FASST: Fast LLM-based Simultaneous Speech Translation

Anonymous ACL submission

Abstract

Simultaneous speech translation (SST) takes streaming speech input and generates text translation on the fly. Existing methods either have high latency due to recomputation of input representations, or fall behind of offline ST in translation quality. In this paper, we propose FASST, a fast large language model based method for streaming speech translation. We propose blockwise-causal speech encoding and consistency mask, so that streaming speech input can be encoded incrementally without recomputation. Furthermore, we develop a two-stage training strategy to optimize FASST for simultaneous inference. We evaluate FASST and multiple strong prior models on MuST-C dataset. Experiment results show that FASST achieves the best quality-latency trade-off. It outperforms the previous best model by an average of 1.5 BLEU under the same latency for English to Spanish translation.

1 Introduction

End-to-end simultaneous speech translation (SST) translates incomplete speech input into text in a different language (Ma et al., 2020b), which is widely used in multilingual conferences, live streaming and etc. Compared to offline ST where speech input is complete, SST needs to decide whether to continue waiting or to generate more translation after receiving new speech input. A common approach in building performant SST streaming models involves pretraining for offline translation and optional finetuning for simultaneous translation (Agarwal et al., 2023; Communication et al., 2023). The quality-latency trade-off of simultaneous streaming models thus heavily depends on its offline performance.

Large language model (LLM) have recently demonstrated its potential to be a strong backbone of offline E2E ST (Huang et al., 2023; Zhang et al., 2023b). However, LLM introduces larger computation overhead compared to regular-sized models when applied to SST. Figure 1 shows that the computation latency of a LLM-based 7B model makes it inferior for real-time application.

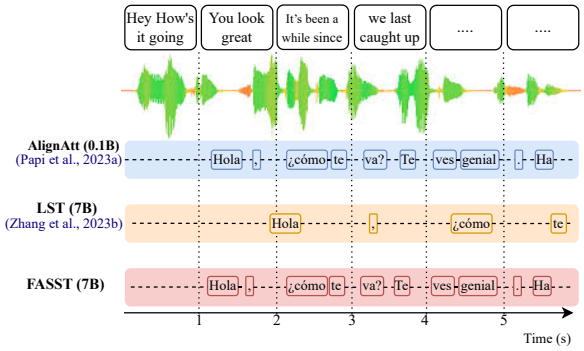

Figure 1: Simultaneous speech translation with AlignAtt-0.1B, LST-7B and our FASST-7B. The LST-7B model generates translation with significantly higher latency than AlignAtt, while our FASST-7B achieves comparable latency with it.

The computation overhead of SST models comes from both encoding new speech input and decoding new translation. While the latter one has been heavily optimized for LLM (Pope et al., 2022; Kwon et al., 2023; Dao, 2024), the former one has not been optimized for SST. As new speech input arrives, most SST models re-encode the entire speech and start autoregressive decoding afterwards, ignoring the incremental nature of streaming speech input. More importantly, the LLM decoder needs to recompute hidden states due to the updated speech features, significantly slowing down the computation.

In this work, we propose a **FA**st LLM-based **SST** (FASST) method to avoid recomputation while maintaining its translation quality. We develop a blockwise-causal speech encoding technique that incrementally encodes new speech input and introduce incremental LLM decoding with consistency mask. We also design an 2-stage training strategy for FASST: 1) aligning speech encoder outputs

with LLM embeddings using word-aligned contrastive loss (Ouyang et al., 2023) and 2) finetuning for SST using wait-k-stride-n policy (Zeng et al., 2021). Experiments on MuST-C dataset (Di Gangi et al., 2019) shows that our 7B model maintains competitive computation aware latency compared to 115M baselines while achieving consistent quality improvement of at least 1.5 BLEU score on English-Spanish direction.

Our contributions are:

- We propose FASST, one of the first efficient LLM-based methods for simultaneous speech translation.
- We verify FASST on MuST-C dataset and it outperforms strong prior methods by 1.5 BLEU at the same latency on English-Spanish direction.
- We further demonstrate that FASST can be generalized to other policies like hold-n and policies spending more time on encoding benefit more from FASST.

2 Related Works

End-to-End SST translates partial speech input into text in another language without generating intermediate transcription. A variety of speech segmentation techniques and policies have been proposed to optimize the quality-latency trade-off. Ren et al. (2020); Dong et al. (2022); Zeng et al. (2023); Zhang et al. (2023c) learn to segment streaming speech input by word boundaries. Zhang and Feng (2023) further learns to segment speech at moments that are beneficial to the translation. On the policy side, Ma et al. (2020b) adapts wait-k and monotonic multihead attention (MMA) from simultaneous text translation to SST model. Ma et al. (2023) further improves the numerical stability of MMA. Papi et al. (2023b) constructs source-target alignment with attention information to guide the simultaneous inference. Zhang and Feng (2022) decides whether to translate based on accumulated information of source speech. Polák et al. (2023) conducts blockwise beam search when doing incremental decoding. The translation quality of SST models depend on not only their policies, but also their offline performance (Agarwal et al., 2023). Recently LLM has been shown as a strong backbone of offline ST (Zhang et al., 2023b; Huang et al., 2023), but its computation overhead prevents it from being used in SST scenarios. FASST is one of the first LLM-based SST models with a reasonable quality-latency trade-off.

