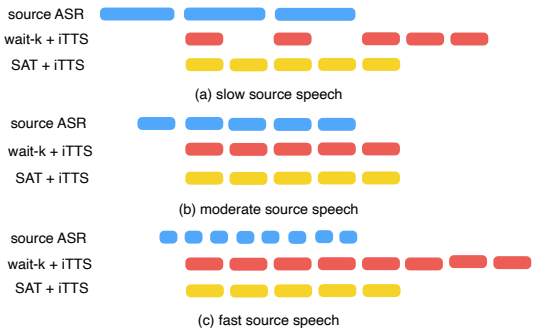


Figure 6: Illustration of conventional wait-k (red) and SAT-k (yellow) training policy. In SAT, we force the length of tail to be k which equals the latency k. In the above example, we have $k = 1$.

amount of initial wait with delay in the tail during translation of each sentence. Following this design, the translation tail of previous sentence will fit perfectly into the beginning delay window for the following sentence, and will not cause any extra latency and intermittent speech.

Based on the above observation, we propose to use different training policies for different sentences with different tgt/src ratios. As shown in Fig. 6, We start from a fixed delay of k tokens and then force the model to have the same number of tokens in initial wait and final tail by amortizing the extra tokens into the middle steps. More specifically, when we have longer tail than the fixed initial wait, we move extra words into former steps, and some steps before tail will decode more than one word at a time. As a result, there will be some one-to-many policies between source and target and the model will learn to generate longer translations with shorter source text. On the contrary, when we have shorter tail, we perform extra reading on the source side and the model will learn to generate shorter translation through this many-to-one policy. Formally, we define our SAT training policy as follows:

$$g_{\text{wait-}k,\,c}(t) = \min\{k + t - 1 - \lfloor ct \rfloor,\ |\boldsymbol{x}|\} \quad (3)$$

where c is the compensation rate which is decided by the tgt/src length ratio after deduction of k tokens in source initial and target tail. For example, when tgt/src length ratio is 1.25, then $c = \frac{|tgt|-k}{|src|-k} - 1 = 1.25 - 1 = 0.25$, representing to decode 5 target words for every 4 source words, and model learn to generate wordy translation. When target side is shorter than source side, c becomes negative, and model learn to decode less tokens than source side.

Note that the tgt/src length ratio in our case is determined by the corresponding sentence it-

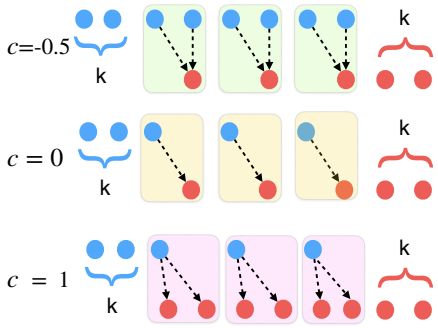
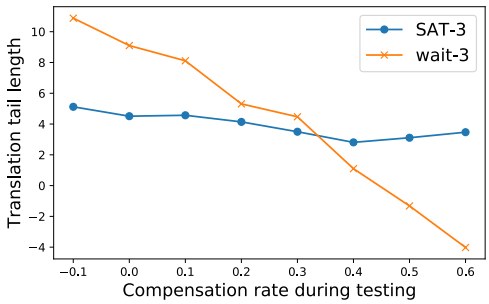


Figure 7: Different translation policy with different choice of c. Green boxes represent many-to-1 policy; yellow boxes denote 1-to-1 policy; purple boxes show 1-to-many translation policy.

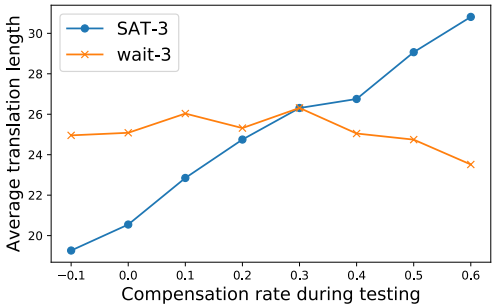
self instead of the corpus level tgt/src length ratio, which is a crucial different from catchup algorithm from (Ma et al., 2019) where some short translations is trained with inappropriate positive c. It seems to be a minor difference, but it actually enables the model to learn totally different thing other than catchup.

