Investigating Omission as a Latency Reduction Strategy in Simultaneous Speech Translation

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) requires balancing translation quality and latency. While most SiST systems follow machine translation assumptions that prioritize full semantic accuracy to the source, human interpreters often omit less critical content to catch up with the speaker. This study investigates whether omission can be used to reduce latency while preserving meaning in SiST. We construct a dataset that includes omission using large language models (LLMs) and propose a Target-Duration Latency (TDL), target-based latency metric that measures the output length considering the start and end timing of translation. Our analysis shows that LLMs can omit less important words while retaining the core meaning. Furthermore, experimental results show that although standard metrics overlook the benefit of the model trained with proposed omission-involving dataset, alternative evaluation methods capture it, as omission leads to shorter outputs with acceptable quality.

1 Introduction

Simultaneous speech translation (SiST) is generating translations before the speaker has finished speaking, requiring a balance between translation quality and latency (Ma et al., 2019; Zheng et al., 2020). Despite the real-time characteristic of this task, most prior work adopts conventional machine translation assumptions during evaluation, leading to output that resembles offline translation (Fujita et al., 2013; Oda et al., 2014; Zhang and Feng, 2022; Papi et al., 2023). In contrast, human interpreters use distinct strategies to manage this trade-off, resulting in a different output style (He et al., 2016; Zhao et al., 2021; Wein et al., 2024). One such strategy is *omission* (Pym, 2009; Napier, 2004), where interpreters intentionally skip less important information to keep up with the speaker while preserving the speaker's intended message.

Inspired by recent work on incorporating human translation and interpretation strategies into machine translation (Briakou et al., 2024; He et al., 2024; Makinae et al., 2024; Sakai et al., 2024), this study investigates whether omission can be used in SiST to reduce latency without significantly sacrificing translation quality. By aligning more closely with simultaneous interpretation strategies used by human interpreters, our goal is to move beyond conventional offline-style translation, which aims for full semantic accuracy.

To explore the potential of omission in SiST, we first construct a training dataset that incorporates omission by leveraging large language models (LLMs). LLM is prompted to remove less important information while preserving the core meaning, producing shorter target outputs that mimic interpreter behavior. This allows us to examine whether models can learn to generate such translations in simultaneous setting. Second, we introduce a Target-Duration Latency (TDL), a target-based latency metric that captures the impact of output length. Existing metrics focus mainly on when translation starts, overlooking how long the system takes to complete the output. Our metric instead measures how many source tokens are consumed to generate each target token semantically, providing an estimate of utterance duration and better reflecting the potential benefits of omission.

Our analysis shows that LLMs can omit less important words, generating shorter translations that still preserve the core meaning. Furthermore, experimental results revealed the gap between standard metrics and alternative evaluations. Models trained with omission scored lower on standard quality metrics, which penalize translations that are not fully covered the source. In contrast, Natural Language Inference (NLI) evaluations showed omission achieves acceptable meaning preservation with the number of entailment. Similarly, standard latency metrics failed to capture the benefit

Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 2238–2258

of shorter outputs, while Target Duration Latency (TDL), which is our proposed latency metric, could capture length of shorter outputs.

Our contributions are as follows:

- We construct an omission-involving dataset using large language models by removing less important information from source sentences.
- We propose a Target-Duration Latency (TDL), a target-based latency metric that reflects the impact of output length.
- Our findings revealed that although standard metrics fail to capture the benefit of omission, the alternative measurement can confirm its advantage, producing shorter outputs with acceptable quality.

2 Background and Related Work

2.1 Simultaneous Speech Translation

In SiST, models process partial source inputs and generate partial target outputs step by step, guided by decoding policies (Ren et al., 2020; Zeng et al., 2021; Ahmad et al., 2024), which are typically categorized as either fixed or adaptive. In fixed policies, wait-k (Ma et al., 2019) reads k tokens before alternating between reading and writing. In adaptive policies (Liu et al., 2021; Zhang and Feng, 2022; Papi et al., 2023), it dynamically decides when to read or write. Among them, Local Agreement (Liu et al., 2020) segments input into fixed-size chunks and decodes each using prior chunk outputs to guide generation based on source and target prefixes. Another research direction focuses on constructing training data tailored to simultaneous constraints. This includes generating pseudo references that align with wait-k decoding policy (Chen et al., 2021), applying grammar and meaning preserving syntactic transformations (He et al., 2015), and by using LLMs to rewrite sentences following interpreter-like strategies (Sakai et al., 2024; Makinae et al., 2024).

2.2 Application of Human Tactics into Computational Approach

With recent progress in LLMs, there's growing interest in replicating human translation strategies into a computational approach. Briakou et al. (2024) draws from translation studies to model translation as a multi-step process, pre-drafting research, drafting, refining, and proofreading, rather than a simple source-to-target mapping. By following this structured workflow, LLMs showed improvement in translation quality. Similarly, He et al. (2024) investigates whether LLMs can engage in preparatory steps similar to those used by professional translators, such as identifying key terms, related topics, and relevant information, showing that LLMs can enhance translation quality by incorporating such frameworks into their generation process. In SiST, Sakai et al. (2024); Makinae et al. (2024) have examined whether LLMs can imitate human segmentation strategies, where interpreters break input into smaller meaningful units and translate them incrementally while preserving word order (Jones, 2015; Gillies, 2013).

2.3 Omission in Simultaneous Interpretation

In simultaneous interpretation studies, omission refers to the deletion of source words or phrases that do not appear in the target output (Barik, 1971; Altman, 1994; Kopczyski, 1980; Pym, 2009; Napier, 2004). Barik (1971) classifies omission into four types: "skipping", "comprehension", "delay", and "compounding" omission. Among these, skipping omission involves removing elements that results in minimal meaning loss. Such omission viewed as an acceptable as strategic tactic for managing time and efficiency (Pym, 2009; Napier, 2004). In contrast, other studies frame omission negatively (Altman, 1994; Kopczyski, 1980), as a sign of performance limitations. For instance, comprehension omission, as defined by Barik (1971), results in a loss of meaning and is considered detrimental to translation quality.

2.4 Latency Metrics in Simultaneous Setting

In human interpretation scenario, the well-known latency metrics is Ear-Voice-Span (EVS) (Hanna, 1957) which measures the delay between the speaker's words and the interpreter's corresponding translation in meaning.

In SiST, the latency is calculated by latency metrics such as Average Lagging (AL) (Ma et al., 2019), Length Adaptive Average Lagging (LAAL) (Papi et al., 2022), and Average Token Delay (ATD) (Kano et al., 2023). Among them, ATD, inspired by EVS, is the only metric that considers both the start and end timing of translation. Since this study focuses on the end timing, we explain how ATD works. We define an input sentence as $\boldsymbol{x} = x_1, x_2, \ldots, x_m$ and its translation as

$\boldsymbol{y} = y_1, y_2, \ldots, y_n$, ATD is formulated as follows:

$$\mathrm{ATD}(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{|\boldsymbol{y}|} \sum_{t=1}^{|\boldsymbol{y}|} \left(T(y_t) - T(x_{a(t)}) \right), \quad (1)$$

where

$$a(t) = \min\left(a(t-1) + 1, g(t)\right). \quad (2)$$

$T(\cdot)$ represents the end time of each input and output token. Each token is a subsegment of speech, character or word in the text, and is tied to speech segment. $a(t)$ means the index of the input token corresponding to y_t. $g(t)$ is directly related to the policy employed in decoding algorithms, such as wait-k, representing the number of input tokens read until the prediction of t-th output token y_t. Therefore, ATD is calculated by comparing the translation timing of source tokens based on the target output time, assuming a 1:1 alignment between input and output tokens (i.e., $\mathrm{ATD}(x, y)$ in Equation 1), and taking into account both the ideal case from the previous timestep and actual translation delays.

However, ATD has several limitations. First, it may not properly capture the duration of the utterance. For example, when the target sequence $\boldsymbol{y}$ is longer than the source $\boldsymbol{x}$, ATD forces the alignment of excess target tokens to the final source token. This leads to inaccurate latency estimates in cases where the model generates longer outputs than the source (Kano et al., 2024). Such cases, where the target is either longer or shorter than the source, are more common than 1:1 source–target alignment, due to differences in word order, morphology, or information density between languages, easily violating the assumption in ATD. Second, ATD mechanically calculates the distance between input and output alignments following the policy employed when decoding. It does not reflect actual alignments between input and output tokens, but instead relies on the assumption defined in Equation 2. For more details and comparisons with other delay metrics, please refer to (Kano et al., 2024).

3 Proposed Method

3.1 Dataset Creation

Following recent trends in simulating human translation and interpretation strategies (Briakou et al., 2024; He et al., 2024; Sakai et al., 2024; Makinae et al., 2024), we focus on *omission*, a technique used by human interpreters to shorten output for time management (Pym, 2009; Napier, 2004). We leverage LLMs[1] to generate the omitted source sentence using prompts designed to remove less important words without significantly affecting the overall meaning. The generated concise source sentences are then translated into the target language, with the goal of producing shorter target sentences than those in existing datasets.

