REDUCING PROMPT SENSITIVITY IN LLM-BASED SPEECH RECOGNITION THROUGH LEARNABLE PROJECTION

Sergio Burdisso[1,⋆], *Esaú Villatoro-Tello*[1,⋆], *Shashi Kumar*[1,2], *Srikanth Madikeri*[4], *Andrés Carofilis*[1]
Pradeep Rangappa[1], *Manjunath K E*[3], *Kadri Hacioglu*[3], *Petr Motlicek*[1,5], *Andreas Stolcke*[3]

[1]Idiap Research Institute [2]EPFL [3]Uniphore [4]University of Zurich [5]Brno University of Technology

ABSTRACT

LLM-based automatic speech recognition (ASR), a well-established approach, connects speech foundation models to large language models (LLMs) through a speech-to-LLM projector, yielding promising results. A common design choice in these architectures is the use of a fixed, manually defined prompt during both training and inference. This setup not only enables applicability across a range of practical scenarios, but also helps maximize model performance. However, the impact of prompt design remains underexplored. This paper presents a comprehensive analysis of commonly used prompts across diverse datasets, showing that prompt choice significantly affects ASR performance and introduces instability, with no single prompt performing best across all cases. Inspired by the speech-to-LLM projector, we propose a prompt projector module, a simple, model-agnostic extension that learns to project prompt embeddings to more effective regions of the LLM input space, without modifying the underlying LLM-based ASR model. Experiments on four datasets show that the addition of a prompt projector consistently improves performance, reduces variability, and outperforms the best manually selected prompts.

***Index Terms*—** LLM-based speech recognition, prompt sensitivity, speech-to-LLM projection, prompt projector, ASR robustness

1. INTRODUCTION

Integrating speech capabilities into large language models (LLMs) is a key research area, enabling seamless voice interaction and enhancing multimodal understanding [1, 2, 3, 4]. A conventional approach uses a cascaded architecture, first with an automatic speech recognition (ASR) system followed by an LLM to process the recognized text [5, 6, 7, 8]. However, this pipeline suffers from error propagation and prevent the LLM from having access to prosodic information that could be relevant for downstream tasks.

A promising alternative is LLM-based ASR, which directly connects a speech foundation model to an instruction-tuned LLM via a lightweight *speech projector* [9, 10, 3]. This projector maps speech-derived embeddings into the LLM's input space, enabling direct conditioning on the audio signal alongside textual prompts to perform ASR. While large audio language models (LALMs) are designed for general audio understanding with diverse prompts, LLM-based ASR systems focus specifically on transcription, relying on one *fixed*, manually defined prompt to be used during both training and inference [3, 11, 12, 13, 14, 15, 4, 16, 17]. This fixed-prompt setup ensures alignment between training objectives and inference behavior, making it well-suited for applications where high-accuracy transcription is the primary goal.

While prior work has explored different speech encoders and projectors [2, 18, 19], a critical component has been largely overlooked: the *manually chosen prompt*. To the best of our knowledge, no prior work has systematically studied its impact on ASR performance. In this paper, we take a first step toward understanding and improving prompt robustness in LLM-based ASR. We address two key research questions: (i) How important is the choice of the prompt for ASR performance? and (ii) Can we naturally extend a typical LLM-based ASR architecture to improve its robustness to prompt choice?

To answer the first question, our comprehensive evaluation of prompts from recent literature across multiple datasets reveals that prompt choice can drastically impact ASR output, with certain prompts yielding significantly better performance even under identical model and data conditions.

To address the second question, we propose a *prompt projector* as a simple, yet effective, architectural extension. Inspired by the success of the speech projector, we hypothesize that a similar projection function can align prompt embeddings to a more effective region of the LLM input space. Our approach is distinct from soft-prompt learning, as it learns a common projection rather than individual embeddings.

Our main contributions are the following: (1) We provide the first systematic analysis of prompt sensitivity in LLM-based ASR systems; (2) we extend the original architecture by introducing the prompt projector; (3) we validate it across five evaluation sets, yielding consistently increased robustness and mitigating prompt-induced performance variance; (4) we release our source code for reproducibility.[1]

2. METHODOLOGY

2.1. Base Model

As our base model, we adopt the recently proposed SLAM-ASR architecture for LLM-based ASR [3].This model showed that a simple speech projection module is sufficient to achieve competitive performance, outperforming more complex LLM-based ASR systems. Its strong results and minimal design make it an ideal foundation for our experiments.