The blue line in Fig. 5 represents the tgt/src length ratio for the ideal simultaneous speech-to-speech translation examples in our training set which have the same speech time between source and target side. When we have the same speech time between source and target side, there will be no accumulated latency from previous sentences to the following sentences. As we notice, our training data covers all the tgt/src length ratio distribution for the ideal cases, indicating that by adjusting the compensation rate c from our training corpus, our model learns to generate appropriate length of translation on the target side to avoid accumulated latency.

As shown in Fig. 5, there are many different choices of c for different sentences, and each sentence is trained with their own corresponding compensation rate which makes the training policy different from others with different c. Hence, As shown in Fig. 7, our trained model is implicitly learned many different policies, and when you choose a compensation rate c during inference, the model will generate certain length of translation corresponding to that compensation rate c in training. More specifically, assume we have a source sentence, for example in Chinese, with length of m, and the conventional full-sentence or wait-k model normally would translate this into a English sentence with length of $1.25 \times m$. However, the output length from SAT can be changed



(a) Tail length vs. test-time compensation rate



(b) Translation length $|\boldsymbol{y}|$ vs. test-time compensation rate

Figure 8: Translation length analysis on Chinese-to-English task using one SAT-3 and wait-k model.

by c following the policy in Eq. 3 during decoding. When c is negative, SAT generates shorter translation than $1.25 \times m$. On the contrary, if we choose c that is positive, SAT generates longer translation than $1.25 \times m$. The compensation rate c functions as the key of model selection to generate outputs of different lengths.

Fig. 8(a)–8(b) show the effectiveness of our proposed model which has the ability to adjust the tail length of the entire translation with different c's.

3.3 Self-Adaptive Inference

The above section discusses the importance of c, which is easily to obtain during training time, but at inference time, we do not know the optimal choice of c in advance since the fluency and latency criteria also rely on the finish time for each word on both sides. Therefore, the streaming ASR and iTTS plays important roles here to determine the decoding policy to form fluent and low-latency translation speech and we use the knowledge of streaming ASR and iTTS for selecting the appropriate policy on the fly.

When we have a faster speech, streaming ASR will send multiple tokens to SAT at some steps. But SAT only generates one token at a time on the target side and pass it to iTTS instantly. This decoding policy has a similar function to a negative c, which has many-to-one translation policy. In this case, SAT will generate succinct translation

and iTTS therefor can finish the translation speech with shorter time since there are less tokens.

Contrarily, when the speaker talks with slower pace, there is only one token that is feed into SAT. SAT translates it into a different token and delivers it to iTTS. This is one-to-one translation policy. When iTTS is about to finish playing the newly generated speech, and there is no new incoming token from steaming ASR, SAT will force the decoder to generate one extra token, which becomes one-to-many translation policy (including the decoded token in previous step), and feeds it to iTTS. When the speaker makes a long pause, and there is still no new tokens from streaming ASR, the decoder of SAT continues to translate until a pause token (e.g. any punctuations) generated. This pause token forms a necessary, natural pause on speech side and will not change the understanding for the translated speech.

4 Paragraph-Based Boundary Aware Latency

As mentioned frequently above, latency is another essential dimension for simultaneous speech-to-speech translation performance. However, the measurement of speech delay from source speech to each synthesized word in target speech is challenging and there is no direct metric that is suitable for simultaneous speech-to-speech translation.

Human interpreters use Ear-Voice-Span (EVS) (Gumul, 2006; Lee, 2002) to calculate translation delay for some landmark words from source speech to target source. However, this requires the target-to-source word correspondence. In practice, the translation model sometimes makes errors during translation which includes miss translation of some words or over translated some words that source does not include. Thus, an automatic fully word-to-word alignment between target and source is hard to be accurate. Human annotation is accurate but expensive and not practical.

Inspired by Ari et al. (2020) who proposed Translation Lag (TL) to ignore the semantic correspondence between words from target to source side and only calculate each target delay proportionally to each source words regardless the actually meaning of word in the task of simultaneous "speech-to-text" translation, we use a similar method to calculate the latency for each sentence.

