Simul-MuST-C: Simultaneous Multilingual Speech Translation Corpus Using Large Language Model

Mana Makinae, Yusuke Sakai, Hidetaka Kamigaito, Taro Watanabe
Nara Institute of Science and Technology
{makinae.mana.mh2, sakai.yusuke.sr9, kamigaito.h, taro}@is.naist.jp

Abstract

Simultaneous Speech Translation (SiST) begins translating before the entire source input is received, making it crucial to balance quality and latency. In real interpreting situations, interpreters manage this simultaneity by breaking sentences into smaller segments and translating them while maintaining the source order as much as possible. SiST could benefit from this approach to balance quality and latency. However, current corpora used for simultaneous tasks often involve significant word reordering in translation, which is not ideal given that interpreters faithfully follow source syntax as much as possible. Inspired by conference interpreting by humans utilizing the salami technique, we introduce the Simul-MuST-C[1], a dataset created by leveraging the Large Language Model (LLM), specifically GPT-4o, which aligns the target text as closely as possible to the source text by using minimal chunks that contain enough information to be interpreted. Experiments on three language pairs show that the effectiveness of segmented-base monotonicity in training data varies with the grammatical distance between the source and the target, with grammatically distant language pairs benefiting the most in achieving quality while minimizing latency.

1 Introduction

Simultaneous speech translation (SiST) begins translating before the source inputs are fully received (Luong and Manning, 2015; Ma et al., 2019; Arivazhagan et al., 2019; Ren et al., 2020; Zeng et al., 2021). As waiting time increases, translation quality improves with more available inputs,

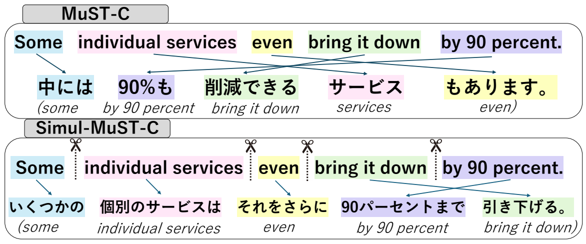

Figure 1: An example of an English-Japanese parallel sentence. In translations from MuST-C, the word order changes frequently, resulting in a reversed order compared to the source, as indicated by the arrows. On the other hand, translations from Simul-MuST-C, where the salami technique is applied to maintain monotonicity, preserve the source's word order as much as possible, as shown by the arrows.

but latency impacts negatively. Starting translation immediately reduces latency but limits available inputs and damages quality.

To address this trade-off between quality and latency, one might consider using a method by simultaneous interpreters, as they also process inputs in real-time. This technique, i.e., "salami technique" (Camayd-Freixas, 2011; Jones, 2015; Gillies, 2013; Yagi, 2000), divides a sentence into units that are as short as possible while ensuring each unit contains enough information to be interpreted clearly. Interpreters translate each segment into the target language, keeping that the output mirrors the source input syntax, which helps to speed up the translation process. This syntax manipulation on the target side is effective because the syntax is more flexible than word order across different languages (Camayd-Freixas, 2011). SiST could benefit from this technique by using simultaneous interpretation corpora made by professional interpreters, allowing a model to learn the segmented-base monotonicity through training with such real simultaneous interpretation data (Ko et al., 2023).

Despite the availability of several simultaneous interpretation corpora (Doi et al., 2021; Zhao et al.,

[1]The code is available at https://github.com/naist-nlp/SimulST. Please note that we provide dataset creation prompts and experimental code at this stage, with concerns regarding license terms as outlined in the License of Source Dataset under Section 10. Once we receive approval from the original dataset owner, we will also release the Simul-MuST-C corpus.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 22185–22205
November 12-16, 2024 ©2024 Association for Computational Linguistics

2024a; Matsubara et al., 2002), their sizes remain limited for effective model training. Collecting new data is challenging and costly because it requires simultaneous human interpreters. Moreover, interpreters employ tactics, e.g., summarization, and they make mistakes due to the intense time pressure and high cognitive load during interpretation (Shimizu et al., 2014; Camayd-Freixas, 2011). Relying on real simultaneous interpretation data is challenging due to frequent summarizations and omissions, which are unsuitable for model training. However, the data's monotonicity is necessary to balance latency and quality.

Therefore, we introduce a segment-base monotonic dataset of Simul-MuST-C (Simultaneous Multilingual Speech Translation Corpus) by rewriting existing multilingual speech translation corpora, MuST-C (Di Gangi et al., 2019) in Figure 1. Based on Sakai et al. (2024), we utilize the salami technique, used in conference interpreting, when prompting Large Language Models (LLMs) with GPT-4o. This technique involves dividing original sentences into shorter segments that contain enough information to be interpreted, reducing the word order changes in the target language. We investigate the effectiveness of salami technique in a computational approach for simultaneous tasks for multiple language pairs. Training models with Simul-MuST-C in speech-to-text settings improves latency minimization and translation quality for language pairs that are grammatically distant, whereas the improvement is less evident for pairs that are grammatically similar. Our contributions are as follows:

- We constructed Simul-MuST-C, a new large-scale training dataset for SiST, using the segment-based monotonic method, i.e., salami technique, across multiple language pairs: English-to-Japanese (En-Ja), English-to-German (En-De), and English-to-Chinese (En-Zh). Leveraging an LLM facilitated this process, indicating LLMs' potential understanding of its technique.

- We found that improving monotonicity correlates with improvements in quality and latency in SiST.

- We show effectiveness of the salami technique varies based on the grammatical distance between source and target languages. Grammatically distant language pairs benefit the most in achieving quality-latency tradeoff, indicating its potential applicability to other language pairs.

2 Background and Related Work

2.1 Simultaneous Speech Translation

In a SiST task, the model processes parts of the source inputs and produces parts of the target outputs step-by-step based on its decoding policies (Ren et al., 2020; Zeng et al., 2021; Agarwal et al., 2023). The policies are mainly categorized as fixed and adaptive. In fixed policies, e.g., wait-k policy (Ma et al., 2019), the model initially reads k tokens, and then changes reading token and writing token operation. In adaptive policies (Zheng et al., 2020; Liu et al., 2021; Zhang and Feng, 2022; Papi et al., 2023), the model reads and writes tokens according to its current source and target prefix. Among adaptive policies, local agreement (Liu et al., 2020) is the incremental decoding framework that splits an utterance into fixed-size chunks. When decoding each new chunk, it uses outputs from the previous chunk to guide the process, depending on prior predictions that align with the current output.

2.2 Handling Word Order Issue for Simultaneous Task

Unlike speech translation, which waits until all inputs are received, SiST starts translating with partial inputs. Despite this difference in translation timing between the two, speech translation corpora (Di Gangi et al., 2019) have been utilized for simultaneous speech translation shared task (Agarwal et al., 2023). Meanwhile, several studies highlight that translation data often requires significant word order reordering (Doi et al., 2021; Sakai et al., 2024; He et al., 2015, 2016).

This reordering is inappropriate for simultaneous tasks, as excessive reordering could result in forced anticipation and other undesirable outcomes. To deal with such word order issues, some studies have proposed rearranging sentences to align with the word order of the source language (He et al., 2015; Chen et al., 2021; Guo et al., 2023; Sakai et al., 2024). He et al. (2015) uses a rule-based method to rewrite sentences, adjusting reference translations to match the source language's word order. Applied to Japanese-to-English translation, this approach resulted in faster and better translations with more monotonic reference translations. Chen et al. (2021) proposes training the Simultane-

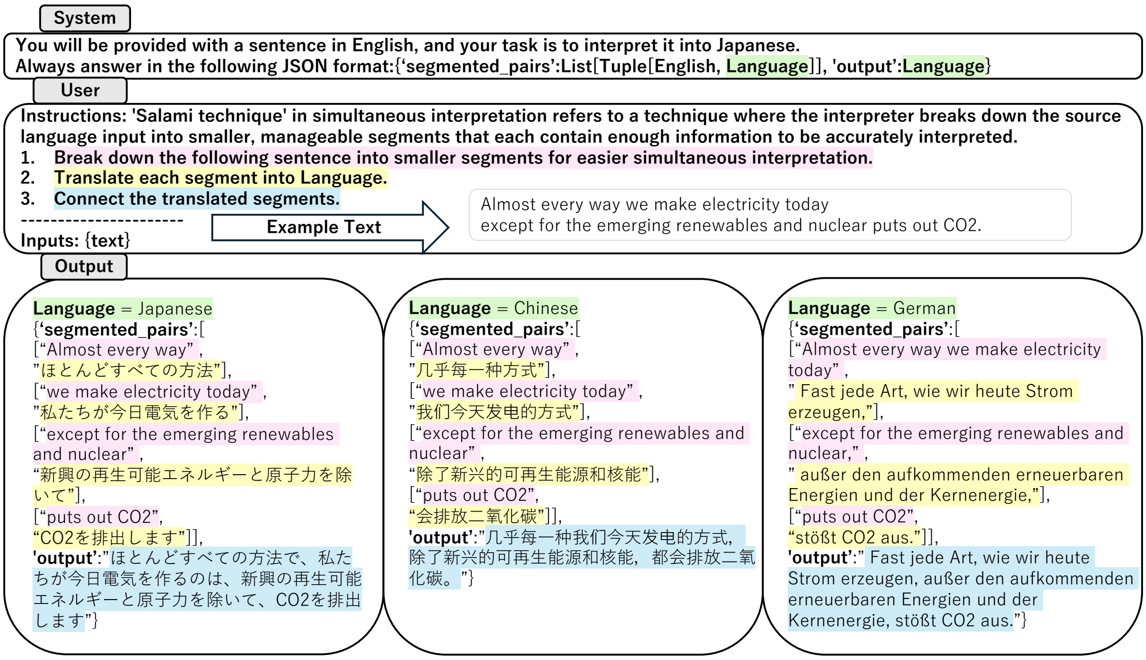

Figure 2: The prompt template and its example for constructing the Simul-MuST-C. The segmentation method is based on the salami technique used by simultaneous interpreters. Each colored line indicates each language, its prompt and corresponding outputs.

ous Machine Translation (SiMT) model with appropriate reference translations for each latency. This involves generating references using various wait-k policies and selecting the best pseudo-references through beam search, applied to both Chinese-to-English and Japanese-to-English translations. Guo et al. (2023) uses reinforcement learning with two reward functions to generate tailored references, managing word reordering and ensuring high-quality translations. This method, applied to English-to-Vietnamese, English-to-Romanian, and German-to-English, proved effective for both fixed and adaptive policies. Sakai et al. (2024) addresses the word order problem for En-Ja SiMT and SiST using LLM to rewrite references into a more monotonic form, based on Chunk-wise monotonic translation (CWMT) work (Okamura and Yamada, 2023; Fukuda et al., 2024), which segments sentences according to grammatical characteristics.

