ExPosST: Explicit Position Allocation for LLM-Based Simultaneous Machine Translation

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have achieved remarkable performance in simultaneous machine translation (SimulMT) via attention mask and positional reordering strategies. However, these approaches have strict constraints on positional encoding methods, such as ALiBi, which limit their general application. In this work, we introduce ExPosST, a simple and general framework to apply decoder-only LLMs to SimulMT tasks. ExPosST explicitly allocates the position range in the source and translation tokens, allowing decoding with KV cache under all positional methods. Experiments on multiple models show that ExPosST has comparable performance with state-of-the-art approaches in LLMs using ALiBi, while outperforming them in mainstream RoPE-based LLMs.

1 Introduction

Simultaneous Machine Translation (SimulMT) generates target language output in real time as portions of the source sentence are received. Due to its significant application value in real-world scenarios, such as international conferences and academic lectures, it has garnered much attention recently (Ma et al., 2019; Zhang and Feng, 2023; Agostinelli et al., 2024). As Large Language Models (LLMs) have achieved great performance in Neural Machine Translation (NMT) (Alves et al., 2023; Xu et al., 2024), some studies have attempted to leverage LLMs for SimulMT (Wang et al., 2024b; Agostinelli et al., 2024). Prior work usually constructs prefix-to-prefix datasets for fine-tuning (Koshkin et al., 2024). Nevertheless, as shown in Figure 1, when a new source word arrives, the positions of target tokens shift, and the KV cache must be recomputed due to the positional information no longer matching actual ones, leading to increased computational overhead. To solve this issue, Raffel et al. (2024) propose SimulMask, which uses a modified ALiBi positional embedding (Press et al., 2022) to store positional information separately from the KV cache. However, mainstream LLMs such as LLaMA 3 (Dubey et al., 2024) and Qwen 2.5 (Yang et al., 2024) often adopt Rotary Positional Embedding (RoPE) (Su et al., 2024), in which positional information is encoded directly into the KV cache. Therefore, SimulMask does not apply to mainstream LLMs.

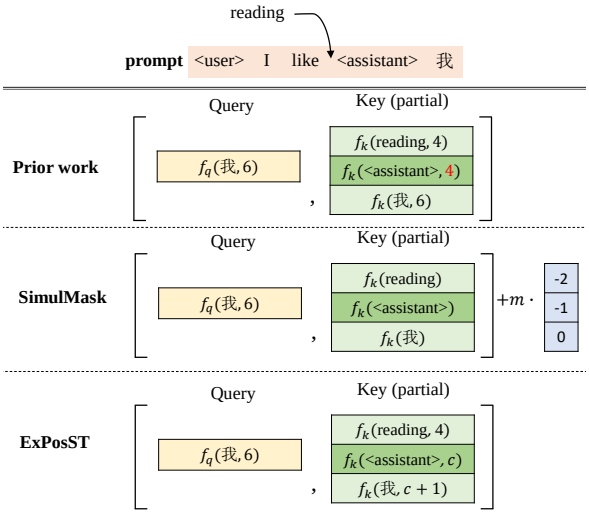


Figure 1: An example of attention score computing in a certain step of SimulMT in Prior works, SimulMask, and ExPosST methods. The Key cache is marked as dark green.

To address this issue, we propose ExPosST, an **Ex**plicit **Pos**itioning Allocation Framework for **S**imultaneous Machine **T**ranslation with LLM. During inference, we pre-allocate a fixed-length chunk for source-language tokens and generate the translation after the chunk. When the number of source tokens exceeds the length of the current chunk, a new chunk is allocated after the latest target output. We also adjust the format and attention mask during fine-tuning to avoid a mismatch between fine-tuning and inference. This framework prevents the target positions from shifting as the source input increases.

The contributions of this paper are summarized as follows:

- We propose ExPosST, a novel framework that avoids output token shifts caused by the expansion of source language input. This framework is applicable across all LLMs.
- Experiments on different LLMs show that our method achieves better performance in mainstream RoPE-based LLMs, and achieves comparable performance with SimulMask in ALiBi-based LLMs.