Nevertheless, TL is only designed for single-sentence latency ,while we need to measure the latency of a paragraph of speech. Thus, we propose paragraph based Boundary Aware Latency (pBAL) to compute the latency of long speech simultaneous translation. In pBAL, we first align the each sentence, make each word's correspondence within the sentence boundary. Then we compute the time differences of the finished time between each target word's audio and its proportion corresponding source word's finish time in source side. In experiments, we determine the finish time of each source and target words by forced aligner (Yuan and Liberman, 2008) and align the translation and source speech by using the corresponding streaming ASR as a bridge.

5 Experiments

5.1 Datasets and Systems Settings

We evaluate on two simultaneous speech-to-speech translation directions: Chinese↔English. For training, we use the text-to-text parallel corpora available from WMT18[1] (24.7M sentence pairs). We also annotate a portion of Chinese and English speeches from LDC United Nations Proceedings Speech [2] (LDC-UN) as a speech-to-text corpus. This corpus includes speeches recorded in 2009-2012 from United Nations conferences in six official UN languages. We transcribe the speeches and then translate the transcriptions as references. The speech recordings include not only source speech but also corresponding professional simultaneous interpreters' interpretation in the conference. Thus, we also transcribe those human simultaneous interpretation of En→Zh direction which will not be used in our model but compared to in the following experiments.

		En→Zh	Zh→En
Train	# of speeches	58	119
	# of words	63650	61676
	Total time	6.81 h	9.68 h
Dev	# of speeches	3	6
	# of words	1153	2415
	Total time	0.27 h	0.35 h
Test	# of speeches	3	6
	# of words	3053	1870
	Total time	0.39 h	0.30 h

Table 3: Statistics of LDC-UN dataset (source-side).

Table 3 shows the statistics of our speech-to-text dataset. We train our models using both the WMT18 training set and the LDC UN speech-to-text training set. We validate and test the models only in the LDC-UN dataset. For Chinese side

[1] http://www.statmt.org/wmt18/translation-task.html

[2] https://catalog.ldc.upenn.edu/LDC2014S08

text, we use jieba [3] Chinese segmentation tool. We apply BPE (Sennrich et al., 2015) on all texts in order to reduce the vocabulary sizes. We set the vocabulary size to 16K for both Chinese and English text. Our Transformer is essentially the same with base Transformer model (Vaswani et al., 2017).

As mentioned in Section 2.1, we use an anonymous real-time speech recognizer from a well-known cloud platform as the speech recognition module for both English and Chinese. During speech-to-speech simultaneous translation decoding, after receiving an ASR input, we first normalize the punctuations and tokenize (or do Chinese segmentation for Zh→En translation) the input. The last tokens are always removed in the encoder of translation model because they are very unstable. In the latency measurement we use Penn Phonetics Lab Forced Aligner (P2FA) (Yuan and Liberman, 2008) as the forced aligner to automatically annotate the time-stamp for both Chinese and English words in source and target sides.

For the incremental Text-to-speech system, we follow Ma et al. (2020a) and take the Tacotron 2 model (Shen et al., 2018) as our phoneme-to-spectrogram model and train it with additional *guided attention loss* (Tachibana et al., 2018) which speeds up convergence. Our vocoder is the same as that in the Parallel WaveGAN paper (Yamamoto et al., 2020), which consists of 30 layers of dilated residual convolution blocks with exponentially increasing three dilation cycles, 64 residual and skip channels and the convolution filter size 3. For English, we use a proprietary speech dataset containing 13,708 audio clips (i.e., sentences) from a female speaker and the corresponding transcripts. For Chinese, we use a public speech dataset[4] containing 10,000 audio clips from a female speaker and the transcripts.

