ASR DATA IS ALL YOU NEED TO BUILD INSTRUCTION-FOLLOWING SPEECHLLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

SpeechLLMs can follow instructions like text LLMs by connecting a pretrained speech encoder to an LLM through an adapter. Current approaches typically rely on a two-stage training procedure including fine-tuning on instruction-following (IF) speech data, which is expensive to collect. In this work, we investigate whether IF speech data can be avoided by using Sequence-Level Knowledge Distillation (SeqKD) from a pretrained LLM operating on automatic speech recognition (ASR) data, exploring both prompt-conditioned SeqKD, where the same prompt is provided to the student and teacher, and prompt-free SeqKD, where no prompt is provided. With experiments on three tasks (ASR, speech translation, and spoken question answering) and four benchmarks including text instructions and outputs in languages unseen during training, we show that SeqKD effectively transfers the LLM's linguistic and IF capabilities but suffers from poor ASR performance. We address this limitation with a single-stage training strategy combining prompt-free SeqKD with limited ASR and prompt-conditioned data. Our approach outperforms a cascade of ASR and LLM and a two-stage approach, and remains competitive across tasks with SpeechLLMs trained on substantially larger amounts of data.

1 INTRODUCTION

Recent advances have extended Large Language Models (LLMs) beyond text-based interaction to spoken input (SpeechLLMs), enabling users to access their knowledge and reasoning capabilities through a more natural communication modality (Latif et al., 2023; Gaido et al., 2024b; Cui et al., 2025). Early solutions adopted a cascaded architecture (Arora et al., 2025), in which an ASR module transcribes the speech and passes the transcript to a text-based LLM (Zhang et al., 2023; Shu et al., 2023). This design is straightforward, as it reuses existing ASR and LLM components without additional training. However, recognition errors propagate to the LLM, and running two models increases inference latency (Sperber & Paulik, 2020). As a solution, current SpeechLLMs either rely on speech units (Wang et al., 2025a) or project speech-encoder outputs into the LLM input space, generating responses from acoustic representations (Chu et al., 2024; Tang et al., 2024; Lu et al., 2026; Liu et al., 2025; Ding et al., 2025; Fang et al., 2025; Xie & Wu, 2024). Since speech-unit-based solutions tend to perform worse than projection-based methods (Wang et al., 2025b), in this work we focus on projection-based SpeechLLMs[1].

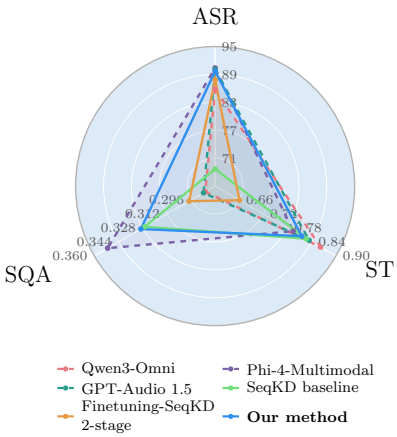

Figure 1: With only ∼3K hours of ASR data, our single-stage method substantially improves speech recognition over SeqKD while preserving its ST and SQA capabilities, achieving performance comparable to substantially larger multimodal models.

Within this paradigm, training a SpeechLLM that can follow instructions across diverse speech-language tasks typically involves two sequential stages: (i) a pre-training stage on ASR data to align speech and text representations, followed by (ii) supervised fine-tuning on speech

[1]Note that the same strategies can be applied to other architectures, although we do not evaluate them.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

inputs paired with instructions and task-specific responses (Zhang et al., 2023; Tang et al., 2024; Microsoft et al., 2025) to unlock instruction following (IF) capabilities and handle diverse speech-related tasks. Optionally, a third post-training stage can leverage Reinforcement Learning techniques to align the SpeechLLM with human preferences (Kong et al., 2026). The first stage benefits from the large availability of open ASR data (Gaido et al., 2024a; Rao Koluguri et al., 2025; Li et al., 2023). Differently, the second and optionally third stages typically rely on proprietary speech datasets (Ding et al., 2025; Chu et al., 2024), as open datasets comprising speech, instructions, and target responses remain scarce and expensive to collect, particularly for multilingual scenarios (Wang et al., 2026b). This hinders the development of open-source multilingual SpeechLLMs that can match the IF capabilities of larger, massively multilingual proprietary models.

To overcome this limitation, we investigate whether **SpeechLLMs with IF capabilities can be trained without dedicated speech IF data, using ASR data as the sole source of speech supervision**. Motivated by recent works effectively applying Knowledge Distillation (KD; Hinton et al. 2015) to train SpeechLLMs, where the underlying text-only LLM serves as the teacher by processing the speech transcript (Fathullah et al., 2024; Shao et al., 2026; Lu et al., 2026; Cuervo et al., 2026; Wang et al., 2024; 2026a), we explore several training strategies based on Sequence-Level KD (SeqKD; Kim & Rush 2016). These operate on ASR data, as the LLM teacher is fed with the transcript to generate the supervision of the SpeechLLM student fed with the corresponding audio. As illustrated in Figure 2, we explore SeqKD both without additional instructions (prompts) for teacher and student (prompt-free SeqKD, hereinafter **PF-SeqKD**) and with the same prompt supplied to both models (prompt-conditioned SeqKD, hereinafter **PC-SeqKD**). We evaluate these strategies on three tasks: ASR, speech translation (ST), and open-ended spoken question answering (SQA) across four benchmarks, also from domains not represented in training. For ST and SQA, we additionally evaluate multiple target languages, unseen during training. Our results show that:

- **SeqKD transfers IF capabilities but lacks reliable speech recognition.** Current SeqKD methods produce SpeechLLMs capable of performing semantic tasks and leveraging the underlying LLM textual processing capabilities, with PF-SeqKD outperforming PC-SeqKD. However, we unveil that such SpeechLLMs are unable to properly perform ASR, as they do not preserve the exact wording of the input.
- **Limited ASR supervision bridges the ASR-IF gap.** We solve this limitation by adding 10% ASR and 10% prompt-conditioned data to PF-SeqKD. Our best model improves ASR by 67.11% over the SeqKD Baseline while maintaining comparable performance on SQA and ST. It also improves SQA and ST by 9.69% and 23.42%, respectively, over a two-stage strategy, in which PF-SeqKD training is followed by a fine-tuning on the same 10% ASR and 10% prompt-conditioned data.
- **ASR speech supervision enables multilingual, competitive SpeechLLMs.** Despite using only $\sim$3K hours of English (EN) and Spanish (ES) ASR data for SpeechLLM training, our best model strictly outperforms a cascade made of the same speech model and LLM, and even approaches the results of strong, larger models (`Qwen3-Omni`, `Phi-4-Multimodal`, and `GPT-Audio-1.5`) on the evaluated benchmarks (Figure 1). These include languages unseen during the SpeechLLM training (German—DE, Italian—IT, and Chinese—ZH) on the textual (prompt and generated text) side, demonstrating that our strategy enables not only IF on various tasks, but also cross-lingual transfer of the LLM capabilities even to languages of families and scripting systems different from those used during training (e.g. Chinese).

2 BACKGROUND

KD was originally introduced for classification tasks to transfer knowledge from a teacher to a smaller student model by training it to reproduce the teacher's output probability distribution (Hinton et al., 2015). Subsequent work extended KD to sequence-to-sequence models with two main variants of similar effectiveness (Kim & Rush, 2016): Word-level KD computes the KL divergence between the teacher and student output distributions for each token (under teacher forcing), while Sequence-level KD (SeqKD) trains the student to reproduce the teacher-generated sequence using standard cross-entropy loss. KD was later also used to train speech models, enabling students to mimic stronger textual counterparts, e.g., by training ST models from text-to-text machine translation (MT) models (Liu et al., 2019). In this case, given a speech segment and its corresponding transcript, the

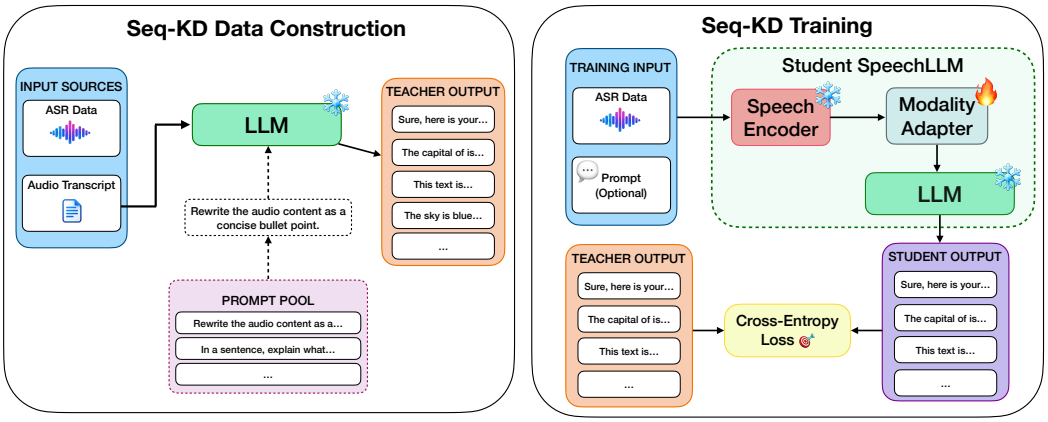

Figure 2: SeqKD transfers a text LLM's outputs to speech: the student learns from teacher-generated targets, optionally conditioned on prompts, by training only the modality adapter.

ST model takes the speech as input and learns to reproduce the output of a frozen MT model given the transcript.

3 SpeechLLM Training with Sequence-KD

In the context of SpeechLLM training, the simplest SeqKD approach feeds a speech transcript to a frozen text-only teacher LLM to obtain its response or continuation. A student SpeechLLM then learns to produce the same output from the corresponding speech input. This mirrors the use of SeqKD in encoder-decoder architectures (MT/ST); in §1, we referred to this setting as **PF-SeqKD**. Formally, let $(x_i, t_i) \in D$ denote a speech input and its corresponding transcription, and f the frozen text-only LLM. For each sample, the teacher receives only the transcript and generates a response or continuation that serves as the SpeechLLM training target. The resulting training dataset D_{PF} is defined as:

$$D_{PF} = \{x_i, y_i^{\text{PF}}\}, \text{where } y_i^{\text{PF}} = f(t_i), \tag{1}$$

Since it is the simplest approach, most works exploring SeqKD for SpeechLLM training have relied on it (Fathullah et al., 2024; Shao et al., 2026; Cuervo et al., 2026; Wang et al., 2024; 2026a). However, SpeechLLMs differ from traditional encoder-decoder models in that they also accept prompts that steer their behavior. We therefore also consider providing both the teacher and student with an additional prompt q_i. In §1, we referred to this approach as **PC-SeqKD**. Under this definition, DeSTA (Lu et al., 2026), which stands for Descriptive Speech-Text Alignment, can be interpreted as a form of PC-SeqKD. Accordingly, we use the prompts defined in DeSTA 2.5 (Lu et al., 2026), which are open-ended audio-understanding requests (see Appendix A).[2] Denoting by P the set of prompts taken from DeSTA 2.5, the training set D_{PC} for the SpeechLLM in PC-SeqKD is defined as:

$$D_{\text{PC}} = \{x_i, q_i, y_i^{\text{PC}}\}, \text{where } y_i^{\text{PC}} = f(q_i, t_i), q_i \sim \text{Uniform}(P) \tag{2}$$

ASR Augmentation. As we will see in §5.1, the main weakness of PF-SeqKD and PC-SeqKD is poor ASR performance. To address this issue, we augment the training data with a small percentage p of speech-transcript pairs sampled from D. Each pair is associated with a transcription prompt sampled from an ASR prompt pool containing k distinct prompts.[3] Varying p and k, we study

[2]Available at: `https://hf.co/datasets/DeSTA-ntu/DeSTA-AQA5M-FROM-Llama3.1-8B-Instruct`

[3]The ASR prompt pool contains five manually crafted prompts (Appendix B) and is expanded with paraphrases generated using `Claude Sonnet 4.6` to obtain pools of $k \in 20, 200$.

the effects of supervision quantity and prompt diversity. The resulting examples are added to the corresponding PF-SeqKD or PC-SeqKD dataset. We denote these data mixtures as $D_{\mathrm{PF+ASR}}$ and $D_{\mathrm{PC+ASR}}$, respectively.

Adding prompt-conditioned supervision. As we will see in §5.2, ASR data may induce overfitting in PF-SeqKD, as the SpeechLLM learns to perform ASR whenever a prompt is present rather than following its content. To address this issue, we introduce an additional prompt-conditioned augmentation (PC-Aug), in which both $D_{\mathrm{PF+ASR}}$ and $D_{\mathrm{PC+ASR}}$ are augmented with a percentage p of prompt-conditioned samples from D_{PC}, matching the percentage used for the ASR augmentation. We denote the resulting datasets as $D_{\mathrm{PF+ASR+PC\text{-}Aug}}$ and $D_{\mathrm{PC+ASR+PC\text{-}Aug}}$, respectively.

Single-step training vs. fine-tuning. Since training on data that combines speech inputs with task prompts resembles the second stage of traditional SpeechLLM training, i.e., IF fine-tuning, we compare our single-stage strategy with a two-stage alternative. In the latter, the model first undergoes PF-SeqKD or PC-SeqKD baseline training and is then fine-tuned on the combined $\mathrm{ASR} + \mathrm{PC\text{-}Aug}$ data. The two strategies use the same training data but differ in when the ASR and PC-Aug samples are presented to the model. In single-stage training, these samples are mixed into the PF-SeqKD or PC-SeqKD training data; in the two-stage approach, they are introduced only after PF-SeqKD or PC-SeqKD training, during fine-tuning. Further details on fine-tuning are provided in Appendix C.

4 Experimental setup

4.1 Data

We use the EN and ES portions of two publicly available and widely used ASR corpora for training: **VoxPopuli** (Wang et al., 2021), sourced from European Parliament recordings, and **Common Voice** (Ardila et al., 2020),[4] a crowd-sourced multilingual speech corpus. The training dataset D contains 2,958 hours (2,295 in EN and 663 in ES). We report pre-processing details in Appendix D. The mixtures described in §3 use $p \in \{1\%, 5\%, 10\%, 20\%\}$ and $k \in \{1, 20, 200\}$.

4.2 Models and training

Our SpeechLLM follows an encoder-adapter-LLM architecture (Gaido et al., 2024b; Arora et al., 2025), where a frozen speech encoder processes the raw audio, a trainable adapter bridges the speech and text representation spaces, the resulting speech representations are appended to the textual prompt embeddings, and a frozen LLM generates textual output. We detail each component below, while we report the training hyperparameters in Appendix E.

Speech Encoder. We use `canary-1b-v2` (Sekoyan et al., 2025) as frozen speech encoder. Canary is a ASR and ST model trained on a multilingual dataset of ∼1.7M hours of speech.

Adapter. We experiment with two MLP-based adapters which differ in size: a 22M- and a 127M-parameter adapter. Both consist of a stack of linear layers with GELU activations (Hendrycks & Gimpel, 2016). We report architecture details in Appendix F. In the body of the paper, we only report results using the 127M adapter, as it performs slightly better than the 22M adapter while showing similar trends. The results with the 22M-parameter adapter are available in Appendix H.

LLM. We use `Qwen3-4B-Instruct-2507` (Yang et al., 2025) as both the frozen SpeechLLM backbone and SeqKD teacher, following prior work (Fathullah et al., 2024) and the evidence that students learn more effectively from similar teachers rather than stronger teachers (Kujanpää et al., 2025; Lu & Liu, 2026; Yang et al., 2026). We report a comparison with other candidate LLMs in Table 5 in Appendix G.

4.3 Evaluation

We evaluate the trained SpeechLLMs on three tasks (ASR, ST, SQA) and four benchmarks.

[4]We use the Common Voice Scripted Speech 25.0 English and Spanish subsets.

Model	WER↓		COMET↑		BERTScore↑
	🇺🇸	🇪🇸	🇺🇸→ X	🇪🇸→ X	
`Qwen3-Omni`	14.40%	13.87%	0.8664	0.8517	0.2865
`Phi-4-Multimodal`	10.64%	8.02%	0.8135	0.7542	0.3512
`GPT-Audio 1.5`	10.14%	9.60%	0.8432	0.8236	0.2877
PF-SeqKD Baseline	22.71%	44.23%	0.8318	0.8154	0.3266
PC-SeqKD Baseline	27.20%	61.70%	0.8310	0.8180	0.2810
Cascade	–	–	0.8042	0.8232	0.2750

Table 1: PF-SeqKD baseline offers stronger SQA and comparable ST capabilities than PC-SeqKD, while ASR remains the main weakness of both, particularly in Spanish.