2 Background

Simultaneous Machine Translation (SimulMT) aims to generate target translations in real-time as the source sentence is being received. Formally, given a full source sentence $(s_1, \ldots, s_{|S|})$ and a target sentence $(t_1, \ldots, t_{|T|})$, SimulMT needs to generate each target token t_j based only on a prefix of the source sentence $(s_1, \ldots, s_i)$, where $i \leq |S|$.

To balance translation quality and latency, SimulMT employs a policy to decide whether to wait for additional source words (READ) or to generate translations (WRITE). An example of such a policy is wait-k (Ma et al., 2019), which reads k words at first, then alternates between writing one target word and reading one source word.

To apply LLMs to SimulMT, previous works usually construct prefix-to-prefix training data from offline translation pairs to train the LLM's ability to begin translating with partial input. These methods often adopt an offline translation prompt format <user> $s_1, \ldots, s_{|S|}$ <assistant> $t_1, \ldots, t_{|T|}$ (Agostinelli et al., 2024; Koshkin et al., 2024). However, Raffel et al. (2024) points out that this method introduces mismatches between fine-tuning and inference in KV cache usage, target tokens shifting during incremental decoding, and additional computational overhead.

To overcome these limitations, Raffel et al. (2024) introduces SimulMask, which uses an attention mask scheme to mimic real-time READ/WRITE decisions, and integrates a modified ALiBi positional encoding method. However, this method cannot be used in other position embeddings, such as RoPE.

3 Methodology

In this section, we propose ExPosST, a novel framework designed for all decoder-only LLMs. ExPosST allocates the range of positions of input tokens to avoid the shift of positions in target tokens.

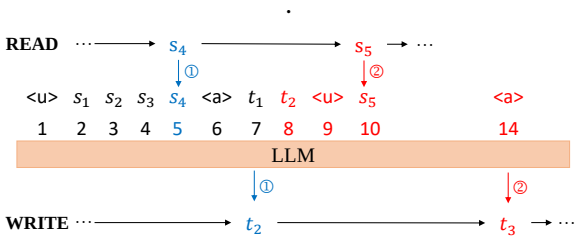


Figure 2: The two-step examples of the inference process in ExPosST.

3.1 Inference

To avoid the shift of positions in target tokens when using the KV cache during Simultaneous Translation, we allocate $chunk\_size$ positions to input tokens. The newly source input words will be added in the chunk, while the target word will be generated after the previous output (step ① in Figure 2). Once the length of the new received source word exceeds the chunk size (step ② in Figure 2), we will allocate another source input chunk after the target output, and set the new start position of the target after the chunk. To help LLM separate between the source and target languages, we use the conversational prompt format of LLM, where the source language is in <user> segment and the target language is generated in <assistant> segment.

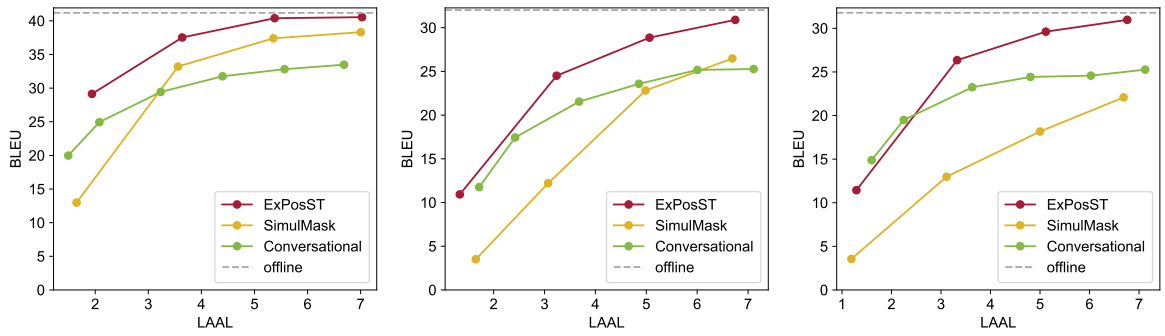
3.2 Fine-tuning

Based on the inference process in Section 3.1, to avoid the mismatch between fine-tuning and inference, we first segment the source language sentence into multiple parts based on the predefined chunk size. Then, for each part, we collected the new output tokens generated during the input of the entire chunk based on the policy, and put them in the output in this round. If the source token has not yet been received when the target token for the next prediction during inference, the keys of these source tokens will be masked in the query of the output tokens, similar to Raffel et al. (2024).