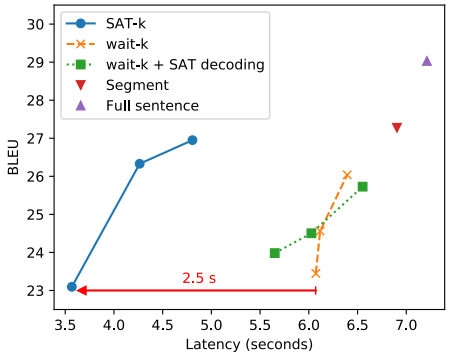
5.2 Speech-to-Speech Simul. Translation

Fig. 9 show the final results of our proposed models and baselines. For translation quality measurement, we use the "multi-bleu.pl" [5] script to calculate BLEU scores. Since different punctuations are soundless, we remove all of them before BLEU evaluation for both hypotheses and references. We follow (Xiong et al., 2019) to concatenate the translations of each talk into one sentence to measure BLEU scores.

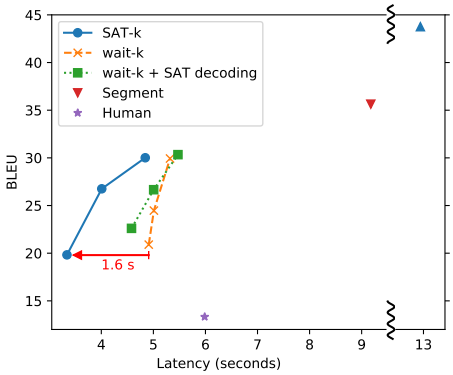
[3] https://github.com/fxsjy/jieba

[4] https://www.data-baker.com/open_source.html

[5] https://github.com/moses-smt//mosesdecoder/blob/master/scripts/generic/multi-bleu.perl



(a) Chinese-to-English simultaneous translation



(b) English-to-Chinese simultaneous translation

Figure 9: Translation quality and latency (pBAL) of proposed simultaneous speech-to-speech translation systems compared with baselines. For all those SAT-k and wait-k models, $k = \{3, 5, 7\}$ from bottom to top.

For Chinese-to-English simultaneous translation, we compare our models with naive wait-k, wait-k with SAT decoding (only use Self-adaptive inference in Sec. 3.3), segment based models (Oda et al., 2014; Xiong et al., 2019) and full sentence translation model. All these models share one iTTS system. For segment based model, since our streaming ASR API doesn't provide any punctuation before the final step, we use the final punctuations to segment the partial streaming inputs and then use a full-sentence translation model to translate the partial segment as a full sentence. The results show that our proposed SAT-k models can achieve much lower latency without sacrificing quality compared with those baselines.

Fig. 9(b) shows the results of En→Zh simultaneous translation. Besides the baselines used in Zh→En experiments, we also compare our system with professional human interpreters' translation. Our proposed models also outperform all the baselines and human interpreters. Our models reduce more latency in Zh→En than En→Zh compared with wait-k because English sentences is always longer than Chinese thus it's more easily to accumulate latency in Zh→En (also shown in Fig. 10).

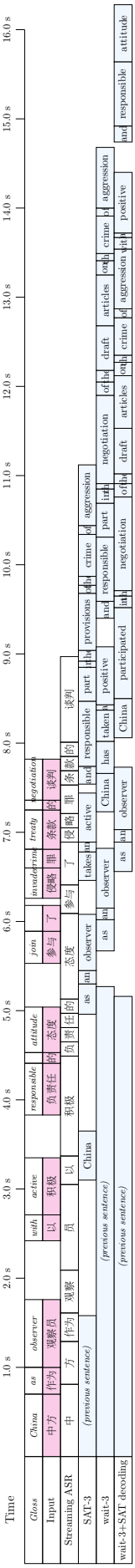


Figure 11: Decoding results of proposed simultaneous speech-to-speech Chinese-to-English translation system and baselines.

Figure 12: Decoding results of proposed simultaneous speech-to-speech English-to-Chinese translation system and baselines.

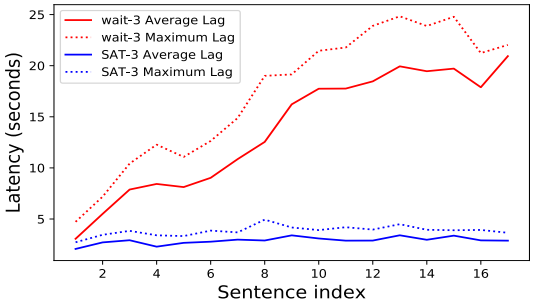


Figure 10: Latency for sentences at different indices in Chinese-to-English dev-set.