Specifically, the speech projector, $sp(\cdot)$, is defined as:

$$\mathbf{e}_i = sp(\mathbf{z}_i) = \text{ReLU}(\mathbf{z}_i W_1 + b_1)W_2 + b_2, \quad (1)$$

where $\mathbf{z}_i$ denotes the i-th downsampled audio feature, obtained by concatenating k consecutive encoded frames along the temporal dimension. The resulting projection $\mathbf{e}_i$ matches the dimensionality of the LLM input embeddings.

⋆ Corresponding authors: {*sergio.burdisso, esau.villatoro*}*@idiap.ch*

[1] https://github.com/idiap/llm-asr-prompt

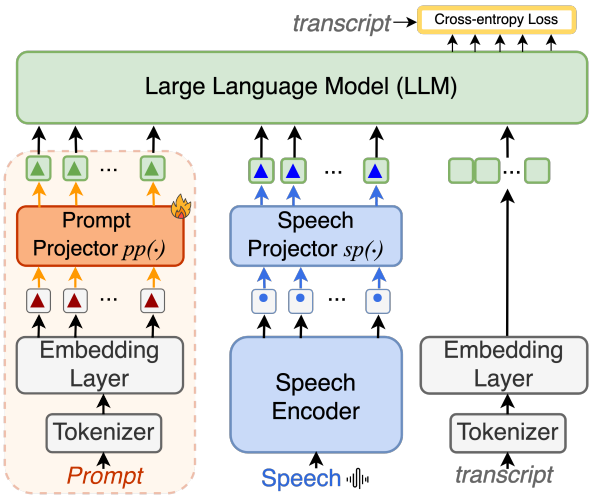


Fig. 1: Typical LLM-based ASR system composed of a fixed prompt, a speech encoder, and an LLM connected by a speech projector, $sp(\cdot)$. The proposed extension, highlighted in orange, introduces a learnable prompt projector, $pp(\cdot)$, into the original (frozen) architecture. The $pp(\cdot)$ learns a common/single projection to transform all the original prompt embeddings (▲) into more effective ones (▲).

In the original work [3], an input audio consisting of n downsampled features, $\mathbf{z}_1, \ldots, \mathbf{z}_n$, is fed to the LLM using the following prompt, which we refer to as the *"base"* prompt:[2]

```
{speech}<s>USER: Transcribe speech to text.
ASSISTANT: {transcript}</s>
```

where {transcript} is the transcription corresponding to the audio, which is either provided during training or forced to be generated by the LLM at inference time. Here, {speech} is replaced by the sequence of n projected speech embeddings, $sp(\mathbf{z}_1), \ldots, sp(\mathbf{z}_n)$, when the prompt is provided to the LLM.

We use the same speech encoder and LLM as in the original work, as this configuration has been shown to outperform models using larger and more recent LLMs [3, 16, 12], offering a good trade-off between ASR performance and computational efficiency:

• **WavLM-large**[3]: speech encoder trained on 94k hours of unlabeled data using self-supervision [20, 21, 22, 23].

• **Vicuna-7B**[4]: LLM fine-tuned from LLama with SFT and optionally RLHF [24].[5]

Throughout all experiments, we use the *WavLM-large + Vicuna-7B* combination. This configuration was shown to be optimal in WER without fine-tuning the speech encoder [3]. We refer to this model as the *"vanilla"* model and, as in the original work, it is trained by only learning the projector (Equation 1) while keeping the speech encoder and the LLM frozen.

[2]The original paper places speech embeddings after "USER:", but the released code prepends them. Our prompt here reflects their implementation.

[3]https://huggingface.co/microsoft/wavlm-large

[4]https://huggingface.co/lmsys/vicuna-7b-v1.5

[5]We tested Llama 3 8B, but Vicuna 7B still outperformed it on 3 of the 5 evaluation sets, so we stick to the original paper's setup.

Table 1: The set of 10 prompts considered for experimentation. The {speech} indicates where the (projected) speech embeddings are located within the prompt. {transcript}s are omitted for simplicity.

No.	Prompt Template
empty	{speech}
base	{speech}<s>USER: Transcribe speech to text.\n ASSISTANT:
1	<s>USER: Transcribe speech to text. {speech}\n ASSISTANT:
2	<s>USER: Transcribe speech to text. Speech: {speech}.\n ASSISTANT:
3	<s>USER: Transcribe the following speech to text: {speech}.\n ASSISTANT:
4	<s>USER: Transcribe accurately speech to text. English speech: {speech}.\n ASSISTANT:
5	<s>USER: Audio: {speech}.\n Transcribe the preceding audio.\n ASSISTANT:
6	<s>USER: Audio: {speech}.\n What is being said in the preceding audio?\n ASSISTANT:
7	<s>USER: Transcribe the following audio: {speech}.\n ASSISTANT:
8	<s>USER: What is being said in the following audio? Audio: {speech}.\n ASSISTANT:

2.2. The Prompt Projector: $pp(\cdot)$

To reduce variability caused by prompt choice (Section 3), we introduce $pp(\cdot)$, a *prompt projector* that projects original prompt embeddings into a more effective region of the LLM input space (Figure 1).

