Conversational SIMULMT: Efficient Simultaneous Translation with Large Language Models

Minghan Wang[1], Thuy-Trang Vu[1], Yuxia Wang[2], Ehsan Shareghi[1], Gholamreza Haffari[1]
[1]Department of Data Science & AI, Monash University [2]MBZUAI
{minghan.wang,trang.vu1,ehsan.shareghi,gholamreza.haffari}@monash.edu
yuxia.wang@mbzuai.ac.ae

Abstract

Simultaneous machine translation (SIMULMT) presents a challenging trade-off between translation quality and latency. Recent studies have shown that LLMs can achieve good performance in SIMULMT tasks. However, this often comes at the expense of high inference costs and latency. In this paper, we propose a conversational SIMULMT framework to enhance the inference efficiency of LLM-based SIMULMT through multi-turn-dialogue-based decoding where source and target chunks interleave in translation history, enabling the reuse of Key-Value cache. To adapt LLMs to the proposed conversational decoding, we create supervised fine-tuning training data by segmenting parallel sentences using an alignment tool and a novel augmentation technique to enhance generalization. Our experiments with `Llama2-7b-chat` on three SIMULMT benchmarks demonstrate that the proposed method empowers the superiority of LLM in translation quality, meanwhile achieving comparable computational latency with specialized SIMULMT models.[1]

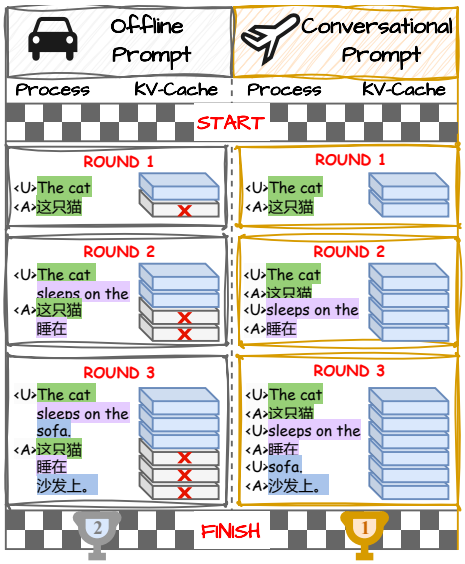


Figure 1: Comparison of offline prompt (left) and conversational prompt (right). Offline prompt inserts tokens mid-sequence, preventing KV-cache reuse (red X), while conversational prompt appends content sequentially, enabling efficient cache utilization (blue blocks).

1 Introduction

Simultaneous machine translation (SIMULMT) systems provide real-time translation of text input stream (Gu et al., 2017). This task plays an important role in real-world applications, such as facilitating communication in online conferences and generating live subtitles with strict latency requirements.

Although large language models (LLMs) have shown the potentials in machine translation (Hendy et al., 2023; Zhu et al., 2023), their applications to SIMULMT is non-trivial, as they are not inherently designed for simultaneous decoding. Recent works have attempted to adapt LLMs for SIMULMT with prefix fine-tuning, incremental decoding (Wang et al., 2023b) and learning to wait for more source tokens before translation (Koshkin et al., 2024). These works show LLMs, with careful prompt-engineering, could approach the performance of specialized SIMULMT models. However, high computational cost, slow inference, and high latency render these approaches impractical for real-world applications (Yuan et al., 2024). This is primarily due to the use of *offline prompting*, where arriving source tokens are inserted at the end of the source sequence, disrupting the continuity of the translation history (Figure 1 left). This prevents reusing cached target history states and requires re-computation of source and target representations.

To mitigate this issue, we propose *conversational prompt* that resemble the multi-turn dialogue nature of LLMs. Specifically, user inputs are treated as the source tokens to be read, while the LLM's responses are considered the predicted

[1]Code, weights, and data will be released with publication.

Proceedings of the 22nd International Conference on Spoken Language Translation (IWSLT 2025), pages 93–105

target tokens to be written. In our conversational SIMULMT, newly arrived source form the current instruction, while previous source tokens and their translations are treated as conversation history (Figure 1 right). This conversational prompt enables the reuse of Key-Value cache (Pope et al., 2023), as all content is appended incrementally without modifying the translation history. However, conversational SIMULMT poses new challenges for LLMs to comprehend the segmented source content and produce a coherent translation via multi-turn conversation.

To adapt the LLM to the conversational decoding format, we opt to perform supervised fine-tuning (SFT) on the pretrained LLM. But the challenge is the lack of the conversational SIMULMT data for SFT. Interleaving incomplete source and target segments in the dialogue history is unnatural (see Figure 1). This code-switching style is exhibited in some languages (Yong et al., 2023); however, it is the *continuation* rather than the *translation* of the previous content, making it challenging to leverage existing code-switched datasets for training. Therefore, we propose to curate the training data by segmenting parallel sentence pairs into smaller chunks based on a transformation of the word alignments. The segmented chunks are further augmented to handle different latency requirements.

Experiments on three SIMULMT benchmarks demonstrate the effectiveness of our proposed conversational SIMULMT in balancing the trade-offs between accuracy, speed and flexibility to different latency requirements. Compared to offline prompting, our method not only maintains strong performance, but also benefits from reduced latency. Notably, our method attains similar decoding speed to the LLM-based OFFLINEMT.

Our contributions are summarized as follows,

- We introduce conversational prompting to reduce the inference cost of LLM-based SIMULMT by leveraging its multi-turn dialogue capability and enabling efficient reuse of Key-Value cached computations.
- We present an automated training data curation pipeline that can turn any offline translation parallel corpus into the conversational prompt format and generalize with a novel augmentation strategy into any inference setting.
- Experiments demonstrate that the proposed conversational SIMULMT obtains up to 2× acceleration compared to the offline-prompting baseline while maintaining comparable translation quality, emphasizing its value in practical applications.

2 Background

Simultaneous Machine Translation (SIMULMT) Unlike *offline* machine translation (OFFLINEMT), where models generate target translation $\mathbf{y} = (y_1, ..., y_J)$ given a complete source sentence $\mathbf{x} = (x_1, ..., x_I)$, SIMULMT incrementally translates with partial source context $\mathbf{x}_{\leq t} = (x_1, ..., x_t)$ where $t \leq I$. A core component of SIMULMT is a read-write policy that decides whether to wait for new source tokens (`READ`) or generate target tokens (`WRITE`), balancing translation quality and latency.