Dataset Construction Pipeline Figure 1 illustrates the prompt design and data creation pipeline. In the first pipeline, we used the term *conscious strategic omission*, prompting LLMs to produce concise English outputs compared to the original source. We adopted the term *conscious strategic omission* (Napier, 2004), as preliminary studies indicated that this terminology guided the model to remove less important words without compromising overall meaning. Both the input and output languages were English, and sentences were processed independently without context, under the assumption that less important words can be identified without relying on surrounding discourse. While human interpreters omit information based on broader context, this study focuses on the initial step of omission, leaving context dependent omission for future work. In the second pipeline, we make the translation monotonic with respect to the source, following Simul-MuST-C (Makinae et al., 2024), using LLMs. To this end, we used the omitted English outputs from the first step as input to generate monotonic translations in the target languages during the second step. We refer to the resulting dataset as "Omission". A two step pipeline was necessary, as a single prompt could not effectively achieve both omission and monotonic translation at the same time.

Dataset Selection We applied our method to MuST-C (Di Gangi et al., 2019) for three language pairs: English–Japanese (En–Ja), English–German (En–De), and English–Chinese (En–Zh). MuST-C is a multilingual speech translation corpus consisting of English TED Talk recordings aligned at the sentence level with manual transcriptions and translations, which are originally for subtitles. These three language pairs were selected to maintain consistency with the setting of Simul-MuST-C (Makinae et al., 2024)[2], enabling fair comparisons and

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

[2] Since the dataset only provides the prompt code for data creation, we used it to produce an equivalent version of the Simul-MuST-C dataset, and applied the code in the second

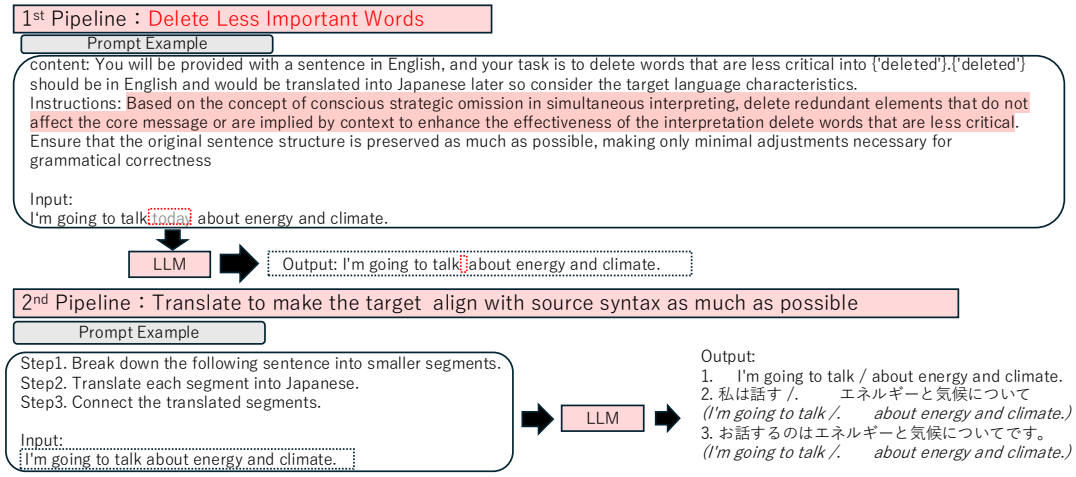

Figure 1: The prompt and workflow for constructing the proposed dataset. The first pipeline aims to generate more concise English outputs, and the second pipeline aims to make the target monotonic to the source.

allowing us to investigate whether shortening target sentences contributes to latency reduction.

3.2 TDL: Target-Duration Latency

Inspired by EVS, we proposed Target-Duration Latency (TDL), a metric that measures how long a model must wait, measured in source token steps, to produce each target token based on source token availability to compute total latency, considered as utterance duration. To compute this, we first obtain the semantic alignment $\boldsymbol{a} = a_1, ..., a_n$ using Awesome Align (Dou and Neubig, 2021) between the source and target tokens, in which a_t denotes an index to a source token for y_t with 0 indicating no-alignment. Based on this alignment and the generated target tokens, TDL is defined as follows:

$$\mathrm{TDL}(\boldsymbol{y}) = \sum_{t=1}^{|\boldsymbol{y}|} \ell_t \tag{3}$$

$$\ell_t = \begin{cases} T(y_{|\boldsymbol{y}|}) - T(y_t) & \text{if } a_t = 0 \\ |T(x_{a_t}) - T(y_t)| & \text{if } a_t > d_t \\ T(x_{d_t}) - T(x_{a_t}) & \text{if } 0 < a_t \leq d_t \end{cases} \tag{4}$$

where

$$d_{t+1} = \max(d_t, a_t). \tag{5}$$

d_t is the index of the longest source token that has been delivered so far at t. ℓ_t is the latency for each target token t. Figure 2 describes how the TDL works, and the examples below illustrate each case division. The green indicates that t_4 is not aligned to any source token, so its semantic distance cannot be computed. However, even though y_4 is unaligned, it still appears in the output sequence y, and its insertion and shift influences the generation of later tokens y_5 and y_6. This shift should therefore be treated as latency, calculated as $6 - 4 = 2$, serving as an alternative to semantic distance calculation. The first blue line indicates that y_1 is aligned to $a_1 = 5$. To produce y_1, the model must read the fifth token of the source, so the latency is calculated as $|5 - 1| = 4$. The first red line indicates that y_2 is aligned to $a_2 = 3$. To generate y_2, the model must read the third token of the source. However, since it has already read up to the fifth token to produce y_1, no additional reading is required to generate y_2. In this case, latency is computed as the difference between the furthest source token read so far and the token aligned to y_2, calculated as $5 - 3 = 2$.

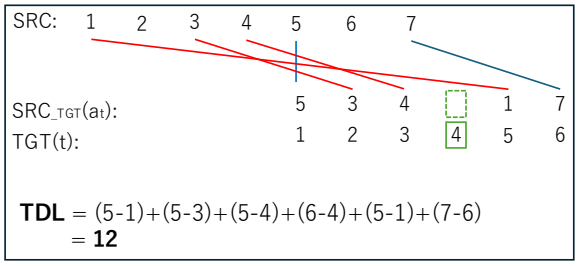

Figure 2: The overview of Target-Duration Latency (TDL). Green represents when $a_t = 0$, blue represents when $a_t > d_t$, and red represents when $0 < a_t \leq d_t$.

There are two distinct difference from ATD. First, this method properly get the word alignment between the source and the target so that this

pipeline also.

	Dataset	src-src deletion ratio (words)	tgt-tgt deletion ratio (words)
En-Ja	MuST-C	1 (20.3)	1 (**20.8**)
	Simul-MuST-C	1 (20.3)	1.39 (28.8)
	Omission	0.830(16.9)	1.18 (24.5)
En-Zh	MuST-C	1 (22.9)	1 (18.9)
	Simul-MuST-C	1 (22.9)	1.01 (19.1)
	Omission	0.813 (18.6)	0.883 (**16.7**)
En-De	MuST-C	1 (20.5)	1 (21.6)
	Simul-MuST-C	1 (20.5)	1.03 (22.4
	Omission	0.803 (16.5)	0.887 (**19.2**)

Table 1: Sentence length comparison between MuST-C and the outputs from the first pipeline step, and target in across MuST-C, Simul-MuST-C, Omission which is translations generated in the second pipeline step.

Lang	entailment	neutral	contradiction	COMET-QE
En-Ja	1369	0	0	0.818
En-Zh	1349	0	0	0.782
En-De	1414	0	0	0.816

Table 2: Quality comparison using NLI between the source before and after the omission in dev sets.

Lang	MuST-C	simul-MuST-C	Omission
En-Ja	0.776	0.836	0.815
En-Zh	0.757	0.798	0.777
En-De	0.802	0.833	0.806

Table 3: Monotonicity comparison in dev sets.

method is more close to EVS. Second, the calculation of this score is based on the target so that it properly counts the length of target from the start to the end, and we don't do cut-offs even if the target gets longer than the source.

4 Dataset Analysis

4.1 Length Analysis

We conducted a sentence length analysis to assess whether LLMs can identify and remove non-essential words from the source, thereby producing more concise source sentences in dev data. These shortened sources are then translated into shorter target outputs, with the aim of reducing overall translation length, in comparison with two existing datasets: MuST-C (Di Gangi et al., 2019) and Simul-MuST-C (Makinae et al., 2024).

Source Sentences Length Difference Analysis We compared the original source from MuST-C with those generated in the first step of our pipeline (Omission) to analyze how many words were deleted, what coarse semantic roles those deleted words served, and which words were most frequently omitted. The analysis method is described in Appendix B. Table 1 shows approximately four words were deleted per sentence on average across all language pairs, resulting in an output length of around 0.8 compared to the original. The most frequently deleted words were function words such as "that", "the", "and", and "of". These serve grammatical purposes rather than contributing semantic content, suggesting their omission does not alter meaning, therefore it's not yet clear whether deleting such function words improves latency. Modifiers such as "really," "actually," "very," and "all" were also omitted. These words add emphasis or nuance but are not critical to the core message. Their removal likely contributed to meaningful length reduction with minimal quality degradation, highlighting the model's ability to prioritize semantically important content. The details regarding deleted words and coarse semantic roles those deleted words served are in Appendix A.

Target Sentences Length Difference Analysis Table 1 compares target sentence length across three datasets, and Omission refers to the translation of source sentences generated in the first pipeline. Word counts were computed after the sentences is tokenized using SpaCy[3]. The analysis revealed that Simul-MuST-C produced the longest target sentences across all three language pairs. This is likely due to its design Simul-MuST-C outputs were generated without explicit instructions to limit sentence length. For En–Ja, the Omission output ranked in the middle, with MuST-C producing the shortest translations. We hypothesize that this is due to MuST-C's subtitle based structure, which is constrained by word limits. In En–De and En–Zh, a different trend emerged compared to En-Ja. Despite these MuST-C pairs being subtitle-based, Omission was the shortest among the three datasets. We attribute this to the deletion step in the Omission pipeline: certain source words were intentionally omitted and thus didn't translate, resulting in reduced output length.