2.3 Salami Technique: Segmentation in Simultaneous Interpretation

The salami technique and its variant segmentation or chunking method is used by human simultaneous interpreters (Jones, 2015; Gillies, 2013; Yagi, 2000). This technique segments a long or complicated sentence into smaller, manageable chunks during the interpreting process, ensuring that each segmented unit contains adequate information for clear understanding. This method follows the original sentence structure as closely as possible and start translating so that it allows interpreters to avoid the extra time and concentration required for complex syntactic rearrangements. As a result, interpreters can translate each segment quickly and smoothly, making it possible to keep up with the speaker. Segmentation is crucial in simultaneous tasks, and several computational approaches in simultaneous translation have also addressed the segmentation issue in various ways (Shavarani et al., 2015; Siahbani et al., 2018; Fujita et al., 2013; Oda et al., 2014; Yarmohammadi et al., 2013).

A similar method, CWMT (Okamura and Yamada, 2023), is used for the En-Ja. It breaks sentences into manageable chunks based on grammatical features like clauses and conjunctions, translating them sequentially while preserving their order. This approach aims to balance translation latency and quality in simultaneous interpretation. Fukuda et al. (2024) describes a chunking workflow and creates a test dataset based on Okamura and Yamada (2023) rules.

3 Simul-MuST-C Construction with LLM

3.1 Prompt by Salami Technique

Inspired by the CWMT technique for dataset construction using an LLM (Sakai et al., 2024), we

Language Pair	Train	Dev	Test
En-Ja	328,639	1,369	2,841
En-De	250,942	1,415	2,580
En-Zh	358,853	1,349	2,841

Table 1: The overview of MuST-C v2 in En-Ja, En-De, En-Zh pairs. Each number indicates the number of lines. MuST-C v2 is used for Simul-MuST-C.

Language Pair	Data	Train	Dev	Test
En-Ja	MuST-C	0.572	0.552	0.522
	Simul-MuST-C	**0.815**	**0.826**	**0.803**
En-Zh	MuST-C	0.862	0.842	0.875
	Simul-MuST-C	**0.945**	**0.953**	**0.948**
En-De	MuST-C	0.923	0.935	0.938
	Simul-MuST-C	**0.972**	**0.971**	**0.970**

Table 2: The number shows the extent to which word order monotonicity has been achieved against the source. In all language pairs, word order monotonicity improved with the Simul-MuST-C dataset.

constructed Simul-MuST-C based on the salami technique used by a real simultaneous interpreter to handle simultaneous inputs. Our method involves a task definition and three steps (Figure 2).

Task Definition First, we define the task using the salami technique (Jones, 2015; Gillies, 2013; Yagi, 2000) to segment sentences into shorter ones containing enough information to be interpreted as in *Instructions*. We included this task definition to refine the prompt and make the request more specific and focused. In our preliminary study, we asked LLMs about the salami technique in simultaneous interpretation. We received detailed explanations similar to those found in Jones (2015). The example of its response is in the Appendix A. Based on this finding, we believe we could generate suitable monotonic text by utilizing the "salami technique" keyword and its knowledge.

Detailed Instructions Next, there are three steps to convert the translation to segmented-base monotonic translation. We specify the target language by adjusting the prompt in *System*, highlighted in green. First, the LLM[2] breaks down the segments into shorter ones to make simultaneous interpretation easier, colored in pink. Second, the LLM translates each segment, colored in yellow. Third, the LLM combines the translated segments into one sentence, colored in blue. We integrated the task definition and steps into a single prompt. The output is JSON to obtain results for each input[3].

3.2 Dataset Creation

We used MuST-C v2.0 (Di Gangi et al., 2019) for three language pairs: En-Ja, En-De, and En-Zh. These language pairs were selected from the eight available in MuST-C because these pairs represent varying degrees of word order differences from English. In addtion to that, they are covered in the IWSLT 2023 simultaneous speech-to-text translation task (Agarwal et al., 2023)[4]. For each target language, MuST-C consists of audio recordings from English TED Talks, which are automatically aligned at the sentence level with their manual transcriptions and translations (Must-C). This allows us to compare word order reordering between translations in Must-C and translations in Simul-MuST-C. Table 1 shows the number of datasets in the Simul-MuST-C for train, dev, and test for three language pairs. The total cost of data creation was 1,134 dollars.

4 Word Order Monotonicity Analysis

We compared word alignments between source and target sentences in both MuST-C and Simul-MuST-C translations to investigate word order differences. We used Awesome-Align (Dou and Neubig, 2021) for this comparison and evaluated word order monotonicity using Spearman's correlation coefficient. As shown in Table 2, Simul-MuST-C has improved monotonicity compared to translations in MuST-C across all three language pairs. However, the extent of this improvement varies among language pairs.

En-Ja Table 2 shows that word order monotonicity in Simul-MuST-C training data is 81.5%, whereas it is 57.2% in MuST-C training data for En-Ja, which demonstrates the most improvement in word order monotonicity. Table 3 in En-Ja provides an example of word order monotonicity between MuST-C and Simul-MuST-C, in which the semantically similar phrase (4) "at the 60 to 80 percent level" appears at the beginning for MuST-C, indicating excessive reordering, whereas in Simul-MuST-C, (4) "at the 60 to 80 percent level" appears later, closer to its position in the source.

[2]We used GPT-4o (OpenAI et al., 2024) (2024-05-13 ver.).

[3]We used batch API (`https://platform.openai.com/docs/guides/batch`) for cost-effective creation.

[4]`https://iwslt.org/2023/simultaneous`

En-Ja	Source	(1) Now, / (2) we have some pilot things / (3) that do this / (4) at the 60 to 80 percent level.
	MuST-C	(4) 60%から80%のレベルで（*at the 60 to 80 percent level*）/ (3) この処理を行う（*do this*）/ (2) 試験運用を（*pilot things*）/ (3) 行っています（*do*）。
	Simul-MuST-C	(1) 今 (*now*)、/ (2) いくつかの試験的なものがあり (*we have some pilot things*)、/ (3) これを（*this*）/ (4) 60から80パーセントのレベルで（*at the 60 to 80 percent level*）/ (3) 行います（*do*）。
En-Zh	Source	(1) I / (2) grew up / (3) on a steady diet of / (4) science fiction.
	MuST-C	(1) 我是 (I) / (4) 在科幻小说 (*science fiction*) / (3) 的陪伴下 (*accompanied by*) / (2) 长大的(*grew up*)。
	Simul-MuST-C	(1) 我 (*I*) / (2) 长大在 (*grew up*) / (3) 稳定的饮食 (*a steady diet*) / (4) 科幻小说(*science fiction*)。
En-De	Source	(1) These are / (2) what people / (3) often / (4) refer to as / (5) the renewable sources.
	MuST-C	(1) Es sind (*there are*) / (5) die Erneuerbaren Energien (*renewable energies*), / (3) wie sie oft (*as they often*) / (4) genannt warden (*be called*).
	Simul-MuST-C	(1) Dies sind (*These are*), / (2) was die Leute (*what people*) / (3) oft als die (*often*) / (5) erneuerbaren Quellen (*renewable energies*) / (4) bezeichnen (*describe*).

Table 3: An example of word order monotonicity between MuST-C and Simul-MuST-C in En-Ja, En-Zh, En-De.

En-Zh Similarly, Table 2 shows that word order monotonicity in Simul-MuST-C's training data is 94.5%, while MuST-C's training data is 86.2%, for En-Zh. This monotonicity improvement is relatively small when compared to the En-Ja pair. The En-Zh example in Table 3 shows that the phrase (4) "science fiction" appears at the front, indicating word reordering for MuST-C, whereas in Simul-MuST-C, (4) "science fiction" appears later, matching its position in the source.

En-De The monotonicity for Simul-MuST-C and MuST-C are 97.2% and 92.3%, respectively, for En-De. The monotonicity improvement is the smallest among the three language pairs, but monotonicity is already high in MuST-C. The En-De example in Table 3 shows that, in MuST-C, the semantically similar phrase (5) "the renewable sources" appears at the beginning, indicating reordering, whereas in Simul-MuST-C, (5) "the renewable sources" appears later, closer to its position in the source. Simul-MuST-C successfully aligns to the source word order more, even though monotonicity is already high in MuST-C.

5 Experimental Setup

To evaluate the contribution of Simul-MuST-C to improving the quality-latency trade-off, we compare two models: one trained with MuST-C and the other with Simul-MuST-C. For clarity in our analysis, we present the results of the wait-k (Ma et al., 2019) policy. We also evaluated based on the Local Agreement (Liu et al., 2020). We describe its differences from wait-k and provide corresponding analyses in Appendix E.

Dataset For the training dataset, we used MuST-C v2.0 (Di Gangi et al., 2019) for three language pairs: En-{Ja, Zh,-De} as the baseline, and Simul-MuST-C, which is built upon on MuST-C v2.0, applying the salami technique. For evaluation, we used the tst-COMMON from MuST-C v2.0.

Training and Decoding We implemented an end-to-end speech-to-text model initialized with two pre-trained models for its speech encoder and text decoder using Fairseq (Ott et al., 2019), integrated into a Transformer architecture (Vaswani et al., 2017). Following the settings from Fukuda et al. (2023), we used HuBERT-Large (Hsu et al., 2021) as speech encoder, and mBART50 (Tang et al., 2021) as text decoder. We tokenized all text data in the corpora using a multilingual SentencePiece tokenizer (Kudo and Richardson, 2018) with a vocabulary of 250,000 subwords, distributed with the mBART50 pre-trained model. We validate the trained model every 500 steps and set 8 as the early stopping. For the SimulST decoding policy, we employed wait-k values ranging from $\{3, 5, 7, 9, 11, 13, 15, 17\}$, with one unit set to 160 frames, adjusting the trade-off between quality and latency. Hypotheses for input chunks were generated using a beam search with the size of five. We also included the offline model performance for each decoding policy for comparison purposes.

Evaluation For quality, We used four distinct metrics, which were chosen because each evaluates using different criteria: BLEU (Papineni et al., 2002), BLEURT (Sellam et al., 2020), COMET (Rei et al., 2020), and COMET-QE (Rei et al., 2021). For latency, we evaluated latency using the SimulEval (Ma et al., 2020) toolkit. We selected Average Lagging (AL) (Ma et al., 2019), Length Adaptive Average Lagging (LAAL) (Papi et al., 2022), and Average Token Delay (ATD), following the standard practice in IWSLT 2024 [5]. Each metric's features and criteria on both quality and latency are described in Appendix B.