Incremental Decoding Studies have explored adapting OFFLINEMT models for simultaneous decoding by performing offline decoding on incrementally updated histories (Liu et al., 2020; Nguyen et al., 2021; Polák et al., 2022; Guo et al., 2023). This involves a chunk-wise `READ` policy that reads n tokens per round and a `WRITE` policy that commits stable partial translations using the longest common prefix (LCP) (Polák et al., 2022) algorithm. LCP often causes high latency when candidates lack common initial tokens. Relaxed Agreement LCP (RALCP) (Wang et al., 2023b) was proposed to vote for accepting prefixes with candidate agreement above threshold γ.

SIMULMT with LLMs Since incremental decoding essentially repeats offline decoding, using offline-style translation prompts with LLMs is straightforward and aligns with their instruction-following capabilities (Xu et al., 2023). During each round, a source chunk is `READ` and appended to source history $\mathbf{x}$. LLMs generate translations using offline prompts as shown in Figure 1, which are then `WRITTEN` to target history $\mathbf{y}$.

3 Conversational SIMULMT

While incremental decoding with offline prompt enables LLMs to perform simultaneous decoding, it faces high computational latency due to the insertion of newly arrived source tokens in the middle of the prompt, disrupting the reuse of cached target history states. In this section, we propose conversational prompts to improve the decoding efficiency and balance quality-latency trade-off.

Setting	N-Shot	SacreBLEU	COMET
OFFLINEMT	0-Shot	30.99	84.95
Convers. SIMULMT	0-Shot	7.14	58.76
Convers. SIMULMT	5-Shot	13.51	69.03
Convers. SIMULMT 0-Shot Failure Case			
Chunk 1 Input:	Die Flugdaten zeigten, dass das		
Chunk 1 Response:	The flight data showed that the plane was flying at an altitude of 35,000 feet.		
Chunk 2 Input:	Flugzeug auch bei einem zweiten		
Chunk 2 Response:	The plane was also flying during the second flight.		
Reference	Flight data showed the plane had to pull out of second		

Table 1: Performance comparison of `Llama2-7b-chat` on WMT15 De->En test set in zero-shot and few-shot conversational SIMULMT settings. OFFLINEMT results are included as a baseline. The example failure case demonstrates how the LLM hallucinates completions (shown in red) when translating partial sentences, leading to compounding errors in subsequent chunks.

3.1 Decoding with Conversational Prompt

The efficiency improvement in LLMs hinges on maintaining the Key-Value (KV-) cache reuse, i.e. the decoding process must consistently add new tokens at the end of the sequence without altering the middle elements. When LLMs are performing multi-turn dialogues, the prompt for each turn is composed of a user input and assistant response separated by special tokens, and conversation histories are simply concatenated as the context (Touvron et al., 2023). Drawing parallels to multi-turn dialogues in LLMs, SIMULMT can also be viewed similarly, where user inputs and assistant responses are equivalent to `READ` and `WRITE` action. At round t, LLM reads a source context chunk X_t and writes its translation Y_t: "`<U>` X_t `<A>` Y_t". The already processed chunks are concatenated as contexts, serving the latest translation round of new incoming chunks. As all contents are appended incrementally, the reuse of KV-cache becomes feasible again like in multi-turn dialogue (see Figure 1). Our approach also adapts the hypothesis selection strategy e.g. RALCP (Wang et al., 2023b) to prune the unstable suffixes in each response. Algorithm 1 in Appendix A presents the detailed decoding process.

We conducted a pilot experiment to assess LLMs' zero-shot and few-shot capabilities with conversational prompts. Using `Llama2-7b-chat` (Touvron et al., 2023) on the WMT15 De->En test set with chunk size $n = 5$, we tested both zero- and five-shot settings. As shown in Table 1, conversational SIMULMT performed poorly even with 5-shot prompting. The failure analysis reveals that LLMs, trained primarily on complete sentences, struggle with partial source translation and tend to hallucinate completions when presented with fragments in a multi-turn dialogue format. To address this limitation, we propose to SFT LLMs on conversational SIMULMT data. The following section details our approach to converting a normal bi-text corpus into conversational prompt format.

3.2 SFT on Conversational SIMULMT Data

As conversational SIMULMT data is not naturally available, we propose to synthesize `READ` / `WRITE` chunks by segmenting sentence pairs from parallel corpora. Inspired by Arthur et al. (2021) which generates the oracle policy from word alignments, we further extend the approach by carefully addressing the impact of word reordering and improving the generalizability of the oracle policy. Specifically, we first build *monotonic dependency graph* from the alignment of a sentence pair. We then segment the graph and convert these segments into `READ` / `WRITE` pairs, followed by augmentation to improve its generalization across various latency demands (Figure 2). The process is explained below.

Alignment Graph Generation Given a sentence pair, we employ `fastalign` (Dyer et al., 2013) to obtain word alignment between source and target tokens (Step 1 in Figure 2). The obtained alignment is a set $\mathcal{A}$ of pairs (i, j) denoting the source token x_i is aligned with its corresponding target token y_j. We define the *sufficient* source token set to generate a given target token y_j as $\mathbf{a}_j = \{i|(i, j) \in \mathcal{A}, \forall i \in [0, I]\}$.

A source and target sentences have a monotonic translation relationship if the previous target tokens only aligned with the previous source tokens, i.e. $\forall j > k \quad \min(\mathbf{a}_j) \geq \max(\mathbf{a}_k)$ (Koehn et al., 2005; Ling et al., 2011). This condition ensures that the relative order of words is preserved between the source and target sentences. In that case, the optimal minimum-latency policy that retains sufficient source information is to produce the monotonic translation that follows the word order of the source sequence, i.e. `WRITE` target token y_j immediately after reading the final required source token $x_{\max(\mathbf{a}_j)}$, and then `READ` the next source tokens.