The MCIF benchmark (Papi et al., 2026), featuring ACL talks that are out of our training data domain, covers all three tasks and provides EN audio inputs with prompts in four languages: EN, DE, IT, and ZH, with the latter three unseen during our training. It includes long- and short-form context settings, and two prompt types: fixed (MCIF_{FIX}), which employs a single prompt per macro-task, and mixed (MCIF_{MIX}), randomly selecting from ten alternatives per task. We use the short-context setting with mixed prompts to reduce the bias introduced by selecting a single prompt. Fixed-prompt results are reported in Appendix H.4.

For ASR, we also evaluate on the EN and ES test sets of Common Voice (Ardila et al., 2020) and Europarl-ST (Iranzo-Sánchez et al., 2020). For ST, we also evaluate on Europarl-ST (Iranzo-Sánchez et al., 2020) for EN-DE, EN-ES, EN-IT, ES-DE, ES-EN, and ES-IT translations, and on CoVoST 2 (Wang et al., 2020) for EN-DE, EN-ZH, and ES-EN translations. These three test sets contain data drawn from the same domain of our training data.

As metrics, we rely on those used in MCIF: *i)* Word Error Rate (WER) for ASR; *ii)* BERTScore (Zhang et al., 2020),[5] for SQA; *iii)* COMET (Rei et al., 2020)[6] for ST.

5 Results

In this section, we first assess the strengths and weaknesses of plain SeqKD (§5.1). We then analyze the effects of adding ASR data (§5.2). Lastly, we show that our proposed method produces a SpeechLLM with strong IF abilities, while also outperforming two-stage training strategies (§5.3).

5.1 PF-SeqKD vs PC-SeqKD

Table 1 compares PF-SeqKD and PC-SeqKD. To contextualize their performance, we also report results for three high-capacity multimodal models as external references: the open-weight `Qwen3-Omni` (Xu et al., 2025) and `Phi-4-Multimodal` (Microsoft et al., 2025), trained on substantially larger datasets and the closed model `GPT-Audio-1.5`. Given the substantial differences in model scale, architecture, and training data, these models serve as reference points rather than baselines. Appendix H.1 reports detailed results for each benchmark and language direction, along with results using the 22M adapter, which exhibit consistent trends.

The results show that PF-SeqKD outperforms PC-SeqKD on almost all tasks, with ST from ES being the only exception. Nonetheless, both SeqKD variants transfer useful IF and generation capabilities: they both outperform the cascade solution—in which `Canary-1B-v2` transcribes the speech before `Qwen3-4B-Instruct-2507` processes the transcription and task instruction—on all tasks except for ES→X ST, in which they are anyway close. PF-SeqKD also outperforms `Qwen3-Omni` and `GPT-Audio 1.5` on SQA (where it is surpassed only by `Phi-4-Multimodal`), while on ST it outperforms `Phi-4-Multimodal` and has a limited gap to `GPT-Audio 1.5`, although `Qwen3-Omni` emerges as the best model on this task. Looking at ASR performance, however, PF-SeqKD and PC-SeqKD obtain high WER, and all models outperform them by a large margin.

[5]We use the Rescaled BERTScore definition `https://github.com/Tiiiger/bert_score/blob/master/journal/rescale_baseline.md`

[6]Using: `https://huggingface.co/Unbabel/wmt22-comet-da`.

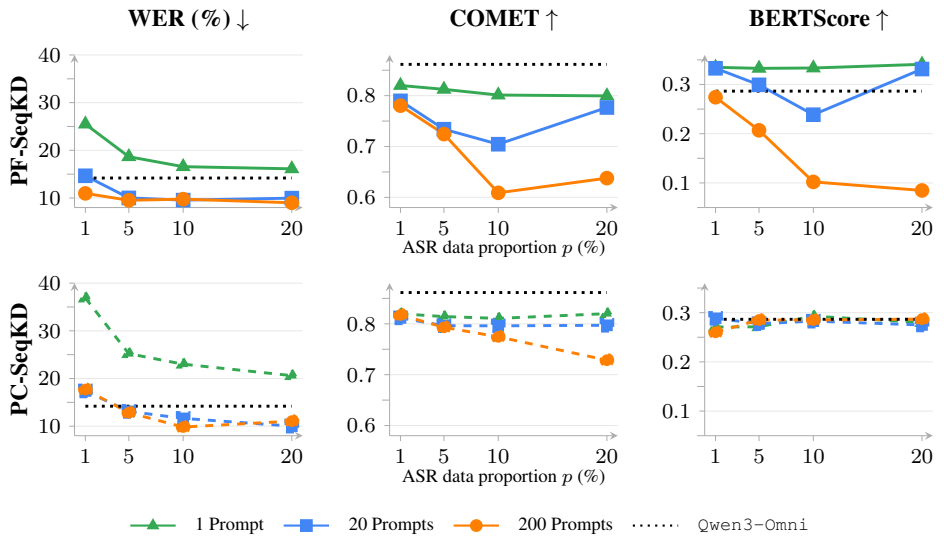

Figure 3: The top and bottom rows show PF-SeqKD and PC-SeqKD, respectively. Increasing ASR supervision improves recognition for both, but can degrade ST and SQA, particularly for PF-SeqKD, revealing an ASR task-overfitting problem.

Paraphrasing dominates recognition errors. To investigate the source of the weak ASR performance, we manually inspected 166 English sentences uniformly drawn from the three test sets. We identified five error categories: **(i)** paraphrasing, **(ii)** named-entity errors, **(iii)** hallucinations, **(iv)** omissions, and **(v)** recognition errors. Interestingly, paraphrasing—i.e., conveying the same semantic content with different wording—was by far the most common phenomenon (110 cases), followed by true recognition errors (25), named-entity errors (23), omissions (12), and hallucinations (5).[7]

Overall, these findings indicate that SeqKD effectively transfers linguistic and IF capabilities, including to languages unseen during SpeechLLM training, while preserving the semantic content of the speech input. However, it does not provide reliable speech recognition. In the next section, we investigate whether explicit ASR supervision can address this limitation.

5.2 Effect of ASR supervision and prompt diversity

To improve the model's ASR capabilities, we include an increasing percentage p of training samples paired with an ASR instruction and reference transcription. We also analyze the impact of prompt diversity by sampling ASR instructions from pools of different sizes (k). Figure 3 reports ASR, ST, and SQA performance for both PF-SeqKD and PC-SeqKD in these settings; additional results are reported in Appendix H.2, showing consistent trends.

ASR supervision improves WER but harms ST and SQA. As expected, increasing p generally lowers WER for both PF-SeqKD and PC-SeqKD. Increasing the prompt pool size from a single fixed prompt to 20 yields substantial gains, while increasing it to 200 does not consistently improve WER. However, these gains come with a trade-off. For PF-SeqKD, increasing p improves WER but can reduce ST and SQA performance, particularly at $k = 200$. For PC-SeqKD, downstream scores are less sensitive to changes in p and k, although ST declines in some settings. This pattern suggests over-specialization to ASR in PF-SeqKD+ASR (Tang et al., 2024). A plausible explanation is a train-test mismatch: during training, PF-SeqKD examples lack prompts, whereas ASR examples include transcription prompts. The model may therefore associate prompt presence with ASR, regardless of its content. At evaluation time, as all samples include a prompt, the model produces transcription-like responses even for other tasks as it interprets non-ASR requests as transcription instructions.

PC-SeqKD offers stability; PF-SeqKD maximizes performance. Overall, PC-SeqKD remains relatively stable across values of p and k, whereas PF-SeqKD generally achieves better SQA and

[7]Appendix I describes the manual analysis and reports annotated examples to showcase these errors.

Method	p	k	WER↓		COMET↑		BERTScore↑
			🇺🇸	🇪🇸	🇺🇸→ X	🇪🇸→ X	
Baselines							
PF-SeqKD	0%	0	22.71%	44.23%	0.8318	0.8154	0.3266
PC-SeqKD	0%	0	27.20%	61.70%	0.8310	0.8180	0.2810
Cascade	–	–	–	–	0.8042	0.8232	0.2750
Fine-tuning: SeqKD followed by ASR+Prompt-Conditioned samples							
PF-SeqKD	5%	20	15.93%	11.23%	0.7859	0.7404	0.3041
	5%	200	12.56%	10.61%	0.7197	0.7069	0.2979
	10%	20	12.68%	10.44%	0.7936	0.7496	0.3043
	10%	200	12.87%	11.03%	0.6565	0.6687	0.2973
PC-SeqKD	5%	20	17.17%	11.21%	0.7641	0.7201	0.3032
	5%	200	14.49%	11.31%	0.6417	0.6725	0.2934
	10%	20	13.34%	9.01%	0.7838	0.7208	0.2893
	10%	200	13.23%	10.48%	0.7219	0.6683	0.3013
SeqKD+ASR+Prompt-Conditioned samples							
Our Approach: PF-SeqKD	5%	20	10.91%	9.30%	0.8162	0.7842	0.3360
	5%	200	10.72%	9.89%	0.7874	0.7400	0.3218
	10%	20	11.66%	9.88%	0.7944	0.7260	0.3442
	10%	200	10.42%	10.13%	0.8212	0.8030	0.3292
PC-SeqKD	5%	20	18.81%	11.93%	0.8143	0.7957	0.2837
	5%	200	11.63%	12.62%	0.8121	0.7811	0.2847
	10%	20	12.73%	11.60%	0.8009	0.7660	0.2755
	10%	200	12.96%	10.26%	0.7986	0.7512	0.2784
External models							
`Qwen3-Omni`	–	–	14.40%	13.87%	0.8664	0.8517	0.2865
`Phi-4-Multimodal`	–	–	10.64%	8.02%	0.8135	0.7542	0.3512
`GPT-Audio 1.5`	–	–	10.14%	9.60%	0.8432	0.8236	0.2877

Table 2: The introduction of both ASR and Prompt Conditioned Samples solves the task-overfitting problem and en training SpeechLLMs with PF-SeqKD that show balanced performance across all the tasks while also outperforming both the PC-SeqKD and the two-stage fine-tuning approaches.

ASR scores but is more vulnerable to downstream degradation at large prompt-pool sizes. Thus, PC-SeqKD trades performance for stability, while PF-SeqKD achieves stronger IF performance but is more susceptible to ASR-induced over-specialization. In the next section, we solve this trade-off by augmenting the ASR-SeqKD mixture with additional prompt-conditioned examples, combining the advantages of both approaches.

5.3 Combining ASR and Prompt-Conditioned Samples

To prevent the model from associating prompts exclusively with ASR, we add an equal proportion p of D_{PC} samples (PC-Aug) to the $D_{PF} + ASR$ training set. These samples encourage the model to follow prompt content rather than associate its presence with ASR. This added data (prompt-conditioned and ASR) resembles the IF data used in the traditional two-stage procedure, where speech-text alignment is followed by IF fine-tuning. We therefore compare our single-stage training with a two-stage fine-tuning. In the latter, we first train the model on D_{PF} (PF-SeqKD) or D_{PC} (PC-SeqKD) data and then fine-tune it on $\mathrm{ASR} + \mathrm{PC\text{-}Aug}$ data. Both approaches use the same data but differ in when we introduce ASR and prompt-conditioned samples.

Table 2 reports the scores for $p \in \{5\%, 10\%\}$ and $k \in \{20, 200\}$, as §5.2 demonstrated that a single prompt and 1% ASR data are not enough to learn ASR, while increasing ASR from 10% to 20% yields no consistent improvement. For completeness, we also include the PC-SeqKD variants.

PF-SeqKD mixtures achieve peak overall performance. Across all four configurations, the PF-SeqKD mixtures consistently outperform the corresponding PC-SeqKD mixtures on EN and ES ASR,

Model	p	k	WER↓		COMET↑		BERTScore↑
					→ X	→ X	
PF-SeqKD Baseline	-	-	74.94%	72.42%	0.8259	0.8161	0.1724
PC-SeqKD Baseline	-	-	117.99%	213.67%	0.8054	0.8084	0.1552
PF-SeqKD+ASR+PC-Aug	10%	200	10.94%	28.57%	0.8289	0.8112	0.1750

Table 3: PF-SeqKD and PC-SeqKD baselines vs. our proposed training strategy using `EuroLLM-9B` as the LLM backbone. Trends are consistent with those obtained with `Qwen3-4B-Instruct-2507`, despite the more constrained training setting.

and on SQA, while ST results are more mixed. Within the PF-SeqKD mixtures, the configuration with $p = 10\%$ and $k = 200$ yields the best EN ASR (10.42% WER) and ST scores (0.8258 and 0.8124 COMET), while the best ES ASR (9.30% WER) and SQA (0.3423 BERTScore) scores are obtained with $p = 10\%$ and $k = 20$. These configurations go beyond the performance-stability trade-off observed for PC-SeqKD in § 5.2. They also achieve performance comparable to open-weight models trained on substantially larger datasets: our best model approaches `Phi-4-Multimodal` on ASR and ST performance while trailing it on SQA. Compared to `Qwen3-Omni 30B`, it achieves better ASR, comparable SQA, and lower ST performance. Notably, our single-stage models match or outperform the cascade system on downstream tasks, while directly processing speech without an intermediate transcription step. Overall, these results demonstrate that our approach can produce a competitive IF SpeechLLM capable of performing ASR, ST and SQA using only ASR data.

Single-stage training outperforms two-stage fine-tuning. Fine-tuning from PF-SeqKD consistently underperforms direct training on the corresponding mixtures across all tasks, with worse ASR, SQA, and ST scores in each configuration. Compared to the PF-SeqKD baseline, fine-tuning also introduces a trade-off between SQA, ST and ASR improving WER at the cost of lower SQA and ST performance. The PC-SeqKD initialization shows a different pattern. Fine-tuning improves SQA but decreases ST across all configurations, while its effect on ASR varies across configurations. Overall, direct training outperforms fine-tuning, with PF-SeqKD yielding the best results among the evaluated strategies.

Robust performance across tasks and unseen languages. To assess performance consistency across datasets and languages, we compare how scores vary across the evaluated settings for our PF-SeqKD SpeechLLM with those of `Qwen3-Omni`, `Phi-4-Multimodal` and `GPT-Audio-1.5`. We quantify this variation using the range between the minimum and maximum scores observed across datasets and language settings (detailed scores are available in Appendix H.3). For ASR, the ranges are 4.53-23.21% for `Qwen3-Omni`, 6.03-12.50% for `Phi-4-Multimodal`, 6.81-12.08% for `GPT-Audio-1.5`, and 6.75-13.50% for our PF-SeqKD SpeechLLM. For ST, the ranges are 0.8128-0.8905 for `Qwen3-Omni`, 0.6612-0.8544 for `Phi-4-Multimodal`, 0.7872-0.8576 for `GPT-Audio-1.5` and 0.7461-0.8588 for our PF-SeqKD SpeechLLM. For SQA, the ranges are 0.1981-0.3217 for `Qwen3-Omni`, 0.3289-0.3845 for `Phi-4-Multimodal`, 0.1902-0.3284 for `GPT-Audio-1.5` and 0.3099-0.3400 for our PF-SeqKD SpeechLLM. Although our model is trained only on EN and ES speech, its narrow ranges indicate consistent performance across datasets, comprising the **out-of-domain** MCIF benchmark, and languages, including **target languages unseen** during training.

5.4 Generalization to other LLM Backbones

To assess whether the proposed training strategy generalizes beyond the original training configuration, we conduct an additional experiment with a different LLM backbone and training setting. Specifically, we train a second SpeechLLM using `EuroLLM-9B-Instruct-2512` (Martins et al., 2025) as both the SeqKD teacher and the SpeechLLM backbone.

Beyond changing the backbone, we consider a substantially lower-resource setting with different training languages: the model is trained on only ∼900 hours of EN and Portuguese (PT) speech, compared with the ∼3K hours of EN and ES used in our main experiments. Further details on the datasets and training setup are provided in Appendix J.

Our method generalizes across backbones and low-resource settings. Table 3 reports results for the PC-SeqKD and PF-SeqKD baselines and our proposed training strategy using 10% ASR data

with 200 prompts and 10% prompt-conditioned samples. Consistently with our previous results, PF-SeqKD+ASR+PC-Aug retains the same ST and SQA scores as PF-SeqKD (which outperforms PC-SeqKD) while also learning to perform the ASR task in which PF-SeqKD is poor. These results provide evidence that the benefits of our training strategy are not specific to the backbone, training languages, or data scale used in our main experiments. Rather, they persist with a different LLM backbone and under lower-resource training scenarios.

6 Related Work

Other recent studies have leveraged SeqKD to transfer the generation capabilities of text LLMs to the speech modality without requiring manually annotated speech-instruction responses (Fathullah et al., 2024; Lu et al., 2026; Shao et al., 2026; Wang et al., 2024; 2026a; Cuervo et al., 2026), with different strategies in producing the teacher target.

