Towards Fine-Grained Code-Switch Speech Translation with Semantic Space Alignment

Yan Gao[1], **Yazheng Yang**[1], **Zhibin Lan**[1], **Yidong Chen**[1], **Min Zhang**[2], **Daimeng Wei**[2], **Derek F. Wong**[3] and **Jinsong Su**[1*]
[1]School of Informatics, Xiamen University, China
[2]Huawei Translation Services Center, Beijing, China
[3]NLP[2]CT Lab, Department of Computer and Information Science, University of Macau
gaoyan@stu.xmu.edu.cn, yazhengy.cs@gmail.com, jssu@xmu.edu.cn

Abstract

Code-switching (CS) speech translation (ST) aims to translate speech that alternates between multiple languages into a target language text, posing significant challenges due to the complexity of semantic modeling and the scarcity of CS data. Previous studies mainly rely on the models themselves to implicitly learn semantic representations and resort to costly manual annotations. To mitigate these limitations, we propose enhancing Large Language Models (LLMs) with a Mixture-of-Experts (MoE) speech projector composed of language expert groups, where each group specializes in the semantic space of a specific language for fine-grained speech feature modeling. A language-specific loss and an intra-group load balancing loss are jointly introduced to guide efficient token routing across and within expert groups. Furthermore, we introduce a multi-stage training paradigm that utilizes readily available automatic speech recognition (ASR) and monolingual ST data, facilitating speech-text alignment and improving translation performance. To bridge the data gap for smooth domain transfer, a transition loss is employed to improve adaptation to CS scenarios. Extensive experiments on widely used datasets demonstrate the effectiveness and generality of our approach, achieving average improvements of 0.86 BLEU and 0.93 COMET over SeamlessM4T, with maximum improvements of 1.49 BLEU and 1.41 COMET across different test sets. Our code and supplementary appendices are available at https://github.com/XMUDeepLIT/CSST-SSA.

1 Introduction

Code-switching (CS), the practice of alternating between multiple languages within a single utterance or discourse, is a pervasive phenomenon in multilingual speech and poses significant challenges to speech and language processing models [Scotton and Ury, 1977]. The task of CS speech translation (ST) aims to translate such speech into text in a target language. With globalization [Winata *et al.*, 2023], CS has become increasingly prevalent, extending beyond multilingual communities to predominantly monolingual settings, thus drawing much attention recently.

While CS has been extensively explored in machine translation (MT) [Xu and Yvon, 2021; Gupta *et al.*, 2021; Vavre *et al.*, 2022; Gaser *et al.*, 2023; Pengpun *et al.*, 2024; Borisov *et al.*, 2025] and automatic speech recognition (ASR) [Winata *et al.*, 2020; Chi and Bell, 2022; Dhawan *et al.*, 2023; Kronis *et al.*, 2024], its research in ST remains under-explored. This is primarily due to two key challenges: 1) *semantic modeling complexity* introduced by language alternation, which poses significant challenges for the model in capturing effective representations from CS speech; 2) *data scarcity*, as high-quality and large-scale CS speech translation datasets are limited and difficult to construct, which limits effective model training. Prior methods tend to ignore the semantic complexity arising from language switching, simply leaving it to the models themselves to implicitly learn and resolve during training [Huber *et al.*, 2022; Weller *et al.*, 2022; Alastruey *et al.*, 2023; Shankar *et al.*, 2025], which hinders overall performance. To mitigate data scarcity, previous studies resort to manual annotations [Alastruey *et al.*, 2023; Shankar *et al.*, 2025], which are inefficient and costly, producing datasets of limited scale. Other approaches focus on synthetic data, which often suffer from unnatural switching, disrupted word order, and grammatical inconsistencies [Xie *et al.*, 2025].

Recently, large language models (LLMs) have demonstrated remarkable performance across diverse tasks. To equip LLMs with cross-modal capabilities, prior approaches typically introduce a projector that connects a pretrained speech encoder to the LLM [Chen *et al.*, 2024; Zhang *et al.*, 2024a; Wu *et al.*, 2025; Li *et al.*, 2026]. This architecture effectively integrates the speech encoder's strength in extracting acoustic features with the LLM's powerful language modeling ability. However, existing studies reveal that despite being pretrained on large-scale multilingual corpora, LLMs still exhibit limited cross-lingual capability, resulting in inferior performance on CS data relative to monolingual data [Zhang *et al.*, 2023; Mohamed *et al.*, 2025]. Moreover, current research on LLMs in the context of CS primarily focuses on MT [Zhang *et al.*, 2024b; Gupta *et al.*, 2024], leaving their potential in CS speech translation largely unexplored.

*Corresponding author.

In this work, to address the aforementioned challenges, we explore the use of LLMs for CS speech translation, improving performance from the perspectives of both model architecture and data utilization. 1) Based on the above-mentioned architecture, we introduce a novel Mixture-of-Experts (MoE) [Shazeer *et al.*, 2017] speech projector, where experts are organized into language-aware groups to enable fine-grained modeling of speech features from different languages, thereby better capturing the complex cross-lingual semantics in CS speech. 2) To mitigate the scarcity of high-quality CS speech translation data, we propose a novel multi-stage training paradigm that leverages high-resource ASR data to strengthen semantic comprehension and monolingual ST data to establish robust translation capability, followed by a gradual transition to CS speech translation data that enables smooth domain adaptation. Specifically, our training paradigm comprises four stages. In the first stage, ASR data are used to pretrain individual projectors for each language involved in the target CS scenario. In the second stage, these projectors of different languages form individual expert groups, which are further aggregated to construct a MoE speech projector. Additionally, a language-specific loss and an intra-group load balancing loss are introduced to better guide the learning of the MoE speech projector and promote effective expert specialization. The first two stages jointly align the speech and text modalities, facilitating the learning of cross-modal representations. In the third stage, we progressively transition from ASR to monolingual ST data using a transition loss to enhance the model's translation ability. In the final stage, the model is further adapted from monolingual ST to CS speech translation data, thus enabling effective translation of CS speech.