[3] https://spacy.io/

4.2 Quality Analysis

Table 2 presents the quality evaluation of the Omission dataset. We conducted a two-step analysis, recognizing that standard quality metrics such as BLEU and COMET are designed for fully faithful translations that contradict our goal of producing shorter outputs that still preserve the core meaning.

First, we used a Natural Language Inference (NLI) model (Conneau et al., 2020)[4] to assess whether the essential meaning of the source is retained after omission, which outputs one of three labels, entailment, neutral, or contradiction, to indicate the logical relationship between the source and target. This analysis was performed on the development sets of all three language pairs. Table 2 shows that a ll results were labeled as entailment, suggesting that the core meaning remains, despite the omission.

Second, we measured the quality using COMET-QE (Rei et al., 2021), a reference-free metric that assesses semantic similarity between the source and hypothesis. Using the omitted source sentences and their corresponding translations, COMET-QE consistently reported high quality scores across all three language pairs.This suggests that the shortened outputs are still semantically adequate.

Taken together, these results support our central claim: that translation output can be shortened with maintaining the core message in the source. Example analysis of the created dataset is in Appendix C.

4.3 Monotonicity Analysis

We compared the monotonicity between source and target sentences to assess the degree of word reordering. This analysis reveals how closely the target follows the source word order, an important factor in SiST, where reduced reordering can contribute to lower latency. The calculation steps are described in Appendix D. Table 3 shows all three language pairs followed the same trend, though the level of monotonicity varies. The gap was the largest for En–Ja and smallest for En–De. As expected, Simul-MuST-C achieved the highest monotonicity, MuST-C the lowest, and Omission ranked in between. Although slightly lower than Simul-MuST-C, the monotonicity of Omission is substantially higher than that of MuST-C.

[4] https://huggingface.co/joeddav/xlm-roberta-large-xnli

5 Experimental Setup

To investigate whether shorter target lengths can reduce latency without severely compromising translation quality, we compare three models trained on different datasets: MuST-C (Di Gangi et al., 2019), Simul-MuST-C (Makinae et al., 2024), and our proposed dataset "Omission". The experiment covers three language pairs: En→{Ja, Zh, De}, which cover all English-to-many directions included in the IWSLT 2024 simultaneous track[5]. Further details on the language selection are provided in the Limitations section.

Dataset Selection MuST-C is a speech translation dataset based on TED Talks, consisting of audio recordings, transcriptions, and target translations. The target side is subtitle-style translations and serves as reference. Simul-MuST-C is built on MuST-C but applies word order manipulation to make target syntax as monotonic to the source. Omission, also derived from MuST-C, is characterized by both shorter target lengths and high monotonicity. These characteristics reflect strategies used by human interpreters to manage latency, making it well-suited for investigating the impact of omission in a computational setting. For evaluation, we use tst-COMMON from MuST-C.

Tranining and Decoding We implemented an end-to-end speech-to-text model initialized with two pre-trained components: a HuBERT-based encoder (Hsu et al., 2021) and an mBART50-based decoder (Tang et al., 2021). The model architecture follows the Transformer (Vaswani et al., 2017) and is implemented using Fairseq (Ott et al., 2019). Text data is tokenized using a multilingual SentencePiece tokenizer (Kudo and Richardson, 2018) with a 250,000 subword vocabulary, consistent with the mBART50. Model validation was performed every 500 steps, with early stopping applied after 8 validations without improvement. We evaluated the model's performance under Local Agreement (Liu et al., 2020) as a representative adaptive policy. The input chunk sizes were set to $400, 600, 800, 1000$. Hypotheses for each input chunk were generated using beam search with a beam width of five. We also evaluated the model with wait-k (Ma et al., 2019) as a fixed policy. The k values were set to $3, 5, 7, 9, 11, 13, 15, 17$, with one unit corresponding to 160 frames. For For comparison purposes, we also report the offline setting performance.

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

Lang	System	Ent.	Neu.	Con.
En-Ja	MuST-C	2274	386	181
	Simul-MuST-C	2337	397	107
	Omission	**2398**	319	124
En-Zh	MuST-C	2394	306	141
	Simul-MuST-C	2466	268	107
	Omission	**2506**	208	127
En-De	MuST-C	2267	230	83
	Simul-MuST-C	2309	209	62
	Omission	**2370**	149	61

Table 4: NLI results across language pairs (En-Ja, En-Zh, En-De) when the chunk size is `400` in local agreement. Ent. stands for Entailment, Neu. stands for Neutral, and Con. stands for Contradiction. The result in all chunk size is in Appendix 10

Lang	System	la-400	la-600	la-800	la-1000
En-Ja	MuST-C	2,096	1,606	1,444	1,409
	Simul-MuST-C	3,114	2,709	2,610	2,552
	Omission	2,559	2,300	2,145	2,032
En-Zh	MuST-C	1,259	924	824	799
	Simul-MuST-C	1,050	880	843	879
	Omission	817	690	664	649
En-De	MuST-C	11,488	8,222	7,504	7,341
	Simul-MuST-C	9,203	7,990	7,690	7,495
	Omission	7,381	6,537	6,205	6,050

Table 5: Target-Duration Latency for local agreement under different chunk-size settings across language pairs.

Evaluation The performance is evaluated using the SimulEval toolkit (Ma et al., 2020), which measures both translation quality and latency. For quality, we report BLEU (Papineni et al., 2002), BLEURT (Sellam et al., 2020), COMET (Rei et al., 2020), and COMET-QE (Rei et al., 2021), and Natural Language Inference (NLI) (Conneau et al., 2020)[6]. For latency, we report Length-Adaptive Average Lagging (LAAL) (Papi et al., 2022), Average Token Delay (ATD) (Kano et al., 2023), and Differentiable Average Lagging (DAL) (Cherry and Foster, 2019). We also measured TDL; our proposed latency metric described in Section 3.2.

6 Experimental Results

We focus on En-Ja results for Local Agreement, as all language pairs exhibited similar trends that highlight the gap between standard metrics and alternative evaluations. Results for other language pairs are in Appendix E. The analysis for wait-k is in Appendix F, and analysis of generated sentences is included in Appendix G.

En-Ja We observe different trends depending on the evaluation method used. Figure 3 is the result in En-Ja. Under standard quality metrics such as BLEU and COMET, MuST-C achieves the highest in BLEU due to its close surface-level match with the subtitle-based reference. However, MuST-C performs the worst when evaluated with embedding-based metrics like BLEURT, COMET, and COMET-QE. In contrast, Simul-MuST-C achieves the highest, while Omission falls in between. For COMET in particular, the overall scores across the three models are relatively close, with Simul-MuST-C slightly outperforming the others. Table 4 shows that using NLI to evaluate whether the generated output preserves the core meaning reveals a different pattern. MuST-C achieves the lowest number of entailment, which is consistent with its lower performance on standard embedding-based metrics. Between Simul-MuST-C and Omission, the trend reverses compared to standard metrics: Omission achieves a higher number of entailment than Simul-MuST-C. This suggests that, despite scoring slightly lower on standard evaluations, Omission could preserves the intended meaning of the source.

In standard latency latency metrics, Simul-MuST-C demonstrates the best performance, as presented in Table 3. It demonstrates the lowest scores in both LAAL and DAL, which focus on the start timing of translation, as well as in ATD, which considers both the start and end timings of the output, suggesting that Simul-MuST-C appears most efficient when evaluated using these latency metrics. However, Table 5 shows that when applying TDL, which measure utterance length by calculating the begin and end semantically, a different trend emerged. MuST-C achieves the shortest output durations, Simul-MuST-C the longest, and Omission in the middle. This aligns with the target sentence length analysis in Table 1, reflecting the target length in the training data align with the generated outputs.

Summary While omission performed worse under standard quality metrics (e.g., BLEU, COMET), which penalizes translations that lack full fidelity to the source, NLI results indicate that omission can preserve the intended meaning. This suggests that omission is still semantically included in the reference, as shown by the number of entailment

