

Align, Integrate, and Fire: Efficient Token-Level Alignment for Zero-Shot SpeechLLMs

Abderrahmane Issam Yusuf Can Semerci Jan Scholtes Gerasimos Spanakis

Department of Advanced Computing Sciences

Maastricht University

{abderrahmane.issam, y.semerci, j.scholtes, jerry.spanakis}@maastrichtuniversity.nl

Abstract

While Large Language Models excel in natural language processing, efficiently extending their capabilities to spoken input remains a significant challenge. Existing methods for building SpeechLLMs often rely on computationally expensive full-model fine-tuning, or employ parameter-efficient projectors that suffer from inefficient token sequence lengths and costly full-model supervision. In this paper, we introduce Aligned Continuous Integrate-and-Fire, a highly efficient framework for zero-shot speech processing. Our method dynamically compresses continuous acoustic frames into the exact discrete token length of the target text utilizing explicit Dynamic Time Warping alignments. This allows our initial training stage to establish a robust acoustic-to-semantic bridge using lightweight distance metrics, entirely bypassing the computationally expensive LLM forward pass. For subsequent fine-tuning, we propose a memory-efficient knowledge distillation objective that targets a single LLM layer, performing competitively with full-model cross-entropy training at a fraction of the computational cost. Through extensive evaluations on Automatic Speech Recognition and Speech Translation, we demonstrate that our method achieves superior performance compared to prior parameter-efficient baselines.¹

1 Introduction

Large Language Models (LLMs) have achieved unprecedented success in natural language processing (OpenAI et al., 2024; Comanici et al., 2025; Grattafiori et al., 2024; Qwen et al., 2025; Guo et al., 2025), yet a significant portion of high-performing open-weights LLMs remain text-only. While cascaded systems (speech recognition model followed by an LLM) offer a straightforward way to process spoken input, they suffer from error propagation and prevent end-to-end downstream fine-

tuning. To natively equip LLMs with auditory capabilities, recent works connect a speech encoder to the LLM and fine-tune the system on speech-to-text data (Wu et al., 2023; Gong et al., 2024; Tang et al., 2024; Fathullah et al., 2024a; Hu et al., 2024; Das et al., 2025). Although effective, this paradigm is computationally expensive and prone to task-specific overfitting (Tang et al., 2024).

To improve efficiency, recent approaches freeze the LLM and train only a speech projector to map acoustic features to text embeddings (Fathullah et al., 2024b; Deng et al., 2025; Tan et al., 2025; Mohapatra et al., 2026). However, these methods still face significant limitations. Wav2Prompt (Deng et al., 2025) dynamically matches token lengths using a Continuous Integrate-and-Fire (CIF) module (Dong and Xu, 2020) but relies on expensive Cross Entropy (CE) supervision from the full LLM. SSR Tan et al. (2025) requires external alignment tools and extensive distillation training followed by full LLM fine-tuning. SpeechMapper (Mohapatra et al., 2026) avoids LLM forward passes but pads text embeddings to match the longer length of speech sequences, wasting computational resources on uninformative tokens and hindering both training and inference efficiency.

To overcome these limitations, we introduce ACIF (Aligned Continuous Integrate-and-Fire), a highly efficient framework for zero-shot speech processing that is trained exclusively on Automatic Speech Recognition (ASR) data. ACIF dynamically compresses continuous acoustic frames into the exact discrete token length of the target text. During our initial training phase (Stage 1), we supervise the CIF module using explicit token-level alignments generated via Dynamic Time Warping (DTW) (Sakoe and Chiba, 1978) directly from the speech and text embeddings. This allows the projector to learn a robust acoustic-to-semantic mapping through lightweight distance metrics (Mean Squared Error and cosine distance), bypassing the

¹Our code: <https://github.com/issam9/ACIF>

computationally expensive LLM forward pass.

For Stage 2 fine-tuning, standard full-model CE training on ASR data risks overfitting and incurs heavy computational costs. To address this, we propose a memory-efficient Knowledge Distillation (KD) objective that distills hidden states from a single LLM layer, performing competitively with full-model CE at a fraction of the cost. Through an analysis of KD layer depth, we reveal that ASR relies heavily on shallow, localized representations optimally captured at the first layer, whereas Speech Translation (ST) benefits from the abstract semantic representations developed in intermediate layers.

We evaluate ACIF on Automatic Speech Recognition (LibriSpeech, VoxPopuli, FLEURS) and zero-shot Speech Translation (Europarl-ST, CoVoST-2) using Llama 3.1 (8B) and Qwen 2.5 (7B) backbones. Our main contributions are:

- We propose ACIF, combining a CIF module with DTW-guided supervision to project continuous speech frames into exact LLM token representations.
- We demonstrate that our Stage 1 training establishes a high-quality semantic bridge with vastly greater efficiency than existing baselines, requiring no LLM forward passes.
- We introduce a highly memory-efficient, single-layer KD objective for Stage 2 fine-tuning that competes robustly with full-model CE training.
- We achieve superior zero-shot performance compared to prior methods, offering a substantially more lightweight and computationally efficient alternative for developing Speech-LLMs.

2 Related Works

2.1 Speech Large Language Models

While recent LLMs natively support multi-modal inputs (OpenAI et al., 2024; Comanici et al., 2025), a large portion of modern LLMs remain text-only, making the development of efficient audio adapters crucial. Previous works have equipped LLMs with auditory capabilities by connecting a speech encoder and fine-tuning the full model or utilizing LoRA adapters on task-specific datasets (Wu et al.,

2023; Gong et al., 2024; Tang et al., 2024; Fathullah et al., 2024a; Hu et al., 2024; Das et al., 2025). Recent parameter-efficient approaches have moved beyond this by training only a lightweight projector and keeping the LLM entirely frozen (Fathullah et al., 2024b; Deng et al., 2025; Tan et al., 2025; Mohapatra et al., 2026). Closely related to our approach, Deng et al. (2025) introduced Wav2Prompt, which converts speech frames to text tokens using a CIF mechanism (Dong and Xu, 2020). Although this successfully avoids updating the LLM parameters, it still relies on CE supervision from the LLM during training, introducing significant computational overhead. Alternatively, Mohapatra et al. (2026) proposed SpeechMapper, which makes the LLM forward pass optional. However, SpeechMapper resolves the length mismatch between modalities by introducing padding tokens to the text embeddings, which renders training and inference inefficient. In contrast, our work trains a CIF module to project speech frames to the exact text token length. Furthermore, unlike Wav2Prompt which requires full LLM propagation for training, we supervise this projection using explicit token-level alignments generated via DTW.