AudioChatLLaMA (Fathullah et al., 2024), BLSP-KD (Wang et al., 2024), and AZeroS (Shao et al., 2026) match our PF-SeqKD setting, while DeSTA2.5 (Lu et al., 2026) and (Yu et al., 2024) are closer to our PC-SeqKD setting. Similarly, Zhou et al. (2026) generate teacher continuations from textual descriptions of the audio, obtained either from paired text or an audio captioning model. However, none of these studies investigate the ASR capabilities of the resulting SpeechLLM, which we identify as a key limitation of plain SeqKD (see §5.1) and address with our proposed training strategy. Moreover, prior studies primarily focus on EN speech, whereas we study multilingual training and cross-lingual generalization to languages unseen during speech training.

A related line of work leverages ASR data to transfer text-based capabilities to the speech modality through speech-text representation alignment. Several works (Deng et al., 2025; Held et al., 2025; Mohapatra et al., 2026; Züfle & Niehues, 2025) align speech and text representations at both the input and output of the LLM by minimizing the Euclidean or Wasserstein distance between their embeddings. This approach has also been extended to multilingual speech through language-aware projections (Gopal et al., 2026). However, some methods still require a fine-tuning stage on IF speech data (Mohapatra et al., 2026; Züfle & Niehues, 2025). In addition, these approaches require an additional LLM forward pass with the transcript as input during training to obtain target text embeddings, increasing computational and memory costs. Lastly, most of them (Deng et al., 2025; Held et al., 2025; Mohapatra et al., 2026; Gopal et al., 2026) rely on compression mechanisms to reduce the speech sequence length, requiring exact or approximate alignment with the text sequence. Besides tightening their training methodology with a specific adapter architecture, this forces aggressive compression of the speech sequence, which has been demonstrated to be detrimental for task performance, in particular ASR (Verdini et al., 2025).

7 Conclusion

We investigated whether SpeechLLMs with instruction-following (IF) capabilities can be trained without dedicated speech IF data, by using ASR data and Sequence-Level Knowledge Distillation (SeqKD). Our experiments, covering both prompt-free and prompt-conditioned solutions, show that SeqKD effectively transfers a text LLM's linguistic and IF capabilities to speech. However, we demonstrated that ASR performance is poor for SpeechLLMs trained with this method, mainly due to semantically faithful paraphrasing rather than recognition errors, indicating that SeqKD encourages preserving meaning over exact wording. This finding motivated our proposed training strategy: retain prompt-free SeqKD as the main training strategy while introducing limited ASR and prompt-conditioned data to recover faithful speech recognition without compromising IF abilities. A single-stage training with 10% ASR data and 10% prompt-conditioned data reduces average WER by 67.11% relative to pure prompt-free SeqKD while preserving ST and SQA performance, and outperforms a two-stage training schedule in which prompt-free SeqKD is followed by fine-tuning on the same data. The resulting capabilities also extend beyond the languages represented in speech training. Despite training on only $\sim$3K hours of English and Spanish speech, our models perform ST and SQA involving unseen output languages, including German, Italian, and Chinese. In addition, they outperform a cascade solution built with the same components (speech model and LLM). Future work should confirm these findings in settings with broader language coverage on the speech side, and larger-scale training schedules.

AI USE STATEMENT

Generative AI tools (Claude Sonnet 4.6) were used to generate paraphrases of five manually authored ASR prompts, producing the 20- and 200-prompt pools described in Appendix B. An AI assistant (GPT Luna 5.6) was used to improve the readability of the paper and to assist the authors in the development of the codebase. Additionally, Grammarly was used to check the text for errors. All AI-assisted outputs were reviewed by the authors, and the authors take responsibility for the final content, claims, and artifacts of this work.

8 ETHICS STATEMENT

The experiments use publicly available datasets and do not collect new human-subject data.

9 REPRODUCIBILITY STATEMENT

We provide the source code and experimental configurations as additional material, including instructions for creating the SeqKD datasets and training the SpeechLLMs. The LLMs and speech encoder used in our experiments are publicly available on Hugging Face.

REFERENCES

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 Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis (eds.), *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pp. 4218–4222, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL https://aclanthology.org/2020.lrec-1.520/.

Siddhant Arora, Kai-Wei Chang, Chung-Ming Chien, Yifan Peng, Haibin Wu, Yossi Adi, Emmanuel Dupoux, Hung-Yi Lee, Karen Livescu, and Shinji Watanabe. On the landscape of spoken language models: A comprehensive survey. *arXiv preprint arXiv:2504.08528*, 2025.

Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen2-audio technical report, 2024. URL https://arxiv.org/abs/2407.10759.

Santiago Cuervo, Skyler Seto, Maureen de Seyssel, Richard Bai, Zijin Gu, Tatiana Likhomanenko, Navdeep Jaitly, and Zakaria Aldeneh. Closing the gap between text and speech understanding in llms. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 146313–146335, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/ec4c05921d70202023898568a2a8c002-Paper-Conference.pdf.

Wenqian Cui, Dianzhi Yu, Xiaoqi Jiao, Ziqiao Meng, Guangyan Zhang, Qichao Wang, Steven Y. Guo, and Irwin King. Recent advances in speech language models: A survey. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13943–13970, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.682. URL https://aclanthology.org/2025.acl-long.682/.

Keqi Deng, Guangzhi Sun, and Phil Woodland. Wav2Prompt: End-to-end speech prompt learning and task-based fine-tuning for text-based LLMs. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *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)*, pp. 6940–6956, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.354. URL https://aclanthology.org/2025.naacl-long.354/.

Ding Ding, Zeqian Ju, Yichong Leng, Songxiang Liu, Tong Liu, Zeyu Shang, Kai Shen, Wei Song, Xu Tan, Heyi Tang, et al. Kimi-audio technical report. *arXiv preprint arXiv:2504.18425*, 2025.

Qingkai Fang, Shoutao Guo, Yan Zhou, Zhengrui Ma, Shaolei Zhang, and Yang Feng. LLaMA-omni: Seamless speech interaction with large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=PYmrUQmMEw.

Yassir Fathullah, Chunyang Wu, Egor Lakomkin, Ke Li, Junteng Jia, Yuan Shangguan, Jay Mahadeokar, Ozlem Kalinli, Christian Fuegen, and Mike Seltzer. AudioChatLlama: Towards general-purpose speech abilities for LLMs. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5522–5532, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.309. URL https://aclanthology.org/2024.naacl-long.309/.

Marco Gaido, Sara Papi, Luisa Bentivogli, Alessio Brutti, Mauro Cettolo, Roberto Gretter, Marco Matassoni, Mohamed Nabih, and Matteo Negri. Mosel: 950,000 hours of speech data for open-source speech foundation model training on eu languages. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 13934–13947, 2024a.

Marco Gaido, Sara Papi, Matteo Negri, and Luisa Bentivogli. Speech translation with speech foundation models and large language models: What is there and what is missing? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14760–14778, 2024b.

Shreyas Gopal, Donghang Wu, Ashutosh Anshul, Yeo Yue Heng, Yizhou Peng, Haoyang Li, Hexin Liu, and Eng Siong Chng. Language-aware distillation for multilingual instruction-following speech llms with asr-only supervision, 2026. URL https://arxiv.org/abs/2603.07025.

William Held, Yanzhe Zhang, Minzhi Li, Weiyan Shi, Michael J Ryan, and Diyi Yang. Distilling an end-to-end voice assistant without instruction training data. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7876–7891, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.388. URL https://aclanthology.org/2025.acl-long.388/.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.

Javier Iranzo-Sánchez, Joan Albert Silvestre-Cerda, Javier Jorge, Nahuel Roselló, Adria Giménez, Albert Sanchis, Jorge Civera, and Alfons Juan. 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)*, pp. 8229–8233. IEEE, 2020.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In Jian Su, Kevin Duh, and Xavier Carreras (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1317–1327, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1139. URL https://aclanthology.org/D16-1139/.

Zhifeng Kong, Sang gil Lee, Jaehyeon Kim, Boxin Wang, Zihan Liu, Sungwon Kim, Yang Chen, Arushi Goel, Rajarshi Roy, Wenliang Dai, Zhuolin Yang, Yangyi Chen, Dongfu Jiang, Sreyan Ghosh, Tuomas Rintamaki, Andrew Tao, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. Unified audio intelligence without regressing on text intelligence, 2026. URL https://arxiv.org/abs/2607.05196.

Kalle Kujanpää, Pekka Marttinen, Harri Valpola, and Alexander Ilin. Efficient knowledge injection in llms via self-distillation. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=drYpdSnRJk.

Siddique Latif, Moazzam Shoukat, Fahad Shamshad, Muhammad Usama, Yi Ren, Heriberto Cuayáhuitl, Wenwu Wang, Xulong Zhang, Roberto Togneri, Erik Cambria, et al. Sparks of large audio models: A survey and outlook. *arXiv preprint arXiv:2308.12792*, 2023.

Xinjian Li, Shinnosuke Takamichi, Takaaki Saeki, William Chen, Sayaka Shiota, and Shinji Watanabe. Yodas: Youtube-oriented dataset for audio and speech. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 1–8, 2023. doi: 10.1109/ASRU57964.2023.10389689.

Alexander H Liu, Andy Ehrenberg, Andy Lo, Clément Denoix, Corentin Barreau, Guillaume Lample, Jean-Malo Delignon, Khyathi Raghavi Chandu, Patrick von Platen, Pavankumar Reddy Muddireddy, et al. Voxtral. *arXiv preprint arXiv:2507.13264*, 2025.

Yuchen Liu, Hao Xiong, Jiajun Zhang, Zhongjun He, Hua Wu, Haifeng Wang, and Chengqing Zong. End-to-End Speech Translation with Knowledge Distillation. In *Interspeech 2019*, pp. 1128–1132, 2019. doi: 10.21437/Interspeech.2019-2582.

Ke-Han Lu, Zhehuai Chen, Szu-Wei Fu, Chao-Han Huck Yang, Sung-Feng Huang, Chih-Kai Yang, Chee-En Yu, Chun-Wei Chen, Wei-Chih Chen, Chien-yu Huang, Yi-Cheng Lin, Yu-Xiang Lin, Chi-An Fu, Chun-Yi Kuan, Wenze Ren, Xuanjun Chen, Wei-Ping Huang, En-Pei Hu, Tzu-Quan Lin, Yuan-Kuei Wu, Kuan-Po Huang, Hsiao-Ying Huang, Huang-Cheng Chou, Kai-Wei Chang, Cheng-Han Chiang, Boris Ginsburg, Yu-Chiang Frank Wang, and Hung-yi Lee. Desta2.5-audio: Toward general-purpose large audio language model with self-generated cross-modal alignment. *IEEE Transactions on Audio, Speech and Language Processing*, 34:2062–2076, 2026. doi: 10.1109/TASLPRO.2026.3675792.

Taiming Lu and Zhuang Liu. Strong teacher not needed? on distillation in llm pretraining, 2026. URL https://arxiv.org/abs/2605.23857.

Pedro Henrique Martins, João Alves, Patrick Fernandes, Nuno M. Guerreiro, Ricardo Rei, Amin Farajian, Mateusz Klimaszewski, Duarte M. Alves, José Pombal, Nicolas Boizard, Manuel Faysse, Pierre Colombo, François Yvon, Barry Haddow, José G. C. de Souza, Alexandra Birch, and André F. T. Martins. Eurollm-9b: Technical report, 2025. URL https://arxiv.org/abs/2506.04079.

Microsoft, :, Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, Dong Chen, Dongdong Chen, Junkun Chen, Weizhu Chen, Yen-Chun Chen, Yi ling Chen, Qi Dai, Xiyang Dai, Ruchao Fan, Mei Gao, Min Gao, Amit Garg, Abhishek Goswami, Junheng Hao, Amr Hendy, Yuxuan Hu, Xin Jin, Mahmoud Khademi, Dongwoo Kim, Young Jin Kim, Gina Lee, Jinyu Li, Yunsheng Li, Chen Liang, Xihui Lin, Zeqi Lin, Mengchen Liu, Yang Liu, Gilsinia Lopez, Chong Luo, Piyush Madan, Vadim Mazalov, Arindam Mitra, Ali Mousavi, Anh Nguyen, Jing Pan, Daniel Perez-Becker, Jacob Platin, Thomas Portet, Kai Qiu, Bo Ren, Liliang Ren, Sambuddha Roy, Ning Shang, Yelong Shen, Saksham Singhal, Subhojit Som, Xia Song, Tetyana Sych, Praneetha Vaddamanu, Shuohang Wang, Yiming Wang, Zhenghao Wang, Haibin Wu, Haoran Xu, Weijian Xu, Yifan Yang, Ziyi Yang, Donghan Yu, Ishmam Zabir, Jianwen Zhang, Li Lyna Zhang, Yunan Zhang, and Xiren Zhou. Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras, 2025. URL https://arxiv.org/abs/2503.01743.

Biswesh Mohapatra, Marcely Zanon Boito, and Ioan Calapodescu. Speechmapper: Speech-to-text embedding projector for llms. In *ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 16277–16281, 2026. doi: 10.1109/ICASSP55912.2026.11463274.

Sara Papi, Maike Züfle, Marco Gaido, Beatrice Savoldi, Danni Liu, Ioannis Douros, Luisa Bentivogli, and Jan Niehues. MCIF: Multimodal crosslingual instruction-following benchmark from scientific talks. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=PtPYZYfa0h.

Vineel Pratap, Qiantong Xu, Anuroop Sriram, Gabriel Synnaeve, and Ronan Collobert. Mls: A large-scale multilingual dataset for speech research. *ArXiv*, abs/2012.03411, 2020.

Nithin Rao Koluguri, Monica Sekoyan, George Zelenfroynd, Sasha Meister, Shuoyang Ding, Sofia Kostandian, He Huang, Nikolay Karpov, Jagadeesh Balam, Vitaly Lavrukhin, Yifan Peng, Sara Papi, Marco Gaido, Alessio Brutti, and Boris Ginsburg. Granary: Speech Recognition and Translation Dataset in 25 European Languages. In *Interspeech 2025*, pp. 3923–3927, 2025. doi: 10.21437/Interspeech.2025-540.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. COMET: A neural framework for MT evaluation. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2685–2702, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.213. URL https://aclanthology.org/2020.emnlp-main.213/.

Monica Sekoyan, Nithin Rao Koluguri, Nune Tadevosyan, Piotr Zelasko, Travis Bartley, Nikolay Karpov, Jagadeesh Balam, and Boris Ginsburg. Canary-1b-v2 and parakeet-tdt-0.6b-v3: Efficient and high-performance models for multilingual asr and ast, 2025. URL https://arxiv.org/abs/2509.14128.

Yiwen Shao, Wei Liu, Jiahong Li, Tianzi Wang, Kun Wei, Meng Yu, and Dong Yu. Azeros: Extending LLM to speech with self-generated instruction-free tuning. *CoRR*, abs/2601.06086, 2026. doi: 10.48550/ARXIV.2601.06086. URL https://doi.org/10.48550/arXiv.2601.06086.

Yu Shu, Siwei Dong, Guangyao Chen, Wenhao Huang, Ruihua Zhang, Daochen Shi, Qiqi Xiang, and Yemin Shi. Llasm: Large language and speech model, 2023. URL https://arxiv.org/abs/2308.15930.

Matthias Sperber and Matthias Paulik. Speech translation and the end-to-end promise: Taking stock of where we are. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7409–7421, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.661. URL https://aclanthology.org/2020.acl-main.661/.

Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. Salmonn: Towards generic hearing abilities for large language models, 2024. URL https://arxiv.org/abs/2310.13289.

Francesco Verdini, Pierfrancesco Melucci, Stefano Perna, Francesco Cariaggi, Marco Gaido, Sara Papi, Szymon Mazurek, Marek Kasztelnik, Luisa Bentivogli, Sebastien Bratières, Paolo Merialdo, and Simone Scardapane. How to Connect Speech Foundation Models and Large Language Models? What Matters and What Does Not. In *Interspeech 2025*, pp. 1813–1817, 2025. doi: 10.21437/Interspeech.2025-2245.

Changhan Wang, Anne Wu, and Juan Pino. Covost 2 and massively multilingual speech-to-text translation. *arXiv preprint arXiv:2007.10310*, 2020.

Changhan Wang, Morgane Riviere, Ann Lee, Anne Wu, Chaitanya Talnikar, Daniel Haziza, Mary Williamson, Juan Pino, and Emmanuel Dupoux. 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)*, pp. 993–1003, 2021.

Chen Wang, Minpeng Liao, Zhongqiang Huang, and Jiajun Zhang. Blsp-kd: Bootstrapping language-speech pre-training via knowledge distillation. *arXiv preprint arXiv:2405.19041*, 2024.