[6] https://huggingface.co/joeddav/xlm-roberta-large-xnli

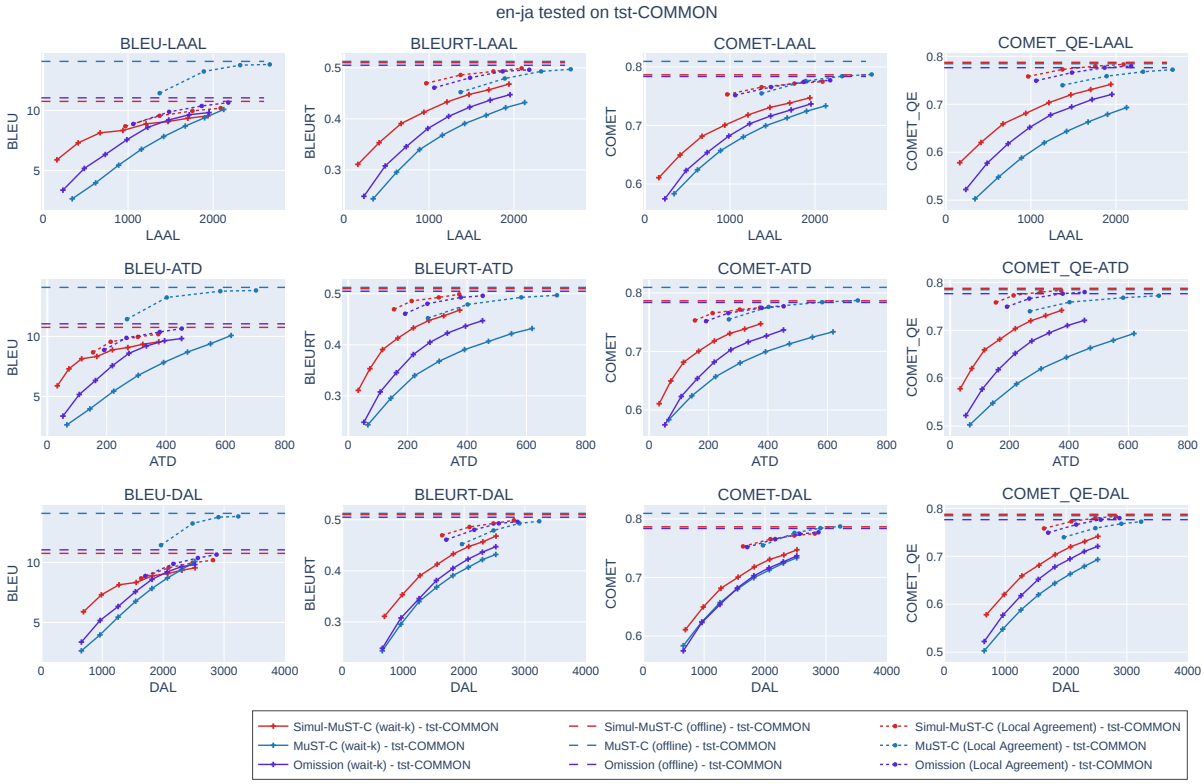

Figure 3: Results for En–Ja on tst-COMMON. The dotted line with circles represents Local Agreement. The dashed line shows the offline results. The solid line with crosses shows wait-k decoding.

cases across all language pairs. Similarly, ATD did not fairly account for the length of shorter outputs, as its did not align with the target lengths observed in the training data. In contrast, TDL captured output length more accurately, with scores that corresponded well to those target lengths.

7 Discussion

This work investigates whether omission, a strategy used by human interpreters, can be effective in SiST. Our findings indicate that while certain components of SiST show potential in handling omission, current model architectures such as transformers, decoding policies like local agreement, and LLMs with their generated translation have demonstrated this capability. However, the potential of omission is not well captured by existing standard evaluation metrics. These metrics remain limited in their ability to reflect the benefits of omission, whereas the alternative method, NLI, demonstrates its value. This suggests that SiST could more closely resemble a human interpreter's translation style than conventional offline-style translations, but achieving this requires shifting away from conventional evaluation methods in order to recognize the value of omission.

Simultaneous interpreters reduce delays by shortening their translations, and this work aims to mimic that behavior. Therefore, it is also important to evaluate whether the translation output is indeed shorter. ATD is based on the assumption that it captures both the start and end timing of translation to reflect output length. However, there are several limitations in ATD's settings that may prevent it from accurately measuring output length. As shown in the experimental results, ATD scores did not align with the target lengths observed in the training data. In contrast, TDL captured output length more accurately, with scores that corresponded well to the target lengths in the training data. This suggests that ATD may overlook the influence of output length, despite its intended design, while TDL handles this aspect more reliably.

8 Conclusion

This work explored whether omission, a strategy used by human interpreters to reduce latency while preserving the original meaning, can be leveraged in SiST. To investigate this, we constructed a dataset involving omission using LLMs and in-

troduced TDL, latency metric that measures output length. The analysis indicates that LLMs are capable of omitting less important words while preserving the essential meaning. Furthermore, experimental results show that although standard metrics fail to justify the advantage of omission, alternative evaluation approaches reveal its value, as omission results in shorter while maintaining acceptable quality. These findings suggest that SiST could adopt a more human-interpreter-like-translation style rather than the conventional offline-translation style, pointing to a promising direction for future work.

9 Limitation

Omission Has No Single Answer This paper investigates whether training models on data with relatively short target sentences, compared to existing datasets, can help reduce latency with only minor degradation in translation quality in simultaneous settings. This approach, which shortens output for time saving purposes, mirrors strategies commonly employed by human interpreters. In human interpretation, omission is context-dependent and has no single correct answer that it often varies based on factors such as speech rate or information density. The same can be said of our dataset, there is no definitive ground truth for which words should be omitted. While LLMs showed the ability to omit less important words such as really and very, they also tended to remove function words like that, of, and is, leading to shorter outputs with seemingly minimal quality loss.

Scalability to Other Architectures and Decoding Policies This study focuses on a Transformer-based architecture, using wait-k as a representative fixed decoding policy and Local Agreement as a representative adaptive policy. While we acknowledge that there are other model architectures and decoding methods, some of which explicitly control output length during training, these are beyond the scope of this work. Our goal in this study is to take an initial step toward exploring whether training on datasets with shortened target outputs can effectively reduce latency without significantly compromising quality. Our contribution in the experiment is although standard metrics overlook the benefit of the model trained with proposed omission-involving dataset, alternative evaluation methods capture it, as omission leads to shorter outputs with acceptable quality. Future research could explore more scalable solutions, such as architectures and decoding policies that incorporate output length control directly into the training or decoding process for SiST.

Language Pair Selection We selected the language pairs En–Ja, En–Zh, and En–De to align with prior work (Makinae et al., 2024), which investigated monotonicity as a latency reduction strategy in simultaneous interpretation. Building on that, our goal was to explore whether omission for length control in addition to monotonicity, could further reduce latency. Therefore, our proposed dataset incorporates both strategies. We believe that our corpus construction method could be applied to other language pairs.

Versatility Across LLMs This study uses GPT-4o and leverages the OpenAI batch API for dataset construction, with prompts specifically designed for this model. Applying the same method to other LLMs would likely require prompt adjustments to accommodate each model's capabilities and response behavior. While our current implementation is optimized for GPT-4o, the underlying approach is designed to be broadly applicable. The prompts are crafted to retain a degree of flexibility, making it feasible to adapt the method for other language models. Thus, despite being tailored to a specific system, our methodology remains aligned with the broader goal of developing tools that can generalize across languages and model architectures.

Human Evaluation for Latency Metrics While human evaluation is always desirable, most prior works on latency metrics in simultaneous speech translation have not included human validation, as such evaluation is extremely challenging. Moreover, existing studies have not established clear methodologies for conducting human evaluation of latency, making it unclear how such an assessment should be designed or standardized. Our proposed latency metric is grounded in the Ear Voice Span (EVS) from interpretation studies, providing a theoretically interpretable and empirically motivated approach to investigate omission strategies and construct the corresponding dataset. While human evaluation could offer additional insight, it falls outside the scope of this study and may serve as a valuable direction for future work.

Ethics Statement

License of Source Dataset Our proposed dataset is derived from MuST-C[7], which is licensed under the CC BY-NC-ND 4.0 license[8]. This license includes a "NoDerivatives", meaning that modified, remixed, or transformed versions of the dataset cannot be redistributed. As a result, while we may make internal modifications and include examples within this paper, we cannot publicly release the modified dataset. MuST-C itself is based on TED Talk content and inherits the same CC BY-NC-ND 4.0 license. Out of ethical and legal considerations, we will only release our dataset after obtaining explicit permission or reaching an agreement with the MuST-C administrators. Until then, we will refrain from making the corpus publicly available. However, we will provide the experimental code, which will allow others to reproduce them independently.

Ownership rights about Proposed dataset Proposed dataset was created using 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 Proposed dataset is free of harmful information, sourced from TED Talks. Moreover, our check with OpenAI Moderation APIs[10] found no harmful content.

Acknowledgments

This work is supported by JSPS KAKENHI under Grant Number JP21H05054 and JST SPRING under Grant Number JPMJSP2140.

References

Ibrahim Said Ahmad, Antonios Anastasopoulos, Ondřej Bojar, Claudia Borg, Marine Carpuat, Roldano Cattoni, Mauro Cettolo, William Chen, Qianqian Dong, Marcello Federico, Barry Haddow, Dávid Javorský, Mateusz Krubiński, Tsz Kin Lam, Xutai Ma, Prashant Mathur, Evgeny Matusov, Chandresh Maurya, John McCrae, and 25 others. 2024. FINDINGS OF THE IWSLT 2024 EVALUATION CAMPAIGN. In *Proceedings of the 21st International Conference on Spoken Language Translation (IWSLT 2024)*, pages 1–11, Bangkok, Thailand (in-person and online). Association for Computational Linguistics.

Janet Altman. 1994. Error analysis in the teaching of simultaneous interpretation. In *Bridging the Gap: Empirical research in simultaneous interpretation*, pages 25–38.

Henri C. Barik. 1971. A description of various types of omissions, additions, and errors of translation, encountered in simultaneous interpretation. In *META*, volume 16-4, pages 199–210.

Eleftheria Briakou, Jiaming Luo, Colin Cherry, and Markus Freitag. 2024. Translating step-by-step: Decomposing the translation process for improved translation quality of long-form texts. In *Proceedings of the Ninth Conference on Machine Translation*, pages 1301–1317, Miami, Florida, USA. Association for Computational Linguistics.

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.