6 Experimental Results on Wait-k Policy

En-Ja Figure 3 shows the results for En-Ja. With a focus on COMET-QE_ATD, the latency gap in ATD between MuST-C and Simul-MuST-C widens as k increases, indicating that Simul-MuST-C not only starts but also finishes translations faster compared to MuST-C. Despite finishing translations faster, Simul-MuST-C's translation quality, as shown by COMET-QE, is better than MuST-C. In SiST scenarios, where delays in translation can negatively impact subsequent inputs, Simul-MuST-C enables faster completion of translations while maintaining the quality observed in the results.

When evaluated offline using COMET-QE, both models achieve similar quality. This suggests that COMET-QE assesses performance directly from the source and target without requiring references, making it unaffected by offline translation style in the reference. However, when using reference-based metrics, a significant quality gap exists. Specifically, with BLEU, the quality difference between MuST-C and Simul-MuST-C is around 5 points, suggesting that BLEU may be strongly influenced by translation style in reference. This discrepancy between reference-free and reference-based metrics highlights the need for references better suited to simultaneous translation settings.

En-Zh Figure 4 shows that Simul-MuST-C outperforms MuST-C for En-Zh. When focusing on BLEU and BLEURT, COMET, COMET-QE, the quality gap in BLEU is larger than in BLEURT, COMET, COMET-QE. Since both the training and evaluation data originate from MuST-C, MuST-C might be expected to align more closely with the test, potentially enhancing BLEU. However, the results show that Simul-MuST-C achieves a closer surface-level match to the test than MuST-C across all k. With a focus on BLEU, COMET-QE_ATD, translation by Simul-MuST-C starts and ends faster while maintaining quality. This is the same trend we observed in En-Ja, which is ideal for SiST.

When focusing on offline, the results are relatively similar, except that MuST-C performs better in BLEU. However, Simul-MuST-C outperforms MuST-C in all wait-k settings, indicating that Simul-MuST-C is better suited for simultaneous translation, while MuST-C is better for offline translation. Additionally, En-Zh may also be affected by offline translation style in the reference, similar to the En-Ja. This is because there is almost no quality gap in reference-free metrics, whereas a slight gap appears in BLEU. However, compared to the En-Ja, the quality gap between the two types of metrics is smaller, probably due to the lesser difference in word order.

En-De Figure 5 shows the results for En-De. Focusing on BLEU and COMET-QR, Simul-MuST-C shows a slight advantage, especially as k increases. This trend is consistent with our findings in En-Zh. While surface-based evaluation metrics and semantic similarity evaluation metrics could show different tendencies sometimes, they correlates in this case. These results suggest that Simul-MuST-C slightly but consistently outperforms MuST-C in quality. With a focus on ATD, both MuST-C and Simul-MuST-C achieve nearly the same latency level, indicating similar handling of the start and end timing of translation. This suggests that Simul-MuST-C does not provide an improvement, as its results are comparable to MuST-C.

In terms of offline quality, performance is relatively comparable, with MuST-C outperforming Simul-MuST-C in BLEU, while Simul-MuST-C shows a slight advantage in COMET-QE. However, in the simultaneous setting, Simul-MuST-C consistently performs better. This pattern is also evident in En-Ja and En-Zh, though the quality gap in En-De is the smallest of the three language pairs, likely due to differences in word order. The word order gap is smallest in the en-de pair, which may explain why Simul-MuST-C is effective, although its impact is limited, as reflected by the slight word order improvement shown in Table 2.

Summary In terms of quality, Simul-MuST-C showed better across all three language pairs in

[5]https://iwslt.org/2024/simultaneous

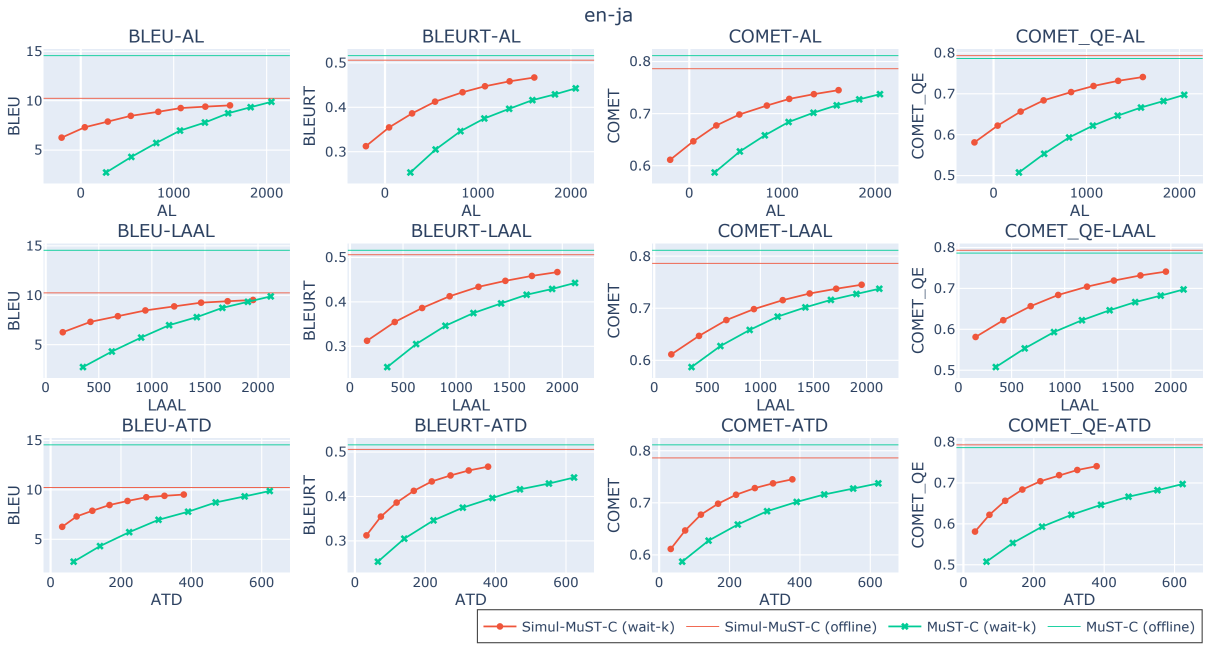

Figure 3: The results for En-Ja on the tst-COMMON. Each plot, from left to right, represents wait-k values ranging from $3, 5, 7, 9, 11, 13, 15, 17$.

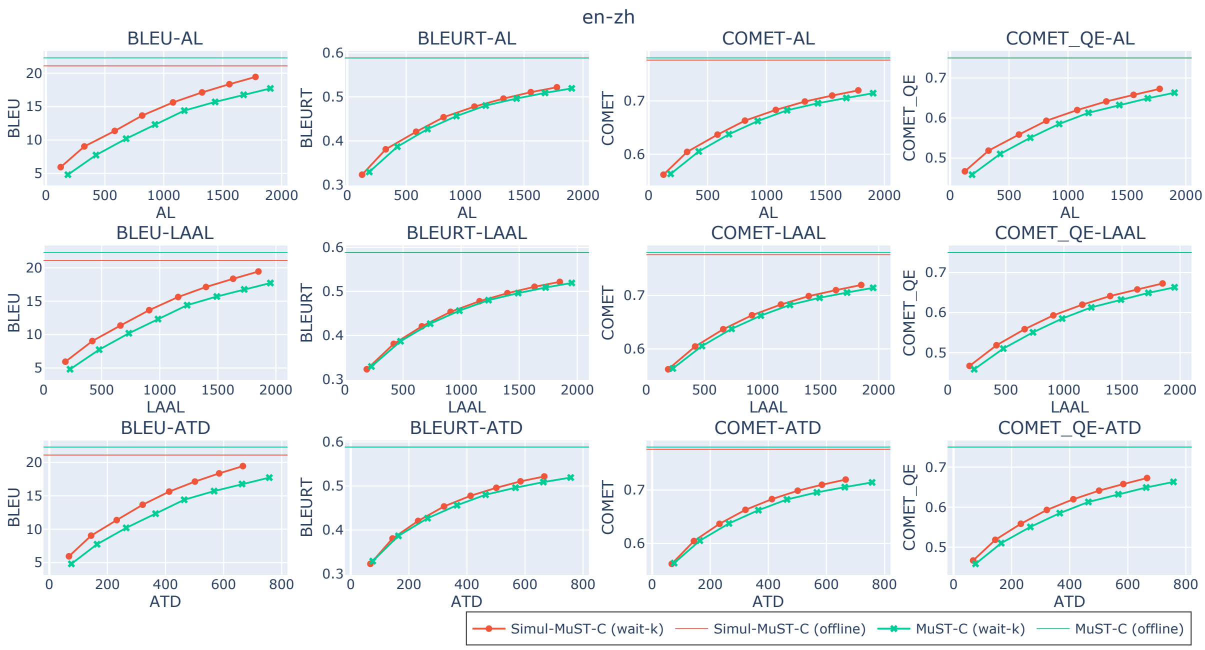

Figure 4: The results for the En-Zh the tst-COMMON. Each plot, from left to right, represents wait-k values ranging from $3, 5, 7, 9, 11, 13, 15, 17$.

reference-free metrics. However, in metrics that require a reference, the results varied depending on the language pair and the specific metric. Some results in BLEURT tend to show MuST-C is better, while others showed that Simul-MuST-C was better. Reference-based metrics may favor the offline translation style because the references used for evaluation do not need to maintain monotonicity between the source and target languages. Moreover, tst-COMMON is also from the same source, MuST-C, suggesting that the provided references are also from offline translations. Given that the comparison involves MuST-C, which was trained on the same source data as the tst-COMMON test data used in this evaluation, it's possible that MuST-C results appear more domain-adapted when using