Monotonic Dependency Graph Monotonic dependency enables effective implementation of optimal `READ` /`WRITE` policies. However, translations often require reordering to produce grammat-

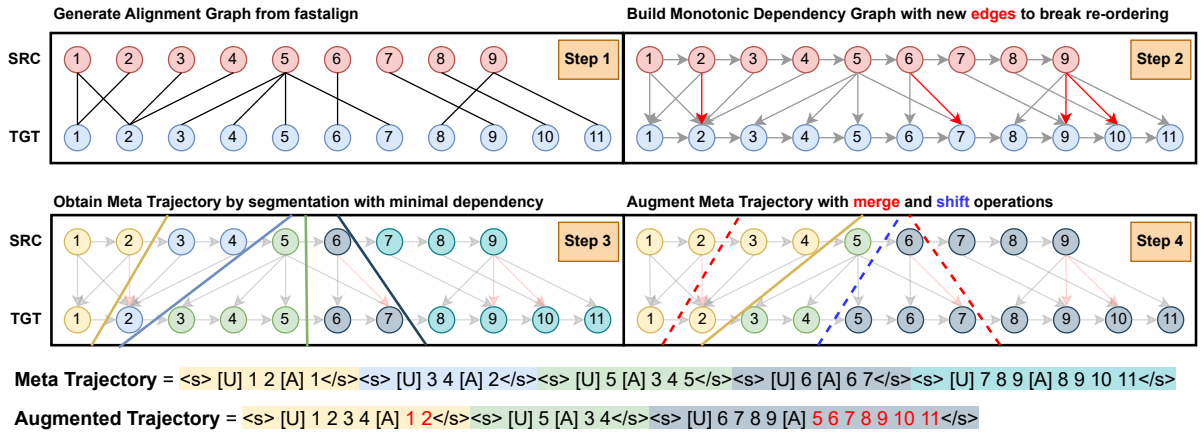


Figure 2: The illustration of the data curating process. The first graph is obtained from `fast_align`, it is then modified into a monotonic dependency graph by adding additional edges. The Meta Trajectory can be derived by segmenting the monotonic dependency graph with minimal dependency (segment with the colored solid line in step 3). Finally, Policy Generalization is applied to augment the segmented graph with merge (red dotted lines will be removed) and shift (blue dotted lines are shifted) operations. Chunks in the trajectories derived from the third and fourth graphs are highlighted with different colors.

ically correct output, especially between languages with different syntactic structures. To address this, we propose constructing a monotonic dependency graph $\overrightarrow{\mathcal{A}}$ from alignment set $\mathcal{A}$ (Step 2 in Figure 2) such that the monotonic condition is met.

For each target token y_j violating the monotonic condition $\min(\mathbf{a}_j) < \max(\mathbf{a}_{j-1})$, we add a new edge from the last sufficient source token $x_{\max(\mathbf{a}_{j-1})}$ to y_j, eliminating the need for reordering. In Figure 2, y_2 violates monotonicity as its earliest required source token $\min(\mathbf{a}_2) = 1$ precedes the last required source token for the previous target $\max(\mathbf{a}_1) = 2$. Thus, we add an edge from x_2 to y_2.

Meta Trajectory We then segment the monotonic dependency graph and convert these segments into `READ` / `WRITE` pairs, representing the *meta trajectory* of the oracle policy with minimum latency (Step 3 in Figure 2). We examine each target token to identify its exclusive corresponding source tokens with minimal dependency. Each subgraph $\overrightarrow{\mathcal{A}_j}$ corresponds to a pair (R_j, W_j) where $W_j = \{y_j\}$ is a target token and $R_j = \{x_i | i \in \mathbf{a}_j \setminus \mathbf{a}_{j-1}\}$ contains new source tokens required since the previous target. When consecutive target tokens depend on the same source token, we combine their `WRITE` actions, assigning the shared source token to $R_j = \{x_i\}$ and forming $W_j = \{y_j, ..., y_{j+n}\}$. This generates a meta trajectory $RW^\star = [(R_1, W_1), ..., (R_C, W_C)], C \leq I$, with C chunks.

Trajectory Augmentation Since the meta trajectories are tailored for minimal latency, they may not generalize well to different lengths of the input chunk, corresponding to different levels of latency. To improve the LLM's adaptability across various latency demands, we augment the meta-trajectory $RW^\star$ with a series of **merge** and **shift** operations (Step 4 in Figure 2). We first traverse $RW^\star$ and randomly merge δ consecutive `READ` and `WRITE` actions, forming new pairs $([R_c, ..., R_{c+\delta}], [W_c, ..., W_{c+\delta}])$, where $[\cdot]$ is the string concatenation operation. Here, δ is a variable re-sampled from a uniform distribution $\mathcal{U}(\delta_{\min}, \delta_{\max})$ where $\delta_{\min}$ and $\delta_{\max}$ are predefined hyperparameters.

Additionally, with a probability of β, we shift a portion of tokens from a `WRITE` action W_c to the next one W_{c+1} in the merged trajectory. More specifically, we split W_c at a proportion ρ and transfer the latter part to the next pair, resulting in $(R_c, W_c^{<\rho}), (R_{c+1}, [W_c^{>\rho}, W_{c+1}])$ where ρ is sampled from $\mathcal{U}(\rho_{\min}, 0.9)$ where $\rho_{\min}$ is a hyperparameter.

This augmentation enhances the LLM's context conditioning and suits incremental decoding where prediction endings are often truncated by hypothesis selection algorithms. The resulting trajectory consists of `READ` /`WRITE` chunks of varying lengths, formatted with conversational prompts for SFT. During training, we apply cross-entropy loss only on target tokens within unshifted `WRITE` chunks.

Trajectory	Dimension	De→En	En→Vi	En→Zh
Meta-Trajectory	#Chunk	10.69 ± 5.5	12.98 ± 8.1	11.94 ± 7.3
	#SRC word/Chunk	1.74 ± 0.8	1.38 ± 0.4	1.68 ± 0.5
	#TGT word/Chunk	1.79 ± 0.8	1.73 ± 0.5	1.53 ± 0.5
Aug-Trajectory	#Chunk	2.74 ± 1.2	3.12 ± 1.6	2.95 ± 1.4
	#SRC word/Chunk	7.01 ± 3.9	5.83 ± 2.8	7.02 ± 3.6
	#TGT word/Chunk	7.18 ± 3.9	7.35 ± 3.5	6.40 ± 3.2

Table 2: Statistics of curated conversational SIMULMT training data across all benchmarks, showing chunk counts and source/target tokens per chunk (mean±std) for both meta and augmented trajectories.