Colin Cherry and George Foster. 2019. Thinking slow about latency evaluation for simultaneous machine translation. *Preprint*, arXiv:1906.00048.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. 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.

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.

Tomoki Fujita, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. 2013. Simple, lexicalized choice of translation timing for simultaneous speech translation. In *Proc. Interspeech*, pages 3487–3491.

[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

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

Blake Hanna. 1957. An investigation into conference interpreting. In *Journal des traducteurs / Translators' Journal*, volume 3(1), pages 35–38.

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.

Zhiwei He, Tian Liang, Wenxiang Jiao, Zhuosheng Zhang, Yujiu Yang, Rui Wang, Zhaopeng Tu, Shuming Shi, and Xing Wang. 2024. Exploring human-like translation strategy with large language models. *Transactions of the Association for Computational Linguistics*, 12:229–246.

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.

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010. Automatic evaluation of translation quality for distant language pairs. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 944–952, Cambridge, MA. Association for Computational Linguistics.

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

Yasumasa Kano, Katsuhito Sudoh, and Satoshi Nakamura. 2023. Average Token Delay: A Latency Metric for Simultaneous Translation. In *Proc. Interspeech*, pages 4469–4473.

Yasumasa Kano, Katsuhito Sudoh, and Satoshi Nakamura. 2024. Average token delay: A duration-aware latency metric for simultaneous translation. In *Journal of Natural Language Processing*, volume 31-3, pages 1049–1075.

Andrzej Kopczyski. 1980. Conference interpreting: Some linguistic and communicative problem. In *UAM,Poznan*.

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*, pages 3620–3624.

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.

Mana Makinae, Yusuke Sakai, Hidetaka Kamigaito, and Taro Watanabe. 2024. Simul-MuST-C: Simultaneous multilingual speech translation corpus using large language model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22185–22205, Miami, Florida, USA. Association for Computational Linguistics.

Jemina Napier. 2004. Interpretation omissions: A new perspective. In *Interpreting*, volume 6-2, pages 117–142.

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.

OpenAI, Josh Achiam, and 1 others. 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.

Anthony Pym. 2009. On omission in simultaneous interpreting: Risk analysis of a hidden effort. *Efforts and Models in Interpreting and Translation Research*, pages 83–105.

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. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22375–22398, Miami, Florida, USA. Association for Computational Linguistics.

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.

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.

Shira Wein, Te I, Colin Cherry, Juraj Juraska, Dirk Padfield, and Wolfgang Macherey. 2024. Barriers to effective evaluation of simultaneous interpretation. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 209–219, St. Julian's, Malta. Association for Computational Linguistics.

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.

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.

Word	Counts
that	177
the	175
and	138
of	129
to	71
a	64
you	51
really	49
very	47
actually	42
it	40
we	40
so	37
is	37
in	35
just	35
it's	35
i	35
all	28
was	25

Coarse role	Counts
TMP	67
ARG1	962
CAU	72
ARG2	629
O	494
MNR	38
V	192
ARG0	127
EXT	14
ADV	154
DIS	55
ARG3	11
ARG4	11
PRP	22
MOD	2
DIR	14
LOC	36
PRD	5
PNC	2
GOL	2
REC	1
COM	2

Table 6: Top 20 most frequently deleted words in source sentence and deleted coarse role counts in En-Ja.

Word	#Words
and	210
that	194
the	189
of	137
a	72
to	70
you	63
really	50
so	49
actually	46
very	45
it	42
is	41
i	41
in	40
we	40
just	38
it's	35
all	34
was	28

Coarse role	Counts
TMP	75
ARG1	1027
CAU	65
ARG2	681
O	579
V	217
MNR	43
ARG0	144
EXT	12
ADV	165
DIS	96
ARG3	16
ARG4	10
LOC	47
PRP	13
MOD	2
DIR	15
PRD	9
PNC	2
GOL	2
REC	1
COM	2

Table 7: Top 20 most frequently deleted words in source sentence and deleted coarse role counts in En-Zh.

A Top 20 Deleted Words and Coarse Semantic Roles

Table 6 is Top 20 Deleted Words and Coarse Semantic Roles in En-Ja. It shows that the deleted words are function words.

Table 7 is Top 20 Deleted Words and Coarse Semantic Roles in En-Zh.It shows that the deleted words are function words.

Table 8 is Top 20 Deleted Words and Coarse Semantic Roles in En-De. It shows that the deleted words are function words.

B Source Sentences Length Difference Analysis Method

To quantify deletions, we tokenized each sentence using SpaCy[11] and counted the number of tokens in both MuST-C and Omission. We then calculated the average number of deletions per sentence and the deletion ratio, using the original source sentence length as a baseline (1.0). To analyze the functions of deleted words, we used Semantic Role Labeling (SRL) with AllenNLP[12] on the original source sentences. By aligning each omitted sentence with its corresponding original sentence, we identified which words were removed and examined their semantic roles. We then aggregated counts of deleted words by role (e.g., arguments, modifiers, adjuncts) to determine which types of information were most frequently omitted.

C Additional Quality Analysis

Example Analysis of Created Dataset Table 9 shows an example from the created dataset alongside the corresponding line from existing datasets. We first compare *Source* and *Omission-src* to observe which words were deleted during the first pipeline step, where less important words were removed. We then compare the target sentence lengths across *MuST-C*, *Simul-MuST-C*, and *Omission-tgt* to examine whether the omission performed in the first step contributes to shorter translations in the second pipeline.

[11] https://spacy.io/

[12] https://github.com/masrb/Semantic-Role-Labeling-allenNLP-

Word	#Words
that	221
the	218
and	209
of	159
to	90
a 2	85
you	63
really	57
actually	56
so	53
very	52
is	51
i	49
just	46
in	45
we	43
it	42
it's	41
all	37
have	33

Coarse role	Counts
TMP	92
ARG1	1229
CAU	84
ARG2	792
O	658
MNR	51
V	233
ARG0	168
EXT	13
ADV	183
ARG3	17
ARG4	11
DIS	76
PRP	25
LOC	43
DIR	15
PRD	16
GOL	2
REC	1
ADJ	6
COM	3

Table 8: Top 20 most frequently deleted words in source sentence and deleted coarse role counts in En-De

En-Ja Focusing on the underlined differences between Source and Omission-src, we find that the LLM is capable of deleting words such as it's and really. Since these words are no longer present in the source, their corresponding translations do not appear in Omission-tgt, resulting in a shorter output. In contrast, Simul-MuST-C, which preserves a 1:1 source-target correspondence, includes these words in the translation. In MuST-C, although omission is not explicitly applied, we assume that subtitle constraints naturally limit sentence length, which may also result in the exclusion of such words.

En-Zh The first pipeline step demonstrates effective deletion of less important words such as *so* and *really*, similar to the underlined differences observed between Source and Omission-src in En–Ja. Additionally, the phrase *where we had these amazing*, which adds expressive tone but is not essential for core meaning, is also removed. This phrase appears in Simul-MuST-C but is omitted in both Omission-tgt and MuST-C. As a result, the output length in Omission-tgt is the shortest among the three, which is consistent with the results reported in the earlier section.

En-De The LLM demonstrates the ability to apply omission effectively. Comparing the Source and Omission-src, we observe that the deleted words are primarily adverbs that serve to emphasize the upcoming verb. As a result of these deletions, the corresponding translations are omitted in Omission-tgt, leading to a shorter target sentence. This observation aligns with our broader analysis, where Omission-tgt consistently produced the shortest translations among the three datasets in En–De. For example, expressions like very quickly that were removed from the source do not appear in the translation. In contrast, MuST-C retains these expressions in the target, resulting in a longer output. Unlike the pattern observed in En–Ja, where Omission-tgt was not always the shortest, in this case it clearly results in the shortest translation.

Summary Overall, the LLM demonstrates the ability to identify and remove relatively unimportant words within a sentence, resulting in a shorter source. This shortened source, in turn, contributes to more concise target translations. However, the extent of this effect varies by language pair. In En–Ja, Omission-tgt remains longer than MuST-C, while in En-Zh and En–De, Omission-tgt ends up as the shortest among the three datasets.