4 Experiments

4.1 Settings

We conducted experiments on English-French, English-Italian, English-Dutch, English-Romanian, and English-German language pairs from IWSLT 2017 (Cettolo et al., 2017). And we used



(a) English-French (b) English-Italian (c) English-Dutch

Figure 3: Translation quality plotted against latency for `Qwen2.5-1.5B-Instruct` on the English-French, English-Italian and English-Dutch language pairs.

`Qwen2.5-1.5B-Instruct` (Yang et al., 2024) with RoPE as the base model. We evaluated with the following baselines. The hyperparameters and prompts are shown in Appendix A.

- **ExPosST:** We used the wait-k policy and set *chunk_size* to 16 during fine-tuning and inference.
- **SimulMask:** Raffel et al. (2024) utilizes a novel attention mask approach that models simultaneous translation during fine-tuning by masking attention for a desired decision policy. We chose the wait-k policy. Specifically, for the models without the ALiBi position embeddings, we did not use modified ALiBi during fine-tuning.
- **Conversational:** Wang et al. (2024a) builds up a conversational prompt structure for incremental decoding, and creates supervised fine-tuning training data by segmenting parallel sentences using an alignment tool and a novel augmentation technique to enhance generalization. It uses "read-n & incremental decoding" policy (Wang et al., 2024b) during evaluation, which reads n words at each step and subsequently continues translating until the end-of-sequence token is generated. In this experiment, n is selected from {2,3,5,7,9,11}.
- **offline:** We conducted experiments on fine-tuning on full sentence pairs and offline translation during evaluation as a reference.

We used greedy search for all methods. For approaches with wait-k policy, we set k to {1,3,5,7} during evaluation, and the fine-tuning configuration employed k four higher than those used in the evaluation, as referred to in Ma et al. (2019).

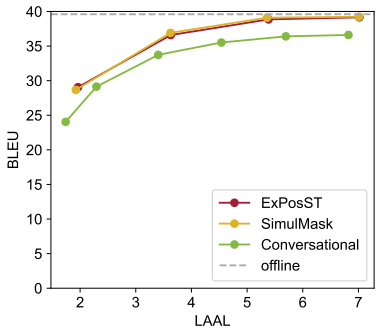


Figure 4: Translation quality plotted against latency for `falcon-rw-1b` on the English-French language pairs.

Our fine-tuning process was implemented in the Simul-LLM (Agostinelli et al., 2024) framework. Inference was in the Simul-LLM agent (Agostinelli et al., 2024) integrated with the SimulEval toolkit (Ma et al., 2020). We used detokenized BLEU with SacreBLEU (Post, 2018) and COMET[1] (Bosselut et al., 2019) for the quality metric. Latency was determined using Length-Adaptive Average Lagging (LAAL) (Papi et al., 2022).

4.2 Main Results

Figure 3 shows the results of BLEU and LAAL on English-French, English-Dutch, and English-Italian language pairs, and other results are shown in Appendix B. ExPosST comprehensively outperformed both SimulMask and Conversational approaches, delivering the best results while preserving the original KV cache without recomputation. Specifically, compared to (Wang et al., 2024a), ExPosST has longer input and output sequences within a single conversational round during training and evaluation. The longer sequence brings more

[1] `https://huggingface.co/Unbabel/wmt22-cometkiwi-da`

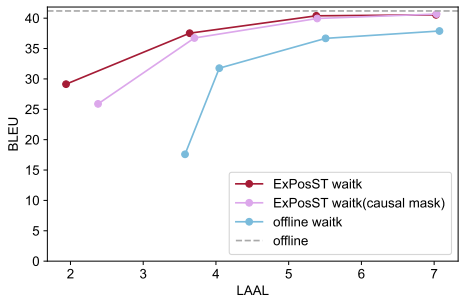


Figure 5: Effect of Inference Process in English-French language pair.