4 Experiments

4.1 Datasets

WMT15 De->En (4.5M training pairs) We use `newstest2013` (3000 pairs) for validation and `newstest2015` (2169 pairs) for testing[2].

IWSLT15 En->Vi (133K training pairs) We employ TED `tst2012` (1553 pairs) and `tst2013` (1268 pairs) as validation and test sets, respectively[3].

MUST-C En->Zh (Di Gangi et al., 2019) (359k training pairs) This TED talk dataset provides 1349 pairs for validation and the `tst-COMMON` (2841 pairs) for testing.

Conversational SIMULMT Datasets For each dataset, we create conversational prompt versions from their training sets using the approach described in §3.2. We employ `fastalign` (Dyer et al., 2013) to obtain initial word alignment graphs. For trajectory augmentation, we set $\delta_{\text{min:max}} = (2, 10)$ for merging operations. For shift operations, both β and ρ_{min} are set to 0.5, meaning we shift at least 50% of tokens in a target segment to the next one with 0.5 probability. Table 2 presents detailed statistics for these datasets.

4.2 Evaluation Metrics

We evaluate translation quality and latency using SacreBLEU[4] (Post, 2018), COMET[5] (Rei et al., 2020), and word-level average lagging (AL) (Ma et al., 2019). To assess computational efficiency, we measure word wall time (WWT) (Wang et al., 2023b), which represents the average time required to predict a word on identical hardware.

[2] `www.statmt.org/wmt15/`

[3] `nlp.stanford.edu/projects/nmt/`

[4] BLEU+nrefs:1+case:mixed+eff:no+tok:{13a,zh} +smooth:exp+version:2.3.1

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

4.3 Model Training

For all LLM-based methods, we use `Llama2-7b-chat` (Touvron et al., 2023) as the backbone following Wang et al. (2023b). We conduct QLoRA-based SFT (Hu et al., 2022; Dettmers et al., 2023) for one epoch with $r = 64$, $\alpha = 16$, learning rate of 2e-4, batch size of 48, and 4-bit quantization on a single A100 GPU. Both offline and conversational prompt models are fine-tuned on identical data sources (standard offline style bitext from the aforementioned training sets), but formatted as offline prompts and conversational prompts respectively.

4.4 Settings

We compare our proposed conversational SIMULMT against the following baselines:

Encoder-Decoder Transformers We evaluate the performance of a series of specialized Encoder-Decoder Transformer models for both OFFLINEMT and SIMULMT:

- **Offline NMT**: Following (Zhang and Feng, 2022), we train vanilla Transformer (Vaswani et al., 2017) (48M parameters for En->Vi; 300M for De->En and Zh->En) with beam size 5 for inference.
- **Wait-k** (Ma et al., 2019): A fixed policy approach that reads k source tokens before alternating read/write operations. We test with k ranging from 1-8 for De->En and Zh->En, 4-8 for En->Vi.
- **ITST** (Zhang and Feng, 2022) An adaptive policy that measures the information transferred from source to target token and determines when to proceed with translation with a threshold (set as 0.1-0.7 for all datasets).
- **Wait-Info** (Zhang et al., 2022) An adaptive policy using token information thresholds ($\mathcal{K}$ from 1-8 for all datasets) to coordinate the timing of translation.

LLM-based SIMULMT We compare our conversational prompt approach against the offline prompt method (Wang et al., 2023b), using identical `READ` policies with chunk sizes n=[3,5,7,9,11,13]. Both approaches are evaluated with RALCP hypothesis selection (beam=5). We also assess greedy decoding (beam=1, no hypothesis selection) with our conversational prompting

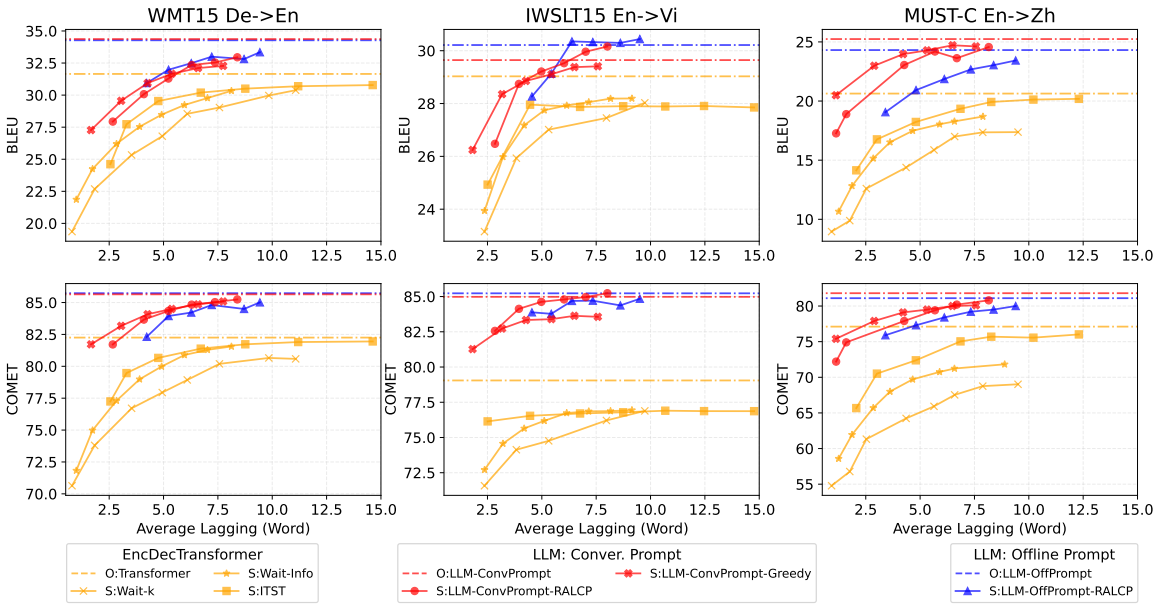


Figure 3: Translation quality and latency results on three benchmarks. Results are presented in three groups with different colors: (i) Encoder-Decoder Transformer baselines (orange), (ii) Offline-Prompt LLMs (blue), and (iii) Conversation-Prompt LLMs (red). Offline and Simultaneous decoding are distinguished by the first letter (O/S).

only (as computational latency baseline), since offline prompting inherently requires hypothesis selection and cannot function with greedy search. For reference, we include results from LLM-based OFFLINEMT as a performance upper bound.