D Monotonicity Calculation

Similar to the method described by (Isozaki et al., 2010), we analyzed monotonicity by computing Spearman's rank correlation coefficient based on word alignments obtained using Awesome Align (Dou and Neubig, 2021). The process involved three steps: (1) each source-target sentence pair was tokenized using SpaCy[13]; (2) word alignments were generated using Awesome Align[14]; and (3) Spearman's rank correlation coefficient was calculated for each pair, with the final monotonicity score derived by averaging across all sentence pairs. A higher score (closer to 1) indicates less reordering and, therefore, greater monotonicity. For MuST-C, we examined alignments between the original source sentences and their corresponding subtitle-based translations from TED Talks. For Simul-MuST-C, we compared the original MuST-C source sentences with the translations produced by Simul-MuST-C, which were specifically designed to follow source word order more closely and achieve higher monotonicity. For Omission, we

[13] https://spacy.io/

[14] https://github.com/neulab/awesome-align

En-Ja	Source	It's selfish, it's ugly, it's beneath us, and we really have to stop it.
	Omission-src	It's selfish, ugly, beneath us, and we have to stop it.
	MuST-C	利己的で醜く我々がやるべきことではありません それはもう止めなければなりません (*It's selfish and ugly and it's not what we should be doing and it has to stop.*)。
	Simul-MuST-C	それは利己的です、それは醜いです、それは私たちにふさわしくありません、そして私たちは本当にそれを止めなければなりません (*It's selfish, it's ugly, it doesn't deserve us, and we really have to stop it*)。
	Omission-tgt	利己的な、醜い、私たちの品位にかかわる、それを止めなければならない (*It's selfish, ugly, it's beneath our dignity, and it has to stop.*)。
En-Zh	Source	So, having done these expeditions, and really beginning to appreciate what was down there, such as at the deep ocean vents where we had these amazing, amazing animals – they're basically aliens right here on Earth.
	Omission-src	Having done these expeditions and beginning to appreciate what was down there, such as at the deep ocean vents with amazing animals basically aliens on Earth.
	MuST-C	所以通过这些探险 我开始真正地欣赏海底的美妙， 比如那些生活在深海裂口处的 奇妙的动物们。 它们算得上就是地球上的外星生物(*So through these expeditions, I began to really appreciate the beauty of the ocean floor, like the amazing animals that live in the rifts of the deep ocean. They're almost alien to Earth.*)。
	Simul-MuST-C	所以，完成了这些探险之后，并且真正开始欣赏那里有什么，比如在深海热泉那里，我们有这些令人惊叹的动物——它们基本上就是地球上的外星人 (*So, having done these expeditions and really starting to appreciate what's out there, like at the hydrothermal vents, we have these amazing animals - they're basically aliens on Earth.*)。
	Omission-tgt	完成了这些探险之后，并开始了解那里有什么，例如在深海热液喷口处有惊人的动物——它们基本上是地球上的外星生物 (*After completing these expeditions and starting to understand what's out there, for example, there are amazing animals at deep-sea hydrothermal vents - they're basically alien creatures on Earth.*)。
En-De	Source	Now, very quickly, another reason we cannot think straight about happiness is that we do not attend to the same things when we think about life, and we actually live.
	Omission-src	Now, another reason we cannot think straight about happiness is that we do not attend to the same things when we think about life and live.
	MuST-C	Nun, ganz kurz, ein anderer Grund, aus dem wir nicht klar über Glück nachdenken können ist, dass wir nicht auf die selben Dinge achten wenn wir über das Leben nachdenken und wenn wir tatsächlich leben (*Well, very briefly, another reason why we cannot think clearly about happiness is that we do not pay attention to the same things when we think about life and when we actually live*).
	Simul-MuST-C	Nun, sehr schnell, ein weiterer Grund, warum wir nicht klar über Glück nachdenken können, ist, dass wir nicht auf die gleichen Dinge achten, wenn wir über das Leben nachdenken, und wir tatsächlich leben (*Well, very quickly, another reason why we can't think clearly about happiness is that we don't pay attention to the same things when we think about life, and we actually live*).
	Omission-tgt	Nun, ein weiterer Grund, warum wir nicht klar über Glück nachdenken können, ist, dass wir nicht auf dieselben Dinge achten, wenn wir über das Leben nachdenken und leben (*Well, another reason why we can't think clearly about happiness is that we don't pay attention to the same things when we think about and live life*).

Table 9: An example of created sentences. Omission-refers to outputs created at first pipeline, and Omission-tgt refers to outputs created at the second pipeline.

aligned the omitted source sentences—generated in the first step of our pipeline with their corresponding translations produced in the second step. We used the modified source (Omission) rather than the original MuST-C source because the omission process disrupts standard source-target correspondence. Aligning against the modified source enables more accurate measurement, as aligning content when the original source and target diverge significantly is inherently more difficult and less meaningful.

E Results on Local Agreement

En-Zh All metrics consistently ranked Simul-MuST-C highest, followed by MuST-C, with Omission performing the lowest. The only exception occurred with BLEU under the chunk size of 400, where Omission slightly outperformed MuST-C. However, this trend reversed as chunk size increased, and MuST-C eventually surpassed Omis-

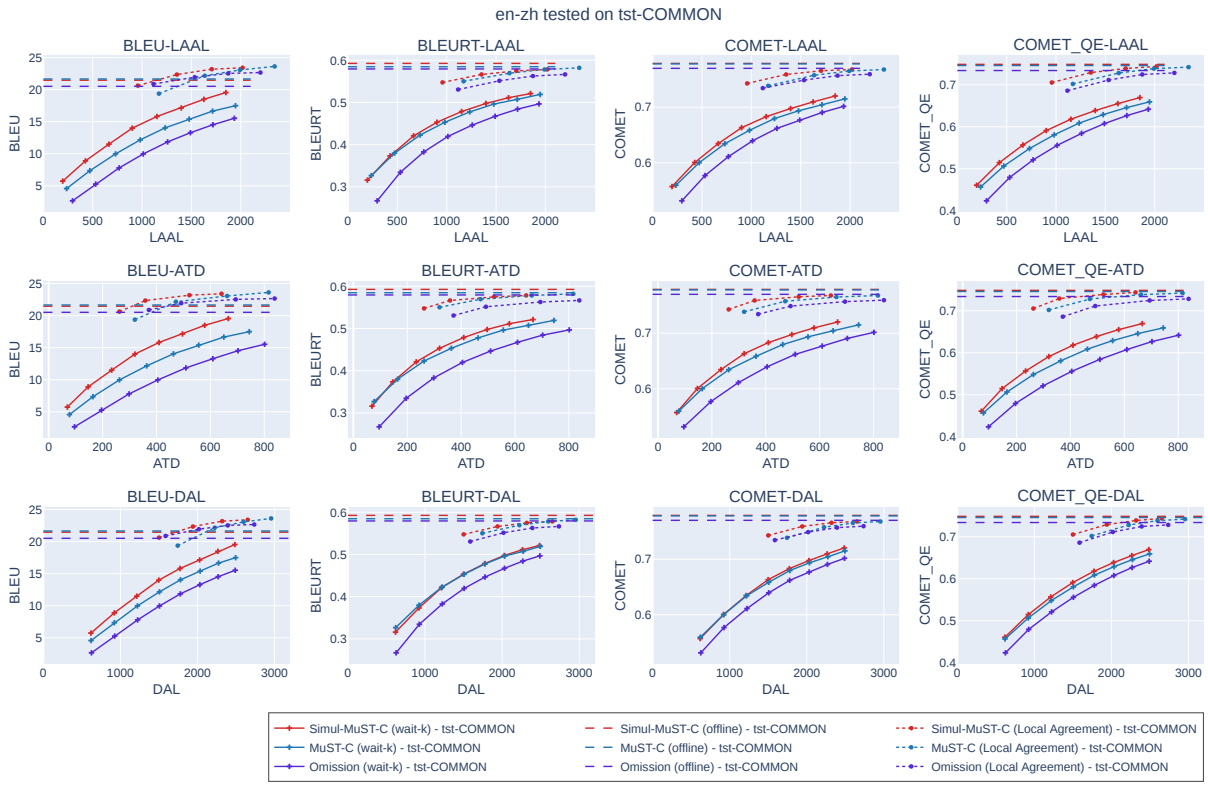

Figure 4: Results for En–Zh on tst-COMMON. The solid line with crosses shows wait-k decoding. The dotted line with circles represents Local Agreement. The dashed line shows the offline results.

sion. Although the overall quality gap among the three models was relatively small, Omission consistently scored the lowest. This may be due to its omission of less important words, which violates the assumption in standard quality metrics.

On the other hand, when evaluated using NLI, however, a different trend emerged. Assuming that a higher number of entailment cases indicates better meaning preservation, Omission achieved the best performance, followed by Simul-MuST-C, with MuST-C performing the worst. This ranking directly contrasts with the results from standard quality metrics, further highlighting that omission-based translations, while penalized by conventional evaluation methods, can still successfully preserve the core meaning of the source.

In a standard latency metrics, Simul-MuST-C achieves the best performance in LAAL and DAL, both of which focus on the start timing of translation. Even in ATD, which considers both the start and end timings of the output, Simul-MuST-C still performs best. Notably, the advantage of Omission, its shorter target output, does not appear to be reflected in ATD, suggesting that ATD may overlook the impact of reduced output length.

Under the proposed target-based latency metric, the trend differs: Omission yields the shortest output durations, followed by Simul-MuST-C, while MuST-C is the longest. We hypothesize that Omission achieves the lowest latency in this metric due to the characteristics of its training data that is shorter translations with stronger monotonicity.

Interestingly, although MuST-C produces shorter target sentences than Simul-MuST-C in traiing data, its latency is higher. This can be explained by its lower monotonicity, which indicates that the system must consume more source input before generating output. This behavior results in longer latency, despite the shorter output length, and aligns with the observed trend between MuST-C and Simul-MuST-C in training data.

En-De In En-De, Omission consistently underperforms when evaluated with standard quality metrics such as BLEU and COMET. Among the three language pairs studied, the performance gap between Omission and the other two models, MuST-C and Simul-MuST-C, is the largest in this case. This suggests that, based on standard metrics, the Omission model may introduce more substantial quality loss in this language pair.