To summarize, the main contributions of our work include the following four aspects:

- To the best of our knowledge, we are the first to explore the use of LLMs in end-to-end CS speech translation, providing baseline results to facilitate future research.
- We propose a novel MoE speech projector that enables fine-grained modeling of speech features from different languages, thereby enhancing the model's ability to capture semantic distinctions in CS speech.
- We propose a multi-stage training paradigm that progressively guides the model to learn speech-text alignment and adapt to CS scenarios, effectively mitigating the scarcity of high-quality CS speech translation data.
- Empirical evaluations on monolingual and CS speech translation datasets validate the effectiveness of our model, and in-depth analysis experiments provide detailed insights into the contribution of each module.

2 Related Works

CS translation in natural language processing (NLP) has garnered much attention and focuses mainly on MT [Xu and Yvon, 2021; Gupta *et al.*, 2021; Vavre *et al.*, 2022; Gaser *et al.*, 2023; Pengpun *et al.*, 2024; Borisov *et al.*, 2025]. However, CS frequently occurs in spoken contexts such as lectures, meetings, and phone conversations, which necessitates handling speech input. To handle speech input, such methods are typically cascaded with ASR models [Winata *et al.*, 2020; Chi and Bell, 2022; Dhawan *et al.*, 2023; Kronis *et al.*, 2024], which suffer from error propagation. Recent research efforts explore end-to-end (E2E) ST to mitigate this issue [Weller *et al.*, 2022; Huber *et al.*, 2022; Alastruey *et al.*, 2023; Yang *et al.*, 2024; Shankar *et al.*, 2025]. A representative approach, COSTA [Shankar *et al.*, 2025], enhances translation quality by aligning speech and text representations via interleaving their respective embeddings during fine-tuning.

Recently, LLMs have demonstrated impressive performance across diverse tasks, leading to increasing research on their use in CS machine translation [Zhang *et al.*, 2023; Zhang *et al.*, 2024b; Gupta *et al.*, 2024; Mohamed *et al.*, 2025] with promising results. Despite these successes, the potential of LLMs for CS speech translation remains largely unexplored. Current studies on monolingual ST generally integrate a pretrained speech encoder with an LLM through a projector [Chen *et al.*, 2024; Wu *et al.*, 2025; Li *et al.*, 2026], thereby equipping the LLM with cross-modal understanding abilities. This framework effectively combines the strength of the speech encoder in extracting speech features with the powerful translation capability of the LLM. Although this framework has shown promising results in monolingual settings, its effectiveness in more challenging CS scenarios remains unclear.

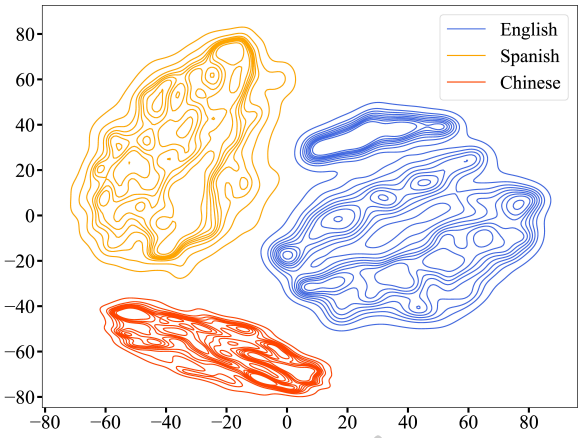

Figure 1: KDE plots of speech features from the validation set for different languages using t-SNE.

Settings	En	Es	zh-CN
Share Projector	7.07	6.01	14.26
Expert Projector	7.05	4.52	13.21

Table 1: WER↓(En, Es) and CER↓(zh-CN) scores of the two settings on the validation set.

3 Preliminary Study

In this section, we explore the complexity of semantic modeling in CS speech from two perspectives. First, we examine whether a semantic space gap exists between different languages. Second, we evaluate the impact of employing a shared projector on model performance.

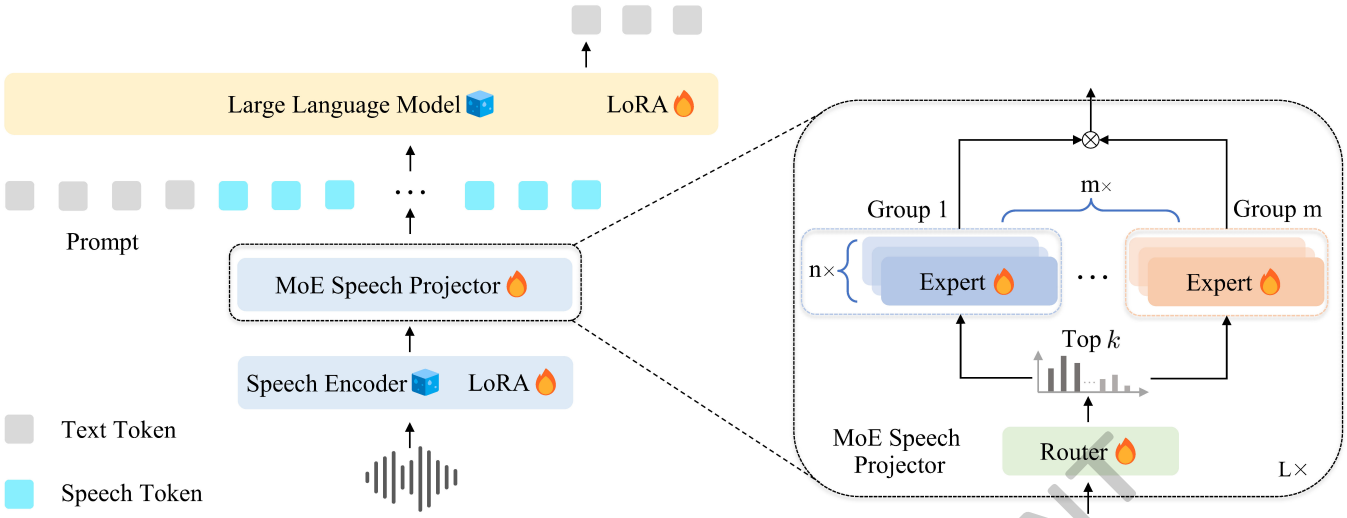

Figure 2: Illustration of the proposed framework. For CS speech involving m different languages, the MoE speech projector assigns one expert group comprising n experts per language, thus the total number of experts $N = n \times m$. During training, LoRA [Hu *et al.*, 2022] is applied to fine-tune the parameters of the LLM and the speech encoder, while fully fine-tuning all parameters of the MoE speech projector.