Dingdong Wang, Junan Li, Mingyu Cui, Dongchao Yang, Xueyuan Chen, and Helen M. Meng. Speech discrete tokens or continuous features? a comparative analysis for spoken language understanding in SpeechLLMs. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 24913–24924, Suzhou, China, November 2025a. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1266. URL https://aclanthology.org/2025.emnlp-main.1266/.

Dingdong Wang, Junan Li, Mingyu Cui, Dongchao Yang, Xueyuan Chen, and Helen M. Meng. Speech discrete tokens or continuous features? a comparative analysis for spoken language understanding in SpeechLLMs. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 24913–24924, Suzhou, China, November 2025b. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1266. URL https://aclanthology.org/2025.emnlp-main.1266/.

Enzhi Wang, Qicheng Li, Zhiyuan Tang, and Yuhang Jia. Cross-modal knowledge distillation for speech large language models. In *ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 16112–16116, 2026a. doi: 10.1109/ICASSP55912.2026.11464272.

Yingzhi Wang, Reem Alhazzani, and Muhammad Alqurishi. Nuha-speech: Building general-purpose arabic speech-llms, 2026b. URL https://arxiv.org/abs/2609.11892.

Zhifei Xie and Changqiao Wu. Mini-omni: Language models can hear, talk while thinking in streaming. *arXiv preprint arXiv:2408.16725*, 2024.

Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, Yuanjun Lv, Yongqi Wang, Dake Guo, He Wang, Linhan Ma, Pei Zhang, Xinyu Zhang, Hongkun Hao, Zishan Guo, Baosong Yang, Bin Zhang, Ziyang Ma, Xipin Wei, Shuai Bai, Keqin Chen, Xuejing Liu, Peng Wang, Mingkun Yang, Dayiheng Liu, Xingzhang Ren, Bo Zheng, Rui Men, Fan Zhou, Bowen Yu, Jianxin Yang, Le Yu, Jingren Zhou, and Junyang Lin. Qwen3-omni technical report, 2025. URL https://arxiv.org/abs/2509.17765.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Yuming Yang, Mingyoung Lai, Wanxu Zhao, Xiaoran Fan, Zhiheng Xi, Mingqi Wu, Chiyue Huang, Jun Zhao, Haijun Lv, Jian Tong, Yunhua Zhou, Yicheng Zou, Qipeng Guo, Tao Gui, Qi Zhang, and Xuanjing Huang. Which reasoning trajectories teach students to reason better? a simple metric of informative alignment. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 42123–42150, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1950. URL https://aclanthology.org/2026.acl-long.1950/.

Tengfei Yu, Xuebo Liu, Zhiyi Hou, Liang Ding, Dacheng Tao, and Min Zhang. Self-powered LLM modality expansion for large speech-text models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 12401–12417, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.690. URL https://aclanthology.org/2024.emnlp-main.690/.

Dong Zhang, Shimin Li, Xin Zhang, Jun Zhan, Pengyu Wang, Yaqian Zhou, and Xipeng Qiu. Speechgpt: Empowering large language models with intrinsic cross-modal conversational abilities, 2023. URL https://arxiv.org/abs/2305.11000.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SkeHuCVFDr.

Xuanru Zhou, Yiwen Shao, Jiahong Li, and Dong Yu. Alignment is all you need: Instruction-free training for general audio-language models, 2026. URL https://arxiv.org/abs/2608.18132.

Maike Züfle and Jan Niehues. Contrastive learning for task-independent SpeechLLM-pretraining. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 8469–8490, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.445. URL https://aclanthology.org/2025.findings-acl.445/.

A DeSTA 2.5 Prompts

Unlike the manually authored ASR prompts above, the prompts used for PC-SeqKD are drawn from the publicly released DeSTA 2.5 (Lu et al., 2026) prompt pool. These prompts are open-ended audio-understanding requests rather than ASR instructions, asking for summaries or descriptions of the audio content instead of verbatim transcription. For each training example, we sample one prompt q_i uniformly from this pool and provide the same prompt to the teacher and the student. The prompt pool is publicly available on HuggingFace.[8]

B ASR Prompts

The following five prompts are handcrafted and are included in the ASR prompt pools:

- Repeat after me, only output the transcription without anything else.
- Please transcribe the following audio.
- Could you please transcribe this for me?
- Run automatic speech recognition on this input.
- Turn the speech into text.

Starting from them, we then used `Claude Sonnet 4.6` to rephrase the requests in different ways and generate the 20- and 200-prompt pools.

B.1 20-Prompt Pool

The complete 20-prompt pool is listed below.

[1] Repeat after me, only output the transcription without anything else.
[2] Transcribe.
[3] Speech to text.
[4] Audio to text.
[5] Generate transcript.
[6] Please transcribe the following audio.
[7] Convert speech to text.
[8] Type out everything that is said.
[9] Provide a text version of the audio.
[10] Write the spoken content.
[11] Turn the speech into text.
[12] Write down what you hear.
[13] Listen and write the words.
[14] Could you please transcribe this for me?
[15] Can you turn this speech into text, please?
[16] Help me turn this audio into text.
[17] Perform ASR on the provided audio stream.
[18] Execute speech-to-text conversion.
[19] Run automatic speech recognition on this input.
[20] Provide a verbatim transcript.

B.2 200-Prompt Pool

The complete 200-prompt pool is listed below.

[1] Repeat after me, only output the transcription without anything else.
[2] Transcribe.
[3] Transcribe this.
[4] Audio to text.
[5] Speech to text.
[6] Dictation.
[7] Write this.
[8] Generate transcript.
[9] Text output.
[10] Write.
[11] Listen and type.
[12] Output text.
[13] Convert to text.
[14] Generate transcription.
[15] Process audio.
[16] Type it.
[17] Transcribe voice.
[18] Transcribe speaker.
[19] Text extraction from audio.
[20] Convert voice recording.
[21] Write the audio down.
[22] Please transcribe the following audio.
[23] Convert speech to text.
[24] Listen and write the words.
[25] Type out everything that is said.
[26] Provide a text version of the audio.
[27] Put into words what you hear.

[8] https://huggingface.co/datasets/DeSTA-ntu/DeSTA-AQA5M-FROM-Llama3.1-8B-Instruct

[28] Write the spoken content.

[29] Write down what you hear.

[30] Give me the transcription of this audio.

[31] Provide the written text for this audio clip.

[32] Write down the exact words spoken here.

[33] Produce a text representation of the speech.

[34] Type out the spoken words.

[35] Translate this audio recording into text.

[36] Turn this voice recording into written words.

[37] Create a transcript from this audio file.

[38] Listen to the recording and write what you hear.

[39] Transcribe the voice content.

[40] Type everything the speaker says.

[41] Output the exact words from the audio.

[42] Change this spoken clip into a written format.

[43] Convert the following voice note to text.

[44] Write down the utterance exactly.

[45] Transform the audio signal into written language.

[46] Render the speech as text.

[47] Capture every word spoken.

[48] Produce a verbatim transcript.

[49] Document the spoken words.

[50] Write the transcript.

[51] Convert to readable text.

[52] Output the spoken sentence.

[53] Write the phrase being spoken.

[54] Type the sentence you hear.

[55] Extract the spoken sentence.

[56] Generate the transcript string.

[57] Write the utterance.

[58] Provide text for the voice.

[59] Generate text from sound.

[60] Could you please transcribe this for me?

[61] I need you to write down what is in this audio.

[62] Turn the speech into text.

[63] Please listen to this and type out the words.

[64] I would appreciate a transcription of this clip.

[65] Could you provide a text version of what's being said?

[66] Please convert this voice note into a written message.

[67] Would you mind writing down the contents of this audio?

[68] I need a transcript of this recording, please.

[69] Can you please transcribe the spoken dialogue?

[70] Please generate a text log of this audio.

[71] Kindly write down what you hear.

[72] Could you translate this audio into a text format?

[73] Please extract the spoken words and write them down.

[74] I'd like you to type out the speech from this file.

[75] Help me turn this audio into text.

[76] Please document what the speaker is saying.

[77] Could you assist me by transcribing this?

[78] I need the written equivalent of this audio, please.

[79] Kindly provide the transcription.

[80] Tell me what this audio says.

[81] What are they saying in this clip?

[82] Can you catch what's being said?

[83] Put this voice note in writing.

[84] Give me the written version.

[85] I need the words from this sound file.

[86] What is the spoken phrase?

[87] What words are spoken here?

[88] Write what you hear please.

[89] Perform ASR on the provided audio stream.

[90] Execute speech-to-text conversion.

[91] Run automatic speech recognition on this input.

[92] Decode the acoustic signal into a text sequence.

[93] Generate ASR output.

[94] Process spoken language input.

[95] Recognize speech and output string.

[96] Initialize transcription sequence.

[97] Parse the audio track into text.

[98] Perform dictate-to-text operation.

[99] Analyze audio and return transcript.

[100] Convert waveform to text sequence.

[101] Run audio decoding.

[102] Transcribe vocalization input.

[103] Return textual representation of the audio tensor.

[104] Acoustic-to-text translation required.

[105] Apply speech recognition module.

[106] System prompt: Transcribe audio input.

[107] Task: Speech Transcription.

[108] Audio decoding requested.

[109] Provide a highly accurate transcription.

[110] Write down every single word spoken.

[111] Transcribe this with exact precision.

[112] Give me a verbatim text of the audio.

[113] Listen closely and type exactly what you hear.

[114] Capture the speech without missing a word.

[115] Provide an exact text match for the audio.

[116] Write down the exact sentence spoken.

[117] Ensure a perfectly accurate transcription.

[118] Output the spoken text exactly as it sounds.

[119] Type out the audio exactly word-for-word.

[120] Deliver a precise transcript of the utterance.

[121] Make sure to catch every word in this recording.

[122] Transcribe the utterance perfectly.

[123] Note down what is vocalized here.

[124] Jot down the speech from this clip.

[125] Put the audio message into words.

[126] Draft a transcript for this sound bite.

[127] Spell out what the person is saying.

[128] Translate the spoken words onto the page.

[129] Make a written record of this sound file.

[130] Copy down the words spoken in the recording.

[131] Write the dialogue you hear.

[132] Supply the text equivalent of this audio.

[133] Generate a readable version of this speech.

[134] Turn the vocal audio into written format.

[135] Construct a text document from the speech.

[136] Let me see the text for this audio.

[137] Provide the script of this recording.

[138] Write down the dialogue.

[139] Provide a full transcript.

[140] I need a verbatim copy of this speech.

[141] Transcribe this vocal track.

[142] Extract the dialogue.

[143] Type out the vocalization.

[144] Convert the spoken content.

[145] Transcribe the attached audio.

[146] Do a speech-to-text conversion.

[147] Write the accompanying text.

[148] Create a text copy of the audio.

[149] Provide the spoken content as a string.

[150] Type out what the voice is saying.

[151] Translate the sound into a sentence.

[152] Write down the audio track.

[153] Transcribe this short clip.

[154] Give me the words for this speech.

[155] Output the transcribed text.

[156] Return the transcript.

[157] Listen and transcribe.

[158] Convert to writing.

[159] Make a text version.

[160] Give me the speech transcript.

[161] Write the speech.

[162] Turn audio into writing.

[163] Provide the transcription of the speech.

[164] Write out the words.

[165] Listen, then write.

[166] Transcribe the vocal recording.

[167] Convert the vocalization to text.

[168] Provide the text for the audio clip.

[169] Write the recorded speech.

[170] Type the spoken audio.

[171] Transcribe the audio file.

[172] Generate the written transcript.

[173] Output the spoken text.

[174] Convert the spoken input.

[175] Provide the audio text.

[176] Write the voice input.

[177] Transcribe the sound.

[178] Convert the sound to words.

[179] Write the spoken text.

[180] Transcribe the voice message.

[181] You are a transcriber; write out this audio.

[182] Act as a speech-to-text engine.

[183] Your task is to transcribe this clip.

[184] As an ASR system, decode this speech.

[185] Pretend you are taking dictation and write this down.

[186] Your job is to convert this audio to text.

[187] Listen like a court reporter and type the speech.

[188] Be my transcriber for this voice note.

[189] Transcribe the audio accurately.

[190] Audio transcription task.

[191] You are an expert transcriber; write down this recording.

[192] Act as an automatic transcription service.

[193] Please serve as my transcription assistant.

[194] Listen to this audio and produce its transcript.

[195] As a transcription system, convert this speech into text.

[196] Transcribe the recording exactly as spoken.

[197] Your task is to write down every word in this audio.

[198] Please decode the speaker's words into text.

[199] Convert this spoken message into a transcript.

[200] Return only the transcript of this recording.

C Fine-Tuning Configuration

Fine-tuning follows the procedure described in E. We initialize from either the PF-SeqKD or PC-SeqKD baseline model, trained on D_{PF} or D_{PC}, respectively, and fine-tune for two epochs on $ASR + PC - Aug$. This dataset combines the ASR and prompt-conditioned samples that are merged with D_{PF} and D_{PC} during single-stage training. We use the same optimization configuration as in single-stage training: AdamW with a learning rate of 5×10^{-4} and weight decay of 1×10^{-3}, together with a cosine learning-rate schedule and 1,000 warmup steps. The total batch size is 128.

D Data Pre-Processing

We filter the Common Voice dataset to remove low-quality samples that could negatively impact model performance. Specifically, we discard samples with more than one downvote or a downvote-to-upvote ratio exceeding 0.3. This led to filtering out 19736 samples, corresponding to 37 hours of speech from the English dataset, and 1929 samples, corresponding to 4 hours of speech, from the Spanish dataset.

Additionally, to limit memory requirements during training, we follow the approach used in the Shao et al. (2026) codebase[9] on all datasets. Specifically, we filter out samples longer than 30 seconds and shorter than 1 second. Moreover, we filter out the samples for which the length of the transcription is greater than the duration of the audio multiplied by 30. This led to filtering out 3716 samples, corresponding to 19 hours of speech for the English samples and 1223 samples corresponding to 6 hours for the Spanish dataset.

E Training Configuration

The adapter was trained for 2 epochs with a total batch size of 128. We used the AdamW optimizer with a learning rate of 5×10^{-4} and a weight decay of 1×10^{-3}. The learning rate followed a cosine annealing schedule with a 1,000-step warm-up, decaying to a minimum learning rate of 1×10^{-6}.

F Adapter Architecture

The modality adapter connecting the ASR encoder to the LLM backbone is implemented as a stack of feed-forward blocks, each consisting of a linear projection followed by a GELU activation and dropout. The first block projects the encoder's frame-level representations into the adapter's hidden

[9] https://github.com/AudenAI/Auden/blob/main/examples/azeros/scripts/data_generation/run_data_generation.py

dimension. We experiment with two adapter sizes, varying the hidden dimension and depth to obtain adapters of approximately 22M and 127M trainable parameters, respectively (Table 4).

Configuration	Small	Large
Adapter input dimension	1024	1024
Perception output dimension	2048	2048
Hidden dimension	2560	4704
Number of layers	3	6
Activation	GELU	GELU
Dropout	0.1	0.1
Trainable parameters	22M	127M

Table 4: Architecture of the two MLP adapters used to connect the frozen Canary-1B ASR encoder to the frozen Qwen3-4B-Instruct LLM backbone.

G BACKBONE LLM

Table 5 compares four candidate LLM backbones on the long-context MCIF tasks Papi et al. (2026), i.e., QA, Summarization, and Machine Translation. The models compared are `Qwen3-4B-Instruct-2507`, `Qwen3.5-4B`, `Qwen3.5-9B`, `Qwen3-14B` and `EuroLLM-9B-Instruct-2512`.

Although `Qwen3-14B` achieves the strongest overall scores, we use `Qwen3-4B-Instruct-2507` (Yang et al., 2025) because it provides a better performance and computational cost trade-off. Its parameters remain frozen throughout SpeechLLM training. Moreover, we also use `EuroLLM-9B-Instruct-2512` to test whether our method generalizes to other LLM Backbones.