4.5 Results

Our preliminary study in Table 1 showed LLMs struggle with zero/few-shot conversational SIMULMT. Here we examine whether fine-tuning on our curated data enables effective conversational SIMULMT, focusing on quality-latency balance.

Translation Quality As shown in Figure 3, LLM-based approaches (red and blue) outperform Transformer baselines (yellow) across all language pairs by up to 3 BLEU/10 COMET points. With sufficient latency allowance, LLM-based SIMULMT even surpasses offline Transformer NMT. At equivalent latency levels, our conversational prompting (red) achieves comparable BLEU scores to offline prompting (blue) while often showing better COMET scores.

Translation Latency Our conversational SIMULMT (red) reduces latency compared to offline prompting (blue), with average reductions of 1.17 and 1.50 AL across all benchmarks. For En->Vi and En->Zh, our approach achieves latency comparable to specialized SIMULMT models. While RALCP (`S:LLM-ConvPrompt-RALCP`) generally provides better quality than greedy decoding (`S:LLM-ConvPrompt-Greedy`), the latter offers lower latency.

Practical Advantages Most significantly, our conversational SIMULMT (red) maintains superior translation quality at low latency levels (AL<4) compared to specialized models (yellow), making it particularly valuable for practical applications requiring both high quality and low latency. In contrast, offline prompting (blue) with identical decoding configurations struggles to operate effectively in the low-latency range, diminishing its quality advantages relative to specialized approaches (yellow). These results demonstrate that our conversational prompting approach effectively addresses the efficiency-quality trade-off in simultaneous translation with LLMs.

5 Analysis

5.1 Decoding Speed

While Average Lagging (AL) effectively quantifies algorithmic delay between translation and source input, it doesn't account for computational costs. In real-world applications, actual inference time critically impacts user experience: a model with low AL might still deliver poor user experience due to high computational overhead. To address this limitation, we evaluate decoding speed using Word Wall Time (WWT), which measures actual

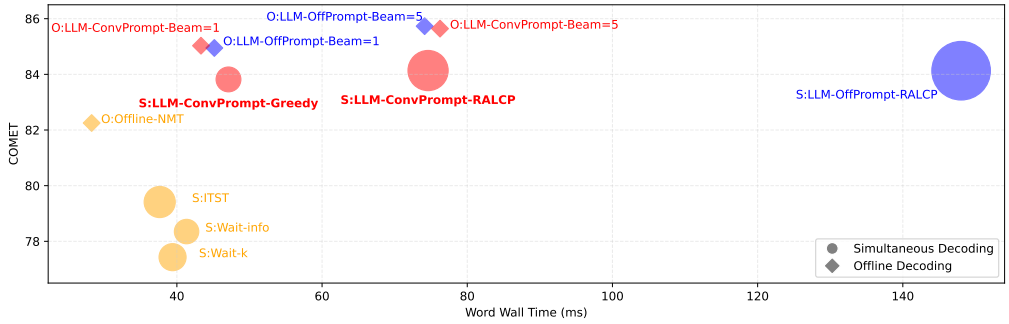


Figure 4: Relationship between computational efficiency (Word Wall Time) and translation quality (COMET score) on WMT15 De->En. Simultaneous decoding settings are shown as circles, with circle size representing variance across different latency control parameters (e.g. n). Offline settings are represented by diamonds. Color coding matches Figure 3, with our proposed approach highlighted in **bold**.

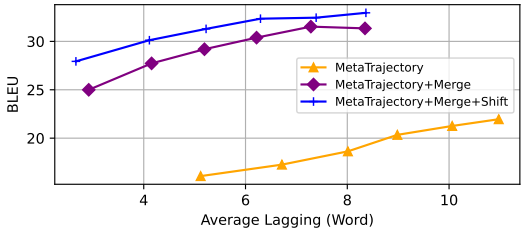


Figure 5: Effect of trajectory augmentation strategies on translation quality (BLEU) and latency (AL) for WMT15 De->En. Results compare models trained on meta-trajectories alone versus with merge and shift operations.

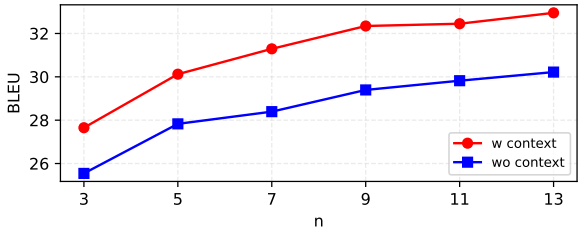


Figure 6: Translation quality (BLEU) on WMT15 De->En when generating the final chunk with vs. without preceding context, across different chunk sizes. The consistent gap demonstrates effective context utilization.

inference time per word (§4.4).

Figure 4 presents detailed WWT results for WMT15 De->En translation. Our analysis reveals that offline prompting with RALCP (`S:LLM-OffPrompt-RALCP`) exhibits the slowest performance, making it impractical despite good translation quality. In contrast, our conversational prompting approach with RALCP (`S:LLM-ConvPrompt-RALCP`) achieves computational efficiency comparable to offline LLM translation (`O:LLM-ConvPrompt-Beam=5`) while maintaining high translation quality.

Most notably, our conversational prompting with greedy decoding (`S:LLM-ConvPrompt-Greedy`) delivers the best efficiency-quality balance—achieving processing speeds comparable to specialized SIMULMT models (yellow) while producing significantly better translations. These results demonstrate that our approach effectively addresses both algorithmic and computational latency concerns, making it suitable for practical deployment.

5.2 Effectiveness of Trajectory Augmentation

To evaluate our trajectory augmentation strategy, we conducted an ablation study comparing models trained on: (*i*) meta trajectories only, (*ii*) meta trajectories with merge operations, and (*iii*) meta trajectories with both merge and shift operations (§3.2). All models used identical hyperparameters, with training data as the only variable.