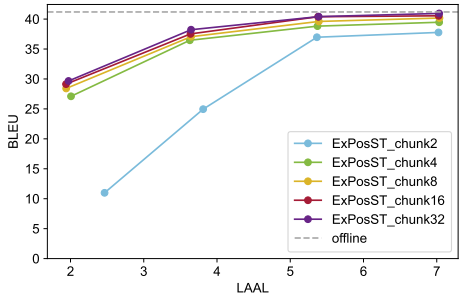


Figure 6: The result of different *chunk_size* on the English-France language pairs.

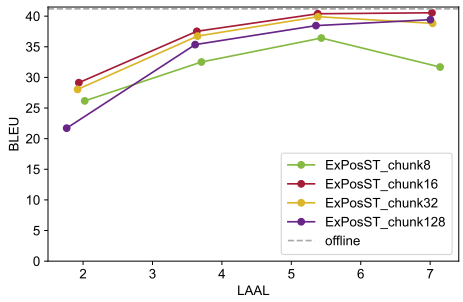


Figure 7: The result of the mismatch in chunks between training and testing in the English-French language pair. The training chunk size is fixed at 16.

semantically coherent text, which is better aligned with natural conversational structure.

Figure 4 shows results on `falcon-rw-1b` with ALiBi position embedding. ExPosST achieved results comparable to those of SimulMask, indicating that the format modifications did not cause performance degradation.

5 Analysis

5.1 Effect of Inference process

To further evaluate the impact of the inference framework, we compare ExPosST with two variants: one using offline training with a causal mask and conversational prompt, and another using ExPosST fine-tuning with a causal mask. As shown in Figure 5, even without a modified attention mask, the framework leads to strong performance, which has the ability to understand the conversational prompt. By adding a simultaneous mask, the performance is further improved.

5.2 Effect of Chunk Size

We examined how varying the *chunk_size* parameters affects translation performance. Figure 6 shows experiments with *chunk_size* in {2,4,8,16,32} during both fine-tuning and inference. The experiment reveals that performance will be stable when *chunk_size* is larger than a threshold. However, larger *chunk_size* brings more <pad> tokens during batch processing in training, giving more computational cost. What's more, if *chunk_size* is too small, more <user> and <assistant> tokens will be added, leading to an increased length in the prompt. Therefore, the length of the chunk will be set to the mid value. A detailed analysis is provided in Appendix C.

5.3 Effect of Mismatch in Chunk Size

We also evaluate the impact of the mismatch of chunk size during fine-tuning and inference. We set the *chunk_size* to 16 for training, and test {8,16,32,128} for testing. From the result in Figure 7, we find that the mismatch of chunks in fine-tuning and inference leads to a drop in translation quality. Moreover, the performance drop becomes more severe as the mismatch increases.

6 Conclusion

In this work, we proposed ExPosST, a novel framework for applying decoder-only Large Language Models (LLMs) to Simultaneous Machine Translation (SimulMT). By explicitly allocating the position range of source language input, our approach ensures stable KV-cache utilization while maintaining compatibility with different LLMs. Experimental results show that ExPosST achieves comparable performance to SimulMask on ALiBi-based LLMs and significantly outperforms existing approaches on mainstream RoPE-based LLMs.

Limitations

Due to computational limits, we conducted experiments primarily on Qwen2.5-1.5B-Instruct and falocn-rw-1b without evaluating our method on other models or across different parameter scales.

References

Victor Agostinelli, Max Wild, Matthew Raffel, Kazi Fuad, and Lizhong Chen. 2024. Simul-LLM: A framework for exploring high-quality simultaneous translation with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10530–10541, Bangkok, Thailand. Association for Computational Linguistics.

Duarte Alves, Nuno Guerreiro, João Alves, José Pombal, Ricardo Rei, José de Souza, Pierre Colombo, and Andre Martins. 2023. Steering large language models for machine translation with finetuning and in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11127–11148, Singapore. Association for Computational Linguistics.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy. Association for Computational Linguistics.