Efficient ST To reduce the computation cost of ST models, Wu et al. (2020); Ma et al. (2020c); Raffel and Chen (2023); Raffel et al. (2023) use segments and explicit or implicit memory banks to calculate self-attention only within the segment. Zhang and Feng (2023); Chen et al. (2021); Wu et al. (2021) adopt unidirectional attention during speech encoding. These methods focus on encoder-side optimization and can be integrated with FASST.

Translation with LLM While LLMs are capable of zero-shot machine translation (Brown et al., 2020; OpenAI, 2023; Touvron et al., 2023a,b), their performance can be further improved via in-context learning (Vilar et al., 2023; Zhang et al., 2023a), supervised and semi-supervised finetuning (Rothe et al., 2020; Yang et al., 2023; Zhang et al., 2023d; Xu et al., 2023). For simultaneous machine translation (SMT), Guo et al. (2024) propose a collaborative translation model with two LLM agents and Koshkin et al. (2024) design a finetuning strategy by adding a special "wait" token. Raffel et al. (2024) propose SimulMask to mask token connections under certain policy. SimulMask is a concurrent work with us and only works on text translation.

3 The FASST Method

In this section, we first review the problem formulation of simultaneous speech translation (SST) and then describe the architecture of our proposed model, FASST, followed by its training and inference strategies.

3.1 Problem Formulation

Simultaneous speech translation (SST) needs to generate translations while receiving streaming speech input. Let $S = (s_1, s_2, \cdots, s_{|S|})$ be a speech waveform where s_i are real numbers. The streaming speech input is cut into segments $S_1, S_2, \cdots$ and the SST model P_θ needs to emit partial translations $T_1, T_2, \cdots$ after receiving each of them,

$$T_i \sim P_\theta(\cdot \mid S_{\leq i}, T_{<i}). \quad (1)$$

T_i can be an empty string, indicating that the SST model needs more speech input to continue the translation. After receiving all inputs $S_1, S_2, \cdots, S_m$ and emitting all translations $T_1, T_2, \cdots, T_m$, we obtain the final translation $T = \bigoplus_{i=1}^{m} T_i$ by concatenating all partial ones.

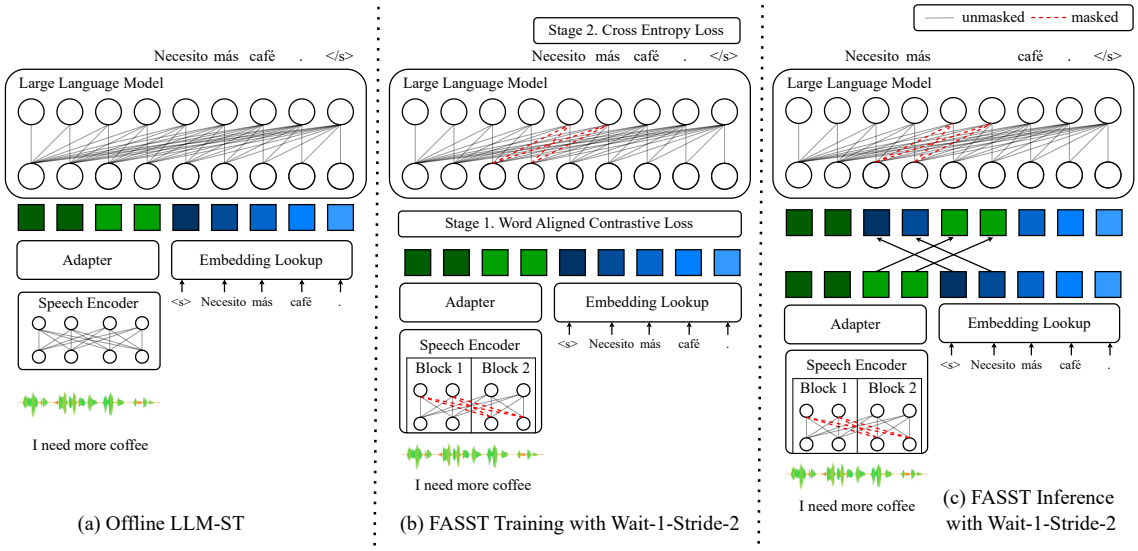

Figure 2: Overview of FASST. (a) shows the offline translation of LLM-based ST model. (b) depicts the 2-stage training pipeline of FASST. Stage 1 aligns adapter output with LLM embedding and stage 2 finetunes for simultaneous translation using wait-k-stride-n policy. (c) illustrates the simultaneous inference procedure of FASST with incremental speech encoding and LLM decoding with consistency mask.

The objective of SST is to emit high-quality translation at a low latency. Quality is evaluated by comparing generated T with the ground-truth T^*, while latency is evaluated based on the amount of lagging of each generated word. In this paper, we consider the computation-aware latency of SST models.

3.2 Model Architecture

As shown in Figure 2, our model is composed of a speech encoder, an adapter and a LLM decoder.

Blockwise-Causal Speech Encoder (BCSE) extracts contextualized acoustic features from the raw waveform incrementally. It consists of several casual convolutional layers as the audio feature extractor and a blockwise-causal Transformer Encoder as the contextual encoder.