2.2 Bridging the Modality Gap between Speech and Text

The goal of aligning acoustic and semantic representations is widely studied under the framework of bridging the modality gap. In speech-to-text tasks, representation misalignment has been shown to severely bottleneck performance (Liu et al., 2020; Wang et al., 2020). Various alignment strategies have proven effective to alleviate this issue (Inaguma et al., 2021; Ye et al., 2022; Fang et al., 2022; Ouyang et al., 2023; Zhang et al., 2025), with techniques relying on explicit word- or token-level boundaries yielding particularly strong results (Fang et al., 2022; Ouyang et al., 2023; Issam et al., 2025). Moving beyond static, pre-computed alignments, Issam et al. (2025) utilized DTW (Sakoe and Chiba, 1978) to dynamically generate alignments during the training of end-to-end speech translation models. We similarly rely on DTW to establish correspondence between acoustic and text representations. However, rather than training a task-specific encoder-decoder architecture with simple mean pooling, we project speech into the embedding space of a frozen LLM to specifically target zero-shot generalization. Additionally, we utilize trainable CIF weights to dynamically pool

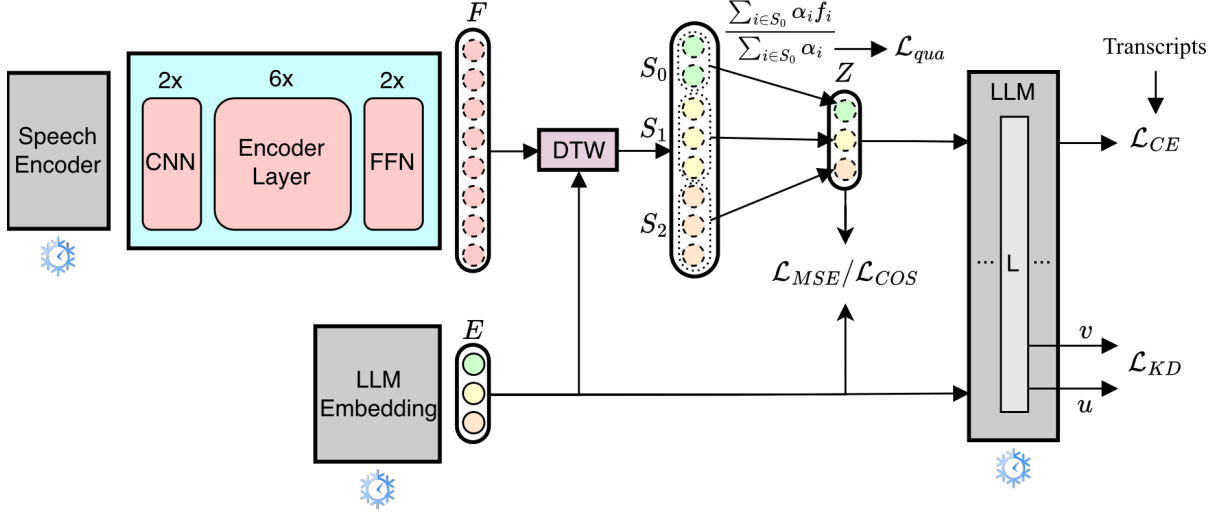


Figure 1: **Overview of the ACIF architecture.** The proposed projector consists of two convolutional layers, a 6-layer Transformer encoder, and a two-layer feed-forward network. The projected speech frames F are aligned with the target text embeddings E using DTW. An additional layer predicts continuous weights α_i to integrate each frame f_i into its corresponding aligned token (Eq. 1). The resulting token-level speech embeddings Z are optimized using normalized MSE ($\mathcal{L}_{MSE}$, Eq. 2) and mean-centered cosine distance ($\mathcal{L}_{COS}$, Eq. 3). Simultaneously, the α_i values are supervised via a quantity loss ($\mathcal{L}_{qua}$, Eq. 4) to ensure they sum to 1 over each text token’s segment. During the fine-tuning stage, we incorporate either a standard cross-entropy loss ($\mathcal{L}_{CE}$) or a knowledge distillation loss ($\mathcal{L}_{KD}$), the latter being computed between the intermediate hidden states of the text and speech embeddings extracted at layer L of the truncated LLM.

and integrate the continuous speech frames into exact, discrete token representations during training and inference.

3 Methodology

In this section, we present our methodology for efficiently bridging a pretrained speech encoder with a frozen LLM, as illustrated in Figure 1. We first introduce the ACIF projector and its training objectives (Section 3.1). To further optimize the projector, Section 3.2 details a memory-efficient fine-tuning strategy. Finally, Section 3.3 outlines how the trained projector dynamically integrates speech embeddings during inference.

3.1 ACIF: Aligned Continuous Integrate and Fire

Connecting a pretrained speech encoder to a frozen LLM requires a bridging projector that resolves the cross-modal gap. To enable the direct injection of speech representations into the LLM context, the projector must map acoustic features to the LLM embedding dimension, compress dense speech frame sequences to match the exact length of textual token sequences, and align the projected frames directly into the semantic embedding space of the LLM. To achieve this multi-step mapping

efficiently, we design a lightweight projector architecture coupled with a dynamic alignment strategy.

Our projector consists of two 1D Convolutional Neural Network layers that perform initial temporal downsampling, and a 6-layer encoder that learns semantic alignment from speech to text. This is followed by two linear layers that map the downsampled features into the LLM hidden dimension, producing intermediate speech frame representations $F = [f_1, f_2, \dots, f_{N'}] \in \mathbb{R}^{N' \times d}$, where N' is the reduced sequence length. Concurrently, a single linear layer operates on these frames to predict a scalar weight $\alpha_i \in [0, 1]$ for each frame, representing its relative contribution toward forming a discrete token embedding.

During training, we compute an optimal alignment path between the intermediate frames F and the target text token embeddings $E = [e_1, e_2, \dots, e_M] \in \mathbb{R}^{M \times d}$ using DTW-Align (Issam et al., 2025) based on cosine distance. DTW partitions the N' frames into M contiguous segments, assigning a set of frames S_j to each target token j . Instead of uniformly averaging the segment embeddings as in standard DTW-Align, we calculate a weighted average using the predicted alpha scores. For each target token j , the pooled representation z_j is computed by multiplying the

aligned frame embeddings by their corresponding scalar weights and dividing the accumulated result by the sum of those weights:

$$z_j = \frac{\sum_{i \in S_j} \alpha_i f_i}{\sum_{i \in S_j} \alpha_i} \quad (1)$$

This token-level integration receives explicit positional signals from the alignment bounds itself, entirely eliminating the requirement for LLM forward passes to regularize the token compression.

The projector is optimized end-to-end using a joint objective comprising normalized MSE, mean-centered cosine distance, and a quantity loss. To ensure the MSE remains stable across different LLM backbones with varying embedding magnitudes, we divide the raw error by the mean squared magnitude of the target LLM embeddings. Given the pooled speech embeddings $Z = [z_1, \dots, z_M]$ and target text embeddings $E = [e_1, \dots, e_M]$, the normalized MSE is computed as:

$$\mathcal{L}_{\text{MSE}} = \frac{\sum_{j=1}^M \|z_j - e_j\|_2^2}{\sum_{j=1}^M \|e_j\|_2^2} \quad (2)$$

Furthermore, raw cosine similarity between embeddings can be inflated by a shared mean vector (Mu and Viswanath, 2018), which might dominate the pairwise similarity and obscure the semantic signal we aim to optimize for. We mitigate this by mean-centering the embeddings prior to computing cosine similarity, subtracting the mean of the target embeddings, $\mu_E = \frac{1}{M} \sum_{j=1}^M e_j$, from both the predicted and target vectors. This removes the dominant shared direction and yields a cosine loss that is more sensitive to true semantic discrepancies:

$$\mathcal{L}_{\text{cos}} = \frac{1}{M} \sum_{j=1}^M \left(1 - \cos(z_j - \mu_E, e_j - \mu_E) \right) \quad (3)$$

where $\cos(\cdot, \cdot)$ denotes the cosine similarity operator.