Models	Long-EN$_{fix}$		Long-IT$_{fix}$			Long-DE$_{fix}$			Long-ZH$_{fix}$		
	QA-BERTS.↑	SUM-BERTS.↑	QA-BERTS.↑	SUM-BERTS.↑	Trans-Comet↑	QA-BERTS.↑	SUM-BERTS.↑	Trans-Comet↑	QA-BERTS.↑	SUM-BERTS.↑	Trans-Comet↑
Qwen3-4B	0.30	0.18	0.32	0.10	0.83	0.33	0.10	0.80	0.25	0.24	0.81
Qwen3.5-4B	0.27	0.21	0.24	0.12	0.84	0.27	**0.12**	0.81	0.20	**0.35**	0.82
Qwen3.5-9B	0.29	0.21	0.26	0.19	0.85	0.26	**0.12**	0.80	0.23	**0.35**	0.82
Qwen3-14B	**0.37**	**0.22**	**0.35**	**0.22**	**0.86**	**0.40**	**0.12**	0.81	**0.36**	0.22	**0.85**
EuroLLM-9B	0.18	0.15	0.20	0.09	0.78	0.19	0.17	**0.85**	0.16	0.24	0.68

Table 5: Performance of different LLMs on the Long Context tasks of the MCIF benchmark. `Qwen3-14B` achieves the highest scores across all metrics and languages. Given the small gains relative to its added computational cost, we adopt `Qwen3-4B-Instruct-2507` for our pipeline, as it provides the best trade-off between performance and computational efficiency.

H ADDITIONAL RESULTS

In this section, we report fine-grained results across all tasks and languages for both the 127M and 22M adapters.

H.1 EXTENDED RESULTS FOR PF-SEQKD VS PC-SEQKD

This subsection extends the analysis in Section 5.1 by reporting the fine-grained results for both adapter sizes, separated by task and language. The discussion below focuses on the two SeqKD baselines, though we also report the results obtained using the external multimodal models (`Qwen3-Omni`, `Phi-4-Multimodal`, and `GPT-Audio 1.5`).

H.1.1 22M ADAPTER

Tables 6, 7, and 8 report the ST, ASR, and SQA baselines for the 22M adapter size.

ST scores are stable across the two SeqKD baselines and language directions, with COMET values between 0.754 and 0.862 depending on the language pairs. SQA scores are more variable, and the

Method	COMET ↑											
	$MCIF_{\text{MIX}}$			Europarl-ST						CoVoST 2		
SeqKD baselines												
PF-SeqKD	0.8122	0.7569	0.8397	0.8386	0.8575	0.8485	0.7542	0.8249	0.8212	0.8195	0.8567	0.8594
PC-SeqKD	0.8136	0.7737	0.8307	0.8434	0.8615	0.8547	0.7583	0.8253	0.8244	0.8169	0.8519	0.8599
External Models												
`Qwen3-Omni`	0.8536	0.8196	0.8584	0.8767	0.8846	0.8845	0.8128	0.8471	0.8603	0.8633	0.8905	0.8864
`Phi-4-Multimodal`	0.7870	0.7538	0.7925	0.8282	0.8463	0.8319	0.6612	0.8033	0.6979	0.8261	0.8424	0.8544
`GPT-Audio 1.5`	0.8419	0.8075	0.8351	0.8508	0.8576	0.8567	0.7872	0.8256	0.8350	0.7938	0.8044	0.8464

Table 6: Speech Translation results for the Baseline SpeechLLMs with 22M adapter measured using the COMET score alongside the external multimodal models `Qwen3-Omni`, `Phi-4-Multimodal` and `GPT-Audio 1.5`. For each dataset, we report the different source- target language pairs.

Method	WER ↓				
	$MCIF_{\text{MIX}}$	Europarl		Common Voice	
SeqKD baselines					
PF-SeqKD	31.02%	52.06%	70.44%	19.63%	68.33%
PC-SeqKD	38.18%	30.08%	75.56%	29.25%	84.92%
External Models					
`Qwen3-Omni`	12.50%	21.20%	23.21%	9.50%	4.53%
`Phi-4-Multimodal`	9.98%	12.50%	10.01%	9.43%	6.03%
`GPT-Audio 1.5`	6.81%	11.54%	9.85%	12.08%	9.34%

Table 7: ASR results for the Baseline SpeechLLMs with 22M adapter measured using WER. We also report the external multimodal models `Qwen3-Omni`, `Phi-4-Multimodal` and `GPT-Audio 1.5`. For each dataset, we report the scores per language.

Method	BERTScore ↑			
	$MCIF_{\text{MIX}}$			
SeqKD baselines				
PF-SeqKD	0.3261	0.2997	0.3288	0.2276
PC-SeqKD	0.2998	0.2536	0.2834	0.2606
External Models				
`Qwen3-Omni`	0.3050	0.3217	0.3212	0.1981
`Phi-4-Multimodal`	0.3845	0.3289	0.3383	0.3531
`GPT-Audio 1.5`	0.3284	0.3064	0.3257	0.1902

Table 8: SQA results for the Baseline SpeechLLMs with 22M adapter measured using BERTScore. We also report the external multimodal models `Qwen3-Omni`, `Phi-4-Multimodal` and `GPT-Audio 1.5`.

PF-SeqKD baseline offers better results. The clearest limitation is ASR: WER ranges from 19.63% to 84.92%, with Spanish consistently producing higher error rates than English on both Europarl-ST and Common Voice. As we underlined in the paper, plain SeqKD transfers translation and generation capabilities more effectively, while ASR capability is not transferred.

H.1.2 127M ADAPTER

Tables 9, 10, and 11 report the corresponding results for the 127M adapter. Increasing the adapter size leaves ST largely unchanged, as the COMET scores remain in the same range as those obtained with the 22M adapter. On SQA, the larger adapter improves the PF-SeqKD scores on $MCIF_{\text{MIX}}$, but does not yield a uniform improvement for PC-SeqKD. ASR improves substantially. For PF-SeqKD, WER decreases from 52.06% to 21.82% on Europarl-ST English and from 68.33% to 27.09% on Common Voice Spanish; for PC-SeqKD, it decreases from 30.08% to 23.80% on Europarl-ST English and from 84.92% to 61.70% on Common Voice Spanish. Nevertheless, the model is still far from offering good recognition abilities.

Method	COMET ↑											
	$MCIF_{\text{MIX}}$			Europarl-ST						CoVoST 2		
SeqKD baselines												
PF-SeqKD	0.8121	0.7716	0.8401	0.8433	0.8604	0.8517	0.7582	0.8228	0.8223	0.8190	0.8559	0.8583
PC-SeqKD	0.8124	0.7678	0.8336	0.8447	0.8603	0.8562	0.7610	0.8261	0.8232	0.8187	0.8546	0.8617
External Models												
`Qwen3-Omni`	0.8536	0.8196	0.8584	0.8767	0.8846	0.8845	0.8128	0.8471	0.8603	0.8633	0.8905	0.8864
`Phi-4-Multimodal`	0.7870	0.7538	0.7925	0.8282	0.8463	0.8319	0.6612	0.8033	0.6979	0.8261	0.8424	0.8544
`GPT-Audio 1.5`	0.8419	0.8075	0.8351	0.8508	0.8576	0.8567	0.7872	0.8256	0.8350	0.7938	0.8044	0.8464

Table 9: Speech Translation results for the Baseline SpeechLLMs with 127M adapter measured using the COMET score alongside the external multimodal models `Qwen3-Omni`, `Phi-4-Multimodal` and `GPT-Audio 1.5`.

Method	WER ↓				
	$MCIF_{\text{MIX}}$	Europarl		Common Voice	
SeqKD baselines					
PF-SeqKD	28.91%	21.82%	61.36%	17.40%	27.09%
PC-SeqKD	34.00%	23.80%	61.70%	23.80%	61.70%
External Models					
`Qwen3-Omni`	12.50%	21.20%	23.21%	9.50%	4.53%
`Phi-4-Multimodal`	9.98%	12.50%	10.01%	9.43%	6.03%
`GPT-Audio 1.5`	6.81%	11.54%	9.85%	12.08%	9.34%

Table 10: ASR results for the Baseline SpeechLLMs with 127M adapter measured using WER. We also report the external multimodal models `Qwen3-Omni`, `Phi-4-Multimodal` and `GPT-Audio 1.5`. For each dataset, we report the scores per language.

Method	BERTScore ↑			
	$MCIF_{\text{MIX}}$			
SeqKD baselines				
PF-SeqKD	0.3411	0.3220	0.3372	0.3059
PC-SeqKD	0.2895	0.2832	0.2921	0.2592
External Models				
`Qwen3-Omni`	0.3050	0.3217	0.3212	0.1981
`Phi-4-Multimodal`	0.3845	0.3289	0.3383	0.3531
`GPT-Audio 1.5`	0.3284	0.3064	0.3257	0.1902

Table 11: SQA results for the Baseline SpeechLLMs with 127 adapters measured using BERTScore. We also report the external multimodal models `Qwen3-Omni`, `Phi-4-Multimodal` and `GPT-Audio 1.5`. For MCIF, we report the scores per language.

H.2 Effect of ASR Supervision and Prompt Diversity

This subsection extends Section 5.2 with task-specific results for the two adapter sizes. The tables vary the proportion of ASR data, $p \in \{1\%, 5\%, 10\%, 20\%\}$, and the prompt-pool size, $k \in \{1, 20, 200\}$. The results show how prompt diversity and supervision quantity affect recognition and the downstream ST and SQA tasks.

H.2.1 22M Adapter

Tables 12, 13, and 14 report the ST, ASR, and SQA results for the 22M adapter. ASR benefits primarily from prompt diversity. For PF-SeqKD with $p = 1\%$, increasing k from 1 to 20 reduces MCIF WER from 23.56% to 15.10%, and increasing it to 200 reduces WER further to 9.21%. This is expected because the case in which $k = 1$ leads the model to associate a single prompt with the ASR task. Increasing k, instead, helps the model to better generalize. Once $k \geq 20$, increasing p provides only modest additional gains, with WER generally remaining around 8-10% across the evaluated benchmarks. A consistent effect can be seen for PC-SeqKD.

p	k	COMET ↑											
		$MCIF_{\text{MIX}}$			Europarl-ST						CoVoST 2		
		en→it	en→de	en→zh	en→de	en→es	en→it	es→de	es→en	es→it	en→zh	es→en	
PF-SeqKD													
1%	1	0.8161	0.7689	0.8389	0.8457	0.8604	0.8541	0.7586	0.8254	0.8232	0.8215	0.8573	0.8625
1%	20	0.7964	0.7441	0.8401	0.7770	0.8222	0.8073	0.7222	0.8050	0.7998	0.8198	0.8528	0.8595
1%	200	0.8043	0.7424	0.8328	0.7577	0.8138	0.8029	0.6514	0.7809	0.7709	0.8167	0.8574	0.8572
5%	1	0.813	0.7643	0.8411	0.8289	0.8486	0.8494	0.7271	0.8131	0.8061	0.8198	0.8540	0.8611
5%	20	0.7249	0.7001	0.8297	0.7350	0.7925	0.7694	0.6547	0.7721	0.7721	0.8039	0.8423	0.8273
5%	200	0.6476	0.5374	0.6655	0.5598	0.6688	0.6473	0.5232	0.7467	0.7198	0.7338	0.7123	0.8108
10%	1	0.8167	0.7739	0.8351	0.7792	0.8227	0.8202	0.7029	0.7976	0.7913	0.8192	0.8555	0.8611
10%	20	0.7362	0.7302	0.8030	0.5733	0.6568	0.6303	0.5399	0.6702	0.7299	0.7565	0.7857	0.7941
10%	200	0.6274	0.5822	0.5888	0.5350	0.6474	0.6108	0.5200	0.6655	0.6655	0.7573	0.6734	0.7919
20%	1	0.8069	0.7656	0.8289	0.8041	0.8371	0.8353	0.6809	0.8062	0.7829	0.8184	0.8531	0.8597
20%	20	0.5423	0.4517	0.3424	0.5198	0.6346	0.5928	0.5171	0.6645	0.6904	0.6613	0.6307	0.7929
20%	200	0.5393	0.456	0.2963	0.5206	0.6349	0.5919	0.5158	0.6631	0.6886	0.6494	0.6296	0.7928
PC-SeqKD													
1%	1	0.8160	0.7713	0.8297	0.8458	0.8614	0.8568	0.7663	0.8248	0.8272	0.8184	0.8537	0.8617
1%	20	0.8061	0.7683	0.8297	0.8430	0.8526	0.8488	0.7515	0.8222	0.8112	0.8189	0.8558	0.8602
1%	200	0.8113	0.7692	0.8355	0.8431	0.8525	0.8485	0.7515	0.8217	0.8179	0.8182	0.8559	0.8605
5%	1	0.8144	0.7742	0.8315	0.8357	0.8505	0.8448	0.7560	0.8248	0.8179	0.8202	0.8557	0.8620
5%	20	0.8112	0.7677	0.8389	0.7997	0.8386	0.8290	0.7245	0.8163	0.8003	0.8195	0.8541	0.8613
5%	200	0.8117	0.7702	0.8345	0.7754	0.8319	0.8269	0.6953	0.7960	0.7949	0.8185	0.8543	0.8612
10%	1	0.8178	0.773	0.8328	0.8027	0.8379	0.8363	0.7273	0.8227	0.8073	0.8200	0.8562	0.8631
10%	20	0.7868	0.7503	0.8282	0.7613	0.8194	0.8152	0.6662	0.7661	0.7789	0.8143	0.8464	0.8589
10%	200	0.7702	0.7464	0.7749	0.7463	0.8203	0.8179	0.6609	0.7797	0.7815	0.8119	0.8378	0.8579
20%	1	0.8125	0.7747	0.8361	0.8225	0.8465	0.8340	0.7256	0.8127	0.8037	0.8199	0.8558	0.8628
20%	20	0.8131	0.7716	0.7921	0.8430	0.8589	0.8545	0.7515	0.8221	0.8205	0.8194	0.8569	0.8630
20%	200	0.6412	0.6062	0.7099	0.5323	0.6419	0.6005	0.5309	0.7071	0.7440	0.7307	0.6972	0.7928

Table 12: Speech Translation results for the SpeechLLMs trained on the SeqKD dataset + ASR data using the 22M adapter. Scores are measured using the COMET score. We also report the external multimodal models `Qwen3-Omni`, `Phi-4-Multimodal` and `GPT-Audio 1.5`. For each dataset, we report the different source-target language pairs.

Notably, these recognition gains come with a trade-off in downstream performance, particularly for PF-SeqKD (see Tables 12 and 14). Increasing p and k substantially degrades ST and SQA performance. We attribute this behavior to task overfitting induced by the ASR samples paired with prompts during training. Specifically, the model associates the presence of a prompt with the ASR task and, at inference time, interprets prompted inputs from other tasks as ASR requests. As a result, the model increasingly produces transcriptions instead of following the requested ST or SQA task.

H.2.2 127M Adapter

Tables 15, 16, and 17 report the corresponding results for the 127M adapter. The larger adapter generally reaches lower WER, but the same pattern emerges. While increasing k and p helps reduce the WER, the downstream trade-off remains visible for PF-SeqKD, confirming that ASR improvements can coincide with a loss of general IF ability.

H.3 Combining ASR and Prompt-Conditioned Samples

This subsection extends Section 5.3 by evaluating mixtures that combine ASR supervision with an equal proportion of prompt-conditioned samples and the two-stage fine-tuning strategy. The appendix reports the configurations with $p \in \{5\%, 10\%\}$ and $k \in \{20, 200\}$, for both adapter sizes.