Our causal convolutional layers are built upon non-causal ones. Denote $H_{in} \in \mathbb{R}^{l \times d}$ as the input vectors to non-causal convolution $\text{Conv}(\cdot)$ with kernel size w. We add additional zero padding $\text{Pad} \in \mathbb{R}^{(w/2-1) \times d}$ to its left so that each output vector only depends on input vectors to its left, and remove the last $w/2-1$ states to keep its output length the same as before,

$$H_{out} = \text{Conv}\left(\text{Pad} \oplus H_{in}\right)_{:-w/2+1}. \quad (2)$$

Besides, we apply blockwise-causal masking to Transformer Encoder. Define attention mask M of speech encoder as follows

$$M_{j_Q, j_K} = \begin{cases} 0 & \left\lfloor \frac{j_Q}{b} \right\rfloor \geq \left\lfloor \frac{j_K}{b} \right\rfloor \\ -\infty & \text{otherwise} \end{cases} \quad (3)$$

where b is the block size, i.e., the number of hidden states of the speech encoder corresponding to one segment, and j_Q, j_K are row indices of query matrix Q and key matrix K. The attention output of speech encoder during training can then be written as

$$O = \text{Softmax}\left(\frac{QK^T}{\sqrt{d}} + M\right)V, \quad (4)$$

where V is the value matrix.

Adapter receives speech encoder outputs and converts them to the LLM embedding space. It consists of two causal convolutional layers to reduce the length of speech encoder outputs by four and one linear layer to project features into the LLM embedding space. We call the adapter outputs as speech embeddings,

$$E^s_{\leq i} = \text{Adapter}(\text{BCSE}(S_{\leq i})). \quad (5)$$

LLM receives speech embeddings and embeddings of previously generated tokens to decode autoregressively according to a wait-k-stride-n policy π.

$$T_i \sim \text{LLM}(\cdot \mid E^s_{\leq i}, T_{<i}, \pi). \quad (6)$$

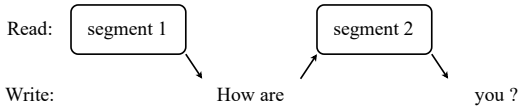

Figure 3: Example of wait-1-stride-2. It waits for 1 segment at the beginning and then alternate between generate 2 words (including punctuations) and reading new segment.

Wait-k-stride-n policy waits for k speech segments at the beginning and then alternate between generating n words and reading new segment. Figure 3 shows an example of wait-1-stride-2.

3.3 Training

As shown in Figure 2 (b), we employ a 2-stage approach to train our model.

Stage 1. Speech-text alignment. We align the speech embedding with LLM input embedding using word-aligned contrastive (WACO) loss. Both transcription embeddings E^t and speech embeddings E^s are grouped into word embeddings W^t and W^s by word boundaries. Word boundaries of speech are obtained through Montreal Forced Aligner [1]. We treat speech and transcription embeddings of the same word as positive pair and others as negative pairs and train the speech encoder and the adapter with contrastive loss,

$$\mathcal{L}_{\text{CTR}} = -\mathbb{E}_i \left[\log \frac{\exp(sim(W_i^s, W_i^t)/\tau)}{\sum_j \exp(sim(W_i^s, W_j^t)/\tau)} \right] \tag{7}$$

where τ is the temperature and $sim()$ is the cosine similarity function. LLM parameters are frozen during stage 1.

Stage 2. Finetuning for simultaneous translation. We finetune the entire model for simultaneous speech translation using wait-k-stride-n policy. Speech input is encoded into speech embeddings E^s. Then we concatenate E^s with embeddings of reference translation and feed them to LLM. Position indices of both speech embeddings and translation embeddings start with the same index and ascend separately, so that text generation during inference does not affect the positional embeddings of speech embeddings.

Then we randomly select $k \in K$ and mask out attentions from translation words with indices from in to $(i+1)n-1$ to speech segments $S_{>i+k}$ for each i, since these words are generated before speech segments $S_{>i+k}$ arrive during inference. Finally, we apply the cross-entropy loss to train the entire model,

$$\mathcal{L}_{\text{CE}} = -\mathbb{E}_i \left[\text{LLM}(T_i^* \mid E^s, T_{<i}^*, \text{mask}) \right], \tag{8}$$

where T^* is the reference translation.

3.4 Efficient Simultaneous Inference

Figure 2 (c) illustrates how we conduct efficient simultaneous inference. FASST waits for k segments at the beginning and then start generating. Suppose now we have received $S_1, S_2, \cdots, S_i$ where $i \geq k$.

Incremental Speech Encoding The blockwise-causal mask of speech encoder allows us to use KV cache of previous speech segments to avoid recomputation. Let $H^s = \left(h_1^s, \cdots, h_{l_i}^s\right)$ be input vectors of the attention. We group them into blocks $B_j = \left(h_{(j-1)b+1}^s, \cdots, h_{jb}^s\right)$ where $1 \leq j \leq i$ and $i \cdot b = l_i$. The query, key and value matrices can be written as follows

$$Q = H^s M_Q = (B_1 M_Q, \cdots, B_i M_Q) \tag{9}$$
$$K = H^s M_K = (B_1 M_K, \cdots, B_i M_K) \tag{10}$$
$$V = H^s M_V = (B_1 M_V, \cdots, B_i M_V) \tag{11}$$

Here the keys and values of previous segments $(B_1 M_K, \cdots, B_{i-1} M_K)$ and $(B_1 M_V, \cdots, B_{i-1} M_V)$ are stored in the KV cache. Now we only need the KV cache and the query $B_i M_Q$, key $B_i M_K$ and value $B_i M_V$ of the latest segment to compute its attention output,

$$O_i^s = \text{Softmax}\left(\frac{B_i M_Q K^T}{\sqrt{d}}\right) V. \tag{12}$$

This results in same output as running attention with full query, key and value matrices and a blockwise-causal mask. In this way, we reduce the time complexity of attention from $O(l_i d^2 + l_i^2 d)$ to $O(bd^2 + l_i bd)$. Here b is a constant while l_i increases with the longer speech input.