Finally, the quantity loss applies an L_1 penalty between the accumulated scalar weights for each token and a target value of 1, enforcing that the predicted weights correctly match the exact target sequence length:

$$\mathcal{L}_{\text{qua}} = \frac{1}{M} \sum_{j=1}^M \left| \sum_{i \in S_j} \alpha_i - 1 \right| \quad (4)$$

The overall objective is the weighted sum of these three components, $\mathcal{L}_{\text{total}} = \lambda_{\text{mse}} \mathcal{L}_{\text{MSE}} + \lambda_{\text{cos}} \mathcal{L}_{\text{cos}} + \lambda_{\text{qua}} \mathcal{L}_{\text{qua}}$.

3.2 Efficient Fine-tuning

Our projector can be efficiently fine-tuned using feedback from the LLM to further improve cross-modal embedding alignment. While the LLM itself could be jointly fine-tuned for downstream speech-to-text tasks, we freeze the LLM to preserve its general capabilities and prioritize computational efficiency. To fine-tune the projector, a standard approach is to incorporate a Cross Entropy loss alongside the alignment objectives (Deng et al., 2025; Mohapatra et al., 2026):

$$\mathcal{L}_{\text{total}} = \lambda_{\text{mse}} \mathcal{L}_{\text{MSE}} + \lambda_{\text{cos}} \mathcal{L}_{\text{cos}} + \lambda_{\text{qua}} \mathcal{L}_{\text{qua}} + \lambda_{\text{ce}} \mathcal{L}_{\text{CE}} \quad (5)$$

where λ_{ce} governs the weight of the autoregressive language modeling loss.

However, while optimizing only the projector minimizes the number of trainable parameters, computing $\mathcal{L}_{\text{CE}}$ still requires a full forward pass through the multibillion-parameter LLM. This imposes a severe GPU memory bottleneck during training. We hypothesize that for zero-shot performance, the primary requirement is successfully mapping speech into the LLM’s initial semantic space; thus, full-depth LLM propagation is unnecessary, and shallow feedback is sufficient to guide the representations.

Guided by this assumption, we propose an efficient alternative to standard CE loss by utilizing early-layer LLM representations (e.g., 1 layer) for fine-tuning. To enforce deeper semantic alignment without the full memory overhead, we introduce a KD objective utilizing the logit lens technique (nostalgebraist, 2020).

Specifically, the pretrained LLM is structurally pruned during training to retain only the first L transformer layers. Both the target text embeddings E and the pooled speech embeddings Z are passed through this truncated network to extract their respective intermediate hidden states. These states are then mapped directly into the vocabulary space by applying the LLM’s unembedding layer, yielding teacher text logits $\mathbf{u}$ and student speech logits $\mathbf{v}$.

The distillation loss $\mathcal{L}_{\text{KD}}$ is computed as the average Kullback-Leibler (KL) divergence between the teacher and student probability distributions over the vocabulary:

$$\mathcal{L}_{\text{KD}} = \frac{1}{M} \sum_{j=1}^M D_{\text{KL}} \left(\text{softmax}(\mathbf{u}_j) \parallel \text{softmax}(\mathbf{v}_j) \right) \quad (6)$$

where M is the number of valid target tokens. In configurations where memory efficiency is paramount, $\mathcal{L}_{KD}$ safely replaces $\mathcal{L}_{CE}$, regularizing the projector by forcing the intermediate speech representations to mirror the predictive trajectory of the text embeddings at a fraction of the computational cost.

3.3 Inference Stage

At inference time, since target transcriptions are unavailable and DTW alignment cannot be applied, the CIF module dynamically collapses the variable-length sequence of speech frame embeddings into discrete token representations. It sequentially aggregates frame embeddings scaled by their predicted scalar weights, progressively accumulating acoustic evidence until the predefined firing threshold of 1 is reached. This threshold signifies the completion of a semantic token. To prevent information loss at token boundaries, the specific frame that breaches the threshold is fractionally partitioned: a remainder portion completes the current token embedding, while the surplus is carried over to initialize the subsequent token. Once all frames are processed, the module resolves any residual acoustic information left in the tail buffer. If the accumulated tail weight exceeds a half-token threshold ($w \geq 0.5$), it is deemed semantically meaningful and emitted as a valid final token. Crucially, this trailing embedding is divided by its accumulated weight to yield a weighted mean, normalizing its magnitude to match that of fully integrated tokens.

4 Experiments

4.1 Datasets

For model training, we exclusively utilize the LibriSpeech dataset (Panayotov et al., 2015), leveraging its paired audio and text transcriptions. We subsequently evaluate ASR performance on the standard LibriSpeech, Voxpopuli (Wang et al., 2021a) and FLEURS (Conneau et al., 2022) test splits. To assess zero-shot ST capabilities, we report results on select test sets from the Europarl-ST (Iranzo-Sánchez et al., 2020) and CoVoST-2 (Wang et al., 2021b) benchmarks. Specifically, we evaluate on the English-to-Spanish, German, Italian, and French directions from Europarl-ST, alongside the English-to-German and Chinese directions from CoVoST-2. Although our proposed alignment method is language-agnostic and applicable to any source language with sufficient ASR data,

we restrict our evaluation to English-centric pairs to ensure direct comparison with prior work.

4.2 Backbones

For our acoustic representations, we use SeamlessM4T-v2-Large (Communication et al., 2023) as a speech encoder. Following SpeechMapper (Mohapatra et al., 2026), we specifically extract features from the 24th layer of the SeamlessM4T encoder. For the core language model, we experiment with Llama-3.1-8B-Instruct (Grattafiori et al., 2024) and Qwen2.5-7B-Instruct (Qwen et al., 2025). To preserve their pretrained capabilities and ensure computational efficiency, the weights of both the speech encoders and the LLMs remain strictly frozen during training.

4.3 Training Details

To optimize GPU memory and accelerate the training process, we pre-compute the acoustic features from the frozen speech encoder. Consequently, only the trainable projector and the frozen LLM embedding layer need to be loaded into memory. Target text transcriptions are normalized using the MMS normalization (Pratap et al., 2024). Specifically, we remove HTML tags, apply NFKC normalization, lowercase, and remove punctuation. Architecturally, the projector consists of two 1D convolutional layers (kernel size 5, stride 2), which reduce the temporal sequence length by a factor of 4. These are followed by 6 standard Transformer encoder layers with a hidden dimension of 1024. A final two-layer feed-forward network projects the representations through an intermediate dimension of 2048 into the specific LLM hidden dimension (4096 for Llama-3.1-8B and 3584 for Qwen2.5-7B).