Mauro Cettolo, Marcello Federico, Luisa Bentivogli, Jan Niehues, Sebastian Stüker, Katsuhito Sudoh, Koichiro Yoshino, and Christian Federmann. 2017. Overview of the IWSLT 2017 evaluation campaign. In *Proceedings of the 14th International Conference on Spoken Language Translation*, pages 2–14, Tokyo, Japan. International Workshop on Spoken Language Translation.

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.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Roman Koshkin, Katsuhito Sudoh, and Satoshi Nakamura. 2024. TransLLaMa: LLM-based simultaneous translation system. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 461–476, Miami, Florida, USA. Association for Computational Linguistics.

Mingbo Ma, Liang Huang, Hao Xiong, Renjie Zheng, Kaibo Liu, Baigong Zheng, Chuanqiang Zhang, Zhongjun He, Hairong Liu, Xing Li, Hua Wu, and Haifeng Wang. 2019. STACL: Simultaneous translation with implicit anticipation and controllable latency using prefix-to-prefix framework. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3025–3036, Florence, Italy. Association for Computational Linguistics.

Xutai Ma, Mohammad Javad Dousti, Changhan Wang, Jiatao Gu, and Juan Pino. 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.

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.

Matt Post. 2018. A call for clarity in reporting bleu scores. *arXiv preprint arXiv:1804.08771*.

Ofir Press, Noah A. Smith, and Mike Lewis. 2022. Train short, test long: Attention with linear biases enables input length extrapolation. *Preprint*, arXiv:2108.12409.

Matthew Raffel, Victor Agostinelli, and Lizhong Chen. 2024. Simultaneous masking, not prompting optimization: A paradigm shift in fine-tuning LLMs for simultaneous translation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18302–18314, Miami, Florida, USA. Association for Computational Linguistics.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Minghan Wang, Thuy-Trang Vu, Yuxia Wang, Ehsan Shareghi, and Gholamreza Haffari. 2024a. Conversational simulmt: Efficient simultaneous translation with large language models. *Preprint*, arXiv:2402.10552.

Minghan Wang, Thuy-Trang Vu, Jinming Zhao, Fatemeh Shiri, Ehsan Shareghi, and Gholamreza Haffari. 2024b. Simultaneous machine translation with large language models. In *Proceedings of the 22nd Annual Workshop of the Australasian Language Technology Association*, pages 89–103, Canberra, Australia. Association for Computational Linguistics.

Haoran Xu, Young Jin Kim, Amr Sharaf, and Hany Hassan Awadalla. 2024. A paradigm shift in machine translation: Boosting translation performance of large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Shaolei Zhang and Yang Feng. 2023. Hidden markov transformer for simultaneous machine translation. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Appendix

A Hyperparameters

A.1 Training Hyperparameters

The fine-tuning hyperparameters of each baseline are shown in Table 1. Empirically, we find that the second epoch achieves the best performance in all methods, so we use the second epoch for evaluation.

Hyperparameter	Value
Weight Precision	bfloat16
Optimizer	AdamW
Learning Rate	$2 \cdot 10^{-4}$
LR Scheduler	Inverse Sqrt
Weight Decay	0.1
Warmup Ratio	0.03
Max Gradient Norm	1
Max Sequence Length	512
Epochs	3
Batch size	64

Table 1: Fine-tuning hyperparameters for all models and all methods.

For the Conversational baseline, we use the Itermax method from the SimAlign toolkit, leveraging XLM-RoBERTa base (Conneau et al., 2020) to align words. The hyperparameter in Conversational is shown in Table 2.

Hyperparameter	Value
δ_{max}	10
β	0.5
ρ_{min}	0.5
ρ_{max}	0.9

Table 2: hyperparameters in Conversational baseline.