Adapting We store the speech encoder outputs of previous segments and concatenate them with encoder outputs of segment i. Then we pass them to the causal convolutional layers and the linear layer to obtain the speech embeddings $E_{\leq i}^s$.

[1] https://github.com/MontrealCorpusTools/Montreal-Forced-Aligner

LLM Decoding with Consistency Mask We partition speech embeddings $E^s_{\leq i}$ into $E^s_1, \cdots, E^s_i$ by speech segment. Following the wait-k-stride-n, the inputs to LLM are organized in the follow way

$$I = E^s_1 \oplus \cdots \oplus E^s_k \oplus \text{Emb}(T_k) \oplus E^s_{k+1} \oplus \text{Emb}(T_{k+1}) \oplus \cdots \oplus E^s_i, \quad (13)$$

where $T_{1:k-1}$ are empty strings and T_j consists of n words for each $k \leq j < i$. Now we need to reuse KV cache of previous $i-1$ speech segments and partial translations to compute LLM hidden states of i_{th} segment. Since speech embeddings are always ahead of text embeddings during training, we design a consistency mask to ensure speech segments can only attend to speech segments before them.

Let $\delta(z)$ be indicator function that equals to 1 if z_{th} position of input I belongs to text and 0 otherwise. Define consistency mask M^c as follows,

$$M^c_{z_Q, z_K} = \begin{cases} 0 & z_Q \geq z_K \text{ and } \delta(z_Q) \geq \delta(z_K) \\ -\infty & \text{otherwise} \end{cases} \quad (14)$$

Let $Q_i, K_i, V_i \in \mathbb{R}^{t_i \times d}$ be query, key and value matrices of segment i and $K_{<i}, V_{<i}$ be cached key and value matrices. We first concatenate K_i and V_i with cache to obtain $K_{\leq i}$ and $V_{\leq i}$. The attention output of segment i can then be computed as follows

$$O^t_i = \text{Softmax}\left(\frac{Q_i K^T_{\leq i}}{\sqrt{d}} + M^c_{-t_i:,:}\right) V_{\leq i}. \quad (15)$$

After computing hidden states for speech segment S_i, the LLM decodes n words autoregressively following the policy.

4 Experiment

4.1 Dataset

We conduct experiments on two language directions of MuST-C v1.0 dataset (Di Gangi et al., 2019): English→Spanish (En-Es) and English→German (En-De). Each language direction contains around 400 hours of audio recordings. The average duration of utterances is less than 10 seconds. To simulate long speech scenarios, we concatenate adjacent utterances in the same TED talk so that each resulting utterance is around 30 seconds. We call the induced dataset as MuST-C-Long[2] and the original one as MuST-C-Short. The duration distribution of both datasets are shown in Figure 4.

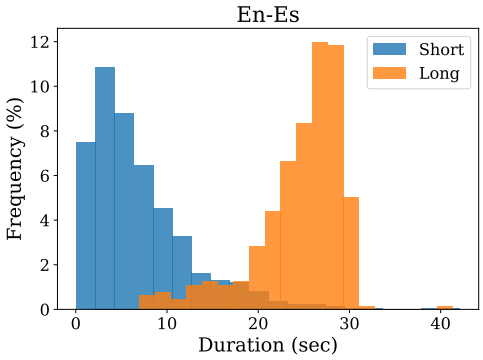

Figure 4: Duration distribution of MuST-C-Short and MuST-C-Long. The average duration of MuST-C-Short is around 5 seconds while that of MuST-C-Long is around 25 seconds.

4.2 Model Configurations

Architecture We intialize our speech encoder with wav2vec 2.0 large model[3] (Baevski et al., 2020) and our LLM with Llama2 7b Base[4] (Touvron et al., 2023a). Wav2vec 2.0 large consists of a 7-layer convolutional feature extractor and a 24-layer Transformer encoder with 1024 hidden units. The block size of speech encoder is set to 50, i.e., around 1 second each block. The adapter connecting wav2vec 2.0 and Llama2 consists of two 1-D convolutional layers with kernel size 3, stride 2 and hidden size 1024 and a linear layer to project hidden size from 1024 to 4096 to match that of LLM embedding. Llama2 7b Base adopts a 32-layer Transformer decoder with hidden size 4096. It uses a vocabulary of size 32000 and rotary positional embedding (Su et al., 2023).

Training We train our model with mixed MuST-C-Short and MuST-C-Long data. The input speech is raw 16-bit 16kHz mono-channel waveform. We filter out speech that is shorter than 320ms during training. The batch size of stage 1 is 16.7 minutes and that of stage 2 is 14 minutes. We use AdamW optimizer with cosine learning rate decay. The warmup steps of stage 1 is 25k and that of stage 2 is 500 steps. The maximum learning rate of stage

[2]The manifest of MuST-C-Long will be released together with the code.