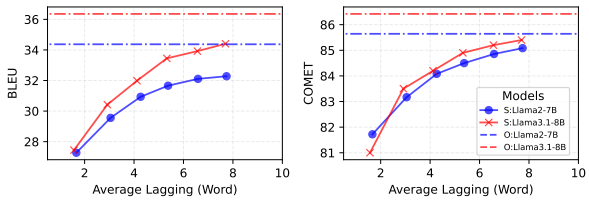
As shown in Figure 5, trajectory augmentation yields notable improvements in translation quality and latency when using RALCP. The merge operation contributes most significantly to these improvements, while models trained solely on meta trajectories perform poorly across all metrics.

This suggests augmentation techniques enhance the model's ability to generalize across different latency conditions. Without augmentation, the model struggles with varying input chunk sizes, causing RALCP to accept less reliable hypotheses and increasing latency. The augmented approach effectively prepares the model for dynamic simultaneous translation scenarios.

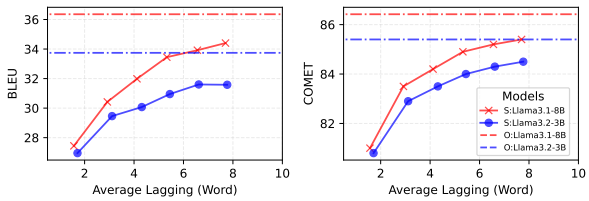
5.3 Ability to Leverage Contextual Information

Effective SIMULMT with conversational prompting requires the model's ability to accurately utilize contextual information. To evaluate this capability, we designed an experiment isolating the model's performance on the final chunk of translation both with and without access to preceding context.

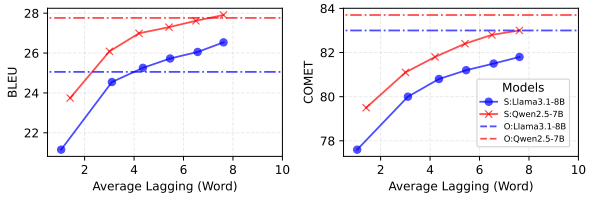
For each test instance, we extracted the com-



(a) Impact of model iteration (Llama-2-7b-chat vs. Llama-3.1-8B-Instruct) on WMT15 De->En.



(b) Effect of model scale (Llama-3.1-8B-Instruct vs. Llama-3.2-3B-Instruct) on WMT15 De->En.



(c) Impact of target language proficiency (Llama-3.1-8B-Instruct vs. Qwen2.5-7B-Instruct) on MUST-C En->Zh.

Figure 7: Performance comparison of different LLM families with our conversational prompt.

plete inference history and separated it into: (i) the source-target dialogue history serving as context, and (ii) the final source chunk representing the latest input. We then tasked our fine-tuned LLM with translating this final chunk under two conditions: with and without access to the preceding conversation history. Performance was evaluated by computing BLEU scores on the concatenation of the generated final chunk with its original history.

As shown in Figure 6, we observed a consistent 2-point decrease in BLEU scores when context was withheld. This performance gap demonstrates our model effectively leverages information from previous conversation turns to produce more accurate translations, confirming the fine-tuned LLM maintains translation coherence.

5.4 Generalizability Across LLM Families

In our main experiments, we used `Llama-2-7b-chat` following Wang et al. (2023b) for consistency. Now, we examine our approach's generalizability across different LLMs, using identical training and inference parameters for fair comparison. We report only greedy simultaneous decoding and offline beam=5 results to eliminate interference with hypothesis selection.

Impact of Model Iteration We compare `Llama-2-7b-chat` with the newer `Llama-3.1-8B-Instruct` (Grattafiori et al., 2024) on WMT15 De->En to assess how model advancements affect performance. As shown in Figure 7a, the newer model demonstrates consistent improvements in both offline and simultaneous modes. This confirms that conversational SIMULMT effectively transfers to newer LLMs, with benefits from improved instruction-following capabilities and enhanced language modeling.

Effect of Model Scale We investigate how model size impacts performance by comparing `Llama-3.1-8B-Instruct` with the smaller `Llama-3.2-3B-Instruct` (Grattafiori et al., 2024) on WMT15 De->En. Figure 7b shows that while the larger model predictably outperforms its smaller counterpart, the 3B model still achieves acceptable translation quality (on par with `Llama-2-7b-chat` in Figure 7a), suggesting our method is viable on resource-constrained devices.

Impact of Target Language Proficiency We evaluate `Llama-3.1-8B-Instruct` against `Qwen2.5-7B-Instruct` (Qwen et al., 2025) on MUST-C En->Zh to investigate the effect of the model's target language capabilities. As shown in Figure 7c, Qwen2.5 consistently outperforms Llama-3.1 for Chinese translation by 1-2 BLEU points across all latency settings, demonstrating that target language proficiency provides additional benefits with our approach.

6 Related Works

Simultaneous Machine Translation (SIMULMT) is the task to provide real-time translation of a source sentence stream where the goal is to minimize the latency while maximizing the translation quality. A common approach is to train an MT model on prefix-to-prefix dataset to directly predict target tokens based on partial source tokens (Ma et al., 2019). Alternatively, Liu et al. (2020) proposed the incremental decoding framework to leverage the pretrained OFFLINENMT model and turn it into a SIMULMT model without further training. A core component of SIMULMT is a read-write policy to decide at every step whether to wait for another source token (`READ`) or to generate a target token (`WRITE`). Previous methods have explored

fixed policy, which always waits for k tokens before generation (Ma et al., 2019; Zhang et al., 2022) and adaptive policy, which trains an agent via reinforcement learning (Gu et al., 2017; Arthur et al., 2021). Re-translation (Arivazhagan et al., 2019) from the beginning of the source sentence at the `WRITE` step will incur high translation latency. Stable hypothesis detection methods such as Local Agreement, hold-n (Liu et al., 2020) and Share prefix SP-n (Nguyen et al., 2021) are employed to commit stable hypothesis and only regenerate a subsequence of source sentence. The goal is to reduce the latency and minimize the potential for errors resulting from incomplete source sentence (Polák et al., 2022; Wang et al., 2021).