The prompt projector module is a simple drop-in extension to an existing LLM-based ASR system: after training the base model—or using a pretrained one—we freeze all components and train only the new projector module. This ensures the training only focuses on learning how to project the prompt embeddings and avoids instability.[6]

Formally, given prompt embeddings $\mathbf{x}_1, \ldots, \mathbf{x}_n$, they are projected via $pp(\cdot)$ and the new sequence $pp(\mathbf{x}_1), \ldots, pp(\mathbf{x}_n)$ is passed to the LLM instead. The projector shares the same architecture as the speech projector $sp(\cdot)$ (Equation 1), differing only in input dimensionality, since it operates on LLM embeddings rather than speech features.

This simple, architecture-consistent design improves robustness to prompt choice without modifying the original system or introducing prompt-engineering parameters or tokens.

2.3. Training & Inference Implementation Details

All experimental settings follow the original work [3]. Specifically, the speech encoder generates the output at 50 Hz with a downsampling rate of $k = 5$, producing downsampled audio features $\mathbf{z}_i$ at the 100 ms rate — i.e., {speech} contains 10 embeddings $sp(\mathbf{z}_i)$ per second of audio. Both speech and prompt projectors have a hidden layer dimension of 2048, and decoding uses beam search with size 4. Training uses AdamW [25] with learning rate $\gamma = 10^{-4}$, batch size 4, and early stopping based on cross-entropy loss on the dev set.

All computations were performed in *bfloat16* format. Overall, our experiments required over 150 train-to-evaluation trials across various settings (10 prompts, 4 datasets, +/- $pp(\cdot)$, +/- LoRA, freezing/unfreezing) on a single NVIDIA H100 (80 GB VRAM). For our

[6]In complementary experiments, we found out that unfreezing the underlying models consistently led to unstable training and degraded performance across all datasets. Detailed results can be found on our Github.

Table 2: WER (%) comparison of the *vanilla* model with and without the prompt projection (+$pp(\cdot)$) across different prompts. Relative improvement ($\Delta\%$) after applying $pp(\cdot)$ is also reported. **Bold** numbers indicate the best performance in each column, while *underlined* values highlight the largest relative improvement.

Prompt	ContactCenter			CallHome			AMI			LibriSpeech-Clean			LibriSpeech-Other		
	vanilla	+$pp(\cdot)$	$\Delta\%$	*vanilla*	+$pp(\cdot)$	$\Delta\%$	*vanilla*	+$pp(\cdot)$	$\Delta\%$	*vanilla*	+$pp(\cdot)$	$\Delta\%$	*vanilla*	+$pp(\cdot)$	$\Delta\%$
empty	12.75	-	-	27.00	-	-	13.88	-	-	2.84	-	-	5.40	-	-
base	13.00	11.23	*(11.3)*	29.26	26.52	*(7.2)*	13.86	13.42	*(3.4)*	3.09	2.34	*(24.3)*	5.85	4.98	*(14.9)*
1	11.91	11.58	*(2.8)*	**25.26**	24.84	*(1.7)*	13.72	12.96	*(5.5)*	2.88	2.39	*(17.0)*	5.59	4.89	*(12.5)*
2	12.27	11.31	*(7.8)*	27.08	**24.73**	*(8.7)*	**13.36**	12.78	*(4.3)*	2.89	2.31	*(20.1)*	5.71	4.84	*(15.2)*
3	**11.81**	11.25	*(4.7)*	25.83	25.90	*(-0.3)*	13.50	13.26	*(1.8)*	**2.72**	2.31	*(15.1)*	**5.30**	4.92	*(7.2)*
4	12.68	12.43	*(2.0)*	27.95	25.94	*(7.2)*	13.83	12.80	*(7.4)*	2.75	**2.28**	*(17.1)*	5.38	**4.79**	*(11.0)*
5	12.71	**11.23**	*(11.6)*	25.77	25.62	*(0.6)*	13.54	13.18	*(2.7)*	2.80	2.29	*(18.2)*	5.42	5.15	*(5.0)*
6	12.44	11.73	*(5.7)*	26.17	25.57	*(2.3)*	13.37	12.77	*(4.5)*	2.80	2.36	*(15.7)*	5.47	5.04	*(7.9)*
7	12.30	11.48	*(6.7)*	26.69	25.60	*(4.1)*	13.49	12.91	*(4.3)*	2.95	2.31	*(21.7)*	5.37	5.06	*(5.8)*
8	12.00	11.44	*(4.7)*	25.56	24.93	*(2.5)*	13.42	**12.74**	*(5.1)*	2.91	2.61	*(10.3)*	5.54	5.14	*(7.2)*