[3]https://dl.fbaipublicfiles.com/fairseq/wav2vec/wav2vec_vox_960h_pl.pt

[4]https://huggingface.co/meta-llama/Llama-2-7b

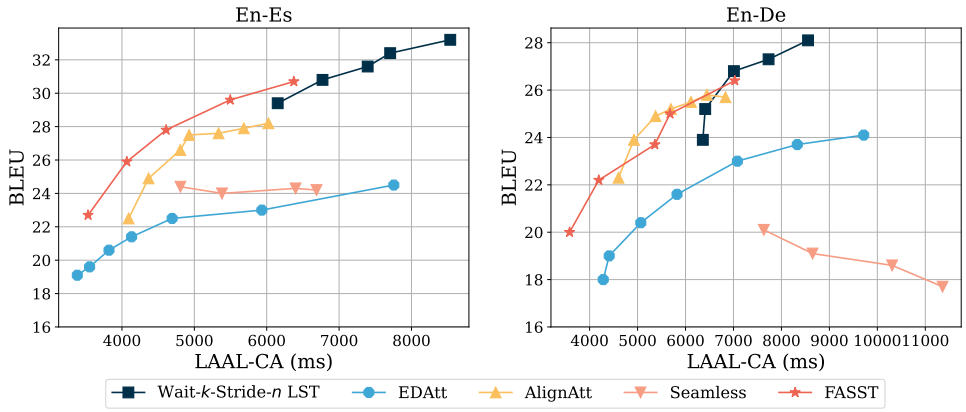

Figure 5: Quality-latency trade-off of FASST and baselines on English-Spanish and English-German direction. Quality is reflected by BLEU and latency is reflected by computation-aware length-adaptive average lagging (LAAL-CA). Given long speech input and large batch size, our model achieves overall the best quality-latency trade-off.

1 is 1e-4 and that of stage 2 is 2e-5. Gradients are clipped by the norm of 10. We train stage 1 for 500k steps and stage 2 for 1 epoch. We choose the checkpoint with the lowest dev loss. All our models are trained on 4 Nvidia A6000 GPUs with fp16. The temperature of WACO loss is 0.2 and we set $K = \{1, 2, 3, 4, 5, 100\}, n = 3$ for stage 2 training.

Inference We set speech segment size to 1 second to match the block size. We wait for $1 \leq k \leq 5$ segments at first. Then as each segment arrives, the speech encoder encodes its speech embedding incrementally and pass it to LLM, where LLM computes hidden states without recomputation and generates $n = 3$ words with greedy decoding as the partial translation.

4.3 Evaluation

We use SimulEval (Ma et al., 2020a) to evaluate our models and baselines. All models are evaluated on MuST-C-Long tst-COMMON with batch size of 8 during inference to simulate heavy workload. Since SimulEval does not support batching multiple instances, we duplicate each instance by 8 during model forwarding. We report SacreBLEU (Post, 2018) for translation quality and computation-aware length-adaptive average lagging (LAAL-CA) (Papi et al., 2022) for latency. All models are evaluated using a single A6000 GPU.

4.4 Baselines

Wait-k-Stride-n LST waits k fixed-length speech segments and translates n words every time (Ma et al., 2020b; Zeng et al., 2021). We run wait-k-stride-n policy on a strong offline LLM-based model LST (Zhang et al., 2023b) trained on the same mixed data as FASST. LST has the Encoder-Adapter-LLM architecture similar to FASST but employs bidirectional speech encoder and requires recomputation every time a new speech segment arrives. We set $k \in \{1, 2, 3, 4, 5\}$, $n = 3$ and segment length 1 second to match the setting of FASST.

EDAtt is an attention-based adaptive policy (Papi et al., 2023a). It leverages the encoder-decoder attention of an offline ST model to decide when to emit partial translations. The intuition is that if the attention is focused on early audio frames, the current translation can be emitted since sufficient information has been received. We use the model checkpoint and settings provided by the authors.

AlignAtt is the current state-of-the-art (SOTA) method that extends EDAtt by explicitly generating audio-translation alignments from encoder-decoder attention (Papi et al., 2023b). While EDAtt emits based on attention scores directly, AlignAtt decides based on whether a predicted token aligns with the latest audio frames, providing a more interpretable latency control. We also use the model checkpoint and optimal settings provided by the authors of AlignAtt.

Seamless is a multilingual streaming speech translation system with efficient monotonic multihead attention mechanism (Ma et al., 2023) to generate low latency translation (Communication et al., 2023). It computes target to source alignment using cross attention and writes translation if the

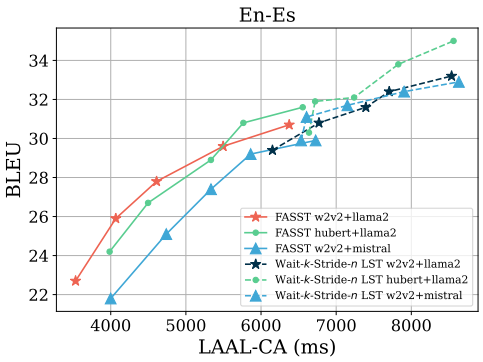

Figure 6: Ablation on the choice of pretrained speech encoder and LLM. We replace wav2vec 2.0 large with HuBERT large and Llama2 7B with Mistral 7B v0.3 base. FASST consistently has lower latency than Wait-k-Stride-n LST while maintaining an acceptable translation quality.