We optimize the model using AdamW with a peak learning rate of 1×10^{-4} and a cosine learning rate scheduler, incorporating a 10% linear warmup period. Training utilizes dynamic batching with a maximum batch size of 8192 speech frames and runs for 250k steps. This effectively matches the total volume of training data seen in SpeechMapper (Mohapatra et al., 2026), adjusting for their comparatively smaller batch size. During the subsequent fine-tuning stage, we maintain identical hyperparameters but reduce the learning rate to 1×10^{-5} and the batch size to 1024 tokens and train for 20k steps. We strictly evaluate the final saved checkpoint in both stages.

During the initial training phase, the weights

for the loss components are set to $\lambda_{\text{mse}} = 0.01$, $\lambda_{\text{cos}} = 10.0$, and $\lambda_{\text{qua}} = 1.0$.² For the subsequent fine-tuning stage, we maintain these base hyperparameters while introducing the additional semantic objective: either standard cross-entropy ($\mathcal{L}_{\text{CE}}$) with $\lambda_{\text{ce}} = 1.0$, or knowledge distillation ($\mathcal{L}_{\text{KD}}$) with $\lambda_{\text{kd}} = 1.0$. When utilizing $\mathcal{L}_{\text{KD}}$, the pretrained LLM is pruned to retain only its first transformer layer.

Our models are implemented using PyTorch Lightning³. Due to the lightweight nature of the projector—comprising 96.6M trainable parameters—and our memory-efficient design, the entire training pipeline fits easily on a single 40GB A100 GPU. This low memory footprint even permits the concurrent training of multiple models on the same hardware. Training completes in approximately 6h30min on a single A100 GPU (or 10h30min on an NVIDIA RTX A5000), representing a drastic reduction in computational overhead compared to prior methods like SpeechMapper, which requires 4 days across 4 V100 GPUs.

4.4 Evaluation

For ASR evaluation, we report the Word Error Rate (WER) and Character Error Rate (CER) following the application of MMS text normalization (Pratap et al., 2024). For Speech Translation, we utilize the COMET⁴ metric (Rei et al., 2022), which evaluates the generated hypothesis against both the source transcription and the reference translation. Additionally, we report case-sensitive, detokenized BLEU (Papineni et al., 2002) computed via SacreBLEU (Post, 2018) in our code repository. During the evaluation of the CoVoST-2 dataset, we filter out audio samples containing fewer than 1,000 or more than 480,000 acoustic frames. To ensure a fair comparison, we adopt closely identical prompt templates to SpeechMapper (Mohapatra et al., 2026) (see Appendix A). All text generation is performed using greedy decoding with a maximum generation length of 150 tokens, implemented via the HuggingFace Transformers library (Wolf et al., 2020).

²The disparity between λ_{cos} and λ_{mse} balances their initial gradient magnitudes: while the cosine loss is naturally bounded in $[0, 2]$, MSE between the randomly initialized projector output and the pretrained LLM embedding space is initially much larger and would otherwise dominate early optimization.

³<https://github.com/Lightning-AI/pytorch-lightning>

⁴Unbabel/wmt22-comet-da

4.5 Baselines

Our primary baseline is SpeechMapper (Mohapatra et al., 2026), which similarly employs a two-stage training pipeline to achieve zero-shot speech capabilities. Their speech projector is architecturally heavier, consisting of two sequential blocks, each containing a single CNN layer and a 6-layer Transformer encoder. Within this structure, a linear layer in the first block projects the acoustic features to a hidden dimension of 2048, and a subsequent linear layer in the second block projects them to 4096, before a final linear layer maps the representations into the LLM’s input space. Consequently, the SpeechMapper projector contains 277M trainable parameters, making it nearly three times larger than our lightweight 96.6M-parameter module. For a direct and fair comparison, we benchmark exclusively against their zero-shot experimental setup. This pipeline consists of Stage 1, which trains the projector independently without any forward passes through the LLM, and Stage 2, which incorporates the frozen LLM to fine-tune the projector using a cross-entropy objective.

5 Results

5.1 ASR Performance

Table 1 presents the ASR performance evaluated via WER and CER on the LibriSpeech (test-clean, test-other), VoxPopuli (VP), and FLEURS datasets. Our proposed ACIF alignment framework demonstrates significant improvements over the SpeechMapper baseline across multiple dimensions.

Most notably, our method exhibits vastly superior sample efficiency and alignment quality during the initial modality-bridging phase. Using the identical Llama 3.1 backbone, our Stage 1 model achieves a WER of 5.1 on the LibriSpeech clean split, nearly halving the 9.4 WER reported by SpeechMapper’s Stage 1. This massive performance gap extends to the more challenging out-of-domain VP and FLEURS datasets, where our Stage 1 model improves upon SpeechMapper’s Stage 1 WER by absolute margins of 8.8 and 13.0 points, respectively. This indicates that our alignment-guided CIF module establishes a highly accurate semantic mapping purely from the acoustic representations, without requiring expensive forward passes through the LLM.

Furthermore, during the Stage 2 fine-tuning phase, our KD objective consistently outperforms

		LS clean	LS other	VP	Fleurs
	Seamless ASR	2.7 / 0.9	5.1 / 2.0	8.9 / 6.2	8.1 / 4.7
Llama 3.1	SpeechMapper Stage 1	9.4 / 6.5	12.0 / 7.9	25.0 / 19.7	30.2 / 27.6
	SpeechMapper Stage 2	4.9 / 2.7	7.8 / 4.1	<u>14.8</u> / 9.2	<u>16.6</u> / 11.1
Llama 3.1	ACIF (stage 1)	5.1 / 3.2	9.1 / 5.8	16.2 / 11.5	17.2 / 12.1
	ACIF (stage 2) [CE]	4.7 / 2.8	7.9 / 4.4	16.3 / 11.4	18.3 / 11.7
	ACIF (stage 2) [KD]	4.1 / 2.2	8.3 / 4.8	<u>14.8</u> / 9.7	17.4 / 11.7
Qwen 2.5	ACIF (stage 1)	<u>3.7</u> / <u>1.7</u>	<u>7.3</u> / <u>3.7</u>	15.2 / 9.2	17.9 / 10.4
	ACIF (stage 2) [CE]	4.9 / 2.5	9.1 / 4.8	18.3 / 12.0	21.6 / 13.5
	ACIF (stage 2) [KD]	3.8 / 1.8	<u>7.2</u> / <u>3.7</u>	15.0 / <u>9.1</u>	17.2 / <u>10.1</u>

Table 1: ASR performance (WER / CER) on LibriSpeech, VoxPopuli (VP), and FLEURS. The overall best results are highlighted in bold, while the highest SpeechLLM results are underlined. Baseline results for SpeechMapper are taken directly from the original paper (Mohapatra et al., 2026).

standard CE. For Llama 3.1, KD training yields a WER of 4.1, surpassing SpeechMapper’s fully fine-tuned Stage 2 result (4.9 WER) on LibriSpeech test-clean. The stabilizing effect of the KD loss becomes especially critical when employing the Qwen 2.5 backbone. While standard CE fine-tuning leads to a noticeable degradation in performance compared to its Stage 1 baseline (e.g., WER increasing from 3.7 to 4.9 on test-clean), the KD objective successfully prevents this. Notably, applying KD safely maintains the performance reached during Qwen 2.5’s Stage 1, yielding the best overall CERs on both VP (9.1) and FLEURS (10.1). Ultimately, these findings demonstrate that employing KD with only a single LLM layer is a highly robust, memory-efficient alternative to full-model CE supervision.