Does a semantic space gap exist between different languages? To investigate this issue, we conduct a preliminary experiment using Spanish, English, and Chinese ASR data from the Common Voice corpus [Ardila *et al.*, 2020]. Specifically, mean-pooled speech features extracted from the Whisper encoder [Radford *et al.*, 2023] are reduced to two dimensions using t-SNE [Hinton and Roweis, 2002] and visualized via a bivariate kernel density estimation (KDE) plot, as illustrated in Figure 1. The visualization reveals that samples from the same language cluster tightly, while those from different languages are clearly separated, indicating clear boundaries between languages. These findings provide empirical evidence of a substantial semantic space gap across languages in speech features.

Does sharing the projector across languages affect performance? Previous studies typically employ a single projector to align the semantic spaces of the speech encoder and the LLM. However, given the observed cross-lingual semantic space gap, the suitability of a shared projector in capturing semantic distinctions across languages remains uncertain. To investigate this issue, we compare two settings: (1) a shared projector for all languages, in which all speech tokens are processed by the single projector; and (2) one expert projector per language, where each speech token is processed by the projector corresponding to its language. We only fine-tune the projector, while keeping the parameters of the speech encoder and the LLM frozen. Since CS speech lacks oracle language labels at the token level, we also evaluate on monolingual ASR data, where all speech tokens are labeled with their corresponding language. First, we pre-train a single projector on the ASR data to mitigate variance from random initialization and establish a basic semantic alignment. Both the shared and expert projectors are then initialized with the pre-trained weights. We subsequently fine-tune both settings on the same ASR data. The experimental results, reported in Table 1, show that the expert projector consistently outperforms the shared projector, especially for Spanish and Chinese. This finding indicates that using expert projectors for different languages better captures semantic distinctions. However, in practical CS scenarios, oracle language labels for speech tokens are unavailable. To address this limitation, we adopt the MoE projector, enabling the model to route speech tokens to appropriate expert projectors in an adaptive manner.

4 Methodology

In this section, we first describe the architecture of our model and then detail the four-stage training strategy, which constructs the MoE speech projector with two auxiliary losses and progressively adapts the model from ASR to monolingual ST and finally to CS speech translation via the transition loss.

4.1 Model Architecture

As shown in Figure 2, our framework primarily consists of a speech encoder, a MoE speech projector, and an LLM. The speech encoder extracts features from the speech input, which are aligned with the representation space of the LLM via the MoE speech projector. These aligned features are concatenated with the prompt embeddings and fed into the LLM to generate the translation.

Speech Encoder. The speech encoder takes a speech input $\boldsymbol{s}$ and encodes it into a representation sequence $\boldsymbol{h}$:

$$\boldsymbol{h} = \text{Encoder}(\boldsymbol{s}), \tag{1}$$

where $\boldsymbol{h} \in \mathbb{R}^{T_s \times D_{\text{enc}}}$ denotes the encoder output, with sequence length T_s and encoder hidden dimension D_{enc}.

MoE Speech Projector. The MoE speech projector connects the speech encoder to the LLM by mapping the representation space of the encoder output to the LLM input.

Each MoE layer consists of a set of N expert linear layers $E = \{E_1, E_2, \ldots, E_N\}$ and a sparse router G. For the l-th MoE layer, the sparse router predicts a probability distribution over experts and selects the top-k experts at the token level. The outputs of the selected experts are then weighted by their corresponding probabilities and summed to produce the output $\boldsymbol{h}_{l+1}$. The complete computation process is formulated as follows:

$$p_i(h_l^t) = \frac{\exp(G_i(h_l^t))}{\sum_{i'=1}^{N} \exp(G_{i'}(h_l^t))}, \tag{2}$$

$$h_{l+1}^t = \sum_{i \in \text{Top-}k(h_l^t)} p_i(h_l^t) \cdot E_i(h_l^t), \tag{3}$$

where i indexes the i-th expert, t indexes the t-th element in $\boldsymbol{h}$, $G(h) = \mathbf{X}_G \cdot h$ and $E(h) = \mathbf{X}_E \cdot h$ denote the linear transformations of the router and an expert, respectively. Specifically, $\mathbf{X}_E^{(1)} \in \mathbb{R}^{D_{\text{enc}} \times D_{\text{LLM}}}$ for the first layer and $\mathbf{X}_E^{(l)} \in \mathbb{R}^{D_{\text{LLM}} \times D_{\text{LLM}}}$ for the remaining layers, where D_{LLM} denotes the hidden dimension of the LLM.

The overall process is summarized as follows:

$$\boldsymbol{h}_s = \text{MoE Speech Projector}(\boldsymbol{h}), \tag{4}$$

where $\boldsymbol{h}_s \in \mathbb{R}^{T_s \times D_{\text{LLM}}}$ denotes the speech projector output.

LLM. The tokenizer and embedding layer process the prompt text to obtain the representation $\boldsymbol{h}_t$, which is concatenated with the speech representation $\boldsymbol{h}_s$ to form the LLM input $\boldsymbol{h}_z$:

$$\boldsymbol{h}_z = \boldsymbol{h}_t \oplus \boldsymbol{h}_s, \tag{5}$$

where the operator $\oplus$ denotes vector concatenation, $\boldsymbol{h}_t \in \mathbb{R}^{T_t \times D_{\text{LLM}}}$ and $\boldsymbol{h}_z \in \mathbb{R}^{(T_t+T_s) \times D_{\text{LLM}}}$, with T_t denoting the sequence length of $\boldsymbol{h}_t$.

4.2 Multi-stage Transition Training

Stage 1. In the initial stage, we aim to align the speech and text modalities to obtain well-initialized parameters.