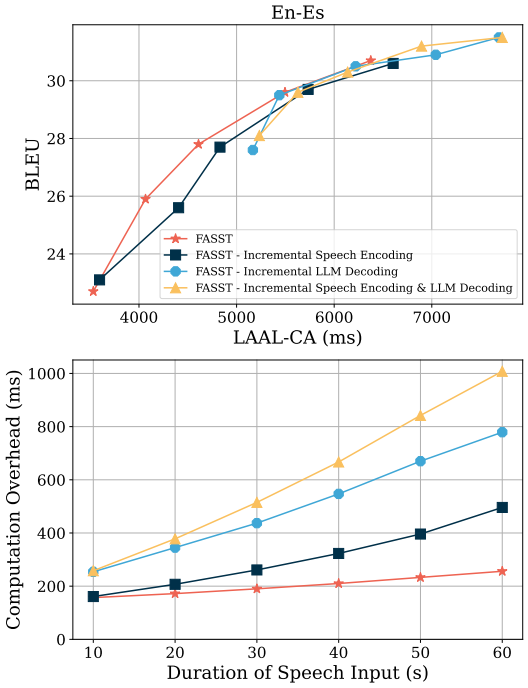

Figure 7: Ablation on incremental speech encoding and LLM decoding. Removing both incremental encoding and decoding can result in 4x larger latency for 60 seconds speech input.

alignment probability is larger than a threshold. We vary the threshold in $[0.2, 0.4, 0.6, 0.8]$ to evaluate its quality-latency trade-off.

4.5 Main Results

Main results are shown in Figure 5. Our model achieves the best quality-latency trade-off for En-Es direction. Although wait-k-stride-n LST has a 2 BLEU score advantage at the latency of 8 seconds, its bidirectional encoding and inefficient use of KV cache prohibit it reaching latency smaller than 6 seconds. Comparing to EDAtt and AlignAtt which do not use LLM and has much less parameters (115M) than our model (7B), our model has similar computation aware latency while achieving a 1.5 BLEU score improvement. For En-De direction, FASST achieves competitive results to AlignAtt, with slightly better quality when latency is smaller than 4 seconds or larger than 6 seconds.

4.6 Ablation Studies

We conduct ablation studies to examine the impact of each component in our model.

Speech Encoder and LLM We replace wav2vec 2.0 large with HuBERT large (Hsu et al., 2021) and Llama2 7B base with Mistral 7B v0.3 base (Jiang et al., 2023) to examine whether FASST is sensitive to the choice of pretrained speech encoder and LLM. We also train Wait-k-Stride-n LST baseline with these configurations as a comparison. Results are shown in Figure 6. For all configurations, FASST has lower latency than the baseline. FASST with HuBERT results in the best quality when the latency is around 5.5∼6.5 seconds and FASST with wav2vec 2.0 becomes the best when the latency is smaller than 5.5 seconds.

Incremental Encoding and Decoding We ablate the incremental speech encoding and the incremental LLM decoding to examine their impact. For encoding, we use the same architecture but recompute the entire speech encoder at each step. For decoding, we recompute the entire LLM hidden states given updated speech input and then incrementally decode the translation as each speech segment arrives. This also provides translation tokens with more context since they can attend to speech embeddings appear after them. Results are shown in Figure 7. Incremental encoding of speech encoder reduces the computational latency consistently by at least 200ms compared to recomputing encoder. Recomputing LLM does improve translation quality ($1 \sim 5$ BLEU), but also introduces significant computation overhead (~ 1.5 second), making it inferior for real-time application.

We also plot the computation cost of each read/write step for each variant in Figure 7 with wait-2-stride-3 policy. FASST scales the best with the speech input length and reduces the overhead by at most 4x compared to the one without incremen-

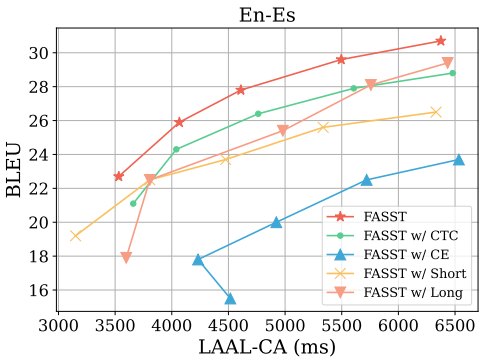

Figure 8: Ablation on the training strategy of FASST. We train stage 1 WACO loss with CTC and cross entropy (CE) loss and also change the training data to only MuST-C-Short and only MuST-C-Long. WACO and mixed-data training achieves the best performance.

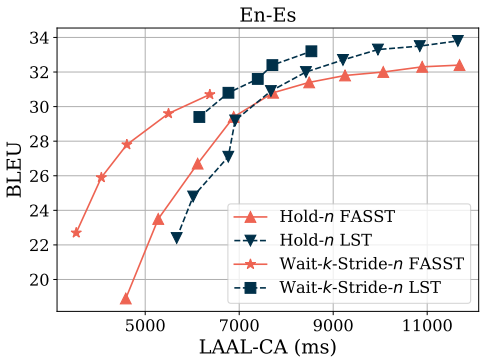

Figure 9: Quality-latency trade-off of FASST when applied to hold-n policy. We observe less improvement with respect to LST than wait-k-stride-n policy.

tal encoding and decoding. Removing incremental decoding results in larger additional computation overhead compared to removing incremental encoding since LLM has far more parameters than the speech encoder.