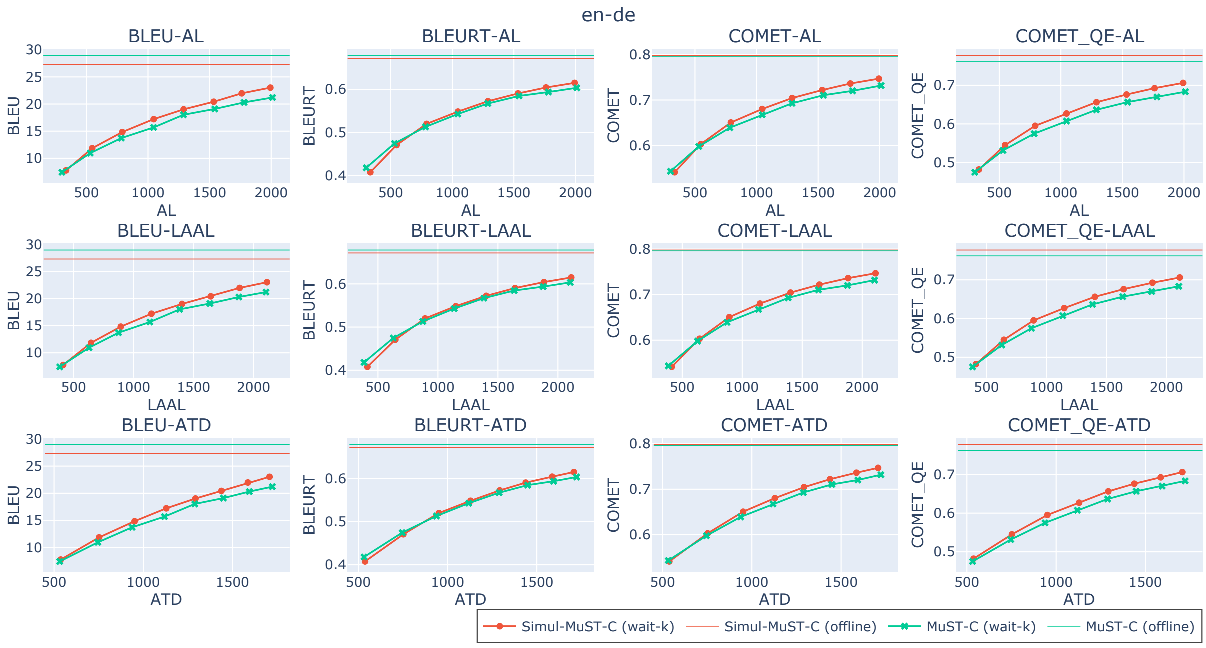

Figure 5: The results for the En-De on the tst-COMMON. Each plot, from left to right, represents wait-k values ranging from $3, 5, 7, 9, 11, 13, 15, 17$.

reference-based evaluation. Regarding latency, it was evident in En-Ja, slightly improved in En-Zh, and not observed in En-De. The result aligns with the degree of word order improvement in the training data, in which the highest improvements were observed for En-Ja, a little improvement was seen for En-Zh, and almost no differences were found for En-De in Table 2. More detailed analyses on each language pair are in Appendix C.

Comparing offline and simultaneous settings, the results across all three language pairs indicate that Simul-MuST-C performs better in simultaneous settings, while MuST-C excels in offline settings, as evidenced by BLEU scores. These findings suggest that Simul-MuST-C is more suited for simultaneous settings, whereas MuST-C is better for offline settings. Additionally, the current test data may be insufficient for evaluating simultaneous translation; test data should more accurately reflect the conditions of simultaneous translation such as word order monotonicity.

Language Pair	Model	Monotonicity	BLEU	BLEURT	COMET -QE
En-Ja	Original	0.565	5.72	0.346	0.593
	Ours	**0.770**	**7.88**	**0.386**	**0.657**
En-Zh	Original	0.878	10.2	**0.427**	0.551
	Ours	**0.912**	**11.36**	0.421	**0.558**
En-De	Original	0.908	13.72	0.513	0.575
	Ours	**0.928**	**14.83**	**0.520**	**0.650**

Table 4: The table shows the word order monotonicity of generated sentences and their corresponding quality with a k value of 7 in the wait-k setting on tst-COMMON. “Original” refers to the model trained with MuST-C, and “Ours” refers to the model trained with Simul-MuST-C.

7 Discussion

7.1 Generated Sentences Analysis

Table 4 shows the difference in word order monotonicity between sentences generated by MuST-C and Simul-MuST-C, and the corresponding quality under the wait-k setting on k = 7. Simul-MuST-C achieved better monotonicity for all language pairs, with varying degrees of improvement across them. En-Ja demonstrated the most significant improvement, followed by En-Zh, while En-De showed the smallest improvement. Table 5 is a generated sentence example for En-Zh. Focusing on the word position of (2) "program", the sentence generated using Simul-MuST-C places it in the same position as in the source, whereas MuST-C places (2) "program" at the end of the sentence, indicating word reordering. This example indicates that Simul-MuST-C contributes to aligning to source word order as much as possible, whereas reordering is more likely to occur in MuST-C. Examples of generated sentences in other language

Source	(1) There is / (2) a program / (3) that some of you / (4) might have heard of.
MuST-C	(1) 有一个 (*there is a*) / (3) 你们 (*you*) / (4)可能听过的 (*might have heard of*) / (2) 项目 (*program*).
Simul-MuST-C	(1) 有一个 (*there is a*) / (2) 项目 (*program*) / (3) 你们中的一些人 (*some of you*) / (4) 可能听说过 (*might have heard of*)).

Table 5: An example of generated sentences focusing word order monotonicity between MuST-C and Simul-MuST-C in En-Zh pair shows that in MuST-C, the semantically similar word (2) "program" appears at the end, indicating excessive reordering, whereas in Simul-MuST-C, the word (2) "program" maintains the same order as in the source.

pairs are in Appendix D. This example suggests that Simul-MuST-C contributes to monotonicity, resulting in latency reduction. However, it's important to note that aligning with the word order of the source language excessively could result in unnatural translations for the target side. This issue becomes more critical when the language pair is grammatically different, although such alignment with the source language's word order was found to be most effective in such grammatically distant pairs, e.g., En-Ja. To address the trade-off between minimizing disparities in word or phrase order between the source and target languages and preserving the naturalness of the target language, future research may consider creating test sets using the salami technique for SiST across multiple language pairs.

7.2 Is segmentation-base monotonicity effective in any language pairs?

The effectiveness of segmentation-based monotonicity on the target side varies among En-Ja, En-Zh, and En-De. The results indicate that this method is effective to balance quality and latency for all language pairs considered. However, the degree of effectiveness depends on the language pair. Among the three, En-Ja benefits the most from segmentation monotonicity. This is due to the significant grammatical differences between English (SVO) and Japanese (SOV), as highlighted by our analysis in Table 2. While En-Zh and En-De pairs also demonstrate effectiveness, the word order differences are not as evident compared to En-Ja. Thus, En-Ja benefits this segmentation the most, whereas, in other language pairs, the effectiveness may vary. Overall, segmentation-based monotonicity proves effective especially when the language pair is grammatically distant, and has the potential to be applied to multiple language pairs and directions.

8 Conclusion

We proposed Simul-MuST-C, a dataset, and a method to rearrange sentences into segmentation-based monotonic data for simultaneous speech translation using LLMs in En-{Ja, Zh, De}. This method, based on the salami technique used in conference interpreting, showed that Simul-MuST-C improves quality and latency, especially in grammatically distant language pairs, indicating a correlation between word order monotonicity and quality-latency improvement. Using LLMs is cost-effective and helps address the scarcity of such datasets, which require extensive human labor. Future work will expand this dataset to end-to-end speech-to-speech translation.

9 Limitations

What is the Ideal Degree of Monotonicity? Simul-MuST-C aims to align closely with the word or phrase order of the source, but not to achieve 100 percent monotonicity, as perfect monotonicity can result in unnaturalness in the target language. To maintain naturalness, some reordering is allowed. This trade-off balances monotonicity with the source and naturalness in the target language. Table 2 shows improvements in monotonicity from MuST-C to Simul-MuST-C, particularly in En-Ja, indicating effective management of the trade-off between monotonicity and naturalness. The optimal level of monotonicity depends on factors like content and input speed, but this study shows that improvements in monotonicity correlate with better latency and quality in SiST.

Scalability of the other language pairs We focused on En-{Ja, zh, De}, following the simultaneous track of IWSLT2023[6]. The proposed corpus construction method for SiST could be applied to many other language pairs. However, our experimental results show it improves quality-latency for grammatically distant pairs (e.g., En-Ja) but have a

[6] https://iwslt.org/2023/simultaneous

limited impact on similar pairs (e.g., En-De). The scarcity of multilingual corpora for SiST remains a challenge for applying the method broadly. Therefore, addressing these constraints is necessary for broader application.

Evaluation Dataset for SiST The evaluation data for the SiST system commonly uses the tst-COMMON from the MuST-C corpus for speech translation. However, such test data is inappropriate for SiST (Sakai et al., 2024; Doi et al., 2024; Zhao et al., 2021). Simultaneous interpretation data curated by humans could be an alternative, but it is also unsuitable for system evaluation (Zhao et al., 2024b; Doi et al., 2024) because it contains critical errors such as omissions or summarizations, due to the high cognitive overload and intense time pressure faced by interpreters. In our research, we used tst-COMMON from the MuST-C corpus, however, tst-COMMON may be inappropriate for SiST evaluation either since the reference for tst-COMMON is offline translation, which includes frequent reordering. Using reference-based metrics with such test data may be biased toward the offline translation style. Therefore, we believe that evaluation data specifically designed for SiST is necessary, and we call for such data to expand this research area.

Applicability for local LLMs We used GPT-4o for dataset construction and designed the prompts specifically for its capabilities. As a result, these prompts may require some adjustments to work effectively with other LLMs. Nonetheless, our study aims to develop methods that could be applied across various languages. Therefore, despite being optimized for GPT-4o, our prompts retain enough flexibility to be useful with other language models, thereby fulfilling our objective.

10 Ethical Considerations

License of Source Dataset Simul-MuST-C originates from MuST-C[7], which is governed by the CC BY-NC-ND 4.0 license [8]. Under this license, "NoDerivatives" implies that any modifications, remixes, or transformations cannot be distributed. Consequently, we can make internal adjustments without distributing them and include examples within the paper. MuST-C itself is from TED Talk data and inherits the same CC BY-NC-ND 4.0 license. When we unveil exclusively the disparities between Simul-MuST-C and MuST-C, we will explicitly outline the source information along with the CC BY-NC-ND 4.0 license. Out of ethical considerations, we intend to release it only after securing permission or arranging with the MuST-C administrators. We will refrain from releasing the Simul-MuST-C corpus until the necessary permissions are obtained. Providing the experiment code poses no issues, enabling the replication of the corpus. Hence, even if making the data publicly available is deemed unfeasible, we are confident in the reproducibility of Simul-MuST-C.