A.2 Prompts

In SimulMask, offline baseline, when using `falcon-rw-1b`, the prompt structure is in the following format:

```
Translate the following sentence from
[SRC] to [TGT]: [SRC-Sentence]
Assistant:[TGT-Sentence]<|endoftext|>
```

And when using `Qwen2.5-1.5B-Instruct`, the prompt structure:

```
Translate the following sentence from
[SRC] to [TGT]: [SRC-Sentence]
<|im_start|>assistant
[TGT-Sentence]<|im_end|>
```

Alternatively, the prompt of Conversational baseline is in the following format when using `falcon-rw-1b`:

```
Translate the following sentence from
[SRC] to [TGT]:
User:[SRC-1]<|endoftext|>
Assistant:[TGT-1]<|endoftext|>
...
User:[SRC-n]<|endoftext|>
Assistant:[TGT-n]<|endoftext|>
```

And in `Qwen2.5-1.5B-Instruct`, the prompt structure in Conversational baseline is:

```
Translate the following sentence from
[SRC] to [TGT]:<|im_start|>user
[SRC-1]<|im_end|><|im_start|>assistant
[TGT-1]<|im_end|>
...
<|im_start|>user
[SRC-n]<|im_end|><|im_start|>assistant
[TGT-n]<|im_end|>
```

In ExPosST, because the end-of-sentence token means end of translation in the wait-k policy, in `falcon-rw-1b` model, the prompt structure is changed to:

```
Translate the following sentence from
[SRC] to [TGT]:
User:[SRC-1]
Assistant:[TGT-1]
...
User:[SRC-n]
Assistant:[TGT-n]<|endoftext|>
```

And in `Qwen2.5-1.5B-Instruct`, the prompt structure in ExPosST is:

```
Translate the following sentence from
[SRC] to [TGT]:<|im_start|>user
[SRC-1]<|im_start|>assistant
[TGT-1]
...
<|im_start|>user
[SRC-n]<|im_start|>assistant
[TGT-n]<|im_end|>
```

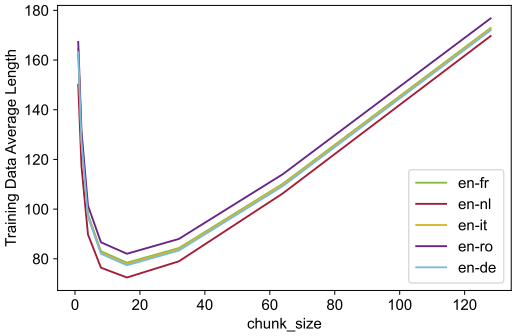


Figure 8: The effect of chunk size on average data length on the IWSLT2017 dataset.

B Numerical Results

We show the numerical BLEU, COMET, and LAAL results in `falcon-rw-1b` and `Qwen2.5-1.5B-Instruct` for ExPosST, Simul-Mask (Raffel et al., 2024), and Conversational (Wang et al., 2024a) in Table 3, Table 4, Table 5, and Table 6.

C Effect of Chunk Size on Training Data Length

As mentioned in Section 5.2, both small and large chunk sizes can increase the length of training data. So we tested the average training data length by different $chunk\_size$ in English-French (en-fr), English-Italian (en-it), English-Dutch (en-nl), English-Romanian (en-ro), and English-German (en-de) of the IWSLT 2017 dataset (Cettolo et al., 2017). As the result shown in Figure 8, the relationship between chunk size and average training data length follows a U-shaped pattern, with the shortest effective length observed when the $chunk\_size$ is around 16.