Finally, our best-performing configuration surpasses the zero-shot baseline of SSR (Tan et al., 2025) on both LibriSpeech evaluation splits. Specifically, our ACIF (Stage 2) [KD] model with a Qwen-2.5 backbone achieves 3.8 and 7.2 WER on test-clean and test-other, respectively, outperforming SSR’s 5.0 and 8.1.

5.2 Zero-Shot Speech Translation Performance

Table 2 presents the zero-shot ST COMET scores on the Europarl-ST and CoVoST-2 evaluation sets. Consistent with our ASR findings, our ACIF method significantly outperforms the SpeechMapper baseline. Using the Llama 3.1 backbone, our Stage 1 model achieves substantial gains over SpeechMapper’s Stage 1 across nearly all eval-

uation directions, with the exception of En-It. This improvement is particularly pronounced on the CoVoST-2 dataset, where our method outperforms the baseline by +12.1 COMET points on En-De and +11.9 points on En-Zh. Furthermore, while SpeechMapper suffers from performance degradation during its Stage 2 fine-tuning, our Stage 2 training consistently enhances the learned representations across both LLM backbones.

When comparing the Stage 2 optimization strategies, the results indicate that both CE and KD provide robust semantic alignment for zero-shot ST. For the Llama 3.1 backbone, KD yields the highest zero-shot performance on En-Es, En-Fr, and En-It directions, peaking at a COMET score of 79.1 on En-Es. Conversely, standard CE fine-tuning proves highly effective on the CoVoST-2 dataset, achieving the top zero-shot scores of 81.1 on En-De and 83.0 on En-Zh. Strikingly, unlike the representation degradation observed during Qwen 2.5 ASR fine-tuning, both CE and KD successfully stabilize and even slightly improve upon the Qwen 2.5 Stage 1 translation baseline. Ultimately, our best zero-shot configurations establish a highly effective semantic bridge. They closely trail the performance of the explicitly supervised cascaded baselines and consistently outperform the in-domain Seamless ST model on the Europarl-ST directions without requiring any translated text during training.

Finally, in comparison to Wav2prompt (Deng et al., 2025), which only reports BLEU on En-Es and En-Fr, we re-evaluate ACIF (Stage 2) [KD] with Llama 3.1 backbone using their exact decoding configuration (beam size of 5 and repetition penalty of 1.5). Under these settings, our method achieves 26.7 BLEU on En-Es and 21.0 BLEU on En-Fr, compared to 25.1 and 21.7, respectively. Notably, our approach performs competitively without requiring full LLM forward propagation or the 10 hours of in-domain Europarl-ST supervision utilized in Wav2prompt’s ASR stage.

6 Analysis

6.1 Some Embeddings are Easier to Align

The geometry of the target embedding space may influence how easily a speech projector can learn the cross-modal mapping. Consequently, training the projector with identical hyperparameters across different LLM backbones may be suboptimal. In Figure 2, we evaluate the impact of training duration by varying the number of training steps for

	Europarl				CoVoST2	
	en-es	en-fr	en-de	en-it	en-de	en-zh
Transcripts + Qwen 2.5 7B (topline)	83.53	79.87	80.55	79.2	83.3	87.3
Transcripts + Llama 3.1 8B (topline)	84.4	80.92	82.87	81.03	85.9	87.1
Seamless + Qwen 2.5 7B (Cascaded)	81.3	77.6	78.3	77.6	81.4	85.8
Seamless + Llama 3.1 8B (Cascaded)	82.2	78.7	80.9	79.1	84.1	85.6
Seamless ST (in-domain)	78.7	72.5	68.6	72.8	86.0	83.7
SpeechMapper (stage 1) (zero-shot)	76.4	73.9	72.3	76.8	67.1	69.3
SpeechMapper (stage 2) (zero-shot)	74.7 ± 2.7	71.0 ± 2.8	66.4 ± 2.6	73.2 ± 2.6	63.7 ± 1.0	68.6 ± 1.5
ACIF (stage 1) (zero-shot)	78.3	75.7	77.8	76.1	79.2	81.2
ACIF (stage 2) [CE] (zero-shot)	78.9	76.2	<u>78.6</u>	76.9	<u>81.1</u>	<u>83.0</u>
ACIF (stage 2) [KD] (zero-shot)	<u>79.1</u>	<u>76.3</u>	78.4	<u>77.1</u>	80.9	82.8
Qwen 2.5	76.6	73.3	75.8	73.9	78.6	82.4
	77.4	73.8	75.9	74.1	78.3	82.9
	77.0	73.7	76.2	74.3	78.6	82.4

Table 2: Speech translation performance (COMET scores) on the Europarl-ST and CoVoST-2 datasets. The overall best results are highlighted in bold, while the highest-performing zero-shot configurations are underlined. Baseline results for SpeechMapper are taken directly from the original paper (Mohapatra et al., 2026).

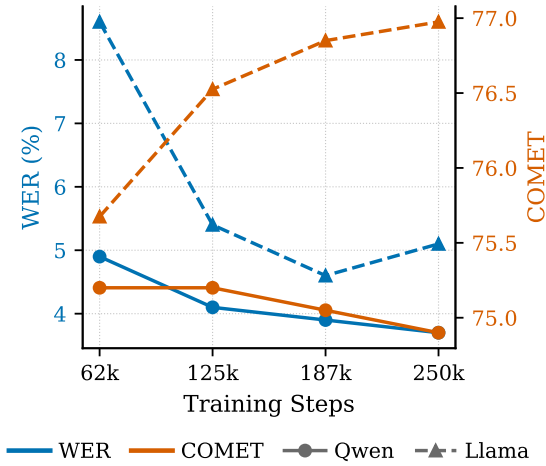


Figure 2: Impact of projector training duration on LibriSpeech test-clean (WER) and average Europarl-ST (COMET) performance. The Qwen 2.5 backbone achieves optimal alignment rapidly, while the performance of Llama 3.1 continues to steadily improve.

both Llama 3.1 and Qwen 2.5 backbones. We report WER on the LibriSpeech test-clean split and average COMET scores across Europarl-ST directions. The results highlight a clear difference in learning dynamics between the two models. While the projector aligned with Qwen achieves most of its performance early in training (within a quarter of the total steps), Llama steadily improves over longer training durations. This indicates that Qwen’s target space is learned much faster by the projector.