5.3 Human Evaluation on Speech Quality

Method	En→Zh	Zh→En
wait-3	3.56 ± 0.09	3.68 ± 0.08
wait-3 + SAT decoding	3.81 ± 0.08	3.96 ± 0.04
SAT-3	3.83 ± 0.07	3.97 ± 0.07
Segment-based	3.79 ± 0.15	3.99 ± 0.07
Full sentence	3.98 ± 0.08	4.03 ± 0.03
Human	3.85 ± 0.05	-

Table 4: MOS evaluations of fluency for different target speeches generated by different methods.

In Table 4, we evaluate our synthesized speeches by Mean Opinion Scores (MOS) with native speakers, which is a standard metric in TTS. Each speech received 10 human ratings scaled from 1 to 5, with 5 being the best. For both Zh↔En directions, wait-3 models have the lowest MOS due to the many unnatural pauses (see Sec. 3.1). Our proposed model SAT-3 and wait-3 with SAT decoding achieve similar fluency to full sentence models and even human interpreters.

5.4 Examples

Fig. 11 shows a Zh→En decoding example. Here the wait-3 models' outputs have a much longer latency compared with SAT-3 because their beginnings are delayed by the translation of previous sentence(s) and their tails are also very long. The En→Zh example in Fig. 12 is similar. While streaming ASR has a very long delay, SAT-3 model still controls the latency to roughly 4.5s; all pauses on the target side are natural ones from punctuations. By contrast, the human interpreter's translation has the longest latency.

6 Conclusions

We proposed Self-Adaptive Translation for simultaneous speech-to-speech translation which flexibly adjusts translation length to avoid latency accumulation and unnatural pauses. In both Zh↔En directions, our method generates fluent and low latency target speeches with high translation quality.

References

Raja Al-Khanji, Said Shiyab, and Riyad Hussein. 2000. On the use of compensatory strategies in simultaneous interpretation. *Meta: Journal des traducteurs*, 45:548.

Naveen Ari, Colin Andrew Cherry, Te I, Wolfgang Macherey, Pallavi Baljekar, and George Foster. 2020. Re-translation strategies for long form, simultaneous, spoken language translation. In *ICASSP 2020*.

Naveen Arivazhagan, Colin Cherry, Wolfgang Macherey, Chung-Cheng Chiu, Semih Yavuz, Ruoming Pang, Wei Li, and Colin Raffel. 2019. Monotonic infinite lookback attention for simultaneous machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy. Association for Computational Linguistics.

Rebecca M. Bae. 2015. *The effects of pausing on comprehensibility*. PhD dissertation, Iowa State University.

Sandra Gordon-Salant, Danielle J Zion, and Carol Espy-Wilson. 2014. Recognition of time-compressed speech does not predict recognition of natural fast-rate speech by older listeners. *The Journal of the Acoustical Society of America*, 136(4):EL268–EL274.

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, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers*, pages 1053–1062.

Ewa Gumul. 2006. Conjunctive cohesion and the length of ear-voice span in simultaneous interpreting. a case of interpreting students. *Linguistica Silesiana*, (27):93–103.

Wei He, Zhongjun He, Hua Wu, and Haifeng Wang. 2016. Improved neural machine translation with smt features. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 151–157. AAAI Press.

Aizhan Imankulova, Masahiro Kaneko, Tosho Hirasawa, and Mamoru Komachi. 2019. Towards multimodal simultaneous neural machine translation. *arXiv preprint arXiv:2004.03180*.

Hirofumi Inaguma, Yashesh Gaur, Liang Lu, Jinyu Li, and Yifan Gong. 2020. Minimum latency training strategies for streaming sequence-to-sequence asr. *ICASSP*.

Tae-Hyung Lee. 2002. Ear voice span in english into korean simultaneous interpretation. *Méta: journal des traducteurs/Méta: Translators' Journal*, 47(4):596–606.

Ryan Lege. 2012. *The Effect of Pause Duration on Intelligibility of Non-Native Spontaneous Oral Discourse*. PhD dissertation, Brigham Young University.