Baseline	en-fr	en-it	en-nl	en-ro	en-de
ExPosST wait-1	29.06 (1.96)	7.67 (1.29)	10.61 (1.28)	7.86 (1.29)	17.20 (1.59)
ExPosST wait-3	36.60 (3.63)	22.35 (3.26)	24.95 (3.31)	21.37 (3.32)	24.12 (3.23)
ExPosST wait-5	38.86 (5.39)	27.93 (5.07)	29.05 (5.13)	25.38 (5.13)	27.17 (4.91)
ExPosST wait-7	39.16 (7.02)	29.37 (6.73)	30.19 (6.77)	26.36 (6.81)	28.15 (6.57)
SimulMask wait-1	28.67 (1.92)	13.86 (1.47)	14.60 (1.36)	9.85 (1.36)	17.61 (1.62)
SimulMask wait-3	36.91 (3.62)	23.76 (3.28)	26.26 (3.33)	21.48 (3.32)	24.14 (3.23)
SimulMask wait-5	39.10 (5.36)	27.97 (5.10)	29.76 (5.14)	25.30 (5.15)	27.28 (4.91)
SimulMask wait-7	39.25 (7.01)	29.48 (6.75)	31.22 (6.77)	26.64 (6.82)	28.29 (6.57)
Conversational read-2	24.04 (1.74)	15.57 (1.93)	16.18 (1.87)	12.87 (1.91)	16.56 (1.82)
Conversational read-3	29.14 (2.29)	20.82 (2.62)	21.41 (2.53)	17.70 (2.53)	20.36 (2.40)
Conversational read-5	33.73 (3.40)	24.53 (3.83)	26.19 (3.77)	22.36 (3.75)	23.97 (3.56)
Conversational read-7	35.52 (4.54)	26.65 (4.95)	28.18 (4.93)	24.22 (4.90)	25.45 (4.72)
Conversational read-9	36.40 (5.70)	27.43 (6.09)	28.70 (6.09)	24.71 (6.09)	26.45 (5.86)
Conversational read-11	36.61 (6.82)	27.67 (7.21)	29.00 (7.21)	24.51 (7.33)	26.67 (7.01)

Table 3: Translation quality and latency results in BLEU and LAAL in `falcon-rw-1b`.

Baseline	en-fr	en-it	en-nl	en-ro	en-de
ExPosST wait-1	70.94 (1.96)	55.31 (1.29)	59.18 (1.28)	57.31 (1.29)	65.28 (1.59)
ExPosST wait-3	79.26 (3.63)	74.94 (3.26)	76.54 (3.31)	77.28 (3.32)	75.06 (3.23)
ExPosST wait-5	81.43 (5.39)	80.47 (5.07)	80.97 (5.13)	81.55 (5.13)	78.21 (4.91)
ExPosST wait-7	81.67 (7.02)	81.80 (6.73)	82.03 (6.77)	82.86 (6.81)	79.06 (6.57)
SimulMask wait-1	71.10 (1.92)	62.76 (1.47)	64.89 (1.36)	59.83 (1.36)	65.68 (1.62)
SimulMask wait-3	79.67 (3.62)	76.55 (3.28)	78.01 (3.33)	77.22 (3.32)	75.57 (3.23)
SimulMask wait-5	81.47 (5.36)	80.42 (5.10)	81.23 (5.14)	81.93 (5.15)	78.33 (4.91)
SimulMask wait-7	81.77 (7.01)	81.97 (6.75)	82.21 (6.77)	82.84 (6.82)	79.22 (6.57)
Conversational read-2	68.89 (1.74)	69.33 (1.93)	68.98 (1.87)	68.30 (1.91)	65.63 (1.82)
Conversational read-3	74.21 (2.29)	75.24 (2.62)	74.92 (2.53)	75.23 (2.53)	70.66 (2.40)
Conversational read-5	77.80 (3.40)	78.27 (3.83)	78.99 (3.77)	79.45 (3.75)	74.86 (3.56)
Conversational read-7	79.09 (4.54)	79.72 (4.95)	79.97 (4.93)	80.81 (4.90)	76.34 (4.72)
Conversational read-9	79.76 (5.70)	80.34 (6.09)	80.62 (6.09)	81.31 (6.09)	77.01 (5.86)
Conversational read-11	79.95 (6.82)	80.71 (7.21)	80.76 (7.21)	80.03 (7.33)	76.77 (7.01)

Table 4: Translation quality and latency results in COMET and LAAL in `falcon-rw-1b`.