Speech-Text Alignment Training We replace WACO loss with CTC loss (Graves et al., 2006) and cross entropy loss in stage 1 to examine its impact on model performance. CTC loss aligns speech and text embeddings by optimizing all possible alignment paths. For cross entropy loss, we pass the speech embeddings to LLM and optimize the cross entropy loss with LLM parameters frozen. As shown in Figure 8, WACO loss consistently outperforms CTC and cross entropy by at least 1 BLEU score at the same latency.

Mixing MuST-C-Long and Short We train our model separately with only MuST-C-Short and only MuST-C-Long for the same number of epochs. As shown in Figure 8, the model trained with both long and short data outperforms the one trained with short data by up to 4 BLEU points. Though we are using LLM, the length extrapolation is still unsatisfactory. Training with long data improves the quality compared to short data at high latency, but still outperformed by mixed-data training.

4.7 Generalizability to Other Policy

We have demonstrated that our method works with wait-k-stride-n policy. However, plenty of policies other than wait-k and its variants have been developed to conduct simultaneous translation. Here we apply our method to hold-n policy (Liu et al., 2020) to exemplify how our method works on a different policy and in the meanwhile explain the factors that influence the effectiveness of our method.

Hold-n policy selects the best hypothesis after each speech segment arrives, discard the last n tokens from it and outputs the rest as the partial translation. Since hold-n is a test-time policy for offline ST model, we train an offline version of our model by replacing stage 2 finetuning with standard offline finetuning using cross entropy loss but keep blockwise-causal encoding. During inference, we still conduct the same incremental encoding and decoding.

As shown in Figure 9, our method has less advantage when applied to hold-n comparing to wait-k-stride-3. The major difference between two policies in terms of computation is that hold-n policy spends more time on autoregressive decoding since it decodes more tokens each time. On average hold-n policy generates more than 6 words each step while wait-k-stride-3 generates at most 3 words. FASST accelerates the encoding of existing features, but for policies like hold-n that involve heavy autoregressive decoding the advantage of our method gets marginalized.

5 Conclusion

In this work, we introduce FASST, a fast LLM-based simultaneous speech translation model. FASST consists of blockwise-causal speech encoding, incremental LLM decoding with consistency mask, and a novel 2-stage training strategy. Experiments on MuST-C dataset show that FASST significantly reduce computation overhead while maintaining its translation quality. Our generalization study shows that policies that spend more time on encoding than decoding benefit more from FASST.

Limitations

- There might be data leakage since LLM is trained on vast amount of text data, so we cannot guarantee LLM does not see the test translation data during pretraining.
- FASST is only tested on two language directions instead of all 8 language directions of MuST-C, so its generalizability to other language directions is unknown.
- There is still a quality gap between blockwise-causal speech encoding and bidirectional speech encoding. It is unclear how to further close the gap.
- We only explore one LLM-ST architecture in the paper and we cannot guarantee that FASST or its idea works on other architectures.

References

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

Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Preprint*, arXiv:2006.11477.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Junkun Chen, Mingbo Ma, Renjie Zheng, and Liang Huang. 2021. Direct simultaneous speech-to-text translation assisted by synchronized streaming ASR. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4618–4624, Online. Association for Computational Linguistics.

Seamless Communication, Loïc Barrault, Yu-An Chung, Mariano Coria Meglioli, David Dale, Ning Dong, Mark Duppenthaler, Paul-Ambroise Duquenne, Brian Ellis, Hady Elsahar, Justin Haaheim, John Hoffman, Min-Jae Hwang, Hirofumi Inaguma, Christopher Klaiber, Ilia Kulikov, Pengwei Li, Daniel Licht, Jean Maillard, Ruslan Mavlyutov, Alice Rakotoarison, Kaushik Ram Sadagopan, Abinesh Ramakrishnan, Tuan Tran, Guillaume Wenzek, Yilin Yang, Ethan Ye, Ivan Evtimov, Pierre Fernandez, Cynthia Gao, Prangthip Hansanti, Elahe Kalbassi, Amanda Kallet, Artyom Kozhevnikov, Gabriel Mejia Gonzalez, Robin San Roman, Christophe Touret, Corinne Wong, Carleigh Wood, Bokai Yu, Pierre Andrews, Can Balioglu, Peng-Jen Chen, Marta R. Costa-jussà, Maha Elbayad, Hongyu Gong, Francisco Guzmán, Kevin Heffernan, Somya Jain, Justine Kao, Ann Lee, Xutai Ma, Alex Mourachko, Benjamin Peloquin, Juan Pino, Sravya Popuri, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, Anna Sun, Paden Tomasello, Changhan Wang, Jeff Wang, Skyler Wang, and Mary Williamson. 2023. Seamless: Multilingual expressive and streaming speech translation. *Preprint*, arXiv:2312.05187.

Tri Dao. 2024. Flashattention-2: Faster attention with better parallelism and work partitioning. In *The Twelfth International Conference on Learning Representations*.

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.

Qian Dong, Yaoming Zhu, Mingxuan Wang, and Lei Li. 2022. Learning when to translate for streaming speech. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 680–694, Dublin, Ireland. Association for Computational Linguistics.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.

Shoutao Guo, Shaolei Zhang, Zhengrui Ma, Min Zhang, and Yang Feng. 2024. Sillm: Large language models for simultaneous machine translation. *Preprint*, arXiv:2402.13036.

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 Trans. Audio, Speech and Lang. Proc.*, 29:3451–3460.