Given m languages involved in the CS scenario, we first randomly initialize an MLP-based projector for each language and pretrain it using the corresponding ASR data. The training objective is to minimize the cross-entropy loss:

$$\mathcal{L}_{\text{ce}} = -\sum_{t=1}^{T} \log p(y_t \mid \boldsymbol{s}, \boldsymbol{y}_{<t}; \theta), \tag{6}$$

where t is the decoding timestep, y_t denotes the target text token at timestep t, and θ refers to the model parameters.

Stage 2. For each language, we initialize an expert group consisting of n experts with the weights of the corresponding pretrained projector to facilitate effective training. The expert groups from all m languages are then aggregated to form the MoE speech projector, with a randomly initialized router inserted between each layer. Thus, the total number of experts is $N = n \times m$. Although all experts within the same group share identical initial weights, the randomly initialized router assigns tokens to different experts, allowing them to gradually specialize through training.

In addition to the cross-entropy loss, we incorporate two auxiliary losses: a language-specific loss and an intra-group load balancing loss. The language-specific loss provides explicit supervision in routing speech tokens to the expert group corresponding to their language. For monolingual ASR data, all speech tokens are labeled with their respective language. Specifically, the loss penalizes the routing of tokens to experts from other language groups and is formally defined as below:

$$\mathcal{L}_{\text{lang}} = -\sum_{t=1}^{T_s} \sum_{l=1}^{L} \sum_{i=1}^{N} \left[(1 - z_i) \log(1 - p_i(h_l^t)) \right], \tag{7}$$

where L is the number of layers, and z_i=1 if expert i belongs to the expert group corresponding to the language of the input speech, otherwise z_i=0.

While the language-specific loss provides supervision at the group level, it lacks regulation over the token distribution among individual experts within each expert group. To address this limitation, the intra-group load balancing loss is introduced to encourage balanced token assignment within each expert group, preventing over-reliance on a subset of experts. The loss is formulated as:

$$\mathcal{L}_{\text{balance}} = \sum_{l=1}^{L} \sum_{j=1}^{m} \sum_{i=1}^{n} f_{ijl} \cdot P_{ijl}, \tag{8}$$

where f_{ijl} is the fraction of tokens in language j dispatched to expert i within the corresponding expert group:

$$f_{ijl} = \frac{\sum_{t \in T_j} \mathbb{1}\{\arg\max_r \ p_r(h_l^t) = i\}}{\sum_{i'=1}^{n} \sum_{t \in T_j} \mathbb{1}\{\arg\max_r \ p_r(h_l^t) = i'\}}, \tag{9}$$

and P_{ijl} is the average routing probability of assigning tokens in language j to expert i within the corresponding expert group:

$$P_{ijl} = \frac{\sum_{t \in T_j} p_i(h_l^t)}{\sum_{i'=1}^{n} \sum_{t \in T_j} p_{i'}(h_l^t)}, \tag{10}$$

where T_j is the set of tokens belonging to language j.

Using the same monolingual ASR data as in Stage 1, the overall objective is summarized as follows:

$$\mathcal{L} = \mathcal{L}_{\text{ce}} + \mathcal{L}_{\text{lang}} + \mathcal{L}_{\text{balance}} \tag{11}$$

Stage 3. This stage focuses on enhancing the translation ability of the model by training on monolingual ST data. However, directly switching from ASR to ST introduces training inconsistency due to the task gap between speech recognition and translation. To enable a smoother transition, we mix ASR data with monolingual ST data and introduce the transition loss:

$$\mathcal{L}_{\text{transition}} = (1 - \lambda)\mathcal{L}_{\text{asr\_ce}} + \lambda \mathcal{L}_{\text{st\_ce}}, \tag{12}$$

where $\lambda = \frac{b}{B}$, b denotes the current batch index, and B is the total number of batches. $\mathcal{L}_{\text{asr\_ce}}$ and $\mathcal{L}_{\text{st\_ce}}$ denote the cross-entropy losses for the ASR and monolingual ST data, respectively.

Additionally, the language-specific loss and the intra-group load balancing loss are also employed in this stage. The overall loss is therefore formulated as follows:

$$\mathcal{L} = \mathcal{L}_{\text{transition}} + \mathcal{L}_{\text{lang}} + \mathcal{L}_{\text{balance}} \tag{13}$$

Dataset	Common Voice						Fisher				NTUML2021			
	En		Es		zh-CN		CS		Mono.		CS		Mono.	
Train	233.0K	364.4H	64.4K	96.6H	7.1K	10.4H	8.2K	14.7H	130.6K	156.8H	7.9K	5.1H	9.9K	5.7H
Valid	15.5K	26.1H	13.2K	21.8H	4.8K	7.9H	0.2K	0.3H	3.4K	4.1H	1.5K	1.0H	1.5K	0.8H
Test	15.5K	24.7H	13.2K	22.7H	4.9K	8.2H	1.0K	1.6H	10.6K	12.2H	6.2K	4.0H	8.7K	4.9H

Table 2: Statistics of various datasets, including the number of pairs in thousands (K) and total duration in hours (H).

Model	BLEU	COMET
Experts-3	38.02	80.95
Experts-5	38.40	80.62
Experts-7	**38.55**	**81.02**
Experts-9	38.35	80.99
Experts-11	37.74	80.77

Table 3: BLEU and COMET scores on the NTUML2021 CS validation set with top-3 and varying numbers of experts n.

Model	BLEU	COMET
Top-1	27.76	76.38
Top-3	38.55	81.02
Top-5	38.53	81.01
Top-7	**38.90**	**81.14**

Table 4: BLEU and COMET scores on the NTUML2021 CS validation set with n=7 and varying top-k values.

Stage 4. In the final stage, we employ CS speech translation data to adapt the translation ability of the model to CS scenarios. To mitigate the distribution gap between monolingual and CS speech translation, we combine monolingual ST with CS speech translation data and apply the transition loss:

$$\mathcal{L}_{\text{transition}} = (1-\lambda)\mathcal{L}_{\text{st\_ce}} + \lambda\mathcal{L}_{\text{csst\_ce}}, \quad (14)$$

where $\mathcal{L}_{\text{csst\_ce}}$ denotes the cross-entropy loss for the CS speech translation data.