Baseline	en-fr	en-it	en-nl	en-ro	en-de
ExPosST wait-1	29.14 (1.94)	10.93 (1.34)	11.44 (1.29)	7.97 (1.29)	17.94 (1.54)
ExPosST wait-3	37.53 (3.64)	24.51 (3.24)	26.35 (3.33)	22.79 (3.31)	25.31 (3.17)
ExPosST wait-5	40.39 (5.38)	28.86 (5.06)	29.62 (5.12)	26.89 (5.16)	28.44 (4.90)
ExPosST wait-7	40.55 (7.03)	30.89 (6.75)	30.97 (6.76)	28.62 (6.82)	29.38 (6.56)
SimulMask wait-1	12.98 (1.65)	3.51 (1.65)	3.55 (1.19)	1.37 (1.09)	8.70 (1.14)
SimulMask wait-3	33.21 (3.56)	12.21 (3.07)	12.98 (3.11)	12.38 (3.11)	20.19 (2.99)
SimulMask wait-5	37.40 (5.36)	22.82 (4.99)	18.17 (5.00)	19.68 (5.05)	25.78 (4.83)
SimulMask wait-7	38.32 (7.00)	26.48 (6.69)	22.08 (6.69)	22.29 (6.77)	28.62 (6.54)
Conversational read-2	19.97 (1.49)	11.76 (1.72)	14.88 (1.60)	11.31 (1.62)	13.81 (1.63)
Conversational read-3	24.94 (2.08)	17.44 (2.42)	19.49 (2.25)	16.13 (2.31)	17.27 (2.18)
Conversational read-5	29.44 (3.23)	21.54 (3.68)	23.24 (3.63)	19.91 (3.61)	20.84 (3.36)
Conversational read-7	31.76 (4.40)	23.58 (4.86)	24.43 (4.81)	22.23 (4.76)	22.70 (4.53)
Conversational read-9	32.82 (5.57)	25.17 (6.01)	24.57 (6.03)	22.93 (5.93)	23.76 (5.68)
Conversational read-11	33.48 (6.69)	25.27 (7.11)	25.26 (7.12)	23.49 (7.08)	24.52 (6.81)

Table 5: Translation quality and latency results in BLEU and LAAL in `Qwen2.5-1.5B-Instruct`.

Baseline	en-fr	en-it	en-nl	en-ro	en-de
ExPosST wait-1	73.99 (1.94)	62.62 (1.34)	61.85 (1.29)	57.48 (1.29)	71.04 (1.54)
ExPosST wait-3	81.05 (3.64)	77.79 (3.24)	79.13 (3.33)	79.98 (3.31)	79.05 (3.17)
ExPosST wait-5	83.45 (5.38)	81.87 (5.06)	82.82 (5.12)	84.08 (5.16)	81.58 (4.90)
ExPosST wait-7	83.84 (7.03)	83.49 (6.75)	83.91 (6.76)	85.22 (6.82)	82.28 (6.56)
SimulMask wait-1	53.34 (1.65)	42.44 (1.65)	48.86 (1.19)	44.80 (1.09)	55.07 (1.14)
SimulMask wait-3	76.66 (3.56)	66.19 (3.07)	67.59 (3.11)	67.50 (3.11)	70.01 (2.99)
SimulMask wait-5	80.41 (5.36)	76.64 (4.99)	73.92 (5.00)	76.50 (5.05)	76.10 (4.83)
SimulMask wait-7	81.67 (7.00)	79.75 (6.69)	76.74 (6.69)	79.21 (6.77)	79.49 (6.54)
Conversational read-2	71.41 (1.49)	72.10 (1.72)	74.47 (1.60)	72.23 (1.62)	70.14 (1.63)
Conversational read-3	75.69 (2.08)	76.94 (2.42)	77.80 (2.25)	77.31 (2.31)	74.04 (2.18)
Conversational read-5	79.16 (3.23)	79.94 (3.68)	80.51 (3.63)	81.17 (3.61)	77.47 (3.36)
Conversational read-7	80.59 (4.40)	80.96 (4.86)	81.55 (4.81)	82.38 (4.76)	79.18 (4.53)
Conversational read-9	81.31 (5.57)	82.19 (6.01)	81.70 (6.03)	83.25 (5.93)	79.95 (5.68)
Conversational read-11	81.71 (6.69)	82.32 (7.11)	82.63 (7.12)	83.58 (7.08)	80.41 (6.81)

Table 6: Translation quality and latency results in COMET and LAAL in `Qwen2.5-1.5B-Instruct`.