Zhichao Huang, Rong Ye, Tom Ko, Qianqian Dong, Shanbo Cheng, Mingxuan Wang, and Hang Li. 2023. Speech translation with large language models: An industrial practice. *Preprint*, arXiv:2312.13585.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

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

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

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

Xutai Ma, Mohammad Javad Dousti, Changhan Wang, Jiatao Gu, and Juan Pino. 2020a. 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.

Xutai Ma, Juan Pino, and Philipp Koehn. 2020b. SimulMT to SimulST: Adapting simultaneous text translation to end-to-end simultaneous speech translation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 582–587, Suzhou, China. Association for Computational Linguistics.

Xutai Ma, Anna Sun, Siqi Ouyang, Hirofumi Inaguma, and Paden Tomasello. 2023. Efficient monotonic multihead attention. *Preprint*, arXiv:2312.04515.

Xutai Ma, Yongqiang Wang, Mohammad Javad Dousti, Philipp Koehn, and Juan Pino. 2020c. Streaming simultaneous speech translation with augmented memory transformer. *Preprint*, arXiv:2011.00033.

R OpenAI. 2023. Gpt-4 technical report. arxiv 2303.08774. *View in Article*, 2:13.

Siqi Ouyang, Rong Ye, and Lei Li. 2023. WACO: Word-aligned contrastive learning for speech translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3891–3907, Toronto, Canada. 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. 2023a. 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.

Sara Papi, Marco Turchi, and Matteo Negri. 2023b. Alignatt: Using attention-based audio-translation alignments as a guide for simultaneous speech translation. In *INTERSPEECH 2023*, interspeech_2023. ISCA.

Peter Polák, Brian Yan, Shinji Watanabe, Alex Waibel, and Ondřej Bojar. 2023. Incremental Blockwise Beam Search for Simultaneous Speech Translation with Controllable Quality-Latency Tradeoff. In *Proc. INTERSPEECH 2023*, pages 3979–3983.

Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Anselm Levskaya, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. 2022. Efficiently scaling transformer inference. *Preprint*, arXiv:2211.05102.

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

Matthew Raffel, Victor Agostinelli, and Lizhong Chen. 2024. Simultaneous masking, not prompting optimization: A paradigm shift in fine-tuning llms for simultaneous translation. *Preprint*, arXiv:2405.10443.

Matthew Raffel and Lizhong Chen. 2023. Implicit memory transformer for computationally efficient simultaneous speech translation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12900–12907, Toronto, Canada. Association for Computational Linguistics.

Matthew Raffel, Drew Penney, and Lizhong Chen. 2023. Shiftable context: addressing training-inference context mismatch in simultaneous speech translation. In *International Conference on Machine Learning*, pages 28519–28530. PMLR.

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.

Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2020. Leveraging pre-trained checkpoints for sequence generation tasks. *Transactions of the Association for Computational Linguistics*, 8:264–280.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2023. Roformer: Enhanced transformer with rotary position embedding. *Preprint*, arXiv:2104.09864.

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, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023a. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

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, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

David Vilar, Markus Freitag, Colin Cherry, Jiaming Luo, Viresh Ratnakar, and George Foster. 2023. Prompting PaLM for translation: Assessing strategies and performance. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15406–15427, Toronto, Canada. Association for Computational Linguistics.

Chunyang Wu, Yongqiang Wang, Yangyang Shi, Ching-Feng Yeh, and Frank Zhang. 2020. Streaming transformer-based acoustic models using self-attention with augmented memory. *Preprint*, arXiv:2005.08042.

Xueqing Wu, Lewen Wang, Yingce Xia, Weiqing Liu, Lijun Wu, Shufang Xie, Tao Qin, and Tie-Yan Liu. 2021. Temporally correlated task scheduling for sequence learning. In *International Conference on Machine Learning*, pages 11274–11284. PMLR.

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. *Preprint*, arXiv:2309.11674.

Wen Yang, Chong Li, Jiajun Zhang, and Chengqing Zong. 2023. Bigtranslate: Augmenting large language models with multilingual translation capability over 100 languages. *Preprint*, arXiv:2305.18098.

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.

Xingshan Zeng, Liangyou Li, and Qun Liu. 2023. AdaTranS: Adapting with boundary-based shrinking for end-to-end speech translation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2353–2361, Singapore. Association for Computational Linguistics.

Biao Zhang, Barry Haddow, and Alexandra Birch. 2023a. Prompting large language model for machine translation: a case study. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Hao Zhang, Nianwen Si, Yaqi Chen, Wenlin Zhang, Xukui Yang, Dan Qu, and Xiaolin Jiao. 2023b. Tuning large language model for end-to-end speech translation. *Preprint*, arXiv:2310.02050.

Linlin Zhang, Kai Fan, Jiajun Bu, and Zhongqiang Huang. 2023c. Training simultaneous speech translation with robust and random wait-k-tokens strategy. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7814–7831, Singapore. Association for Computational Linguistics.

Shaolei Zhang, Qingkai Fang, Zhuocheng Zhang, Zhengrui Ma, Yan Zhou, Langlin Huang, Mengyu Bu, Shangtong Gui, Yunji Chen, Xilin Chen, and Yang Feng. 2023d. Bayling: Bridging cross-lingual alignment and instruction following through interactive translation for large language models. *Preprint*, arXiv:2306.10968.

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.

Shaolei Zhang and Yang Feng. 2023. End-to-end simultaneous speech translation with differentiable segmentation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7659–7680, Toronto, Canada. Association for Computational Linguistics.