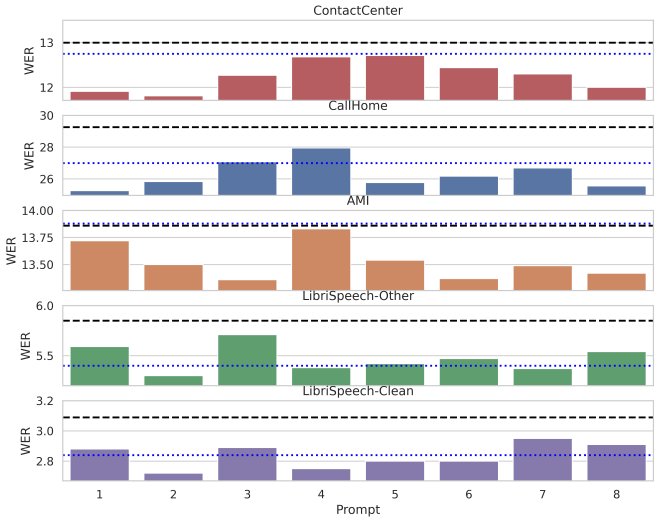


Fig. 2: ASR performance (WER (in %)) across datasets with different prompts. The black dashed line represents the *base* prompt, while the blue dotted line corresponds to the *empty* prompt.

experiments, we trained all models for 5 epochs,[7] and, in experiments involving LLM fine-tuning, we used LoRA adapters with rank 8 and $\alpha = 32$.

2.4. Datasets

We used four datasets spanning diverse speaking styles (read, spontaneous, and conversational), domains, and recording conditions: • **LibriSpeech (LS)**: A 1000 h corpus of read English audiobooks [26]. Models were trained on the 960 h split and evaluated on the 5.4 h `test-clean` (LS-C) and 5.1 h `test-other` (LS-O) sets, with their respective 5.4 h `dev-clean` and 5.3 h `dev-other` sets. • **CallHome (CH)**: 17.5 h of spontaneous telephone conversations [27], split into 13 h training, 3 h dev, and 1.5 h test sets. Its conversational style and short utterances make it challenging for ASR. • **AMI**: A 100 h multi-modal meeting corpus [28]. We used the individual head-mounted microphone (IHM) recordings, comprising 80 h training, 8.5 h test, and 8.8 h dev sets. • **ContactCenter (CC)**: A 48 h proprietary corpus of contact center conversations in health and finance. It is split into 30 h training, 4 h dev, and 6 h test sets, and is difficult due to its domain specificity.

[7]To limit computation given the large number of experiments, experiments using LibriSpeech (about 1k hours) trained for only one epoch, each taking about one day.

3. EXPERIMENTATION

3.1. Assessing the Impact of Manual Prompts

We first examine the effect of fixed prompts on the word error rate (WER%) of our *vanilla* LLM-based ASR model (Section 2.1). For each prompt in Table 1, we train and evaluate an independent *vanilla* model per dataset, isolating the impact of prompt choice. In total, nine prompts beyond the *base* prompt are evaluated, along with a *empty* prompt containing only speech embeddings.

Prompts 1–4 are variations of the *base* prompt with minor wording changes or different placement of the speech embeddings ({speech}). Prompts 5–8 were adapted from prior works, such as SpeechVerse [18] and SpeechLLM [29].

Results are reported in Table 2 (*vanilla* column) and visualized in Figure 2. Even subtle changes, such as the difference between the *base* prompt and prompt 1, yield noticeable improvements: relative WER reductions of 13.6% (CallHome), 8.3% (ContactCenter), 6.7% (LibriSpeech-Clean), and 4.4% (LibriSpeech-Other).