Bo Li, Shuo-Yiin Chang, Tara N. Sainath, Ruoming Pang, Yanzhang He, Trevor Strohman, and Yonghui Wu. 2020. Towards fast and accurate streaming end-to-end asr.

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.

Mingbo Ma, Baigong Zheng, Kaibo Liu, Renjie Zheng, Hairong Liu, Kainan Peng, Kenneth Church, and Liang Huang. 2020a. Incremental text-to-speech synthesis with prefix-to-prefix framework. *Findings of EMNLP*.

Xutai Ma, Juan Pino, James Cross, Liezl Puzon, and Jiatao Gu. 2020b. Monotonic multihead attention. *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, Conference Track Proceedings*.

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

Aaron Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George Driessche, Edward Lockhart, Luis Cobo, Florian Stimberg, Norman Casagrande, Dominik Grewe, Seb Noury, Sander Dieleman, Erich Elsen, Nal Kalchbrenner, Heiga Zen, Alex Graves, Helen King, and Demis Hassabis. 2017. Parallel WaveNet: Fast high-fidelity speech synthesis. In *ICML*.

Wei Ping, Kainan Peng, Andrew Gibiansky, Sercan Ömer Arik, Ajay Kannan, Sharan Narang, Jonathan Raiman, and John L. Miller. 2017. Deep Voice 3: Scaling text-to-speech with convolutional sequence learning. In *ICLR*.

Tara N. Sainath, Yanzhang He, Bo Li, Arun Narayanan, Ruoming Pang, Antoine Bruguier, Shuo-Yiin Chang, Wei Li, Raziel Alvarez, Zhifeng Chen, Chung-Cheng Chiu, David García, Alex Gruenstein, Ke Hu, Minho Jin, A. Kannan, Qiao Liang, I. Mcgraw, Cal Peyser, Rohit Prabhavalkar, Golan Pundak, David Rybach, Yuan Shangguan, Yash Sheth, Trevor Strohman, Mirkó Visontai, Yonghui Wu, Yinyong Zhang, and Ding Zhao. 2020. A streaming on-device end-to-end model surpassing server-side conventional model quality and latency. *ICASSP*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Jonathan Shen, Ruoming Pang, Ron Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, Rif Saurous, Yannis Agiomvrgiannakis, and Yonghui Wu. 2018. Natural TTS synthesis by conditioning WaveNet on MEL spectrogram predictions. In *Interspeech*. Tacotron 2.

Hideyuki Tachibana, Katsuya Uenoyama, and Shunsuke Aihara. 2018. Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4784–4788. IEEE.

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 *Advances in Neural Information Processing Systems 30*.

Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc Le, Yannis Agiomyrgiannakis, Rob Clark, and Rif A. Saurous. 2017. Tacotron: Towards end-to-end speech synthesis. In *Interspeech*.

Hao Xiong, Ruiqing Zhang, Chuanqiang Zhang, Zhongjun He, Hua Wu, and Haifeng Wang. 2019. Dutongchuan: Context-aware translation model for simultaneous interpreting.

Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim. 2020. Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6199–6203. IEEE.

Jiahong Yuan and Mark Liberman. 2008. Speaker identification on the scotus corpus. *Journal of the Acoustical Society of America*, 123(5):3878.

Baigong Zheng, Kaibo Liu, Renjie Zheng, Mingbo Ma, Hairong Liu, and Liang Huang. 2020a. Simultaneous translation policies: from fixed to adaptive. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Baigong Zheng, Renjie Zheng, Mingbo Ma, and Liang Huang. 2019a. Simpler and faster learning of adaptive policies for simultaneous translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1349–1354.

Baigong Zheng, Renjie Zheng, Mingbo Ma, and Liang Huang. 2019b. Simultaneous translation with flexible policy via restricted imitation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5816–5822.

Renjie Zheng, Mingbo Ma, Baigong Zheng, and Liang Huang. 2019c. Speculative beam search for simultaneous translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1395–1402.

Renjie Zheng, Mingbo Ma, Baigong Zheng, Kaibo Liu, and Liang Huang. 2020b. Opportunistic decoding with timely correction for simultaneous translation. In *ACL*.