Ownership rights about Simul-MuST-C The Simul-MuST-C was created GPT-4o and is therefore subject to OpenAI's license terms[9]. OpenAI assigns to us all rights, titles, and interests in and to the output.

Moderations Simul-MuST-C is free of harmful information, sourced from TED Talks. Moreover, our check with OpenAI Moderation APIs[10]found no harmful content.

References

Milind Agarwal, Sweta Agrawal, Antonios Anastasopoulos, Luisa Bentivogli, Ondřej Bojar, Claudia Borg, Marine Carpuat, Roldano Cattoni, Mauro Cettolo, Mingda Chen, William Chen, Khalid Choukri, Alexandra Chronopoulou, Anna Currey, Thierry Declerck, Qianqian Dong, Kevin Duh, Yannick Estève, Marcello Federico, Souhir Gahbiche, Barry Haddow, Benjamin Hsu, Phu Mon Htut, Hirofumi Inaguma, Dávid Javorský, John Judge, Yasumasa Kano, Tom Ko, Rishu Kumar, Pengwei Li, Xutai Ma, Prashant Mathur, Evgeny Matusov, Paul McNamee, John P. McCrae, Kenton Murray, Maria Nadejde, Satoshi Nakamura, Matteo Negri, Ha Nguyen, Jan Niehues, Xing Niu, Atul Kr. Ojha, John E. Ortega, Proyag Pal, Juan Pino, Lonneke van der Plas, Peter Polák, Elijah Rippeth, Elizabeth Salesky, Jiatong Shi, Matthias Sperber, Sebastian Stüker, Katsuhito Sudoh, Yun Tang, Brian Thompson, Kevin Tran, Marco Turchi, Alex Waibel, Mingxuan Wang, Shinji Watanabe, and Rodolfo Zevallos. 2023. FINDINGS OF THE IWSLT 2023 EVALUATION CAMPAIGN. In *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*, pages 1–61, Toronto, Canada (in-person and online). Association for Computational Linguistics.

[7] `https://mt.fbk.eu/must-c`

[8] `https://creativecommons.org/licenses/by-nc-nd/4.0`

[9] `https://openai.com/policies/terms-of-use`

[10] `https://platform.openai.com/docs/guides/moderation`

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*, pages 1313–1323, Florence, Italy. Association for Computational Linguistics.

Erik Camayd-Freixas. 2011. Cognitive theory of simultaneous interpreting and training. In *Proceedings of AMTA*.

Junkun Chen, Renjie Zheng, Atsuhito Kita, Mingbo Ma, and Liang Huang. 2021. Improving simultaneous translation by incorporating pseudo-references with fewer reorderings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5857–5864, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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.

Kosuke Doi, Yuka Ko, Mana Makinae, Katsuhito Sudoh, and Satoshi Nakamura. 2024. Word order in english-japanese simultaneous interpretation: Analyses and evaluation using chunk-wise monotonic translation. *Preprint*, arXiv:2406.08940.

Kosuke Doi, Katsuhito Sudoh, and Satoshi Nakamura. 2021. Large-scale English-Japanese simultaneous interpretation corpus: Construction and analyses with sentence-aligned data. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 226–235, Bangkok, Thailand (online). Association for Computational Linguistics.

Zi-Yi Dou and Graham Neubig. 2021. Word alignment by fine-tuning embeddings on parallel corpora. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2112–2128, Online. Association for Computational Linguistics.

T. Fujita, Graham Neubig, Sakriani Sakti, T. Toda, and Satoshi Nakamura. 2013. Simple, lexicalized choice of translation timing for simultaneous speech translation. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pages 3487–3491.

Ryo Fukuda, Kosuke Doi, Katsuhito Sudoh, and Satoshi Nakamura. 2024. Test data creation in simultaneous machine translation in english to japanese pair: Insights from simultaneous interpretation tactics. *IPSJ SIG Technical Report*. (In Japanese).

Ryo Fukuda, Yuta Nishikawa, Yasumasa Kano, Yuka Ko, Tomoya Yanagita, Kosuke Doi, Mana Makinae, Sakriani Sakti, Katsuhito Sudoh, and Satoshi Nakamura. 2023. NAIST simultaneous speech-to-speech translation system for IWSLT 2023. In *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*, pages 330–340, Toronto, Canada (in-person and online). Association for Computational Linguistics.

Andrew Gillies. 2013. Conference interpreting: A student's practice book. In *Routledge*.

Shoutao Guo, Shaolei Zhang, and Yang Feng. 2023. Simultaneous machine translation with tailored reference. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3070–3084, Singapore. Association for Computational Linguistics.

He He, Jordan Boyd-Graber, and Hal Daumé III. 2016. Interpretese vs. translationese: The uniqueness of human strategies in simultaneous interpretation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 971–976, San Diego, California. Association for Computational Linguistics.

He He, Alvin Grissom II, John Morgan, Jordan Boyd-Graber, and Hal Daumé III. 2015. Syntax-based rewriting for simultaneous machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 55–64, Lisbon, Portugal. Association for Computational Linguistics.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460.

Roderick Jones. 2015. Conference interpreting explained. In *Routledge*.

J. Kahn, M. Rivière, W. Zheng, E. Kharitonov, Q. Xu, P. E. Mazaré, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen, T. Likhomanenko, G. Synnaeve, A. Joulin, A. Mohamed, and E. Dupoux. 2020. Libri-light: A benchmark for asr with limited or no supervision. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7669–7673. `https://github.com/facebookresearch/libri-light`.

Yuka Ko, Ryo Fukuda, Yuta Nishikawa, Yasumasa Kano, Katsuhito Sudoh, and Satoshi Nakamura. 2023. Tagged end-to-end simultaneous speech translation training using simultaneous interpretation data. In *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*, pages 363–375, Toronto, Canada (in-person and online). Association for Computational Linguistics.

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.

Dan Liu, Mengge Du, Xiaoxi Li, Ya Li, and Enhong Chen. 2021. Cross attention augmented transducer networks for simultaneous translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 39–55, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Danni Liu, Gerasimos Spanakis, and Jan Niehues. 2020. Low-Latency Sequence-to-Sequence Speech Recognition and Translation by Partial Hypothesis Selection. In *Proc. Interspeech 2020*, pages 3620–3624.

Minh-Thang Luong and Christopher Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the 12th International Workshop on Spoken Language Translation: Evaluation Campaign*, pages 76–79, Da Nang, Vietnam.

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. 2020. 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.

Shigeki Matsubara, Akira Takagi, Nobuo Kawaguchi, and Yasuyoshi Inagaki. 2002. Bilingual spoken monologue corpus for simultaneous machine interpretation research. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, Las Palmas, Canary Islands - Spain. European Language Resources Association (ELRA).

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.

Yuki Okamura and Masaru Yamada. 2023. Jyun okuri yaku" no kihan to mohan doji tsuyaku wo mohan tosita kyoikuron no shiron. In Hiroyuki Ishizuka, editor, *Word Order in English-Japanese Interpreting and Translation: The History, Theory and Practice of Progressive Translation*, pages 217–250. Hitsuji Syobo.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex

Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

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.

Sara Papi, Marco Gaido, Matteo Negri, and Marco Turchi. 2022. Over-generation cannot be rewarded: Length-adaptive average lagging for simultaneous speech translation. In *Proceedings of the Third Workshop on Automatic Simultaneous Translation*, pages 12–17, Online. Association for Computational Linguistics.

Sara Papi, Matteo Negri, and Marco Turchi. 2023. Attention as a guide for simultaneous speech translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13340–13356, Toronto, Canada. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Ricardo Rei, Ana C Farinha, Chrysoula Zerva, Daan van Stigt, Craig Stewart, Pedro Ramos, Taisiya Glushkova, André F. T. Martins, and Alon Lavie. 2021. Are references really needed? unbabel-IST 2021 submission for the metrics shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 1030–1040, Online. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

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.

Yusuke Sakai, Mana Makinae, Hidetaka Kamigaito, and Taro Watanabe. 2024. Simultaneous interpretation corpus construction by large language models in distant language pair. *Preprint*, arXiv:2404.12299.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Hassan Shavarani, Maryam Siahbani, Ramtin Mehdizadeh Seraj, and Anoop Sarkar. 2015. Learning segmentations that balance latency versus quality in spoken language translation. In *Proceedings of the 12th International Workshop on Spoken Language Translation: Papers*, pages 217–224, Da Nang, Vietnam.

Hiroaki Shimizu, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. 2014. Collection of a simultaneous translation corpus for comparative analysis. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 670–673, Reykjavik, Iceland. European Language Resources Association (ELRA).

Maryam Siahbani, Hassan Shavarani, Ashkan Alinejad, and Anoop Sarkar. 2018. Simultaneous translation using optimized segmentation. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 154–167, Boston, MA. Association for Machine Translation in the Americas.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2021. Multilingual translation from denoising pre-training. In *Findings of the Association* *for Computational Linguistics: ACL-IJCNLP 2021*, pages 3450–3466, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Sane Yagi. 2000. Studying style in simultaneous interpretation. *Meta*, 45(3):520–547.

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.

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

Shaolei Zhang and Yang Feng. 2022. 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.

Jinming Zhao, Philip Arthur, Gholamreza Haffari, Trevor Cohn, and Ehsan Shareghi. 2021. It is not as good as you think! evaluating simultaneous machine translation on interpretation data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6707–6715, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jinming Zhao, Yuka Ko, Kosuke Doi, Ryo Fukuda, Katsuhito Sudoh, and Satoshi Nakamura. 2024a. Naist-sic-aligned: an aligned english-japanese simultaneous interpretation corpus. *Preprint*, arXiv:2304.11766.

Jinming Zhao, Katsuhito Sudoh, Satoshi Nakamura, Yuka Ko, Kosuke Doi, and Ryo Fukuda. 2024b. NAIST-SIC-aligned: An aligned English-Japanese simultaneous interpretation corpus. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 12046–12052, Torino, Italia. ELRA and ICCL.

Baigong Zheng, Kaibo Liu, Renjie Zheng, Mingbo Ma, Hairong Liu, and Liang Huang. 2020. Simultaneous translation policies: From fixed to adaptive. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2847–2853, Online. Association for Computational Linguistics.

A Does LLM Understand Salami Technique?

In our preliminary study, we found that LLM has the potential understanding the salami technique used by simultaneous interpreters. Table 6 shows an example of its prompt and the answer.