Notably, since the language label of each speech token is unavailable in CS speech, both the language-specific loss and the intra-group load balancing loss are excluded at this stage.

5 Experiments

In this section, we conduct extensive experiments to evaluate the effectiveness of our proposed approach. We first describe the experimental setup, including evaluation metrics, datasets, model details, hyperparameters, and baseline methods. We then present the main results and ablation studies to analyze the overall performance and the contribution of each component, followed by in-depth analyses of the experts to better understand their semantic modeling behavior.

5.1 Experimental Setting

Metrics. We report SacreBLEU [Post, 2018] and COMET [Rei *et al.*, 2020] to evaluate translation quality. ASR performance is measured by Word Error Rate (WER) [Morris *et al.*, 2004] for English and Spanish, and Character Error Rate (CER) for Chinese.

Datasets. Following [Weller *et al.*, 2022] and [Yang *et al.*, 2024], we select the Fisher and NTUML2021 datasets to evaluate model performance, as well as the widely used Common Voice dataset for ASR. The dataset details are as follows:

- **Common Voice.** [Ardila *et al.*, 2020] This is a widely used multilingual ASR dataset consisting of monolingual speech and corresponding transcriptions. We select the English, Spanish, and Chinese subsets as our ASR data.
- **Fisher.** [Cieri *et al.*, 2004] This is a CS speech translation dataset containing English–Spanish source speech with English target translations. We follow the splits defined by [Weller *et al.*, 2022] and adopt their separation of monolingual and CS speech translation data.
- **NTUML2021.** [Yang *et al.*, 2024] This dataset consists of Chinese–English source speech with English target translations. We adopt the original data splits and categorize samples whose transcriptions contain both Chinese and English as CS speech translation data, while the remaining samples are treated as monolingual speech translation data.

The numbers of parallel speech-text pairs and total hours for each dataset are reported in Table 2.

Model Details. We use Llama-3-8B-Instruct[1] as the base LLM and adopt Whisper-large-v3 encoder as the speech encoder. The MoE speech projector consists of three layers and employs ReLU [Agarap, 2018] as the activation function.

Hyperparameter Tuning. We tune the number of experts per group n and the top-k value on the CS validation set of the NTUML2021 dataset. For consistency, we use a uniform number of experts across all groups to avoid introducing additional variability, although this number can be configured independently for each group. Specifically, we first fix the top-k value to 3 and vary n over the set $\{3, 5, 7, 9, 11\}$. As shown in Table 3, performance steadily improves as n increases until it peaks at n=7, and gradually declines thereafter. We then fix n=7 and tune the top-k value over the set $\{1, 3, 5, 7\}$. As illustrated in Table 4, activating more experts generally improves performance, with top-7 achieving the best results.

Accordingly, we set n=7 and select the top-7 experts during inference.

5.2 Baselines

Our baselines include the following models:

[1]https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct

Model	Fisher		NTUML2021	
	CS	Mono.	CS	Mono.
Whisper	31.56 / 72.63	31.04 / 75.93	29.12 / 74.90	28.92 / 77.54
SeamlessM4T	36.89 / 76.13	35.04 / 79.71	38.03 / 79.93	35.78 / 81.17
Whisper+LLaMA	29.67 / 70.99	33.20 / 76.08	34.85 / 79.08	35.21 / **81.68**
LLaST	33.78 / 75.35	30.67 / 78.37	32.53 / 78.92	33.81 / 79.03
Ours	**37.51 / 77.54**	**35.87 / 80.14**	**39.52 / 81.33**	**36.27** / 81.65

Table 5: BLEU↑ / COMET↑ scores of various models on the Fisher and NTUML2021 test sets. CS refers to CS speech translation data, and Mono. refers to monolingual ST data.

- **Whisper.** [Radford *et al.*, 2023]. As a large-scale pretrained model, Whisper demonstrates strong performance across a wide range of speech translation tasks and is widely regarded as a robust baseline.
- **SeamlessM4T.** [Communication *et al.*, 2023]. Similar to Whisper, SeamlessM4T also exhibits strong performance across diverse speech translation tasks and serves as a competitive baseline.
- **Whisper+LLaMA.** We construct a cascaded speech translation pipeline by combining Whisper for ASR with LLaMA for MT. Specifically, Whisper is first used to transcribe speech into source language text, which is then translated into the target language by LLaMA. This pipeline represents a strong cascaded baseline for comparison with our E2E speech translation models.
- **LLaST.** [Chen *et al.*, 2024]. As there is currently no work specifically targeting E2E CS speech translation with LLMs, we include LLaST as a representative baseline. It follows a mainstream architecture that integrates a pretrained speech encoder with an LLM and shows competitive performance on monolingual ST tasks.

5.3 Main Results

Table 5 presents the BLEU and COMET scores of various models on the Fisher and NTUML2021 test sets under two evaluation settings: CS speech translation data only (CS) and monolingual ST data only (Mono.).

Under the CS setting, our model achieves state-of-the-art performance with BLEU/COMET scores of $37.51/77.54$ on Fisher and $39.52/81.33$ on NTUML2021, clearly outperforming all baselines. Notably, although primarily designed for CS speech translation, our approach also delivers competitive or even superior performance in the Mono. setting, indicating its effectiveness for both monolingual and CS speech translation scenarios.

Overall, these results demonstrate the advantage of our method in capturing fine-grained speech features and improving translation robustness across diverse scenarios.

5.4 Ablation Studies

As shown in Table 6, we consider the following variants:

- *w/o MoE Speech Projector.* In this variant, the MoE speech projector is replaced with a single MLP projector that adopts the same number of layers and ReLU activation. Accordingly, the language-specific loss and the intra-group load balancing loss are removed, along with Stage 2. As indicated in Line 2, replacing the MoE speech projector with an MLP projector leads to a significant performance drop, highlighting the effectiveness of our proposed MoE speech projector in capturing fine-grained representations for CS speech.