Prior work has shown that pretrained language representations exhibit highly anisotropic structure and considerable redundancy, with much of the variance concentrated in a relatively small number

of principal directions (Mu and Viswanath, 2018; Raunak et al., 2019). Such low effective dimensionality has been argued to simplify optimization in downstream learning tasks (Aghajanyan et al., 2021). Motivated by these observations, we examine whether differences in the geometry of the target embedding space can explain the projector learning dynamics observed above.

Figure 3 shows the cumulative variance explained by the principal components for both models, computed from over one million token embeddings extracted from the LibriSpeech training set. Qwen 2.5 exhibits a noticeably steeper eigenvalue decay than Llama 3.1. Specifically, Qwen requires only 857 principal components to explain 90% of the variance, whereas Llama requires 1,987 (more than twice as many), indicating that the variance of Qwen embeddings is concentrated in considerably fewer directions.

These geometric differences offer a plausible explanation for the training efficiency observed in Figure 2. The more concentrated variance spectrum of the Qwen embedding space suggests that its target representations are easier for the speech projector to learn, which is consistent with the substantially faster convergence observed during training. These results suggest that projector optimization is influenced not only by model size or architecture, but also by the geometry of the target embedding space. Consequently, training schedules and optimization hyperparameters should be adapted to the effective complexity of the target representation rather than transferred unchanged across LLM backbones.

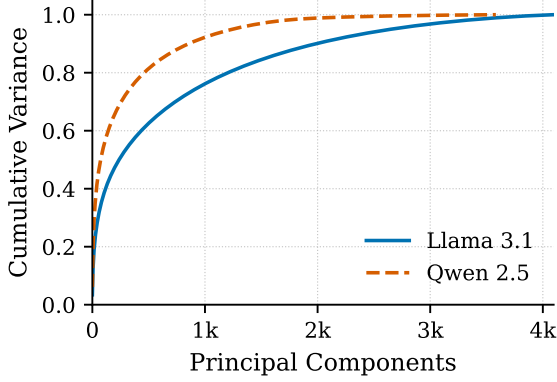


Figure 3: PCA eigenvalue decay of the target embedding spaces for Llama 3.1 and Qwen 2.5. Qwen 2.5 exhibits a substantially steeper decay, capturing 90% of the total variance in 857 principal components compared to 1,987 for Llama 3.1.

6.2 Impact of Distillation Depth

Our initial results demonstrate that KD using only the first layer of the LLM performs comparably to standard CE training with the full model. This raises an important question: how does increasing the distillation depth affect downstream performance? Figure 4 plots the WER on LibriSpeech and the average COMET score on Europarl-ST as we vary the target LLM layer L for the $\mathcal{L}_{\text{KD}}$ objective. The isolated star markers represent the baseline performance using standard CE supervision across the full 32-layer LLM.

The results reveal a stark contrast between the optimal representations for ASR and ST. For ASR (the solid blue line), performance is strictly optimized at the shallowest depth. Distilling only the first LLM layer yields the lowest WER, and as the distillation depth increases, acoustic alignment steadily degrades. This suggests that the ASR task relies more heavily on the localized, surface-level representations found in early layers rather than the highly abstract representations formed deeper in the network.

Conversely, ST performance (the dashed orange line) benefits from intermediate semantic abstraction. Average COMET scores initially improve as the distillation depth increases, peaking at layer 12 before steadily declining. This indicates that the ST task requires the more complex semantic signals that develop at intermediate stages of the model. Importantly, for both tasks, distilling shallow or intermediate representations yields strictly better performance than mimicking the final layers (All

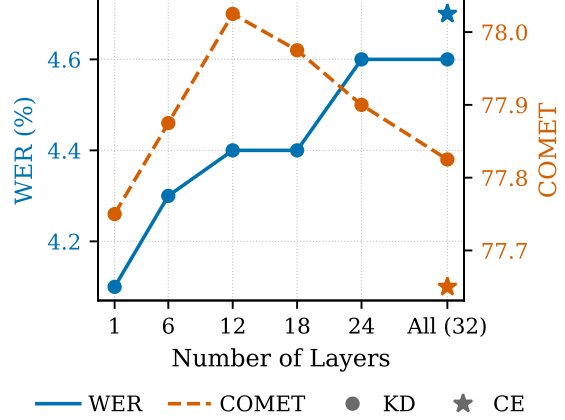


Figure 4: Impact of knowledge distillation layer depth on ASR and ST performance of Llama 3.1. WER on LibriSpeech (test-clean) steadily degrades as the number of distillation layers increases. Conversely, average COMET scores on Europarl-ST initially improve, reaching a peak at 12 layers before declining.

32). Furthermore, our optimal KD configurations (layer 1 for ASR, layer 12 for ST) outperform the full-model CE baselines. This confirms our hypothesis that mapping speech into the LLM’s early or intermediate semantic spaces is sufficient for strong zero-shot performance, eliminating the need for full-model feedback. Finally, in Appendix B, we show that using cosine distance as an alternative to KL divergence is equally effective for zero-shot ASR and ST.

6.3 Sequence Length Alignment and Efficiency

During inference, ACIF relies solely on the predicted accumulation weights α to integrate acoustic frames into discrete token representations. This raises the question of how closely the length of these integrated embeddings matches that of the target reference text. As reported in Table 3, our ACIF mechanism dynamically aggregates acoustic frames into discrete speech representations that closely mirror ground-truth token lengths across both LLM backbones. Specifically, the average length ratio (M_{CIF}/M , where M_{CIF} and M denote the number of emitted ACIF tokens and target text tokens, respectively) remains tightly bounded between 0.994 and 0.996 with minimal variance ($\sigma \leq 0.033$). This demonstrates that the learned α -accumulation accurately triggers at lexical boundaries without systematic under- or over-generation. Remarkably, 82%–84% of test sequences achieve

	Stage	Comp.	Ratio	Exact (%)	$\leq \pm 1$ (%)	$\leq \pm 2$ (%)
Llama 3.1	Stage 1	$4.45\times$	0.996 ± 0.032	84.1	97.8	99.3
	Stage 2 [CE]	$4.45\times$	0.995 ± 0.033	82.1	97.1	99.2
	Stage 2 [KD]	$4.45\times$	0.994 ± 0.032	82.7	97.1	99.2
Qwen 2.5	Stage 1	$4.45\times$	0.996 ± 0.031	83.8	97.7	99.4
	Stage 2 [CE]	$4.45\times$	0.995 ± 0.033	82.8	97.4	99.3
	Stage 2 [KD]	$4.45\times$	0.995 ± 0.033	83.2	97.3	99.2

Table 3: Acoustic sequence compression and length alignment accuracy on LibriSpeech test-clean. **Comp.** denotes the dynamic CIF compression factor (N'/M_{CIF}). **Ratio** indicates the mean length ratio ($M_{\text{CIF}}/M \pm \sigma$). **Exact**, $\leq \pm 1$, and $\leq \pm 2$ report the percentage of utterances matching the target transcript length within 0, 1, and 2 token tolerances. Best overall results are in bold.

an exact length match with the reference token sequence, and over 97% fall within a tight ± 1 token tolerance ($> 99\%$ within ± 2 tokens). Furthermore, this length calibration is strictly preserved throughout Stage 2 fine-tuning under both KD and CE objectives.