H.3.1 22M adapter

Tables 18, 19, and 20 report the results for the 22M adapter. Combining ASR and prompt-conditioned samples keeps WER close to 10% across most configurations. The prompt-conditioned samples introduced into the training dataset preserve the downstream-task performance, avoiding the task-overfitting problem. Across the evaluated settings, PF-SeqKD generally obtains higher scores than

p	k	WER↓				
		$MCIF_{\text{MIX}}$	Europarl		Common Voice	
		🇺🇸	🇺🇸	🇪🇸	🇺🇸	🇪🇸
PF-SeqKD						
1%	1	23.56%	23.23%	27.35%	17.23%	15.06%
1%	20	15.10%	14.87%	13.87%	13.69%	12.06%
1%	200	9.21%	10.01%	10.08%	10.38%	6.41%
5%	1	21.75%	20.71%	19.45%	17.92%	12.74%
5%	20	8.44%	9.75%	9.29%	10.32%	5.58%
5%	200	9.12%	12.72%	9.98%	10.24%	6.69%
10%	1	17.72%	18.22%	17.83%	13.34%	11.81%
10%	20	9.47%	9.85%	9.70%	9.84%	6.03%
10%	200	9.29%	9.94%	9.90%	11.34%	5.76%
20%	1	14.04%	16.43%	14.79%	11.79%	10.50%
20%	20	8.75%	9.59%	9.17%	9.55%	5.40%
20%	200	9.39%	9.74%	9.23%	11.41%	5.24%
PC-SeqKD						
1%	1	37.53%	24.97%	75.11%	24.71%	70.89%
1%	20	22.36%	11.67%	13.89%	12.41%	10.00%
1%	200	21.77%	12.26%	13.70%	11.33%	8.66%
5%	1	40.19%	22.90%	40.15%	20.74%	42.89%
5%	20	13.19%	10.41%	10.59%	10.75%	7.80%
5%	200	9.75%	10.20%	9.90%	11.28%	6.34%
10%	1	28.85%	21.57%	41.91%	18.85%	53.82%
10%	20	11.40%	10.54%	11.29%	10.21%	7.59%
10%	200	9.98%	9.98%	9.63%	11.44%	6.38%
20%	1	33.08%	16.47%	38.50%	19.09%	46.19%
20%	20	9.37%	9.81%	9.81%	10.15%	5.85%
20%	200	11.47%	9.63%	9.61%	9.79%	5.35%

Table 13: ASR results for the SpeechLLMs trained on the SeqKD dataset + ASR data using the 22M adapter. Results are reported using WER. We report the external multimodal models `Qwen3-Omni`, `Phi-4-Multimodal` and `GPT-Audio 1.5`. For each dataset, we report the scores per language.

p	k	BERTScore↑			
		$MCIF_{\text{MIX}}$			
		🇺🇸	🇮🇹	🇩🇪	🇨🇳
PF-SeqKD					
1%	1	0.3370	0.3313	0.3458	0.3067
1%	20	0.3442	0.3124	0.3441	0.2692
1%	200	0.2083	0.1651	0.2047	0.1323
5%	1	0.3441	0.3277	0.3485	0.2967
5%	20	0.1608	0.0883	0.1109	-0.071
5%	200	0.1876	0.1412	0.1592	0.031
10%	1	0.3472	0.3337	0.3575	0.2908
10%	20	0.1551	0.0942	0.0978	0.1478
10%	200	0.1558	0.0894	0.0992	-0.0704
20%	1	0.3579	0.3277	0.3581	0.2737
20%	20	0.1508	0.0822	0.0934	-0.0896
20%	200	0.1523	0.0842	0.0887	-0.0901
PC-SeqKD					
1%	1	0.3029	0.2658	0.3095	0.2763
1%	20	0.3005	0.2741	0.2919	0.2601
1%	200	0.3057	0.2632	0.2979	0.2661
5%	1	0.3016	0.2712	0.3003	0.2576
5%	20	0.2893	0.2594	0.2877	0.2389
5%	200	0.3002	0.2731	0.2978	0.2449
10%	1	0.3011	0.2849	0.2875	0.2565
10%	20	0.3054	0.2594	0.2991	0.2764
10%	200	0.3008	0.2835	0.2952	0.2642
20%	1	0.3308	0.2727	0.3037	0.2504
20%	20	0.3069	0.2698	0.3012	0.2660
20%	200	0.296	0.2691	0.2974	0.2379

Table 14: SQA results for the SpeechLLMs trained on the SeqKD dataset + ASR data using the 22M adapter. Performance is measured using the BERTScores. We also report the external multimodal models `Qwen3-Omni`, `Phi-4-Multimodal` and `GPT-Audio 1.5`. For each dataset, we report the scores per language.

p	k	COMET ↑											
		$MCIF_{\text{MIX}}$			Europarl-ST						CoVoST 2		
		en→it	en→de	en→zh	en→de	en→es	en→it	es→de	es→en	es→it	en→de	en→zh	es→en
PF-SeqKD													
1%	1	0.8103	0.7712	0.8366	0.8368	0.8556	0.8489	0.7239	0.8134	0.8076	0.8192	0.8564	0.8602
1%	20	0.7816	0.7250	0.8374	0.7431	0.8050	0.7835	0.7008	0.7893	0.7835	0.8166	0.8554	0.8559
1%	200	0.7967	0.7438	0.8357	0.7890	0.8270	0.8184	0.5360	0.7839	0.7062	0.8172	0.8533	0.8568
5%	1	0.8073	0.7636	0.8389	0.8070	0.8414	0.8395	0.7064	0.8133	0.7984	0.8182	0.8547	0.8595
5%	20	0.732	0.6813	0.7932	0.5835	0.7085	0.6619	0.6533	0.7691	0.7656	0.7981	0.8331	0.8287
5%	200	0.6806	0.6207	0.5601	0.7339	0.8013	0.7957	0.6238	0.6632	0.7462	0.8118	0.8438	0.8116
10%	1	0.8072	0.7522	0.8369	0.7636	0.8210	0.8130	0.6992	0.8056	0.7861	0.8173	0.8538	0.8592
10%	20	0.7188	0.6404	0.7172	0.5996	0.7400	0.7012	0.5165	0.6624	0.6856	0.8081	0.8258	0.8379
10%	200	0.5436	0.4682	0.4603	0.5218	0.6361	0.5947	0.5137	0.6581	0.6778	0.7130	0.7305	0.7894
20%	1	0.8054	0.7595	0.8335	0.7732	0.8220	0.8184	0.6629	0.8068	0.7868	0.8176	0.8524	0.8558
20%	20	0.7813	0.7301	0.8245	0.7703	0.8194	0.8099	0.5860	0.7683	0.7336	0.8104	0.8409	0.8459
20%	200	0.622	0.5802	0.6337	0.5240	0.6386	0.5974	0.5155	0.6602	0.6798	0.7036	0.7087	0.7901
PC-SeqKD													
1%	1	0.8031	0.7616	0.8290	0.8360	0.8498	0.8471	0.7501	0.8224	0.8165	0.8149	0.8479	0.8576
1%	20	0.8012	0.7631	0.8284	0.8115	0.8372	0.8322	0.7301	0.8056	0.8051	0.8155	0.8519	0.8583
1%	200	0.7929	0.7644	0.8191	0.8380	0.8536	0.8490	0.7485	0.8182	0.8127	0.8113	0.8480	0.8531
5%	1	0.7901	0.7566	0.8196	0.8268	0.8445	0.8413	0.7462	0.8190	0.8096	0.8128	0.8486	0.8555
5%	20	0.7918	0.7513	0.8300	0.7706	0.8274	0.8218	0.6746	0.7820	0.7829	0.8164	0.8492	0.8581
5%	200	0.7867	0.7489	0.8154	0.7884	0.8249	0.8183	0.6925	0.7770	0.7851	0.8080	0.8244	0.8490
10%	1	0.8001	0.7616	0.8236	0.8176	0.8443	0.8373	0.7195	0.8071	0.7983	0.8144	0.8484	0.8565
10%	20	0.7878	0.7517	0.8233	0.7822	0.8354	0.8268	0.6743	0.7816	0.7791	0.8113	0.8419	0.8566
10%	200	0.7434	0.7074	0.8104	0.7412	0.8043	0.7871	0.6474	0.7691	0.7728	0.8096	0.8438	0.8558
20%	1	0.8053	0.7695	0.8261	0.8440	0.8599	0.8555	0.7366	0.8185	0.7991	0.8167	0.8507	0.8586
20%	20	0.7967	0.767	0.8254	0.7817	0.8252	0.8237	0.6630	0.7760	0.7832	0.8165	0.8484	0.8580
20%	200	0.7555	0.755	0.8265	0.6056	0.6921	0.6428	0.6324	0.6324	0.7585	0.7900	0.8101	0.8411

Table 15: Speech Translation results for the SpeechLLMs trained on the SeqKD dataset + ASR data using the 127M adapter. Scores are measured using the COMET score. We also report the external multimodal models `Qwen3-Omni`, `Phi-4-Multimodal` and `GPT-Audio 1.5`. For each dataset, we report the different source-target language pairs.

PC-SeqKD, especially in terms of SQA, while ST scores are comparable and vary by language direction. For fine-tuning from PF-SeqKD, ASR remains strong, with WER between 5.29% and 12.14%, but the downstream scores are generally below those of the corresponding single-stage mixtures. Fine-tuning from PC-SeqKD shows a similar ST trade-off. Overall, fine-tuning requires an additional training stage while not preserving the balanced results among the different tasks offered by the single-stage approach.

H.3.2 127M ADAPTER

Tables 21, 22, and 23 report the corresponding results for the 127M adapter. Overall, the 127M results confirm that adding prompt-conditioned samples allows ASR supervision to be introduced without the broad downstream degradation observed for ASR-only mixtures and that single-stage training is the more reliable strategy.

H.4 $MCIF_{FIX}$ RESULTS

This subsection reports the additional results on the fixed-prompt MCIF benchmark ($MCIF_{FIX}$) separately for the 22M and 127M adapters. Different from the $MCIF_{MIX}$, in the $MCIF_{FIX}$ benchmark, each task is evaluated with one prompt shared across its samples.

H.4.1 22M ADAPTER

The 22M-adapter results are reported in Tables 24, 25, 26, and 27. The plain SeqKD baselines provide moderate ST and SQA scores, but ASR remains weak. Adding ASR supervision reduces WER but can damage downstream-task performance. Adding the same amount of prompt-conditioned data

p	k	WER↓				
		$MCIF_{\text{MIX}}$	Europarl		Common Voice	
		🇺🇸	🇺🇸	🇪🇸	🇺🇸	🇪🇸
PF-SeqKD						
1%	1	30.88%	22.44%	31.10%	17.19%	25.91%
1%	20	21.02%	11.97%	14.05%	13.97%	12.27%
1%	200	12.27%	10.70%	12.44%	10.54%	8.91%
5%	1	21.67%	19.40%	18.70%	15.96%	17.57%
5%	20	11.91%	10.35%	10.61%	10.11%	7.10%
5%	200	9.91%	10.49%	10.84%	9.85%	6.52%
10%	1	21.62%	17.51%	16.95%	14.70%	12.07%
10%	20	10.27%	10.17%	10.19%	10.12%	7.07%
10%	200	9.60%	10.31%	12.55%	10.10%	6.24%
20%	1	19.30%	18.32%	17.37%	13.62%	12.00%
20%	20	10.37%	11.09%	10.97%	10.21%	7.19%
20%	200	8.90%	9.93%	9.78%	10.33%	6.04%
PC-SeqKD						
1%	1	41.33%	27.17%	28.35%	25.32%	62.19%
1%	20	24.49%	15.19%	17.43%	15.54%	14.32%
1%	200	19.39%	17.99%	19.35%	16.85%	14.84%
5%	1	42.29%	23.94%	20.66%	23.65%	15.47%
5%	20	13.13%	12.07%	13.80%	13.83%	12.66%
5%	200	14.42%	12.83%	14.44%	11.91%	10.55%
10%	1	35.87%	21.18%	19.05%	23.68%	15.23%
10%	20	11.35%	11.19%	13.38%	13.30%	8.95%
10%	200	10.68%	9.25%	11.68%	10.41%	7.10%
20%	1	27.81%	20.93%	18.43%	17.64%	18.17%
20%	20	10.45%	10.62%	11.12%	10.92%	6.93%
20%	200	11.69%	10.73%	11.80%	12.99%	8.11%

Table 16: ASR results for the SpeechLLMs trained on the SeqKD dataset + ASR data using the 127M adapter. Results are reported using WER. We report the external multimodal models `Qwen3-Omni`, `Phi-4-Multimodal` and `GPT-Audio 1.5`. For each dataset, we report the scores per language.

p	k	BERTScore↑			
		$MCIF_{\text{MIX}}$			
		🇺🇸	🇮🇹	🇩🇪	🇨🇳
PF-SeqKD					
1%	1	0.3428	0.3276	0.3516	0.3175
1%	20	0.3460	0.3246	0.3433	0.3175
1%	200	0.3467	0.2898	0.3131	0.1466
5%	1	0.3368	0.3348	0.3539	0.3055
5%	20	0.3346	0.3203	0.3251	0.2171
5%	200	0.2656	0.2159	0.269	0.0771
10%	1	0.3328	0.33	0.3529	0.3183
10%	20	0.2562	0.2133	0.2612	0.2238
10%	200	0.1649	0.1069	0.1611	-0.0247
20%	1	0.3412	0.3356	0.355	0.3321
20%	20	0.359	0.3248	0.3517	0.2902
20%	200	0.1657	0.0981	0.124	-0.0492
PC-SeqKD					
1%	1	0.2912	0.2633	0.2855	0.2430
1%	20	0.3045	0.2738	0.2994	0.2759
1%	200	0.2759	0.2498	0.2813	0.2333
5%	1	0.2941	0.2679	0.2856	0.2377
5%	20	0.2934	0.2650	0.2951	0.2569
5%	200	0.2933	0.2789	0.3055	0.2627
10%	1	0.3014	0.2858	0.3085	0.2753
10%	20	0.3049	0.2716	0.2939	0.2605
10%	200	0.2969	0.2793	0.2898	0.2745
20%	1	0.2898	0.2633	0.2967	0.2702
20%	20	0.2914	0.2692	0.2896	0.2496
20%	200	0.2997	0.285	0.2942	0.2687

Table 17: SQA results for the SpeechLLMs trained on the SeqKD dataset + ASR data using the 127M adapter. Performance is measured using the BERTScores. We also report the external multimodal models `Qwen3-Omni`, `Phi-4-Multimodal` and `GPT-Audio 1.5`. For each dataset, we report the scores per language.

p	k	COMET ↑											
		$MCIF_{\text{MIX}}$			Europarl-ST						CoVoST 2		
PF-SeqKD+ASR+Prompt-Conditioned samples													
5%	20	0.8095	0.7723	0.8369	0.8122	0.8367	0.8331	0.7225	0.8095	0.8012	0.8179	0.8541	0.8585
5%	200	0.8176	0.7785	0.8363	0.8089	0.8343	0.8264	0.7227	0.8036	0.8026	0.8194	0.8566	0.8602
10%	20	0.8414	0.776	0.83	0.8434	0.8579	0.8532	0.7520	0.8220	0.8133	0.8198	0.8549	0.8619
10%	200	0.7944	0.7514	0.8284	0.7289	0.8137	0.8080	0.6453	0.7880	0.7755	0.8176	0.8519	0.8572
PC-SeqKD+ASR+Prompt-Conditioned samples													
5%	20	0.8004	0.7655	0.8314	0.8175	0.8467	0.8410	0.7474	0.7839	0.7924	0.8197	0.8557	0.8621
5%	200	0.8017	0.755	0.8282	0.6988	0.8075	0.8039	0.6160	0.7793	0.7772	0.8151	0.8495	0.8574
10%	20	0.8073	0.7731	0.8283	0.7188	0.8147	0.8089	0.6496	0.7858	0.7851	0.8183	0.8530	0.8623
10%	200	0.7151	0.7030	0.7723	0.8422	0.8569	0.8493	0.7503	0.8226	0.8171	0.8200	0.8562	0.8622
Fine-tuning PF-SeqKD													
5%	20	0.8083	0.7701	0.8329	0.8058	0.8337	0.8262	0.7087	0.8019	0.7881	0.8207	0.8570	0.8625
5%	200	0.7674	0.7437	0.8192	0.8363	0.8505	0.8452	0.7202	0.8128	0.7959	0.8206	0.8570	0.8633
10%	20	0.8067	0.7567	0.8339	0.8329	0.8475	0.8442	0.7192	0.8136	0.7997	0.8215	0.8574	0.8632
10%	200	0.7410	0.7242	0.7756	0.7654	0.8271	0.8169	0.6670	0.7881	0.7849	0.8196	0.8566	0.8631
Fine-tuning PC-SeqKD													
5%	20	0.7948	0.7361	0.8347	0.7733	0.8279	0.8265	0.7017	0.7885	0.7881	0.8181	0.8523	0.8587
5%	200	0.8102	0.7701	0.8331	0.7881	0.8215	0.8225	0.6850	0.7893	0.7850	0.8183	0.8534	0.8600
10%	20	0.8085	0.7715	0.8329	0.7915	0.8295	0.8274	0.6959	0.7883	0.7903	0.8191	0.8535	0.8605
10%	200	0.8059	0.7737	0.8335	0.7720	0.8259	0.8246	0.6709	0.7843	0.7850	0.8183	0.8527	0.8608

Table 18: Speech Translation results for the SpeechLLMs trained on the SeqKD dataset + ASR data + Prompt Conditioned Samples using the 22M adapter. Scores are measured using the COMET score. We also report the external multimodal models `Qwen3-Omni`, `Phi-4-Multimodal` and `GPT-Audio 1.5`. For each dataset, we report the different source-target language pairs.