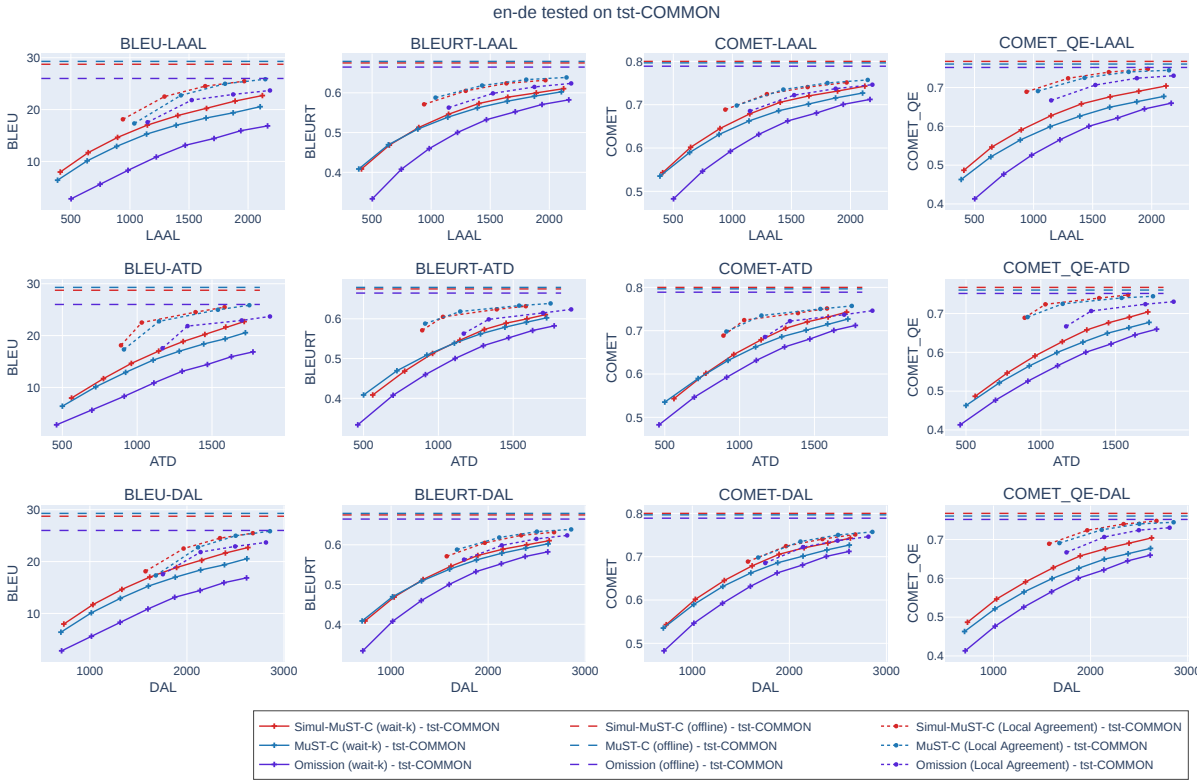

Figure 5: Results for En–De on tst-COMMON.The solid line with crosses shows wait-k decoding. The dotted line with circles represents Local Agreement. The dashed line shows the offline results.

However, when using NLI to assess whether the core meaning is preserved, the evaluation reveals a different outcome. As observed in the En-Ja and En-Zh results, the NLI-based evaluation shows a reversed trend. The number of entailment cases is highest for the Omission, followed by Simul-MuST-C, with MuST-C ranking last. This ordering directly contradicts the findings from standard metrics and highlights a key insight: while surface or embedding level similarity may decline, Omission still succeeds in preserving meaning. This again points out the limitations of conventional metrics in evaluating strategies that mimic human interpretation, such as omission.

In standard latency metrics, Simul-MuST-C achieves the best performance in LAAL and DAL, both of which emphasize the initial timing of translation. It also leads in ATD, which captures both when the translation starts and finishes. In contrast, Omission does not demonstrate any clear latency advantage under these conventional metrics, despite generating more concise translations that preserve core meaning.

In line with the pattern seen in the En–Zh results, Omission yields the lowest latency when evaluated with the proposed target-based metric. Simul-MuST-C follows, and MuST-C shows the highest latency. This result likely stems from the properties of the Omission training data, which encourage more compact and monotonic output.

Notably, even though MuST-C generates shorter outputs than Simul-MuST-C, its latency is greater. This can be explained by its lower monotonicity score, indicating that it requires more source input before producing translations. As a result, MuST-C incurs longer delays, highlighting that shorter output length alone does not guarantee lower latency.

F Results on wait-k

En-Ja Figure 3 shows the results for En-Ja. Under the wait-k policy, MuST-C shows the lowest translation quality across most metrics despite its alignment with tst-COMMON. In contrast, Simul-MuST-C achieves the highest quality across all metrics, including BLEU, despite structural differences from the subtitle style reference. This suggests that its emphasis on monotonicity and lack of length constraints, resulting in strong 1:1 input-output correspondence, achieves the best quality, even though it produces longer outputs. Omission consistently

Lang	System	Latency	Ent.	Neu.	Con.
En-Ja	MuST-C	la-200	2,179	461	201
		la-400	2,274	386	181
		la-600	2,302	374	165
		la-800	2,334	346	161
		la-1000	2,321	351	169
	Simul-MuST-C	la-200	2,262	411	168
		la-400	2,337	397	107
		la-600	2,356	392	93
		la-800	2,386	364	91
		la-1000	2,368	390	83
	Omission	la-200	2,300	389	152
		la-400	2,398	319	124
		la-600	2,390	335	116
		la-800	2,418	314	109
		la-1000	2,411	327	103
En-Zh	MuST-C	la-200	2,358	326	157
		la-400	2,394	306	141
		la-600	2,411	301	129
		la-800	2,447	279	118
		la-1000	2,446	265	130
	Simul-MuST-C	la-200	2,400	299	142
		la-400	2,466	268	107
		la-600	2,497	236	108
		la-800	2,505	232	104
		la-1000	2,510	243	119
	Omission	la-200	2,442	248	151
		la-400	2,506	208	127
		la-600	2,522	204	115
		la-800	2,525	203	113
		la-1000	2,513	209	119
En-De	MuST-C	la-200	2,205	282	93
		la-400	2,267	230	83
		la-600	2,296	210	74
		la-800	2,311	191	78
		la-1000	2,312	195	73
	Simul-MuST-C	la-200	2,245	243	92
		la-400	2,309	209	62
		la-600	2,318	196	66
		la-800	2,328	182	70
		la-1000	2,342	168	70
	Omission	la-200	2,263	207	110
		la-400	2,370	149	61
		la-600	2,378	149	53
		la-800	2,386	133	61
		la-1000	2,390	129	61

Table 10: Entailment classification results across language pairs (En-Ja, En-Zh, En-De), systems, and latency settings. Ent. stands for Entailment, Neu. stands for Neutral, and Con. stands for Contradiction.

ranks between Simul-MuST-C and MuST-C across all metrics, slightly outperforming MuST-C but falling behind Simul-MuST-C. This middle performance likely reflects output length and quality trade-offs, suggesting that longer target lengths in the training data tend to yield better translation quality. In terms of latency, as shown in the data analysis (Section 4.1), target sentences in the Omission had intermediate length, shorter than Simul-MuST-C but longer than MuST-C. If shorter translations directly led to reduced latency, as initially assumed in this work, MuST-C would be expected to yield the lowest latency. However, the results reveal the opposite: Simul-MuST-C, despite having the longest target side training data, achieved the lowest latency across all metrics, followed by Omission, with MuST-C performing the worst.

En-Zh Figure 4 shows the results for En-Zh. Omission produced the weakest results across both quality and latency, contrasting with the trend observed in En–Ja. BLEU revealed a clear quality ranking: Simul-MuST-C > MuST-C > Omission. This pattern held across embedding-based evaluation metrics as well, though the quality gap between Simul-MuST-C and MuST-C was less pronounced than in BLEU. Omission consistently ranked lowest across all metrics, with a substantial quality gap compared to Simul-MuST-C and MuST-C. Regarding latency, as discussed in the target sentence length analysis (Section 4.1), Omission achieves shorter outputs without a significant drop in translation quality. This led to the expectation that shorter translations would reduce latency, especially in metrics like ATD, which reflect both start and end timing. However, contrary to this expectation, omission did not lead to latency improvements. Instead, Simul-MuST-C, despite this model trained with the longest target sentences, achieved the lowest latency among the three, consistent with the experimental results in En–Ja.

En-De Figure 5 shows the results for En-De. Omission showed the weakest performance in both translation quality and latency. The quality drop in En–De was more pronounced than in En–Zh. While the quality difference between Simul-MuST-C and MuST-C was relatively small, Omission lagged significantly behind both. This trend was consistent across all evaluation metrics, surface-based BLEU as well as embedding-based BLEURT, COMET, and COMET-QE. Regarding latency, as noted in the Sentence Length Analysis (Section 4.1), Omission produced shorter translations without a significant loss in quality, leading us to expect lower latency. However, in En–De, latency metrics showed no improvement. Contrary to expectations, reducing sentence length through omission did not result in measurable latency gains, consistent with the results in En-Ja and En-Zh.

En-Ja	Source	You can imagine how startling then it is when you have children who are born who are two people inside of one body.
	MuST-C	想像できますか (*Can you imagine*)?
	Simul-MuST-C	あなたは想像できます、どれほど驚くべきものになるか、それが、子供を持つとき、生まれた、二人の子供である、一対の体の中で (*You can imagine how amazing it would be when you have a child, two children born in a pair of bodies.*)。
	Omission	想像してみてください、どれほど驚くべきか、あなたが持っているとき (*Imagine how amazing it would be if you had*)。
En-Zh	Source	And let me close with three words of my own: I do remember.
	MuST-C	让我来跟你们讲三个词 (*Let me tell you three words*)。
	Simul-MuST-C	而且让我以这个结束三个词 (*And let me end with these three words*)。
	Omission	让我结束 (*Let me finish*)。
En-De	Source	I've come to understand the sentiments of George Burns, who was performing still in Las Vegas well into his 90s.
	MuST-C	Ich habe verstanden, dass es sich um die Gefühle von George Bush handelt (*I understand that these are the feelings of George Bush*).
	Simul-MuST-C	Ich habe verstanden die Gefühle von George Berns, der immer noch in Las Vegas auftrat, bis in seine 90er Jahre (*I understood the feelings of George Berns, who was still performing in Las Vegas until his 90s*).
	Omission	Ich habe verstanden (*I have understood*).