Crucially for computational efficiency, ACIF achieves an average dynamic downsampling rate of $4.45\times$ over the projected speech representations. Compounded with the initial $4\times$ reduction from the projector’s convolutional layers, this yields an effective overall compression ratio of $\sim 17.8\times$ (4×4.45) relative to the speech encoder’s output. In contrast, SpeechMapper relies on a fixed $8\times$ reduction via consecutive frame averaging and strided convolutions. ACIF thus achieves more than double the temporal compression of prior work, substantially shortening the sequence length injected into the frozen LLM and mitigating quadratic self-attention overhead during inference.

7 Conclusion

In this paper, we introduced ACIF, a highly efficient framework that equips text-centric LLMs with zero-shot speech capabilities. By dynamically compressing acoustic frames into exact discrete token lengths via DTW-guided alignments, ACIF enhances both computational efficiency and downstream performance while eliminating the need for full LLM supervision. Our Stage 1 training establishes a robust acoustic-to-semantic bridge using lightweight distance metrics, entirely bypassing the LLM forward pass. For optional Stage 2 fine-tuning, we proposed a single-layer knowledge distillation objective that performs competitively with full-model cross-entropy supervision at a fraction of the memory cost. Ultimately, ACIF consis-

tently outperforms parameter-efficient baselines on ASR and ST benchmarks, providing a lightweight, robust alternative for the future development of SpeechLLMs.

Limitations

While ACIF demonstrates strong zero-shot capabilities, several limitations remain. First, our current training and evaluation pipelines are primarily English-centric. Future work should investigate the framework’s scalability to a broader range of low-resource and typologically diverse languages. Second, our empirical evaluation focuses exclusively on Automatic Speech Recognition and Speech Translation. Exploring the applicability of ACIF to broader downstream tasks, such as spoken question answering or acoustic reasoning, remains an open area for investigation. Third, due to computational constraints, all models were trained using a single random seed. Although training converged reliably across configurations, reporting variance and confidence intervals across multiple seeds would provide a more complete assessment of performance stability. Finally, mapping continuous speech frames directly to discrete text token embeddings runs the risk of discarding valuable prosodic and paralinguistic cues inherent to spoken language. Future research should systematically analyze this information loss and explore alignment mechanisms that retain expressive acoustic features without compromising semantic alignment.

Acknowledgments

The research presented in this paper was conducted as part of VOXReality project⁵, which was funded by the European Union Horizon Europe program under grant agreement No 101070521.

This work used the Dutch national e-infrastructure with the support of the SURF Cooperative using grant no. EINF-16552.

References

- Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. 2021. [Intrinsic dimensionality explains the effectiveness of language model fine-tuning](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7319–7328, Online. Association for Computational Linguistics.

⁵<https://voxreality.eu/>

- Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Naveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, and 3416 others. 2025. [Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities](#). *Preprint*, arXiv:2507.06261.
- Seamless Communication, Loïc Barrault, Yu-An Chung, Mariano Coria Meglioli, David Dale, Ning Dong, Mark Duppenhaler, 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, and 46 others. 2023. [Seamless: Multilingual expressive and streaming speech translation](#). *Preprint*, arXiv:2312.05187.
- Alexis Conneau, Min Ma, Simran Khanuja, Yu Zhang, Vera Axelrod, Siddharth Dalmia, Jason Riesa, Clara Rivera, and Ankur Bapna. 2022. [Fleurs: Few-shot learning evaluation of universal representations of speech](#). *arXiv preprint arXiv:2205.12446*.
- Nilaksh Das, Saket Dingliwal, Srikanth Ronanki, Rohit Paturi, Zhaocheng Huang, Prashant Mathur, Jie Yuan, Dhanush Bekal, Xing Niu, Sai Muralidhar Jayanthi, Xilai Li, Karel Mundnich, Monica Sunkara, Sravan Bodapati, Sundararajan Srinivasan, Kyu J Han, and Katrin Kirchhoff. 2025. [Speechverse: A large-scale generalizable audio language model](#). *Preprint*, arXiv:2405.08295.
- Keqi Deng, Guangzhi Sun, and Phil Woodland. 2025. [Wav2Prompt: End-to-end speech prompt learning and task-based fine-tuning for text-based LLMs](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6940–6956, Albuquerque, New Mexico. Association for Computational Linguistics.
- Linhao Dong and Bo Xu. 2020. [Cif: Continuous integrate-and-fire for end-to-end speech recognition](#). In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6079–6083.
- Qingkai Fang, Rong Ye, Lei Li, Yang Feng, and Mingxuan Wang. 2022. [STEMM: Self-learning with speech-text manifold mixup for speech translation](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7050–7062, Dublin, Ireland. Association for Computational Linguistics.
- Yassir Fathullah, Chunyang Wu, Egor Lakomkin, Junteng Jia, Yuan Shangguan, Ke Li, Jinxi Guo, Wenhan Xiong, Jay Mahadeokar, Ozlem Kalinli, Christian Fuegen, and Mike Seltzer. 2024a. [Prompting large language models with speech recognition abilities](#). In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 13351–13355.
- Yassir Fathullah, Chunyang Wu, Egor Lakomkin, Ke Li, Junteng Jia, Yuan Shangguan, Jay Mahadeokar, Ozlem Kalinli, Christian Fuegen, and Mike Seltzer. 2024b. [AudioChatLlama: Towards general-purpose speech abilities for LLMs](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5522–5532, Mexico City, Mexico. Association for Computational Linguistics.
- Yuan Gong, Hongyin Luo, Alexander H. Liu, Leonid Karlinsky, and James R. Glass. 2024. [Listen, think, and understand](#). In *The Twelfth International Conference on Learning Representations*.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025. [Deepseek-r1 incentivizes reasoning in llms through reinforcement learning](#). *Nature*, 645(8081):633–638.
- Shujie Hu, Long Zhou, Shujie Liu, Sanyuan Chen, Lingwei Meng, Hongkun Hao, Jing Pan, Xunying Liu, Jinyu Li, Sunit Sivasankaran, Linquan Liu, and Furu Wei. 2024. [WavLLM: Towards robust and adaptive speech large language model](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 4552–4572, Miami, Florida, USA. Association for Computational Linguistics.
- Hirofumi Inaguma, Tatsuya Kawahara, and Shinji Watanabe. 2021. [Source and target bidirectional knowledge distillation for end-to-end speech translation](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1872–1881, Online. Association for Computational Linguistics.
- Javier Iranzo-Sánchez, Joan Albert Silvestre-Cerdà, Javier Jorge, Nahuel Roselló, Adrià Giménez, Albert Sanchis, Jorge Civera, and Alfons Juan. 2020. [Europarl-st: A multilingual corpus for speech translation of parliamentary debates](#). In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8229–8233.