p	k	WER↓				
		$MCIF_{\text{MIX}}$	Europarl-ST		Common Voice	
PF-SeqKD+ASR+Prompt-Conditioned samples						
5%	20	11.37%	10.58%	15.10%	11.50%	12.08%
5%	200	9.09%	10.03%	9.54%	9.70%	6.37%
10%	20	8.98%	9.79%	9.51%	10.99%	5.82%
10%	200	9.48%	10.36%	10.13%	10.23%	6.27%
PC-SeqKD+ASR+Prompt-Conditioned samples						
5%	20	9.87%	10.52%	10.35%	12.07%	7.15%
5%	200	12.86%	10.78%	10.20%	10.07%	6.18%
10%	20	9.68%	10.05%	10.99%	10.26%	6.52%
10%	200	9.73%	9.85%	9.54%	10.09%	5.74%
Fine-tuning PF-SeqKD						
5%	20	9.71%	10.33%	9.65%	9.55%	5.48%
5%	200	9.34%	12.14%	10.19%	10.41%	5.94%
10%	20	9.03%	9.57%	9.19%	11.02%	5.33%
10%	200	9.12%	9.48%	9.26%	10.32%	5.29%
Fine-tuning PC-SeqKD						
5%	20	9.64%	9.81%	61.36%	11.28%	7.08%
5%	200	9.81%	10.08%	9.78%	7.08%	6.56%
10%	20	9.56%	9.78%	9.68%	11.63%	6.61%
10%	200	9.45%	9.76%	9.43%	11.77%	6.01%

Table 19: ASR results for the SpeechLLMs trained on the SeqKD dataset + ASR data + Prompt Conditioned Samples using the 22M adapter. Results are reported using WER. We report the external multimodal models `Qwen3-Omni`, `Phi-4-Multimodal` and `GPT-Audio 1.5`. For each dataset, we report the scores per language.

solves the ASR task-overfitting problem. Overall, the results are coherent with the one obtained with the $MCIF_{MIX}$.

p	k	BERTScore↑			
		$MCIF_{\text{MIX}}$			
		🇺🇸	🇮🇹	🇩🇪	🇨🇳
PF-SeqKD+ASR+Prompt-Conditioned samples					
5%	20	0.3595	0.3403	0.3412	0.3232
5%	200	0.3499	0.3235	0.3233	0.2770
10%	20	0.33	0.3475	0.31	0.24
10%	200	0.3393	0.3236	0.3371	0.2855
PC-SeqKD+ASR+Prompt-Conditioned samples					
5%	20	0.2947	0.2690	0.2922	0.2688
5%	200	0.2947	0.2698	0.2968	0.2669
10%	20	0.3017	0.2900	0.2945	0.2568
10%	200	0.3023	0.2766	0.2937	0.2442
Fine-tuning PF-SeqKD					
5%	20	0.3098	0.2819	0.2980	0.3004
5%	200	0.3226	0.2982	0.3243	0.2687
10%	20	0.3209	0.3036	0.3084	0.2675
10%	200	0.3279	0.3062	0.3156	0.3009
Fine-tuning PC-SeqKD					
5%	20	0.3243	0.3070	0.3227	0.2897
5%	200	0.3125	0.2783	0.3051	0.2957
10%	20	0.3134	0.2720	0.2897	0.2877
10%	200	0.3093	0.2794	0.2992	0.2832

Table 20: SQA results for the SpeechLLMs trained on the SeqKD dataset + ASR data + Prompt Conditioned Samples using the 22M adapter. Performance is measured using the BERTScores. We also report the external multimodal models `Qwen3-Omni`, `Phi-4-Multimodal` and `GPT-Audio 1.5`. For each dataset, we report the scores per language.

p	k	COMET ↑											
		$MCIF_{\text{MIX}}$			Europarl-ST						CoVoST 2		
		🇺🇸→🇮🇹	🇺🇸→🇩🇪	🇺🇸→🇨🇳	🇺🇸→🇩🇪	🇺🇸→🇪🇸	🇺🇸→🇮🇹	🇪🇸→🇩🇪	🇪🇸→🇺🇸	🇪🇸→🇮🇹	🇺🇸→🇩🇪	🇺🇸→🇨🇳	🇪🇸→🇺🇸
PF-SeqKD+ASR+Prompt-Conditioned samples													
5%	20	0.7999	0.7615	0.8362	0.8056	0.8292	0.8249	0.7006	0.7931	0.7853	0.8171	0.8551	0.8580
5%	200	0.7940	0.7306	0.8319	0.7456	0.7798	0.7501	0.6349	0.7186	0.7531	0.8175	0.8496	0.8535
10%	20	0.804	0.7503	0.8360	0.7587	0.8190	0.8159	0.5658	0.7542	0.7414	0.8058	0.7655	0.8425
10%	200	0.7973	0.7588	0.8329	0.8353	0.8384	0.8342	0.7461	0.8130	0.7943	0.8180	0.8550	0.8588
PC-SeqKD+ASR+Prompt-Conditioned samples													
5%	20	0.7934	0.7587	0.8222	0.8000	0.8395	0.8365	0.7126	0.8135	0.8000	0.8137	0.8502	0.8565
5%	200	0.7914	0.7536	0.8241	0.8050	0.8351	0.8314	0.6938	0.7910	0.7840	0.8141	0.8419	0.8556
10%	20	0.7973	0.7532	0.8234	0.7389	0.8193	0.8081	0.6670	0.7649	0.7749	0.8163	0.8503	0.8570
10%	200	0.7978	0.7599	0.8036	0.7659	0.8202	0.8178	0.6327	0.7564	0.7672	0.8090	0.8149	0.8488
Fine-tuning PF-SeqKD													
5%	20	0.7670	0.7310	0.8017	0.7434	0.8060	0.8086	0.6024	0.7625	0.7580	0.7976	0.8319	0.8389
5%	200	0.7660	0.6910574474	0.7678	0.5958	0.6986	0.6577	0.5635	0.7208	0.7468	0.7883	0.7922	0.7965
10%	20	0.7913	0.7510	0.8013	0.7601	0.8154	0.8125	0.6157	0.7733	0.7637	0.8020	0.8149	0.8457
10%	200	0.6386	0.5869	0.5818	0.6046	0.7486	0.6993	0.5234	0.6761	0.6929	0.7479	0.6439	0.7823
Fine-tuning PC-SeqKD													
5%	20	0.7655	0.7183	0.6297	0.7507	0.8121	0.8084	0.5682	0.7422	0.7283	0.7978	0.8303	0.8418
5%	200	0.6649	0.6228	0.3931	0.6226	0.7541	0.7115	0.5353	0.6565	0.7185	0.7249	0.6395	0.7798
10%	20	0.7708	0.7259	0.7762	0.7357	0.8094	0.8070	0.5621	0.7426	0.7353	0.8062	0.8395	0.8432
10%	200	0.7386	0.7172	0.5146	0.7052	0.7735	0.7487	0.5275	0.6665	0.6895	0.8038	0.7736	0.7899

Table 21: Speech Translation results for the SpeechLLMs trained on the SeqKD dataset + ASR data + Prompt Conditioned Samples using the 127M adapter. Scores are measured using the COMET score. We also report the external multimodal models `Qwen3-Omni`, `Phi-4-Multimodal` and `GPT-Audio 1.5`. For each dataset, we report the different source-target language pairs.

H.4.2 127M ADAPTER

Similar results are obtained with the 127M-adapter results reported in Tables 28, 29, 30, and 31.

The larger adapter exhibits better performance, with the usual low ASR performance fixed by the introduction of ASR and Prompt-Conditioned samples. Overall, the fixed-prompt results reinforce that

p	k	WER↓				
		$MCIF_{\text{MIX}}$	Europarl-ST		Common Voice	
PF-SeqKD+ASR+Prompt-Conditioned samples						
5%	20	11.28%	10.36%	11.35%	11.10%	7.25%
5%	200	10.38%	10.81%	11.88%	10.96%	7.89%
10%	20	13.63%	11.11%	12.59%	10.23%	7.16%
10%	200	9.88%	10.60%	13.50%	10.77%	6.75%
PC-SeqKD+ASR+Prompt-Conditioned samples						
5%	20	32.83%	12.02%	13.49%	11.57%	10.36%
5%	200	11.81%	11.91%	13.83%	11.16%	11.40%
10%	20	15.67%	11.54%	13.45%	10.97%	9.75%
10%	200	13.27%	11.69%	12.35%	13.93%	8.17%
Fine-tuning PF-SeqKD						
5%	20	18.87%	11.79%	12.39%	17.13%	10.07%
5%	200	12.96%	11.74%	13.12%	12.98%	8.10%
10%	20	11.73%	11.50%	12.77%	14.81%	8.10%
10%	200	13.25%	11.98%	12.92%	13.37%	9.13%
Fine-tuning PC-SeqKD						
5%	20	22.80%	11.89%	13.12%	16.83%	9.30%
5%	200	16.15%	12.40%	12.61%	14.91%	10.01%
10%	20	10.89%	14.12%	10.60%	15.01%	7.42%
10%	200	20.09%	10.78%	12.12%	8.83%	8.83%

Table 22: ASR results for the SpeechLLMs trained on the SeqKD dataset + ASR data + Prompt Conditioned Samples using the 127M adapter. Results are reported using WER. We report the external multimodal models `Qwen3-Omni`, `Phi-4-Multimodal` and `GPT-Audio 1.5`. For each dataset, we report the scores per language.

p	k	BERTScore↑			
		$MCIF_{\text{MIX}}$			
PF-SeqKD+ASR+Prompt-Conditioned samples					
5%	20	0.3541	0.3274	0.3336	0.3289
5%	200	0.3374	0.3153	0.3390	0.2956
10%	20	0.3463	0.3499	0.3486	0.3320
10%	200	0.3400	0.3309	0.3361	0.3099
PC-SeqKD+ASR+Prompt-Conditioned samples					
5%	20	0.2974	0.2744	0.2967	0.2661
5%	200	0.2969	0.2855	0.2954	0.2610
10%	20	0.2941	0.2623	0.2821	0.2634
10%	200	0.2905	0.2766	0.2954	0.2513
Fine-tuning PF-SeqKD					
5%	20	0.3170	0.2961	0.3201	0.2831
5%	200	0.3008	0.2883	0.3088	0.2936
10%	20	0.3110	0.2972	0.3081	0.3010
10%	200	0.3039	0.2885	0.3075	0.2892
Fine-tuning PC-SeqKD					
5%	20	0.3175	0.2964	0.3175	0.2814
5%	200	0.3111	0.2788	0.3086	0.2750
10%	20	0.2987	0.2729	0.3044	0.2811
10%	200	0.3053	0.3018	0.3096	0.2885

Table 23: SQA results for the SpeechLLMs trained on the SeqKD dataset + ASR data + Prompt Conditioned Samples using the 127M adapter. Performance is measured using the BERTScores. We also report the external multimodal models `Qwen3-Omni`, `Phi-4-Multimodal` and `GPT-Audio 1.5`. For each dataset, we report the scores per language.

balanced single-stage training is more reliable than ASR-only supervision or two-stage fine-tuning for retaining all three capabilities.

p	k	COMET↑			WER↓	BERTScore↑			
		en→it	en→de	en→zh	en	en	it	de	zh
PF-SeqKD Baseline									
-	-	0.8066	0.7560	0.8451	19.85%	0.3402	0.3431	0.3484	0.3325
PF-SeqKD + ASR									
1%	1	0.8134	0.7724	0.8391	21.49%	0.3602	0.3623	0.3820	0.3744
1%	20	0.7952	0.7475	0.8405	13.17%	0.3721	0.3461	0.3601	0.3864
1%	200	0.8031	0.7454	0.8342	9.19%	0.2171	0.2503	0.2739	0.1805
5%	1	0.8101	0.7662	0.8425	18.70%	0.3609	0.3617	0.3932	0.3816
5%	20	0.7559	0.7106	0.8235	8.83%	0.1774	0.0931	0.1563	-0.0342
5%	200	0.6222	0.5506	0.6004	8.73%	0.2273	0.1670	0.1665	0.0572
10%	1	0.8104	0.7762	0.8348	16.20%	0.3707	0.3692	0.4001	0.3789
10%	20	0.7723	0.7476	0.8069	9.23%	0.1627	0.1005	0.1086	0.1614
10%	200	0.6090	0.5840	0.5414	9.19%	0.1572	0.0898	0.0956	-0.0709
20%	1	0.8044	0.7672	0.8298	13.68%	0.3769	0.3669	0.4007	0.3801
20%	20	0.5325	0.4485	0.3382	8.76%	0.1539	0.0832	0.0984	-0.0893
20%	200	0.5440	0.4580	0.2938	9.13%	0.1519	0.0843	0.0900	-0.0900

Table 24: PF-SeqKD results for the 22M adapter on the $MCIF_{\text{FIX}}$ benchmark. The baseline results are compared with the setting in which the SpeechLLMs are trained on SeqKD and ASR data.

p	k	COMET↑			WER↓	BERTScore↑			
		en→it	en→de	en→zh	en	en	it	de	zh
PC-SeqKD Baseline									
-	-	0.8198	0.7775	0.8336	26.17%	0.3156	0.2916	0.3336	0.3287
PC-SeqKD + ASR									
1%	1	0.8158	0.7721	0.8293	22.88%	0.3160	0.3125	0.3689	0.3309
1%	20	0.8018	0.7648	0.8299	14.01%	0.3055	0.3037	0.3342	0.3302
1%	200	0.8102	0.7698	0.8337	13.75%	0.3324	0.3096	0.3525	0.3340
5%	1	0.8130	0.7767	0.8335	20.59%	0.3185	0.3130	0.3630	0.3175
5%	20	0.8095	0.7687	0.8382	11.06%	0.3047	0.2981	0.3399	0.3203
5%	200	0.8085	0.7683	0.8352	9.34%	0.3181	0.2931	0.3421	0.3351
10%	1	0.8167	0.7734	0.8351	20.10%	0.3252	0.3158	0.3546	0.3296
10%	20	0.7877	0.7420	0.8292	10.15%	0.3218	0.3140	0.3517	0.3452
10%	200	0.7897	0.7621	0.7789	9.46%	0.3283	0.3314	0.3685	0.3374
20%	1	0.8105	0.7748	0.8347	15.84%	0.3201	0.3036	0.3503	0.3260
20%	20	0.8132	0.7741	0.6900	9.19%	0.3205	0.3064	0.3642	0.3374
20%	200	0.6457	0.6467	0.7026	9.08%	0.3094	0.3035	0.3594	0.3295

Table 25: PC-SeqKD results for the 22M adapter on the $MCIF_{\text{FIX}}$ benchmark. The baseline results are compared with the setting in which the SpeechLLMs are trained on SeqKD and ASR data.

p	k	COMET↑			WER↓	BERTScore↑			
		en→it	en→de	en→zh	en	en	it	de	zh
PF-SeqKD+ASR+Prompt Conditioned Samples									
5%	20	0.8071	0.7682	0.8373	11.05%	0.3767	0.3683	0.3698	0.4018
5%	200	0.8167	0.7801	0.8348	8.97%	0.3666	0.3610	0.3788	0.3698
10%	20	0.8100	0.7700	0.8300	9.40%	0.3600	0.3400	0.3600	0.3300
10%	200	0.7898	0.7495	0.8228	9.40%	0.3592	0.3569	0.3802	0.3825
PC-SeqKD+ASR+Prompt Conditioned Samples									
5%	20	0.7964	0.7676	0.8308	9.18%	0.3164	0.2870	0.3377	0.3353
5%	200	0.8001	0.7505	0.8300	11.28%	0.3099	0.3079	0.3568	0.3368
10%	20	0.8090	0.7690	0.8292	30.63%	0.3179	0.3301	0.3489	0.3355
10%	200	0.7362	0.7073	0.7054	9.65%	0.3227	0.3253	0.3608	0.3350

Table 26: Results for the 22M adapter trained with combined ASR and prompt-conditioned SeqKD data on the $MCIF_{\text{FIX}}$ benchmark.