B Detailed Experimental Settings

Training and Decodings We implemented an end-to-end speech-to-text model initialized with two pre-trained models for its speech encoder and text decoder using Fairseq (Ott et al., 2019), integrated into a Transformer architecture (Vaswani et al., 2017), following the settings from Fukuda et al. (2023)[11], the speech encoder was initialized with HuBERT-Large (Hsu et al., 2021), comprising a feature extractor trained on 60K hours of unlabeled speech data from Libri-Light (Kahn et al., 2020) and Transformer encoder layers. The feature extractor has seven convolutional layers with kernel sizes of (10, 3, 3, 3, 3, 2, 2), strides of (5, 2, 2, 2, 2, 2, 2), and 512 channels. The text decoder was initialized with the decoder of mBART50 (Tang et al., 2021). We tokenized all text data in the corpora using a multilingual SentencePiece tokenizer (Kudo and Richardson, 2018) with a vocabulary of 250,000 subwords, distributed with the mBART50 pre-trained model. We validate the trained model every 500 steps and set 8 as the early stopping. The maximum source length is set to 400,000 frames, and the target length is set to 1,024 tokens. Gradient accumulation and data-parallel computations were employed to achieve a batch size of approximately 32 million tokens. A single NVIDIA A6000 (48GB) was used for three days. The decoder consists of 12 Transformer layers, with embedding and linear projection weights shared, totaling a size of 250,000. Each Transformer layer and feed-forward layer has a size of 1,024 and 4,096, respectively, with 16 attention heads. The activation function is ReLU, and layer normalization is applied before the attention operations. Speech input is provided as waveforms with a 16 kHz sampling rate, normalized to zero mean and unit variance. During training, each source audio was augmented before normalization, with a probability of 0.8. We used Adam with $\beta 1 = 0.99$, $\beta 2 = 0.98$, and the learning rate is 2.5e-4.

[11]We used the NAIST-SimulST framework (`https://github.com/ahclab/naist-simulst`) and followed the default parameter settings.

Query	What is salami technique in simultaneous interpretation?
Answer	The "salami technique" in simultaneous interpretation refers to a method where the speaker breaks down their speech into smaller, more manageable segments or "slices," similar to how a salami might be sliced. This technique is often employed to facilitate the interpreter's work, making it easier to keep up with the speaker and maintain accuracy in real-time translation. Here's how it generally works: 1. **Shorter Phrases:** The speaker consciously uses shorter sentences or phrases, pausing slightly between each one. This allows the interpreter to process and translate each segment without falling behind. 2. **Natural Pauses:** By incorporating natural pauses and breaks into their speech, the speaker provides the interpreter with regular opportunities to catch up and ensure that the audience receives a coherent and accurate translation. 3. **Clarity and Pacing:** The speaker pays close attention to clarity and pacing, avoiding overly complex or lengthy sentences that might overload the interpreter. The salami technique is particularly useful in high-stakes, fast-paced environments where precision and timeliness are critical, such as international conferences, diplomatic negotiations, and business meetings. This approach helps ensure that both the speaker's message and the interpreter's translation remain clear and accurate, benefiting the entire audience.

Table 6: The prompt and its answer for salami technique .

For the SimulST decoding policy, we employed wait-k values ranging from $\{3, 5, 7, 9, 11, 13, 15, 17\}$, with one unit set to 160 ms, adjusting the trade-off between quality and latency. We utilized local agreement with n = 2 (LA-2). The chunk size is from 400, 600, 800, and 1000 frames to balance quality-latency. Hypotheses for input chunks were generated using a beam search with a beam size of five.

Quality Evaluation We evaluate translation quality using four distinct metrics, which were chosen because each evaluates using different criteria: surface-level textual similarity, surface-level-free semantic similarity, and the necessity of a reference or source. BLEU (Papineni et al., 2002) evaluates translations based on surface-level n-gram matching between the reference sentences and generated sentences. BLEURT (Sellam et al., 2020) evaluates the semantic similarity between generated and reference sentences based on embeddings from language models. COMET (Rei et al., 2020) uses sentence-level embeddings of the hypothesis, reference, and input, leveraging a multilingual pretrained model. COMET-QE (Rei et al., 2021), an extension to reference-free evaluation, uses a multilingual embedding model to eliminate dependence on the reference and evaluates the similarity between the source and generated sentences directly.

Latency Evaluation We evaluated latency using the SimulEval (Ma et al., 2020) toolkit. We selected Average Lagging (AL) (Ma et al., 2019), Length Adaptive Average Lagging (LAAL) (Papi et al., 2022), and Average Token Delay (ATD), following the standard practice in IWSLT 2024 [12]. AL measures translation start times. LAAL also evaluates the start timing of its translation but is more length-adaptive compared to AL, meaning it evaluates longer outputs more fairly. Meanwhile, ATD considers both the start and end timings of the translation.

C Detailed Experimental Results Analyses in Each Language Pair in Wait-k.

En-Ja Figure 3 shows the results for the En-Ja. When focusing on BLEURT, COMET, COMET-QE, Simul-MuST-C demonstrates superior performance over MuST-C, showing significant differences. However, MuST-C tends to outperform as k increases in BLEU. This implies that MuST-C is more likely to align with the test data, potentially achieving better BLEU. In terms of latency in AL, Simul-MuST-C outperforms MuST-C with a noticeable difference. However, in LAAL, although Simul-MuST-C still performs better, the gap is smaller compared to that in AL. This suggests that the difference is influenced by the characteristics of the metrics, as LAAL handles longer outputs more fairly.

En-Zh Figure 4 shows the results for En-Zh. When focusing on BLEU and {BLEURT, COMET,

[12] https://iwslt.org/2024/simultaneous

COMET-QE}, the quality gap in BLEU is larger than in {BLEURT, COMET, COMET-QE Simul-MuST-C outperforms MuST-C. This indicates that while Simul-MuST-C outperforms in surface-level textual matching, there is not much difference between MuST-C and Simul-MuST-C when evaluating semantic similarity, despite Simul-MuST-C being slightly better. While trends in surface-based evaluation metrics and semantic similarity evaluation metrics could sometimes differ, however they correlate in this case. These results suggests that Simul-MuST-C is slightly, but consistently, better than MuST-C. For latency, in both AL and LAAL, Simul-MuST-C is slightly faster than MuST-C, with the gap remaining constant even as k increases, suggesting Simul-MuST-C could translate faster.

En-De Figure 5 shows the results for En-De. In terms of quality, as measured by BLEU, Simul-MuST-C is slightly better than MuST-C, and the quality gap increases as wait-k increases. We found a similar pattern that we observed in En-Zh: with both the training and evaluation data are from MuST-C, suggesting that MuST-C is more likely to align with the test data, possibly improving the BLEU. Nevertheless, the outcomes show that Simul-MuST-C achieves a closer surface-level match to the test data than MuST-C. Meanwhile, in BLEURT and COMET, MuST-C performs slightly better when wait-k is small, and the gap narrows as wait-k increases, with Simul-MuST-C eventually surpassing it. In AL and LAAL, MuST-C and Simul-MuST-C are almost the same, indicating both could start translation at the same latency. Similarly, in ATD, MuST-C and Simul-MuST-C achieve nearly the same level of latency. This is different from what we observed in En-Ja and En-Zh, where Simul-MuST-C showed a distinct advantage.

D Analysis of generated sentences under the Wait-k setting on k = 7

En-Ja Table 7 shows an example that sentence generated using Simul-MuST-C aligns with the source phrase order, while the sentence generated using MuST-C reverses its monotonicity compared to the source, shown as (1) to (2). Additionally in Table 8, when the inputs become longer, MuST-C fails to translate all the content from the source, omitting (3), (4), and (5). On the other hand, Simul-MuST-C translates all the content, maintaining alignment with the source order. This indicates that Simul-MuST-C could align with the word order in the source language and also translate more effectively.

En-Zh Similar to the case shown in the En-Ja (Table 8), when the sentence becomes relatively longer, MuST-C cannot translate the entire source content, omitting the phrase (2) "it's a very good media opportunity". However, Simul-MuST-C translates all the content from the source, ensuring word order monotonicity (Table 8). This indicates that En-Zh also gains advantages from Simul-MuST-C, maintaining alignment with the original language's word order and maintaining quality.

En-De Table 7 provides an example of generated output, highlighting the position of the word (2) "at all". In sentences generated with Simul-MuST-C, (2) "at all" aligns its original position from the source, while in those generated with MuST-C, it is placed in the middle of the sentence, indicating word reordering. With longer sentences, MuST-C struggles to fully cover the source inputs in En-Ja and En-Zh pairs, however, both MuST-C and Simul-MuST-C generate all source content while retaining the initial word order in En-De, as illustrated in Table 8.

E Experimental Results on Local Agreement

En-Ja Figure 6 shows that when evaluating with {BLEU, COMET}, MuST-C consistently outperforms Simul-MuST-C, demonstrating superiority. In BLEURT, Simul-MuST-C excels with smaller chunk sizes, whereas MuST-C surpasses Simul-MuST-C as the chunk size increases. Conversely, across all chunk size settings in COMET-QE, Simul-MuST-C consistently exhibits superior performance. These discoveries indicate that MuST-C is better aligned with test data, which may be possible to increase reference-based quality metrics {BLEU, BLEURT, COMET}. Regarding latency, Simul-MuST-C outperforms in {AL, LAAL, ATD}, as it starts translations much faster across all chunk sizes. Additionally, in COMET-QE_ATD, Simul-MuST-C not only starts translating faster but also completes translations faster. This feature is particularly advantageous in SiST scenarios, where delays in translation could detrimentally impact subsequent inputs. Simul-MuST-C facilitates faster completion of translations while maintaining quality, which is the same tendency we observed in wait-k setting on En-{Ja, Zh}. In an offline setting,

En-Ja	Source	(1) And you know / (2) what I've learned?
	MuST-C	(2) 私が学んだことは (*what I've learned*) / (1) 分かりますか (*you know*)?
	Simul-MuST-C	(1) あなたは知っていますか (*you know*) 、/ (2) 私が学んだことを (*what I've learned*)?
En-De	Source	(1) That wouldn't have been a problem / (2) at all.
	MuST-C	(1) Das wäre (*that would be*) / (2) überhaupt (*at all*) / (1) kein Problem gewesen (*no problem*).
	Simul-MuST-C	(1) Das wäre nicht ein Problem gewesen (*that would be not a problem*) / (2) überhaupt (*at all*).

Table 7: Examples of generated sentences with an emphasis on word order monotonicity in Wait-k.