LLM-based NMT Recent research has delved into the potential usage of LLMs in MT (Hendy et al., 2023; Zhu et al., 2023; Robinson et al., 2023), especially in handling discourse phenomena (Wang et al., 2023a; Wu et al., 2024) and linguistic nuances such as idioms (Manakhimova et al., 2023) and proverbs (Wang et al., 2025). While LLMs do exhibit some level of translation capability, prior research has identified that they still lags behind the conventional NMT models, especially for low resource languages (Robinson et al., 2023). Additionally, the translation performance varies depending on prompting strategies (Zhang et al., 2023). Efforts have been made to enhance the LLMs' MT performance by incorporating guidance from dictionary (Lu et al., 2023), further fine-tuning (Zeng et al., 2023; Xu et al., 2023) and augmenting with translation memories (Mu et al., 2023).

LLM-based SIMULMT SimulLLM (Agostinelli et al., 2023) explore the ability to adapt an LLM finetuned on NMT task to simultaneous translation with wait-k strategy. Wang et al. (2023b) adopt hybrid READ/WRITE policy with wait-k and incremental decoding. TransLLaMA (Koshkin et al., 2024) teach LLMs to produce WAIT tokens to preserve the causal alignment between source and target tokens. At each inference round, LLMs only produce a single word or WAIT token, which is very costly due to multiple rounds of LLM calls. Guo et al. (2024) introduce LLM into the SIMULMT task as a translation agent working with a specialized SIMULMT policy agent. An additional memory module stores translation history. The policy agent decides on `READ`/`WRITE` actions, while the LLM translates target segments. They face the same KV-cache reuse challenge noted by Wang et al. (2023b), making the computational cost of collaborating big and small models even more significant.

7 Conclusion

This paper focuses on the feasibility of utilizing LLM for SIMULMT. We found that leveraging the incremental-decoding framework with offline prompting leads to high computational latency, hindering the reuse of the Key-Value cache. To address this, we propose the conversational prompting which allows LLMs to conduct SIMULMT in a multi-turn dialogue manner. The approach significantly speeds up the inference and also preserves the quality superiority, enabling practical LLM-based SIMULMT systems.

Limitations

We summarize the limitations of this study in the following aspects:

Data Our evaluation was conducted on three commonly used benchmarks which may limit the diversity in domains, styles, and languages. There may also be potential data contamination concerns since LLMs might have been exposed to parts of our test sets during pre-training. A more comprehensive evaluation with diverse datasets across more domains and language pairs would strengthen our findings.

Alignment-based Data Curation Our approach relies on word alignment tools like `fast_align` to segment parallel sentences, which has inherent limitations. These tools may struggle with languages having drastically different word orders or grammatical structures, potentially creating suboptimal segmentation points. Furthermore, the alignment quality degrades for distant language pairs or complex sentences with idiomatic expressions and cultural references. While our augmentation strategies help mitigate some issues, they are still constrained by the initial alignment quality.

Ethics Statement

Our work is built on top of an existing LLM. For this reason, we share the similar potential risks and concerns posed by the underlying LLM. Our method is trained on commonly used training resources of the Machine Translation research community and as such we are not expecting our approach to introduce new areas of risks.

References

Victor Agostinelli, Max Wild, Matthew Raffel, Kazi Asif Fuad, and Lizhong Chen. 2023. Simul-llm: A framework for exploring high-quality simultaneous translation with large language models. *arXiv preprint arXiv:2312.04691*.

Naveen Arivazhagan, Colin Cherry, Wolfgang Macherey, Chung-Cheng Chiu, Semih Yavuz, Ruoming Pang, Wei Li, and Colin Raffel. 2019. Monotonic infinite lookback attention for simultaneous machine translation. In *ACL*, pages 1313–1323.

Philip Arthur, Trevor Cohn, and Gholamreza Haffari. 2021. Learning coupled policies for simultaneous machine translation using imitation learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2709–2719, Online. Association for Computational Linguistics.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *CoRR*, abs/2305.14314.

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.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 644–648. The Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, and Ahmad Al-Dahle et al. 2024. The llama 3 herd of models.

Jiatao Gu, Graham Neubig, Kyunghyun Cho, and Victor O.K. Li. 2017. Learning to translate in real-time with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1053–1062, Valencia, Spain. Association for Computational Linguistics.

Jiaxin Guo, Daimeng Wei, Zhanglin Wu, Zongyao Li, Zhiqiang Rao, Minghan Wang, Hengchao Shang, Xiaoyu Chen, Zhengzhe Yu, Shaojun Li, Yuhao Xie, Lizhi Lei, and Hao Yang. 2023. The hw-tsc's simultaneous speech-to-text translation system for IWSLT 2023 evaluation. In *IWSLT@ACL*, pages 376–382.

Shoutao Guo, Shaolei Zhang, Zhengrui Ma, Min Zhang, and Yang Feng. 2024. Sillm: Large language models for simultaneous machine translation.

Amr Hendy, Mohamed Abdelrehim, Amr Sharaf, Vikas Raunak, Mohamed Gabr, Hitokazu Matsushita, Young Jin Kim, Mohamed Afify, and Hany Hassan Awadalla. 2023. How good are GPT models at machine translation? A comprehensive evaluation. *CoRR*, abs/2302.09210.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *ICLR*.

Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *Proceedings of the Second International Workshop on Spoken Language Translation*, Pittsburgh, Pennsylvania, USA.

Roman Koshkin, Katsuhito Sudoh, and Satoshi Nakamura. 2024. Transllama: Llm-based simultaneous translation system. *arXiv preprint arXiv:2402.04636*.

Wang Ling, João Graça, David Martins de Matos, Isabel Trancoso, and Alan W Black. 2011. Discriminative phrase-based lexicalized reordering models using weighted reordering graphs. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 47–55, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Danni Liu, Gerasimos Spanakis, and Jan Niehues. 2020. Low-latency sequence-to-sequence speech recognition and translation by partial hypothesis selection. In *Interspeech*, pages 3620–3624.

Hongyuan Lu, Haoyang Huang, Dongdong Zhang, Haoran Yang, Wai Lam, and Furu Wei. 2023. Chain-of-dictionary prompting elicits translation in large language models. *CoRR*, abs/2305.06575.

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.

Shushen Manakhimova, Eleftherios Avramidis, Vivien Macketanz, Ekaterina Lapshinova-Koltunski, Sergei Bagdasarov, and Sebastian Möller. 2023. Linguistically motivated evaluation of the 2023 state-of-the-art machine translation: Can ChatGPT outperform NMT? In *Proceedings of the Eighth Conference on Machine Translation*, pages 224–245, Singapore. Association for Computational Linguistics.