Table 11: An example of generated sentence when the value of k is three in wait-k setting.

En-Ja	Source	I was the second volunteer on the scene, so there was a pretty good chance I was going to get in.
	MuST-C	私が2番目のボランティアでしただから参加する可能性は大きかったのです (*I was the second volunteer, so my chances of participating were high.*)。
	Simul-MuST-C	それは2番目のボランティアでした、現場で、だから、かなり良いチャンスがありました、私が参加できる (*It was the second volunteer on-site, so there was a pretty good chance I could participate.*)。
	Omission	それは2番目のボランティアでした。だから、良いチャンスがありました。入るための (*It was the second volunteer. So I had a good chance to get in.*)。
En-Zh	Source	So of course, random sequences of instructions are very unlikely to sort numbers, so none of them will really do it.
	MuST-C	当然,随机序列的指令是非常不可能排序的数字。因此,没有一个数字能真正排序 (*Of course, a random sequence of instructions is very unlikely to be a sequence of numbers. Therefore, no number can really be sequenced.*)。
	Simul-MuST-C	所以,当然,随机指令序列是非常不可能的分类数字,所以没有一个会真正做到 (*So, of course, random instruction sequences are extremely unlikely to be class numbers, so none of them will actually do*)。
	Omission	当然,随机序列的指令是不太可能的。所以没有一个会这样做 (*Of course, a random sequence of instructions is unlikely. So no one will do this*)。
En-De	Source	I want to get back to my suitcase and tell you what I packed for this exciting week here at TED.
	MuST-C	Ich möchte zu meinem Koffer zurückkehren. Und Ihnen sagen, was ich getan habe. Ich habe mich für diese aufregende Woche hier bei TED gepackt (*I want to return to my suitcase and tell you what I've done. I've packed myself for this exciting week here at TED*).
	Simul-MuST-C	Ich möchte zurückkommen zu meinem Koffer. Und dir sagen, was ich für diese aufregende Woche hier bei TED gepackt habe (*I want to go back to my suitcase and tell you what I've packed for this exciting week here at TED*).
	Omission	Ich möchte zu meinem Koffer zurückkehren. Und Ihnen sagen, was ich für diese Woche bei TED verpackt habe (*I'd like to return to my suitcase and tell you what I packed for this week at TED*).

Table 12: An example of generated sentence when the chunk size is 400 in Local Agreement setting.

En-Ja	Source	So, not exactly what I was hoping for, but off I went – up the stairs, down the hall, past the 'real' firefighters, who were pretty much done putting out the fire at this point, into the master bedroom to get a pair of shoes.
	MuST-C	私が望んでいたこととは まったく異なり とにかく 階段を登り 実際の消防士を 通り抜けました 消防士は 火を消すのに 苦労していました 主人の寝室に行き 靴を買いました (*So anyway, not at all what I wanted, I walked up the stairs, past actual firefighters who were struggling to put out the fire, and into the master bedroom to get some shoes.*)。
	Simul-MuST-C	だから、私が望んでいたこととは正確には違いますが、しかし、私は出発しました、階段を登り、廊下を通り、本物の消防士たちを通り過ぎ、彼らはこの時点で火を消すのにかなり苦労していました、主人の寝室に入り、靴を手に入れるために (*So, not exactly what I had hoped for, but off I went, up the stairs, down the hallway, past the real firefighters who were having a pretty hard time putting out the fire at this point, into the master bedroom to get my shoes.*)。
	Omission	だから、私が望んでいたことではなく、私は出て行きました、階段を上がって、本当の消防士たちを通り過ぎ、火を消すのに苦労していた、師匠の寝室に、靴を手に入れるために (*So, not what I wanted, I went out, up the stairs, past the real firemen who were struggling to put out the fire, to my master's bedroom, to get my shoes.*)。
En-Zh	Source	If we look at what's really happening in the online world, we can group the attacks based on the attackers.
	MuST-C	如果我们看看网络世界到底发生了什么, 我们可以根据攻击者组织攻击者 (*If we look at what is happening in the cyber world, we can organize attackers by*)。
	Simul-MuST-C	如果我们看在网络世界中真正发生的事情,我们可以组织基于攻击者的攻击 (*If we look at what is really happening in the cyber world, we can organize attacks based on the attackers*)。
	Omission	如果我们看看网络世界中正在发生的事情,我们可以根据攻击者组织攻击 (*If we look at what is happening in the cyber world, we can organize attacks based on the attackers*)。
En-De	Source	So get in the game. Save the shoes.
	MuST-C	Also gehen Sie ins Spiel, sparen Sie die Schuhe (*So go into the game, save the shoes*).
	Simul-MuST-C	Also, steigen Sie ins Spiel, sparen Sie die Schuhe (*So, get in the game, save the shoes*).
	Omission	Also, gehen Sie ins Spiel. Sparen Sie Schuhe (*So, get in the game. Save on shoes*).

Table 13: An example of generated sentence under an offline setting.

G Analysis on Generated Sentences

Local Agreement Table 12 shows an example of generated sentence when the chunk size is 400 in Local Agreement setting, results on tst-COMMON. The overall trend in Table 12 is similar to what was observed in Table 11, but translation quality is generally higher,even for the model trained on the Omission dataset. In En–Ja, all three models produce translations of relatively similar quality, in contrast to the results under wait-k, where quality differences between datasets were more pronounced. In En–Zh, the model trained on Omission performs better than it did with wait-k, aligning with the findings from the previous section, where Local Agreement resulted in smaller quality gaps. Although the underlined part of the source sentence is missing in the output, the translation still captures most of the original meaning and successfully omits *really*,a word that was frequently removed during the Dataset Creation process.

A similar pattern is observed in En–De, where the Omission model also performs better than it did under wait-k. The output preserves most of the source content, omitting *exciting*, a non-essential modifier that adds emphasis but does not impact the core meaning.

These quality gap between wait-k and Local Agreement can be attributed to the flexibility of the Local Agreement decoding policy. Unlike wait-k, which enforces a strict alternation between reading and writing, Local Agreement allows the model to dynamically determine when to read and write. This adaptability is particularly beneficial when the training data includes structural asymmetries, such as proposed dataset that includes omission. Our results suggest that for models trained on unbalanced or selectively reduced data, adaptive decoding policies like Local Agreement are better suited than rigid fixed policies like wait-k.

Wait-k Table 11 example of generated sentence when the value of k is three in wait-k setting, results on tst-COMMON. Similar to the sentence length patterns observed in our created data analysis, the output quality trends also differ across lan-

guage pairs. In En–Ja, Simul-MuST-C produced the highest quality translations, followed by Omission, with MuST-C performing the worst. Notably, only Simul-MuST-C was able to fully translate the underlined portion of the source sentence. Omission translated only the first half, and MuST-C barely captured the content at all.

In contrast, for En–Zh and En–De, the pattern reversed: Omission resulted in the lowest quality outputs. The underlined content was omitted entirely in the Omission outputs, whereas both MuST-C and Simul-MuST-C included it in their translations.

We found that these quality differences correlate with target sentence length. In En–Ja, Simul-MuST-C produced the longest outputs, Omission ranked in the middle, and MuST-C had the shortest, as described in the Created Data Analysis section. The output quality ranking followed the same order: Simul-MuST-C > Omission > MuST-C. In En–Zh and En–De, Simul-MuST-C again produced the longest outputs, MuST-C the second longest, and Omission the shortest—and once again, translation output quality followed this length-based order.

These findings suggest that while our data creation process aimed to remove only less important word, described in the Created Data Analysis section, the models trained on omission-involved data struggled to replicate this behavior during generation. That is, although the training data was designed to omit less critical content, the trained model often failed to reproduce this selective omission. Instead, it frequently omitted key parts of the input, resulting in degraded output quality. This indicates that the omission strategy is difficult to replicate under current training and decoding frameworks.

Offline Table 13 presents an example of generated output in the offline setting, evaluated on tst-COMMON. The quality trend differs from what was observed in Table 11 and Table 12, and instead aligns more closely with the observations from the Created Data Analysis.

Across all language pairs, the three models produce translations of relatively similar quality, with a slight drop observed in the outputs from the Omission. Since omission involves the deletion of less important words, some level of quality degradation is expected. However, the deterioration is not as pronounced as what we observed in the wait-k and Local Agreement results.

In En–Ja, for example, the phrase *pretty much* *done* is omitted, while in En–Zh, the word *really* is missing from the Omission output. Both are emphasis markers that contribute more to tone than to essential meaning. Their absence reflects the goal of our approach to shorten translations by removing less critical words and reduce overall latency. In En–De, all models achieve comparable quality.

From this perspective, the minor quality loss may be acceptable, especially when the goal is faster output. However, this behavior challenges the current assumption that all content in the source must be fully represented in the target. Our findings suggest that even when 1:1 source-target correspondence is not strictly maintained, offline setting results exhibit that translations can still be semantically accurate and practically useful, which highlights the potential for more flexible translation strategies.