En-Ja	Source	(1) Now, I don't know / (2) how you play, / (3) but I want to show you / (4) a couple of unique clips / (5) fresh from the wild.
	MuST-C	(2) 皆さんがどう遊ぶか (*how you play*) / (1) 分かりません (*I don't know*)。
	Simul-MuST-C	(1) いいえ、私はわかりません (*I don't know*)、/ (2) あなたがどのように遊ぶか (*how you play*)、/ (3) しかし、私はあなたに見せたいです (*but I want to show you*)、/ (4) いくつかのユニークなクリップを (*a couple of unique clips*)、/ (5) フローから新鮮な (*fresh from the wild*)。
En-Zh	Source	(1) But that being said, / (2) it's a very good media opportunity.
	MuST-C	(1) 但是那不是说 (*but that is not to say*)。
	Simul-MuST-C	(1) 但这话是说不出来的 (*but word are cannot be said*) / (2) 这是一种非常好的媒介机会 (*this is a very good media opportunity.*)。
En-De	Source	(1) Today, / (2) more than ever, / (3) a little honesty /(4) is going to / (5) go a long way.
	MuST-C	(1) Heute (*today*) / (2) mehr als je zuvor (*more than ever before*), / (3) ein bisschen Ehrlichkeit (*a bit honesty*) / (4) wird (*will*) / (5) weitergehen (*go further*).
	Simul-MuST-C	(1) Heute (*today*), / (2) mehr denn je (*more than ever*), / (3) ein wenig Ehrlichkeit (*a bit honesty*) / (4) wird (*will*) / (5) ankommen (*arrive*).

Table 8: Examples of generated sentences focusing on omission in Wait-k.

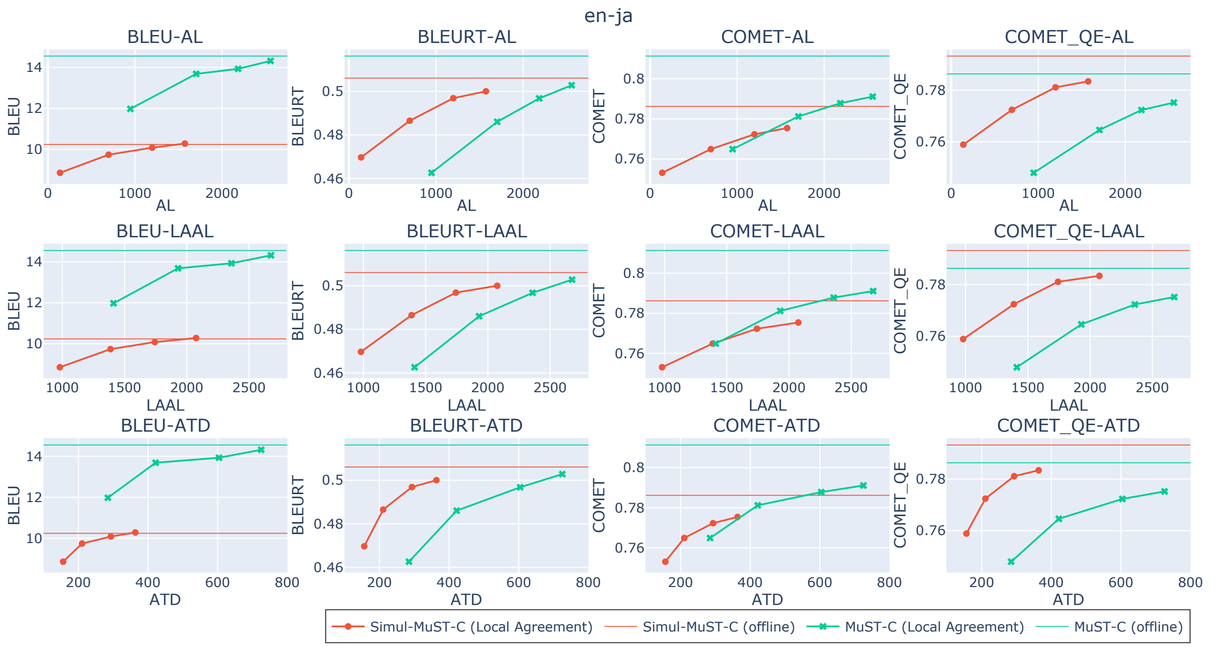

Figure 6: The results for the En-Ja pair on the tst-COMMON. Each plot, from left to right, represents a chunk size ranging from 200, 400, 600, 800, 1000.

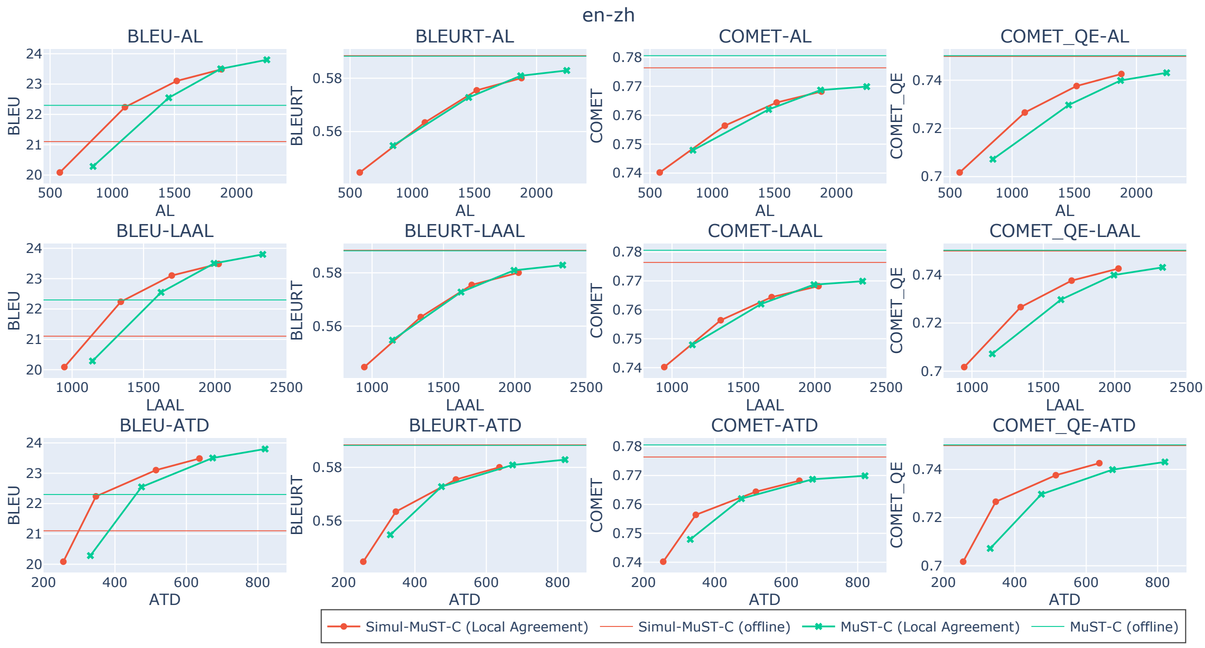

Figure 7: The results for the En-Zh pair on the tst-COMMON. Each plot, from left to right, represents a chunk size ranging from $200, 400, 600, 800, 1000$.

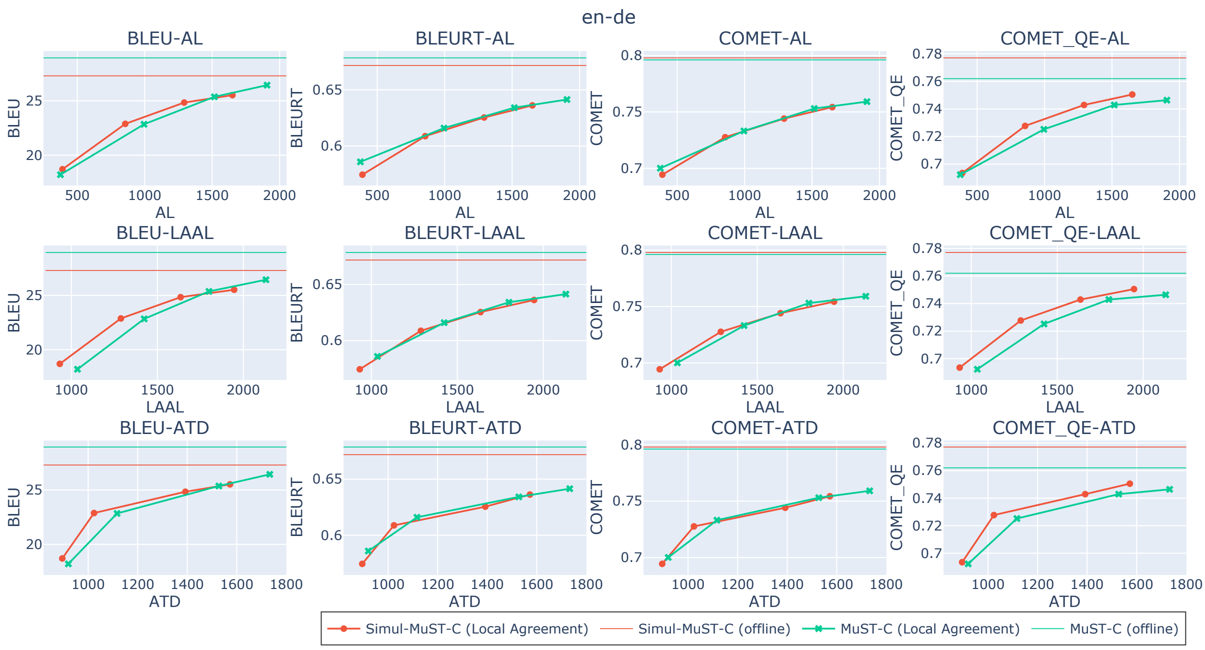

Figure 8: The results for the En-De pair on the tst-COMMON. Each plot, from left to right, represents a chunk size ranging from $200, 400, 600, 800, 1000$.

evaluated with COMET-QE, Simul-MuST-C performs better than MuST-C, with a larger quality gap between the two compared to that observed in wait-k under the same conditions. However, when evaluated with BLEU, MuST-C outperforms Simul-MuST-C. These quality gaps may be due to differences in the evaluation metrics, emphasizing the need for test data that more accurately reflects the specific demands of simultaneous translation.