Yongyu Mu, Abudurexiti Reheman, Zhiquan Cao, Yuchun Fan, Bei Li, Yinqiao Li, Tong Xiao, Chunliang Zhang, and Jingbo Zhu. 2023. Augmenting

large language model translators via translation memories. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10287–10299, Toronto, Canada. Association for Computational Linguistics.

Thai-Son Nguyen, Sebastian Stüker, and Alex Waibel. 2021. Super-Human Performance in Online Low-Latency Recognition of Conversational Speech. In *Proc. Interspeech 2021*, pages 1762–1766.

Peter Polák, Ngoc-Quan Pham, Tuan Nam Nguyen, Danni Liu, Carlos Mullov, Jan Niehues, Ondřej Bojar, and Alexander Waibel. 2022. CUNI-KIT system for simultaneous speech translation task at IWSLT 2022. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 277–285, Dublin, Ireland (in-person and online). Association for Computational Linguistics.

Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Anselm Levskaya, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. 2023. Efficiently scaling transformer inference. In *Proceedings of Machine Learning and Systems*.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers, WMT 2018, Belgium, Brussels, October 31 - November 1, 2018*, pages 186–191. Association for Computational Linguistics.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 technical report.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. Comet: A neural framework for mt evaluation.

Nathaniel R. Robinson, Perez Ogayo, David R. Mortensen, and Graham Neubig. 2023. Chatgpt MT: competitive for high- (but not low-) resource languages. *CoRR*, abs/2309.07423.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*, pages 5998–6008.

Longyue Wang, Chenyang Lyu, Tianbo Ji, Zhirui Zhang, Dian Yu, Shuming Shi, and Zhaopeng Tu. 2023a. Document-level machine translation with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16646–16661, Singapore. Association for Computational Linguistics.

Minghan Wang, Jiaxin Guo, Yuxia Wang, Daimeng Wei, Hengchao Shang, Chang Su, Yimeng Chen, Yinglu Li, Min Zhang, Shimin Tao, and Hao Yang. 2021. Diformer: Directional transformer for neural machine translation.

Minghan Wang, Viet-Thanh Pham, Farhad Moghimifar, and Thuy-Trang Vu. 2025. Proverbs run in pairs: Evaluating proverb translation capability of large language model.

Minghan Wang, Jinming Zhao, Thuy-Trang Vu, Fatemeh Shiri, Ehsan Shareghi, and Gholamreza Haffari. 2023b. Simultaneous machine translation with large language models. *arXiv preprint arXiv:2309.06706*.

Minghao Wu, Thuy-Trang Vu, Lizhen Qu, George Foster, and Gholamreza Haffari. 2024. Adapting large language models for document-level machine translation. *arXiv preprint arXiv:2401.06468*.

Haoran Xu, Young Jin Kim, Amr Sharaf, and Hany Hassan Awadalla. 2023. A paradigm shift in machine translation: Boosting translation performance of large language models. *CoRR*, abs/2309.11674.

Zheng-Xin Yong, Ruochen Zhang, Jessica Zosa Forde, Skyler Wang, Arjun Subramonian, Holy Lovenia, Samuel Cahyawijaya, Genta Indra Winata, Lintang Sutawika, Jan Christian Blaise Cruz, Yin Lin Tan, Long Phan, Rowena Garcia, Thamar Solorio, and Alham Fikri Aji. 2023. Prompting multilingual large language models to generate code-mixed texts: The case of south east asian languages.

Zhihang Yuan, Yuzhang Shang, Yang Zhou, Zhen Dong, Chenhao Xue, Bingzhe Wu, Zhikai Li, Qingyi Gu,

Yong Jae Lee, Yan Yan, et al. 2024. Llm inference unveiled: Survey and roofline model insights. *arXiv preprint arXiv:2402.16363*.

Jiali Zeng, Fandong Meng, Yongjing Yin, and Jie Zhou. 2023. TIM: teaching large language models to translate with comparison. *CoRR*, abs/2307.04408.

Biao Zhang, Barry Haddow, and Alexandra Birch. 2023. Prompting large language model for machine translation: A case study. *ArXiv*, abs/2301.07069.

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, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 992–1013. Association for Computational Linguistics.

Shaolei Zhang, Shoutao Guo, and Yang Feng. 2022. Wait-info policy: Balancing source and target at information level for simultaneous machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2249–2263, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Lingpeng Kong, Jiajun Chen, Lei Li, and Shujian Huang. 2023. Multilingual machine translation with large language models: Empirical results and analysis. *CoRR*, abs/2304.04675.

Algorithm 1 Conversational SIMULMT Decoding

Require: LLM : $\texttt{LLM}_\theta$,
Source chunks: $\mathbf{x} = []$,
Target chunks: $\mathbf{y} = []$,
KV-Cache: $\mathbf{h} = []$,
Chunk index: $c = 0$,
Variables Definition: Source chunk size: n,
Beam-size: B, Agreement-degree: γ

while NOT_FINISH **do**
  $\mathbf{x}_c \leftarrow \texttt{READ}(n)$ //READ n tokens
  $\mathbf{x}$.append($\mathbf{x}_c$)
  $\mathbf{x}_{\text{prompt}} \leftarrow \texttt{PROMPT}(\mathbf{x}, \mathbf{y})$
  $\mathbf{y}'_c, \mathbf{h}' \leftarrow \texttt{LLM}(\mathbf{x}_{\text{prompt}}, B, \mathbf{h}, \text{latest=True})$
  //B candidates with latest tokens in $\mathbf{y}'_c$
  $\mathbf{y}_c, \mathbf{h} \leftarrow \texttt{PREFIX}(\mathbf{y}'_c, \mathbf{h}')$
  //Prune with Prefix selection, e.g. RALCP
  if $\mathbf{y}_c == \emptyset$ **then**
    continue
  else
    $\mathbf{y}$.append($\mathbf{y}_c$)
    $\texttt{WRITE}(\mathbf{y}_c)$
    $c \leftarrow c + 1$
  end if
end while

Appendix

A Conversational SimulMT Decoding

Algorithm 1 presents the details of applying conversational prompts for decoding.