Model	BLEU	COMET
Ours	39.52	81.33
w/o MoE Speech Projector	36.85	79.95
w/o $\mathcal{L}_{lang}$	39.05	80.88
w/o $\mathcal{L}_{balance}$	38.86	81.03
w/o $\mathcal{L}_{transition}$	39.21	81.24
conventional $\mathcal{L}'_{balance}$	38.68	80.79
w/o Stage 1	38.14	80.31
w/o Stage 3	37.74	80.02

Table 6: Ablation studies on the NTUML2021 CS test set.

- *w/o* $\mathcal{L}_{lang}$ *or* $\mathcal{L}_{balance}$. To validate the benefit of our carefully designed loss functions, we evaluate the variant in which $\mathcal{L}_{\text{lang}}$ or $\mathcal{L}_{\text{balance}}$ is excluded from Stages 2 and 3. As shown in Line 3 and 4, we find a notable performance degradation, indicating that these two auxiliary losses play an important role in guiding expert specialization as well as routing consistency.
- *w/o* $\mathcal{L}_{transition}$. We remove $\mathcal{L}_{\text{transition}}$ from Stages 3 and 4 to evaluate its impact. As illustrated in Line 5, the resulting performance decline suggests that the transition loss facilitates smoother adaptation across different domains.
- *conventional* $\mathcal{L}'_{balance}$. For a direct comparison, we replace our proposed $\mathcal{L}_{\text{balance}}$ with the conventional $\mathcal{L}'_{\text{balance}}$ introduced by [Fedus *et al.*, 2022]:

$$\mathcal{L}'_{\text{balance}} = \sum_{l=1}^{L} \sum_{i=1}^{N} f'_{il} \cdot P'_{il}, \tag{15}$$

where $f'_{il} = \frac{1}{T} \sum_T \mathbb{1}\{\arg\max_r \ p_r(h_l^t) = i\}$, and $P'_{il} = \frac{1}{T} \sum_T p_i(h_l^t)$.
This comparison allows us to assess how different balancing objectives affect expert utilization and training stability. Results in Line 6 show a slightly lower performance, suggesting that our customized balance loss better facilitates stable and effective expert utilization within our MoE framework.

Top-k	Fisher		NTUML2021	
	En	Es	En	zh-CN
Top-1	97.62%	95.66%	99.99%	85.21%
w/o $\mathcal{L}_{lang}$	54.42%	51.16%	45.72%	69.28%
Top-7	96.72%	95.76%	99.98%	81.03%
w/o $\mathcal{L}_{lang}$	51.06%	55.00%	45.66%	69.80%

Table 7: Proportion of speech tokens routed to the corresponding language expert group within the top-k experts.

- *w/o Stage 1.* For this variant, training starts from Stage 2 with all expert weights randomly initialized. As shown in Line 7, removing Stage 1 causes a noticeable decline in performance. In addition, we observe that the training loss decreases more slowly, suggesting that Stage 1 helps provide a better initialization for subsequent stages and contributes to improved final performance.
- *w/o Stage 3.* Under this variant, Stage 3 is omitted and Stage 4 is initialized using the weights from Stage 2. Moreover, the training pipeline is further simplified to a direct transition from monolingual ASR to CS speech translation data. As reported in Line 8, the notable performance degradation reveals a considerable task gap between ASR and CS speech translation, validating the necessity of Stage 3 for effective domain adaptation.

5.5 Analysis of Experts Assignment

To further investigate how the MoE projector adaptively routes speech tokens to appropriate experts, we conduct an analysis using monolingual speech data from the Common Voice corpus, as oracle language labels are unavailable in CS speech. We use English and Spanish speech for the model trained on Fisher, whereas for the model trained on NTUML2021, English and Chinese speech are used. Specifically, for each speech token, we record the ranking of experts assigned by the router and compute the proportion of top-k experts that belong to the corresponding language expert group. As shown in Table 7, the majority of speech tokens are routed to their corresponding language expert groups, demonstrating that the MoE projector effectively captures cross-lingual semantic distinctions. Specifically, the routing accuracy exceeds 95% for both English and Spanish tokens, while that for Chinese tokens is slightly lower but still above 80%, which is likely due to data imbalance across languages. Meanwhile, the proportion of correctly routed speech tokens for all languages drops significantly when $\mathcal{L}_{\text{lang}}$ is removed, indicating that $\mathcal{L}_{\text{lang}}$ plays a crucial role in guiding the router to learn appropriate expert assignments. In addition, we compute the expert ranks within the corresponding language expert group for each speech token, and report their mean and variance, which are 4.25 / 0.69 and 4.77 / 0.88 for the models trained on Fisher and NTUML2021, respectively. In contrast, these statistics become 4.14 / 1.05 and 4.67 / 1.21 when $\mathcal{L}_{\text{balance}}$ is excluded. These results indicate that $\mathcal{L}_{\text{balance}}$ substantially reduces the variance of correct expert ranks, thereby promoting more balanced routing.

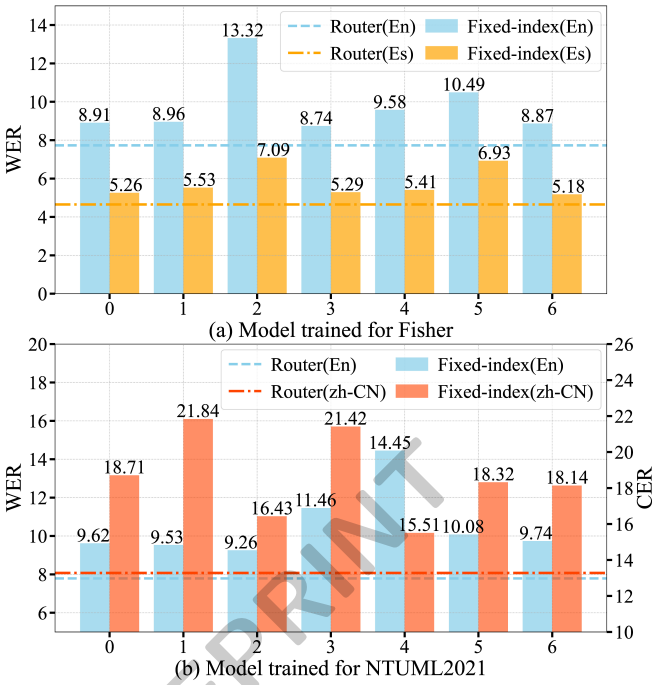

Figure 3: Comparison of WER↓(En, Es) and CER↓(zh-CN) scores for the router and fixed-index variants. The x-axis represents the expert index. Dashed lines and bars indicate the performance of the router and fixed-index variants, respectively, and different colors denote different languages in the ASR test sets.