En-Zh MuST-C consistently outperforms Simul-MuST-C across all quality metrics, particularly noticeable with smaller chunk sizes as shown in Figure 7. However, as the chunk size increases, the quality gap diminishes until both models achieve similar levels of quality. When on COMET-QE-

{AL, LAAL, ATD}, Simul-MuST-C achieves translations faster and reaches the quality upper bound sooner than MuST-C, meanwhile MuST-C achieves better quality when the chunk size is small but translation speed is slower than Simul-MuST-C. Regarding latency, Simul-MuST-C excels in AL, LAAL, and ATD, initiating translations much faster across all chunk sizes. Moreover, in ATD, Simul-MuST-C not only starts translating faster but also completes translations more quickly. This feature is particularly advantageous in SiST scenarios, where delays in translation could adversely affect consecutive inputs. Simul-MuST-C's faster completion of translations is similar to the observed tendency in the wait-k setting for En-Ja and En-Zh and Local Agreement on En-Ja. Evaluated with COMET-QE in offline settings, both MuST-C and Simul-MuST-C achieve similar quality outputs, while MuST-C performs better in BLEU. This may indicate a mismatch in using offline translation-style test data for simultaneous settings, as observed in previous analyses. There is little quality gap between the two models in offline evaluations with COMET-QE, but in simultaneous settings, Simul-MuST-C shows better latency, though not necessarily better quality. In contrast, under the wait-k policy, Simul-MuST-C outperformed in both latency and quality. This suggests that, in this decoding policy, there is room for improvement to enhance quality while minimizing latency for this language pair.

En-De Figure 8 shows when the chunk size is small, Simul-MuST-C achieves comparable quality levels to MuST-C in terms of BLEU. However, as the chunk size increases, MuST-C demonstrates better performance. Similar trends are observed in BLEURT and COMET metrics, with MuST-C consistently outperforming Simul-MuST-C. This may be attributed to the fact that translation similarity between tst-COMMON and MuST-C, enhances reference-based scores. In addition to that, in COMET-QE, both MuST-C and Simul-MuST-C achieve similar quality levels across different chunk sizes, suggesting that Simul-MuST-C might not be as effective in terms of Local Agreement in En-De for improving its quality. On the other hand, Simul-MuST-C contributes to latency improvement as Simul-MuST-C excels in AL, LAAL, and ATD. This speed advantage becomes clear as the chunk size increases. Moreover, in ATD, Simul-MuST-C not only starts translating faster but also completes translations more quickly. In SiST scenarios,

Language Pair	Data	Monotonicity	BLEU	BLEURT	COMET-QE
En-Ja	Original	0.633	**13.69**	0.486	0.765
	Ours	**0.815**	9.74	**0.487**	**0.772**
En-Zh	Original	0.919	**22.55**	**0.573**	0.730
	Ours	**0.954**	22.24	0.563	**0.757**
En-De	Original	0.949	22.84	**0.616**	0.725
	Ours	**0.962**	**22.88**	0.610	**0.728**

Table 9: The table shows the word order monotonicity of generated sentences and their corresponding quality with a chunk-size of 600 in the Local Agreement setting on tst-COMMON. "Original" refers to the model trained with MuST-C, and "Ours" refers to the model trained with Simul-MuST-C.

where delays in translation might impede incoming inputs, these results prove beneficial. Simul-MuST-C's quick translation completion corresponds with the patterns, in the wait-k setting for En-Ja and En-Zh, as well as Local Agreement in En-Ja and En-Zh, although this was not observed in wait-k in En-De. In reference-free metrics like COMET-QE, Simul-MuST-C performs better, while in reference-based metrics such as BLEU, MuST-C shows superior results in offline settings. This discrepancy between different metrics was also observed in previous analyses. When comparing simultaneous and offline settings, Simul-MuST-C demonstrates a significant advantage in terms of latency. However, regarding quality, Simul-MuST-C performs slightly better with smaller chunk sizes, but as the chunk size increases, MuST-C begins to slightly outperform it. These findings suggest, as seen in the En-Zh local agreement setting, that this adaptive decoding policy may not be fully optimized for maximizing quality while maintaining low latency. This trend is evident in language pairs with similar word orders.

Summary Although Simul-MuST-C is effective across all three language pairs under the wait-k policy, its effectiveness in the local agreement setting, which represents adaptive decoding, depends on the language pair. In En-Ja, where the word order gap is significant, the results with COMET-QE suggest that Simul-MuST-C is effective. However, in language pairs with more similar word orders, such as en-zh and en-de, Simul-MuST-C effectively minimizes latency but falls short in achieving comparable quality. These findings suggest that adaptive decoding policy could be further refined, particularly for language pairs with similar word orders,

En-Ja	Source	(1) So / (2) we thought / (3) we would start writing / (4) a brand new chapter of mobility.
	MuST-C	(1) それで (*So*) / (2) 私たちは (*we*) / (4)「移動性」の新しい章を (*a brand new chapter of mobility*) / (3) 書き始めることにしました (*start writing*)。
	Simul-MuST-C	(1) だから (*So*)、/ (2) 私たちは考えました (*we thought*)、/ (3) 始めるだろうと、書くことを、(*would start writing*) / (4) 全く新しい章を (*a brand new chapter*)。
En-Zh	Source	(1) He / (2) robbed / (3) every ounce of hope / (4) from my being.
	MuST-C	(1) 他 (*he*) / (3) 把一切希望 (*puts all hope*) / (4) 从我身上 (*from my being*) / (2) 抹去了 (*erase*)。
	Simul-MuST-C	(1) 他 (*he*) / (2) 剥夺了 (*robbed*) / (3) 每一盎司的希望 (*every ounce hope*) / (4) 从我的存在中 (*from my being*)。
En-De	Source	(1) They / (2) need / (3) to tell / (4) me / (5) about my brand.
	MuST-C	((1) Sie (*you*) / (2) müssen (*must*)/ (4) mir (*me*) / (5) von meiner Marke (*my brand*) / (3) erzählen (*tell*).
	Simul-MuST-C	(1) Sie (*you*) / (2) müssen (*must*) / (4) mir (*me*) / (3) erzählen (*tell*) / (5) von meiner Marke (*my brand*).

Table 10: Word order monotonicity focused example of generated sentences when Loal Agreement is decoding policy.

to better balance quality and latency when using Simul-MuST-C. Additionally, as observed in the wait-k analysis, the current test data tends to favor offline translation-style outputs, as evidenced by the offline quality gap between BLEU and COMET-QE. To ensure fair evaluation in simultaneous settings, test data specifically designed for simultaneous translation is needed.

F Analysis of generated sentences in Local Agreement

Table 9 shows the difference in word order monotonicity between sentences generated by MuST-C and Simul-MuST-C, and the corresponding quality under the Local Agreement setting with a chunk size of 600. Simul-MuST-C demonstrates better monotonicity across all language pairs, displaying differing levels of improvement among them. En-Ja exhibited the most notable enhancement, followed by En-Zh, with En-De showing the least improvement.

En-Ja An example showed in Table 10 demonstrates how sentences generated using Simul-MuST-C aligns to the source word order, while reorderings, as seen in phrases such as (3) "we would start writing" and (4) "a brand new chapter of mobility," occur with MuST-C-generated sentences. On the other hand, in wait-k settings, omission is observed more frequently in sentences generated by MuST-C-trained models (Table 8), whereas this decoding policy decreases the the probability of omitting words, even in sentences produced by MuST-C. This implies that adaptive policy may be more suitable for SiST than fixed policy.

En-Zh An example of a generated sentence is shown in Table 10, we observe examples of how sentences generated using Simul-MuST-C and MuST-C differ. For instance, the word similar to (2) "robbed" appears at the end in MuST-C-generated sentences, while its position in Simul-MuST-C mirrors that of the source. Additionally, omission is less likely to occur in models trained with MuST-C, consistent with observations in the En-Ja pair. Both MuST-C and Simul-MuST-C-generated sentences cover all contents present in the source text, shown in Table 11. This also suggests that an adaptive policy is better suited for SiST than a fixed policy.

En-De In Table 9, the smallest disparity in monotonicity between MuST-C and Simul-MuST-C among the three language pairs is observed in the En-De pair. Table 10 shows the semantically similar word (3) tell in the source appears at the end in MuST-C, whereas the position of the (2) “tell” is relatively close to the order in the source in Simul-MuST-C. In addition to that, Table 11 shows that word order reversal occurs from (3) to (6) in the sentence generated by the MuST-C-trained model, whereas those generated by the Simul-MuST-C-trained model align with the source. However, such word reordering cases are rare occurrences, as indi-

En-Ja	Source	(1) Now, I don't know / (2) how you play, / (3) but / (4) I want to show you / (5) a couple of unique clips / (6) fresh from the wild.
	MuST-C	(2) 皆さんがどう遊ぶか (*how you play*) / (1) 分かりません (*I don't know*) / (3) が (*but*) / (5) いくつかクリップを (*a couple of unique clips*) / (4) お見せしましょう (*to show you*) / (6) 野生のクリップです (*fresh from the wild*)
	Simul-MuST-C	(1) いいえ、私は知りません (*I don't know*)、/ (2) あなたがどのように遊ぶか (*how you play*)。/ (3) しかし (*but*)、/ (4) 私はあなたに見せたいです (*I want to show you*)、/ (5) いくつかのユニークなクリップを (*a couple of unique clips*)、/ (6) 野生から新鮮な (*fresh from the wild*)。
En-Zh	Source	((1) But / (2) that being said, / (3) it's a very good media opportunity.
	MuST-C	(1) 但 (*but*) / (2) 这只是说而已 (*this is just saying*)。/ (3) 这是个非常好的媒体机会 (*this is a very good media opportunity*)。
	Simul-MuST-C	(1) 但是 (*but*), / (2) 话虽如此 (*having said that*), / (3) 这是一个非常好的媒体机会 (*this is a very good media opportunity*)。
En-De	Source	(1) Because / (2) the lesson / (3) I'm trying / (4) to learn / (5)myself / (6) this week is / (7) that it's okay / (8) to let go.
	MuST-C	(1) Denn (*because*) / (2) das ist die Lektion (*the lesson*), / (3) die ich (*I*) / (6) in dieser Woche (*this week*) / (5) selbst (*myself*) / (4) zu lernen (*to learn*) / (3) versuche (*try*), / (7) weil es okay ist (*because it is okay*), / (8) loszulassen (*to let go*).
	Simul-MuST-C	(1) Weil (*because*) / (2) die Lektion (*the lesson*), / (3) die ich (*I*) / (4) zu lernen (*to learn*) / (3) versuche (*try*), / (5) selbst (*myself*) / (6) diese Woche (*this week*), / ist , dass es in Ordnung ist , / (8) loszulassen (*to let go*).

Table 11: Example of generated sentences from Local Agreement. Apart from Wait-k, where omission happens a lot, Local Agreement covers source side contents much better.

cated in the Table 9, where the En-De pair already achieves high word order monotonicity in MuST-C.