- Abderrahmane Issam, Yusuf Can Semerci, Jan Scholtes, and Gerasimos Spanakis. 2025. [DTW-align: Bridging the modality gap in end-to-end speech translation with dynamic time warping alignment](#). In *Proceedings of the Tenth Conference on Machine Translation*, pages 191–199, Suzhou, China. Association for Computational Linguistics.
- Yuchen Liu, Junnan Zhu, Jiajun Zhang, and Chengqing Zong. 2020. [Bridging the modality gap for speech-to-text translation](#). *Preprint*, arXiv:2010.14920.
- Biswesh Mohapatra, Marcey Zanon Boito, and Ioan Calapodescu. 2026. [Speechmapper: Speech-to-text embedding projector for llms](#). In *ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 16277–16281.
- Jiaqi Mu and Pramod Viswanath. 2018. [All-but-the-top: Simple and effective postprocessing for word representations](#). In *International Conference on Learning Representations*.
- nostalgebraist. 2020. interpreting GPT: the logit lens. *LessWrong*.
- OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Madry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024. [Gpt-4o system card](#). *Preprint*, arXiv:2410.21276.
- 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.
- Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. [Librispeech: An asr corpus based on public domain audio books](#). In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. [Bleu: a method for automatic evaluation of machine translation](#). In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.
- 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.
- Vineel Pratap, Andros Tjandra, Bowen Shi, Paden Tomasello, Arun Babu, Sayani Kundu, Ali Elkahky, Zhaoheng Ni, Apoorv Vyas, Maryam Fazel-Zarandi, Alexei Baevski, Yossi Adi, Xiaohui Zhang, Wei-Ning Hsu, Alexis Conneau, and Michael Auli. 2024. [Scaling speech technology to 1,000+ languages](#). *J. Mach. Learn. Res.*, 25(1).
- Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. [Qwen2.5 technical report](#). *Preprint*, arXiv:2412.15115.
- Vikas Raunak, Vivek Gupta, and Florian Metze. 2019. [Effective dimensionality reduction for word embeddings](#). In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 235–243, Florence, Italy. Association for Computational Linguistics.
- Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. 2022. [COMET-22: Unbabel-IST 2022 submission for the metrics shared task](#). In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.
- H. Sakoe and S. Chiba. 1978. [Dynamic programming algorithm optimization for spoken word recognition](#). *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49.
- Weiting Tan, Hirofumi Inaguma, Ning Dong, Paden D. Tomasello, and Xutai Ma. 2025. [SSR: Alignment-aware modality connector for speech language models](#). In *Proceedings of the 22nd International Conference on Spoken Language Translation (IWSLT 2025)*, pages 56–75, Vienna, Austria (in-person and online). Association for Computational Linguistics.
- Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun MA, and Chao Zhang. 2024. [SALMONN: Towards generic hearing abilities for large language models](#). In *The Twelfth International Conference on Learning Representations*.
- Changhan Wang, Morgane Riviere, Ann Lee, Anne Wu, Chaitanya Talnikar, Daniel Haziza, Mary Williamson, Juan Pino, and Emmanuel Dupoux. 2021a. [VoxPopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 993–1003, Online. Association for Computational Linguistics.
- Changhan Wang, Anne Wu, Jiatao Gu, and Juan Pino. 2021b. [CoVoST 2 and Massively Multilingual Speech Translation](#). In *Interspeech 2021*, pages 2247–2251.

Chengyi Wang, Yu Wu, Shujie Liu, Zhenglu Yang, and Ming Zhou. 2020. [Bridging the gap between pre-training and fine-tuning for end-to-end speech translation](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:9161–9168.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Jian Wu, Yashesh Gaur, Zhuo Chen, Long Zhou, Yimeng Zhu, Tianrui Wang, Jinyu Li, Shujie Liu, Bo Ren, Linquan Liu, and Yu Wu. 2023. [On decoder-only architecture for speech-to-text and large language model integration](#). In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–8.

Rong Ye, Mingxuan Wang, and Lei Li. 2022. [Cross-modal contrastive learning for speech translation](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5099–5113, Seattle, United States. Association for Computational Linguistics.

Chengwei Zhang, Yue Zhou, Rui Zhao, Yidong Chen, and Xiaodong Shi. 2025. [Representation purification for end-to-end speech translation](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6255–6269, Abu Dhabi, UAE. Association for Computational Linguistics.

A Evaluation Prompts

For our evaluations, we format the inputs according to the specific chat templates of the underlying LLMs (e.g., Llama-3.1 or Qwen-2.5). The generalized prompt structure consists of a system message establishing the assistant’s role, followed by a user message containing the projected acoustic embeddings and a task-specific instruction.

For ASR, we follow the instruction style of SpeechMapper to prompt verbatim transcription. For ST, the user instruction consists of a target-language-specific instruction which is a translation of "Can you translate the Speech content into [German/Spanish/French/Italian/Chinese] text?", followed by a formatting suffix: "Output only the translation and nothing else.". Table 4 details the exact system and user instructions used across all evaluations.

B Cosine is as Effective as KL

Computing the standard KL divergence requires projecting hidden states through the LLM’s unembedding matrix and applying a softmax operation, which introduces notable computational overhead during training. As a more efficient alternative, we experiment with defining our distillation objective, $\mathcal{L}_{KD}$, as the simple cosine distance between the intermediate hidden states of the speech representations Z and the text embeddings E . Because the cosine distance is strictly bounded between 0 and 2, yielding intrinsically smaller numerical values than standard CE or KL losses, we scale this objective by a factor of $\lambda_{kd} = 10.0$. As demonstrated in Table 5, the cosine distance serves as a highly effective alternative to KL divergence. It achieves identical average COMET scores (77.8) across the Europarl-ST translation directions, while incurring only a marginal performance degradation of 0.1 WER on the LibriSpeech test-clean split.

Objective	LS clean	Europarl-ST
Cosine	4.2 / 2.3	77.8
KL	4.1 / 2.2	77.8

Table 5: Performance comparison between Cosine and KL distillation objectives on LibriSpeech (WER/CER) and Europarl-ST (COMET).

Task	System Prompt	User Instruction (Appended after Speech Embeddings)
ASR	You are a helpful ASR transcription assistant.	Repeat the previous text between the quotes in its entirety just once and nothing else. Do not repeat the text multiple times or correct the text or add punctuation. End the text if you notice a phrase or a text is getting repeated. Ignore the words that do not make any sense.
ST (de)	You are a helpful translation assistant.	Können Sie den Inhalt der Rede in den deutschen Text übersetzen? Output only the translation and nothing else.
ST (es)	You are a helpful translation assistant.	¿Puedes traducir el contenido del discurso al texto en español? Output only the translation and nothing else.
ST (fr)	You are a helpful translation assistant.	Pouvez-vous traduire le contenu du discours en texte français ? Output only the translation and nothing else.
ST (it)	You are a helpful translation assistant.	Puoi tradurre il contenuto del discorso in testo italiano? Output only the translation and nothing else.
ST (zh)	You are a helpful translation assistant.	你能把演讲内容翻译成中文吗? Output only the translation and nothing else.

Table 4: System prompts and user instructions used for zero-shot ASR and ST evaluation. The acoustic embeddings are inserted directly before the user instruction to form the final user message.