I ASR Output Analysis

We manually inspected 166 sentences and annotated the differences between the reference transcription and the model output. We considered five non-mutually-exclusive error categories: *Paraphrasing*, when the model preserves the meaning of the reference but uses a different wording; *Named Entities*, when the output incorrectly transcribes a named entity, such as a name or location; *Hallucinations*, when the model introduces content that is not supported by the reference; *Omissions*, when part of

p	k	COMET↑			WER↓	BERTScore↑			
		en→it	en→de	en→zh	en	en	it	de	zh
Fine-tuning PF-SeqKD									
5%	20	0.8079	0.7693	0.8331	9.63%	0.3303	0.3109	0.3551	0.3519
5%	200	0.7921	0.7397	0.8154	9.25%	0.3347	0.3350	0.3665	0.3612
10%	20	0.8027	0.7568	0.8350	8.94%	0.3307	0.3365	0.3719	0.3540
10%	200	0.7840	0.7054	0.7537	9.10%	0.3345	0.3445	0.3670	0.3572
Fine-tuning PC-SeqKD									
5%	20	0.7960	0.7316	0.8340	9.46%	0.3458	0.3529	0.3718	0.3694
5%	200	0.8100	0.7702	0.8345	9.61%	0.3202	0.3047	0.3619	0.3670
10%	20	0.8058	0.7718	0.8347	9.59%	0.3254	0.3034	0.3608	0.3518
10%	200	0.8062	0.7723	0.8350	9.43%	0.3225	0.3141	0.3677	0.3490

Table 27: Two-stage fine-tuning results for the 22M adapter on the $MCIF_{\text{FIX}}$ benchmark. The baseline models are fine-tuned on ASR and Prompt-Conditioned samples.

p	k	COMET↑			WER↓	BERTScore↑			
		en→it	en→de	en→zh	en	en	it	de	zh
PF-SeqKD Baseline									
-	-	0.8106	0.7767	0.8388	21.55%	0.3683	0.3557	0.3804	0.3648
PF-SeqKD + ASR									
1%	1	0.8027	0.7705	0.8378	19.93%	0.3682	0.3663	0.3850	0.3703
1%	20	0.7754	0.7220	0.8389	13.18%	0.3613	0.3533	0.3771	0.3926
1%	200	0.7978	0.7464	0.8384	11.62%	0.3594	0.3088	0.3245	0.2926
5%	1	0.8017	0.7663	0.8402	19.22%	0.3536	0.3628	0.3864	0.3685
5%	20	0.7528	0.6975	0.7890	11.49%	0.3650	0.3500	0.3746	0.3378
5%	200	0.7118	0.6456	0.4833	9.73%	0.3205	0.3160	0.3127	0.1782
10%	1	0.8065	0.7609	0.8373	18.91%	0.3599	0.3516	0.3967	0.3788
10%	20	0.6875	0.6350	0.7360	10.18%	0.3196	0.2595	0.3072	0.3748
10%	200	0.5680	0.4772	0.4405	9.81%	0.1925	0.1347	0.2220	-0.0126
20%	1	0.8001	0.7622	0.8354	17.76%	0.3704	0.3694	0.4048	0.3748
20%	20	0.7836	0.7358	0.8211	10.29%	0.3912	0.3556	0.3934	0.3992
20%	200	0.6088	0.5439	0.6566	9.04%	0.1722	0.1086	0.1247	-0.0359

Table 28: PF-SeqKD results for the 127M adapter on the $MCIF_{\text{FIX}}$ benchmark. The baseline results are compared with the setting in which the SpeechLLMs are trained on SeqKD and ASR data.

p	k	COMET↑			WER↓	BERTScore↑			
		en→it	en→de	en→zh	en	en	it	de	zh
PC-SeqKD Baseline									
0%	0	0.8098	0.7687	0.8336	23.05%	0.3231	0.3167	0.3678	0.3154
PC-SeqKD + ASR									
1%	1	0.8016	0.7627	0.8290	23.96%	0.2967	0.2906	0.3327	0.3069
1%	20	0.8007	0.7667	0.8296	20.09%	0.3198	0.3096	0.3616	0.3372
1%	200	0.7909	0.7695	0.8175	18.39%	0.3067	0.2910	0.3287	0.3258
5%	1	0.7891	0.7547	0.8196	21.86%	0.3084	0.2901	0.3200	0.3416
5%	20	0.2650	0.7537	0.8302	12.56%	0.3086	0.3026	0.3406	0.3342
5%	200	0.7872	0.7518	0.8119	13.22%	0.3118	0.3253	0.3550	0.3334
10%	1	0.8010	0.7634	0.8226	21.12%	0.3232	0.3268	0.3658	0.3427
10%	20	0.7875	0.7496	0.8260	11.09%	0.3390	0.2953	0.3487	0.3404
10%	200	0.7806	0.7183	0.8045	10.71%	0.3110	0.2999	0.3285	0.3523
20%	1	0.8040	0.7710	0.8253	17.90%	0.3230	0.2966	0.3595	0.3327
20%	20	0.7952	0.7673	0.8258	10.39%	0.3228	0.2970	0.3445	0.3340
20%	200	0.7949	0.7600	0.8255	12.15%	0.3209	0.3236	0.3590	0.3331

Table 29: PC-SeqKD results for the 127M adapter on the $MCIF_{\text{FIX}}$ benchmark. The baseline results are compared with the setting in which the SpeechLLMs are trained on SeqKD and ASR data.

the reference content is missing from the generated transcription; and *Errors*, which captures other transcription errors that do not fall into the previous categories. Sentences without any of these phenomena are labeled as *No errors*.

Overall, we identified 110 instances of paraphrasing, 23 named-entity errors, 5 hallucinations, 12 omissions, and 25 other transcription errors, while 66 sentences were annotated as containing no errors. This analysis confirms that, for the SpeechLLM Baseline trained with

p	k	COMET↑			WER↓	BERTScore↑			
					$MCIF_{\text{FIX}}$				
		en→it	en→de	en→zh	en	en	it	de	zh
PF-SeqKD+ASR+Prompt Conditioned Samples									
5%	20	0.7922	0.7555	0.8358	10.16%	0.3671	0.3480	0.3815	0.3951
5%	200	0.7915	0.7335	0.8297	10.27%	0.3602	0.3521	0.3693	0.3744
10%	20	0.7986	0.7554	0.8364	12.10%	0.3761	0.3675	0.3784	0.3676
10%	200	0.8036	0.7502	0.8331	9.85%	0.3670	0.3594	0.3719	0.3850
PC-SeqKD+ASR+Prompt Conditioned Samples									
5%	20	0.7877	0.7606	0.8221	13.83%	0.3190	0.3095	0.3550	0.3504
5%	200	0.7930	0.7563	0.8252	12.14%	0.3112	0.3076	0.3541	0.3504
10%	20	0.7943	0.7650	0.8247	14.69%	0.3224	0.3085	0.3575	0.3391
10%	200	0.7983	0.7622	0.7976	13.75%	0.3022	0.3046	0.3563	0.3348

Table 30: Results for the 127M adapter trained with combined ASR and prompt-conditioned SeqKD data on the $MCIF_{\text{FIX}}$ benchmark.

p	k	COMET↑			WER↓	BERTScore↑			
					$MCIF_{\text{FIX}}$				
		en→it	en→de	en→zh	en	en	it	de	zh
Fine-tuning PF-SeqKD									
5%	20	0.7674	0.7290	0.8031	19.37%	0.3277	0.3436	0.3692	0.3383
5%	200	0.7613	0.6917	0.7422	12.08%	0.3084	0.3332	0.3614	0.3316
10%	20	0.7879	0.7470	0.8030	11.44%	0.3356	0.3447	0.3588	0.3568
10%	200	0.6643	0.6103	0.5840	16.78%	0.3194	0.3257	0.3629	0.3474
Fine-tuning PC-SeqKD									
5%	20	0.7638	0.7146	0.5700	23.68%	0.3350	0.3288	0.3710	0.3630
5%	200	0.6741	0.6131	0.2750	15.76%	0.3251	0.3218	0.3525	0.3351
10%	20	0.7703	0.7177	0.7322	10.62%	0.3111	0.3215	0.3438	0.3591
10%	200	0.6930	0.7300	0.3778	20.39%	0.3248	0.3385	0.3567	0.3722

Table 31: Two-stage fine-tuning results for the 22M adapter on the $MCIF_{\text{FIX}}$ benchmark. The baseline models are fine-tuned on ASR and Prompt-Conditioned samples.

`Qwen3-4B-Instruct-2507` as the backbone LLM, errors captured by WER do not necessarily correspond to a loss of semantic information. Rather, the semantic information is retained, while fine-grained details on what has been said are not faithfully preserved, harming ASR performance.

The examples illustrated in 33, 32 and 34 show some paraphrasing errors found in the three benchmark datasets, respectively MCIF, Europarl-ST, and Common Voice. In the tables, we report the Reference transcription and the one obtained using the Baseline PF-SeqKD SpeechLLM with the 127M adapter.

J EURO LLM BACKBONE

We compute the largest part of our results evaluating a SpeechLLM using `Qwen3-4B-Instruct-2507` as the LLM backbone. However, to show how the proposed methodology works regardless of the backbone, we also trained three additional SpeechLLMs using `EuroLLM-9B-Instruct-2512` (Martins et al., 2025) as the LLM backbone.

For these SpeechLLMs, we used a 26M adapter, and we trained on two languages, English and Portuguese. The training datasets are a subset of Multilingual LibriSpeech (English and Portuguese) (Pratap et al., 2020), Common Voice (English and Portuguese) (Ardila et al., 2020), and Vox Populi (English) (Wang et al., 2021) for a total of $\sim$ 900 hours of speech.

We report the full ASR results obtained with the `EuroLLM-9B-Instruct-2512` backbone in Table 35, the ST results in Table 36, and the SQA results in Table 37. The results show the same pattern as the main experiments: adding ASR and prompt-conditioned samples substantially improves recognition while preserving the downstream translation and question-answering scores of the prompt-free baseline. In the ASR experiments, the WER is higher than in the experiments performed with `Qwen3-4B-Instruct-2507` as the LLM backbone, and the impact of our proposed method is even more evident.

Reference	Output	Labels
What about skewed pay schemes **with a lack of ethics**, which have **stimulated reckless risk-taking**?	What about skewed pay schemes, which **lack ethics** and have **encouraged reckless risk-taking**?	**Paraphrasing**
We have seen the **development of new financial instruments** that are, however, not transparent enough and do not allow for proper risk assessment.	We have **observed the emergence of new financial instruments** that, however, are not transparent enough and do not enable proper risk assessment.	**Paraphrasing**
Correction **is going to be very painful**, especially in the United States, but Europe is not immune to economic **downturn**.	Correction **will be very painful**, especially in the United States, but Europe is not immune to economic **downturns**.	**Paraphrasing**
Solutions should not be patchy – **consider that** markets are global.	Solutions should not be patchy and **must take into account that** markets are global.	**Paraphrasing**
But we have a problem now, and changing the rules after the problem has arisen **is not going to help us with that problem**.	But we have a problem now, and changing the rules after the problem has arisen **won't help us solve that problem**.	**Paraphrasing**

Table 32: Selected Europarl ASR examples whose sole manual-analysis label is Paraphrasing. References are shown in green, outputs in orange, and red bold spans highlight paraphrased content.

Reference	Output	Labels
This works well, but trees are usually not given and need to be obtained **somehow**.	This works well, but trees are usually not given and need to be obtained **in some way**.	**Paraphrasing**
We introduce a new method to predict **the permutation** that **does not put any hard constraints** on the possible permutations. This makes our approach quite flexible and expressive.	We introduce a new method to predict **a permutation** that **does not impose any hard constraints** on the possible permutations. This makes our approach quite flexible and expressive.	**Paraphrasing**
When compared to human annotations, **the weaker annotations are much cheaper**, yet they are also noisy, meaning that **a certain amount** of the annotations are incorrect.	When compared to human annotations, **weak annotations are significantly cheaper**, yet they are also noisy, meaning that **a certain proportion** of the annotations are incorrect.	**Paraphrasing**
Our second finding is that increasing the number of clean validation samples will help WSL approaches **to achieve better performance**, as shown in the **figure on the left**.	Our second finding is that increasing the number of clean validation samples will help WSL approaches **achieve better performance**, as shown in the **left-hand figure**.	**Paraphrasing**
As we can see, **if we have ten samples per class, direct fine-tuning starts to beat** WSL approaches.	As we can observe, **when we have ten samples per class, direct fine-tuning begins to outperform** WSL approaches.	**Paraphrasing**

Table 33: Selected MCIF ASR examples whose sole manual-analysis label is Paraphrasing. References are shown in green, outputs in orange, and red bold spans highlight paraphrased content.

K LIMITATIONS

We summarize the limitations of this work below.

First, due to limited computational resources, we could only evaluate a limited number of combinations of ASR data proportion and pool size. A broader study over adapter sizes, ASR proportions,

Reference	Output	Labels
Joe Keaton **disapproved of films**, and Buster also had reservations about the medium.	Joel Keaton **disliked films**, and Buster also had reservations about the medium.	**Paraphrasing**
It would be very useful if my web browser could **synchronize** all of its preferences, not just some.	It would be very useful if my web browser could **sync** all of its preferences, not just some.	**Paraphrasing**
It was always **so**.	It was always **like that**.	**Paraphrasing**
She has **shown fine bindings** in exhibitions throughout her career.	She has **demonstrated excellent bindings** in exhibitions throughout her career.	**Paraphrasing**
The drinking water supply comes from an intake pipe **southwest of town**.	The drinking water supply comes from an intake pipe **located southwest of town**.	**Paraphrasing**
Berry's fundraising has also **been called into question** by the Charlotte Observer.	Barry's fundraising has also **been questioned** by the Charlotte Observer.	**Paraphrasing**
I guess, it's a **never-ending story**.	I guess it's an **endless story**.	**Paraphrasing**
A new shopping and dining area has been **created** on the first floor.	A new shopping and dining area has been **established** on the first floor.	**Paraphrasing**
Although she had flowers and applause, Mavis said she was not **content**.	Although she had flowers and applause, Mavis said she was not **satisfied**.	**Paraphrasing**

Table 34: Selected Common Voice ASR examples whose sole manual-analysis label is Paraphrasing. References are shown in green, outputs in orange, and red bold spans highlight paraphrased content.

Method	WER ↓				
	$MCIF_{\text{MIX}}$	Europarl		Common Voice	
	🇺🇸	🇺🇸	🇵🇹	🇺🇸	🇵🇹
SeqKD baselines					
PF-SeqKD	157.47%	35.74%	56.90%	31.62%	87.95%
PC-SeqKD	214.65%	57.25%	123.94%	82.06%	303.40%
SeqKD + ASR + Prompt-Conditioned samples					
PF-SeqKD	9.81%	10.44%	15.69%	12.56%	41.45%

Table 35: ASR results for the EuroLLM SpeechLLMs, measured using WER.

Method	$MCIF_{\text{MIX}}$			Europarl-ST							CoVoST 2		
	COMET ↑			COMET ↑							COMET ↑		
	🇺🇸→🇮🇹	🇺🇸→🇩🇪	🇺🇸→🇨🇳	🇺🇸→🇩🇪	🇺🇸→🇪🇸	🇺🇸→🇮🇹	🇺🇸→🇵🇹	🇵🇹→🇩🇪	🇵🇹→🇩🇪	🇵🇹→🇺🇸	🇵🇹→🇮🇹	🇺🇸→🇩🇪	🇺🇸→🇨🇳	🇵🇹→🇺🇸
SeqKD baselines														
PF-SeqKD	0.7890	0.7729	0.8088	0.7956	0.8515	0.8696	0.8555	0.7482	0.8234	0.8495	0.8404	0.8499	0.8432
PC-SeqKD	0.7479	0.7281	0.7659	0.7895	0.8412	0.8628	0.8555	0.7504	0.8174	0.8480	0.8289	0.8284	0.8179
SeqKD + ASR + Prompt-Conditioned samples														
PF-SeqKD	0.8202	0.7895	0.8240	0.7804	0.8398	0.8551	0.8588	0.7487	0.8228	0.8228	0.8412	0.8512	0.8507	

Table 36: ST results for the EuroLLM SpeechLLMs, measured using COMET.

prompt-pool size, and the relative amount of PC-SeqKD data could support the results in a more robust way. For the same reason, we could not run large-scale training on massive datasets as done for training state-of-the-art SpeechLLMs.

Second, the PC-SeqKD data uses a fixed prompt pool inherited from DeSTA Lu et al. (2026). For each sample in our dataset, we sampled one prompt from this pool. However, it is not guaranteed that the prompt is the most meaningful one for that sample. Future work should investigate dynamically generated prompts conditioned on the speech content, and evaluate whether this strategy improves IF performance.

Method	BERTScore ↑				
	$MCIF_{\text{MIX}}$				LibriSQA
	🇺🇸	🇮🇹	🇩🇪	🇨🇳	🇺🇸
SeqKD baselines					
PF-SeqKD	0.1719	0.1776	0.1871	0.1528	0.1127
PC-SeqKD	0.1502	0.1705	0.1544	0.1456	0.1011
SeqKD + ASR + Prompt-Conditioned samples					
PF-SeqKD	0.1702	0.1830	0.1787	0.1681	0.1151

Table 37: SQA results for the EuroLLM SpeechLLMs, measured using BERTScore.

Third, our training data includes speech in two European languages, English and Spanish. While this limited language coverage represents a limitation of the work, as mentioned in the main paper, the model shows cross-lingual generalization abilities to both understanding (as part of the prompt) and generating text in unseen languages such as Italian, German, and Chinese. However, the generalization does not extend to the speech side, so a model capable of processing speech in a wide range of languages requires a multilingual dataset and training that was not evaluated in this work. While there is no reason to believe that broader multilingual dataset mixtures can represent a challenge or have any specificity for the proposed training strategy, adaptations (e.g., the adapter size) may be needed to obtain a strong model. Future work is required to test our approach on a large scale, both in terms of training hours and in terms of language coverage.