5.6 Expert-Level Semantic Analysis

In this set of experiments, we further analyze the semantic comprehension capability of individual experts using ASR data. Specifically, we disable the router and assign all speech tokens to a single expert, iterating over all expert indices within the corresponding language expert group. As presented in Figure 3, enabling the router consistently achieves the best performance across datasets. Meanwhile, the performance of fixed-index experts does not degrade substantially when operating independently. These results suggest that different experts capture complementary aspects of semantic understanding, and that our router can effectively and adaptively route speech tokens to appropriate experts.

6 Conclusion and Future Work

In this work, we first explore LLMs in end-to-end CS speech translation, and achieve significant improvements in two key aspects. First, we propose a novel MoE speech projector, which enables capturing fine-grained representations of CS speech. Second, we propose a novel multi-stage training paradigm that effectively utilizes diverse data domains while ensuring smooth transitions across training stages. Extensive experiments and analyses verify the effectiveness of our model. For future work, we plan to extend our model to support a wider range of CS language pairs and to investigate more adaptive expert selection mechanisms. Besides, we aim to generalize our model to related CS tasks such as ASR and MT to further validate its robustness and generalizability.

Acknowledgments

This work was supported by the First Batch of Projects for the 2025 "Intergovernmental International Science, Technology and Innovation Cooperation" of the National Key Research and Development Program of China under Grant 2025YFE0121700, Natural Science Foundation of Fujian Province of China (No. 2024J011001), and the Public Technology Service Platform Project of Xiamen (No.3502Z20231043). We also thank the reviewers for their insightful comments.

Contribution Statement

Yan Gao and Yazheng Yang contributed equally.

References

[Agarap, 2018] Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.

[Alastruey *et al.*, 2023] Belen Alastruey, Matthias Sperber, Christian Gollan, Dominic Telaar, Tim Ng, and Aashish Agarwal. Towards real-world streaming speech translation for code-switched speech. In *Proceedings of the 6th Workshop on Computational Approaches to Linguistic Code-Switching*, 2023.

[Ardila *et al.*, 2020] Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. Common voice: A massively-multilingual speech corpus. In *Proc. of LREC*, 2020.

[Borisov *et al.*, 2025] Maksim Borisov, Zhanibek Kozhirbayev, and Valentin Malykh. Low-resource machine translation for code-switched Kazakh-Russian language pair. In *Proc. of NAACL (Volume 4: Student Research Workshop)*, 2025.

[Chen *et al.*, 2024] Xi Chen, Songyang Zhang, Qibing Bai, Kai Chen, and Satoshi Nakamura. LLaST: Improved end-to-end speech translation system leveraged by large language models. In *Proc. of ACL Findings*, 2024.

[Chi and Bell, 2022] Jie Chi and Peter Bell. Improving code-switched ASR with linguistic information. In *Proc. of COLING*, 2022.

[Cieri *et al.*, 2004] Christopher Cieri, David Miller, and Kevin Walker. The fisher corpus: a resource for the next generations of speech-to-text. In *Proc. of LREC*, 2004.

[Communication *et al.*, 2023] Seamless Communication, Loïc Barrault, Yu-An Chung, Mariano Cora Meglioli, David Dale, Ning Dong, Paul-Ambroise Duquenne, Hady Elsahar, Hongyu Gong, Kevin Heffernan, John Hoffman, Christopher Klaiber, Pengwei Li, Daniel Licht, Jean Maillard, Alice Rakotoarison, Kaushik Ram Sadagopan, Guillaume Wenzek, Ethan Ye, Bapi Akula, Peng-Jen Chen, Naji El Hachem, Brian Ellis, Gabriel Mejia Gonzalez, Justin Haaheim, Prangthip Hansanti, Russ Howes, Bernie Huang, Min-Jae Hwang, Hirofumi Inaguma, Somya Jain, Elahe Kalbassi, Amanda Kallet, Ilia Kulikov, Janice Lam, Daniel Li, Xutai Ma, Ruslan Mavlyutov, Benjamin Peloquin, Mohamed Ramadan, Abinesh Ramakrishnan, Anna Sun, Kevin Tran, Tuan Tran, Igor Tufanov, Vish Vogeti, Carleigh Wood, Yilin Yang, Bokai Yu, Pierre Andrews, Can Balioglu, Marta R. Costa-jussà, Onur Celebi, Maha Elbayad, Cynthia Gao, Francisco Guzmán, Justine Kao, Ann Lee, Alexandre Mourachko, Juan Pino, Sravya Popuri, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, Paden Tomasello, Changhan Wang, Jeff Wang, and Skyler Wang. Seamless: Multilingual expressive and streaming speech translation. *arXiv preprint arXiv:2312.05187*, 2023.

[Dhawan *et al.*, 2023] Kunal Dhawan, KDimating Rekesh, and Boris Ginsburg. Unified model for code-switching speech recognition and language identification based on concatenated tokenizer. In *Proceedings of the 6th Workshop on Computational Approaches to Linguistic Code-Switching*, 2023.

[Fedus *et al.*, 2022] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 2022.

[Gaser *et al.*, 2023] Marwa Gaser, Manuel Mager, Injy Hamed, Nizar Habash, Slim Abdennadher, and Ngoc Thang Vu. Exploring segmentation approaches for neural machine translation of code-switched Egyptian Arabic-English text. In *Proc. of EACL*, 2023.

[Gupta *et al.*, 2021] Abhirut Gupta, Aditya Vavre, and Sunita Sarawagi. Training data augmentation for code-mixed translation. In *Proc. of NAACL*, 2021.

[Gupta *et al.*, 2024] Ayushman Gupta, Akhil Bhogal, and Kripabandhu Ghosh. Code-mixer ya nahi: Novel approaches to measuring multilingual llms' code-mixing capabilities. *arXiv preprint arXiv:2410.11079*, 2024.

[Hinton and Roweis, 2002] Geoffrey E Hinton and Sam Roweis. Stochastic neighbor embedding. *Advances in neural information processing systems*, 15, 2002.

[Hu *et al.*, 2022] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *Proc. of ICLR*, 2022.

[Huber *et al.*, 2022] Christian Huber, Enes Yavuz Ugan, and Alexander Waibel. Code-switching without switching: Language agnostic end-to-end speech translation. *arXiv preprint arXiv:2210.01512*, 2022.

[Kronis *et al.*, 2024] Martins Kronis, Askars Salimbajevs, and Mārcis Pinnis. Code-mixed text augmentation for Latvian ASR. In *Proc. of LREC-COLING*, 2024.

[Li *et al.*, 2026] Yi Li, Rui Zhao, Ruiquan Zhang, Jinsong Su, Daimeng Wei, Min Zhang, and Yidong Chen. Plast: Towards paralinguistic-aware speech translation. In *Proc. of AAAI*, 2026.

[Mohamed *et al.*, 2025] Amr Mohamed, Yang Zhang, Michalis Vazirgiannis, and Guokan Shang. Lost in the mix: Evaluating llm understanding of code-switched text. *arXiv preprint arXiv:2506.14012*, 2025.

[Morris *et al.*, 2004] Andrew Cameron Morris, Viktoria Maier, and Phil D Green. From wer and ril to mer and wil: improved evaluation measures for connected speech recognition. In *Proc. of Interspeech*, 2004.

[Pengpun *et al.*, 2024] Parinthapat Pengpun, Krittamate Tiankanon, Amrest Chinkamol, Jiramet Kinchagawat, Pitchaya Chairuengjitjaras, Pasit Supholkhan, Pubordee Aussavavirojekul, Chiraphat Boonnag, Kanyakorn Veerakanjana, Hirunkul Phimsiri, et al. On creating an english-thai code-switched machine translation in medical domain. In *Proc. of EMNLP Findings*, 2024.

[Post, 2018] Matt Post. A call for clarity in reporting BLEU scores. In *Proc. of MT*, 2018.

[Radford *et al.*, 2023] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine Mcleavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *Proc. of ICML*, 2023.

[Rei *et al.*, 2020] Ricardo Rei, Craig Stewart, Ana C. Farinha, and Alon Lavie. COMET: A neural framework for MT evaluation. In *Proc. of EMNLP*, 2020.

[Scotton and Ury, 1977] Carol Myers Scotton and William Ury. Bilingual strategies: The social functions of code-switching. *International Journal of the Sociology of Language*, 1977.

[Shankar *et al.*, 2025] Bhavani Shankar, Preethi Jyothi, and Pushpak Bhattacharyya. CoSTA: Code-switched speech translation using aligned speech-text interleaving. In *Proc. of COLING*, 2025.

[Shazeer *et al.*, 2017] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *Proc. of ICLR*, 2017.

[Vavre *et al.*, 2022] Aditya Vavre, Abhirut Gupta, and Sunita Sarawagi. Adapting multilingual models for code-mixed translation. In *Proc. of EMNLP Findings*, 2022.

[Weller *et al.*, 2022] Orion Weller, Matthias Sperber, Telmo Pires, Hendra Setiawan, Christian Gollan, Dominic Telaar, and Matthias Paulik. End-to-end speech translation for code switched speech. In *Proc. of ACL Findings*, 2022.

[Winata *et al.*, 2020] Genta Indra Winata, Samuel Cahyawijaya, Zhaojiang Lin, Zihan Liu, Peng Xu, and Pascale Fung. Meta-transfer learning for code-switched speech recognition. In *Proc. of ACL*, 2020.

[Winata *et al.*, 2023] Genta Winata, Alham Fikri Aji, Zheng Xin Yong, and Thamar Solorio. The decades progress on code-switching research in NLP: A systematic survey on trends and challenges. In *Proc. of ACL Findings*, 2023.

[Wu *et al.*, 2025] Suhang Wu, Jialong Tang, Chengyi Yang, Pei Zhang, Baosong Yang, Junhui Li, Junfeng Yao, Min Zhang, and Jinsong Su. Locate-and-focus: Enhancing terminology translation in speech language models. In *Proc. of ACL*, 2025.

[Xie *et al.*, 2025] Peng Xie, Xingyuan Liu, Tsz Wai Chan, Yequan Bie, Yangqiu Song, Yang Wang, Hao Chen, and Kani Chen. Switchlingua: The first large-scale multilingual and multi-ethnic code-switching dataset. *arXiv preprint arXiv:2506.00087*, 2025.

[Xu and Yvon, 2021] Jitao Xu and François Yvon. Can you traducir this? machine translation for code-switched input. In *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*, 2021.

[Yang *et al.*, 2024] Chih-Kai Yang, Kuan-Po Huang, Ke-Han Lu, Chun-Yi Kuan, Chi-Yuan Hsiao, and Hung-yi Lee. Investigating zero-shot generalizability on mandarin-english code-switched asr and speech-to-text translation of recent foundation models with self-supervision and weak supervision. In *Proc. of ICASSP Workshops*, 2024.

[Zhang *et al.*, 2023] Ruochen Zhang, Samuel Cahyawijaya, Jan Christian Blaise Cruz, Genta Indra Winata, and Alham Fikri Aji. Multilingual large language models are not (yet) code-switchers. In *Proc. of EMNLP*, 2023.

[Zhang *et al.*, 2024a] Liang Zhang, Zhen Yang, Biao Fu, Ziyao Lu, Liangying Shao, Shiyu Liu, Fandong Meng, Jie Zhou, Xiaoli Wang, and Jinsong Su. Multi-level cross-modal alignment for speech relation extraction. In *Proc. of EMNLP*, 2024.

[Zhang *et al.*, 2024b] Wenbo Zhang, Aditya Majumdar, and Amulya Yadav. Chai for llms: Improving code-mixed translation in large language models through reinforcement learning with ai feedback. *arXiv preprint arXiv:2411.09073*, 2024.