Overall, prompts 1–8 outperform the *base* prompt across all the datasets, suggesting the original *base* prompt is suboptimal and may limit transcription quality. Figure 2 also reveals inconsistent performance: some prompts excel on certain datasets but underperform on others. For instance, prompt 1 performs well on CC and CH but poorly on AMI and LS. In some cases, prompts may even degrade performance relative to having no prompt at all (*e.g.*, *base* or prompt 4 on CH).

Notably, this demonstrates that an LLM-based ASR model can operate effectively using only speech embeddings. The speech projector, even in its simple form (Equation 1), can *implicitly encode prompt-like information while mapping speech features to the LLM input space*. We recommend including a *no prompt* baseline in future LLM-ASR research. It provides a fast, low-cost diagnostic to detect poorly performing prompts—like the *base* prompt—early in development.

These results highlight the high variability of manual prompts: no single prompt performs optimally across all datasets, and even minor changes in wording or embedding placement can lead to large differences in WER.

3.2. Incorporating the Prompt Projector

The results from the previous section highlight the need to reduce sensitivity to prompt choice. We now evaluate whether the *prompt projector* (Section 2.2) can learn a projection, $pp(\cdot)$, that maps prompts into more effective regions of the LLM input space. Following the prior setup, we train and evaluate independent models for each prompt and dataset, now including $pp(\cdot)$. The projector is trained on frozen *vanilla* model checkpoints reported in previous section.[8] Results are reported in Table 2 under the "+$pp(\cdot)$" column.

Table 3 summarizes key results, comparing the *vanilla* model to its variant with the *prompt projector* for both the base and best prompts per dataset, and contextualize the findings relative to other recent work. It also includes *empty* and *base* prompts, along with a "best" row showing the top-performing manual prompt per dataset. "+*LoRA*" results report performance when models are fine-tuned with LoRA adapters.

Comparing *base* and *best* under *vanilla* highlights the substantial variability due to prompt choice (e.g., 13.00 vs. 11.81 WER% on CC, 29.26 vs. 25.26 on CH). Optimal prompts (*best* under $pp(\cdot)$) can approach the best published results, whereas suboptimal prompts may underperform the *empty* prompt (e.g., *base* vs. *empty* on CH and LS). Prompt choice also affects LLM fine-tuning (*base+LoRA* vs. *best+LoRA*), creating a potential bottleneck (e.g., 28.18 vs. 24.74 WER% on CH).

Incorporating $pp(\cdot)$ reduces the gap between *base* and *best* prompts, producing WERs competitive with recent works. Crucially, $pp(\cdot)$ mitigates the impact of suboptimal prompts: even the worst-performing prompt outperforms the best manual prompt alone (e.g., *base*+$pp(\cdot)$ vs. *best*). Figure 3 visualizes the before-and-after effect of using the prompt projector across the datasets, showing that $pp(\cdot)$ consistently stabilizes and enhances performance regardless of the prompt.

Overall, our results show that the new module is able to learn prompt projections that improve performance for all original prompt embeddings, while mitigating the impact of unlucky prompt choices.

4. CONCLUSIONS

In this work, we took a first step toward understanding and improving prompt robustness in LLM-based ASR. Through a systematic evaluation of manual prompts across multiple datasets, we showed that prompt choice has a large impact on transcription quality, with even minor wording or placement changes leading to notable differences in WER. This variability highlights the limitations of relying on fixed, hand-crafted prompts in practical systems.

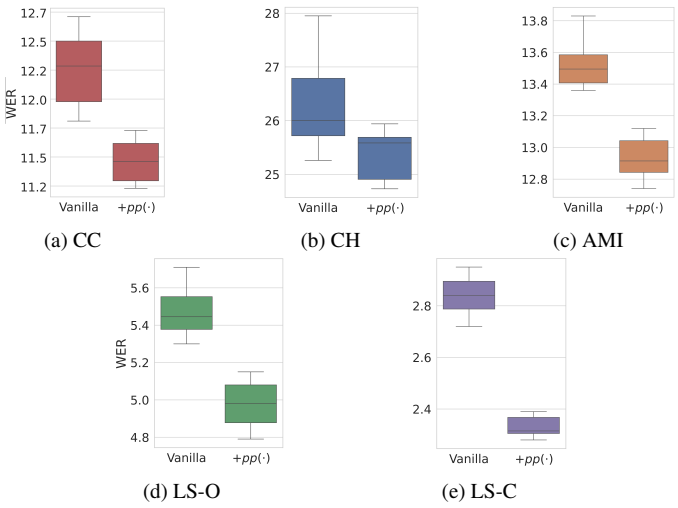
To address this, we introduced the *prompt projector*, a lightweight, architecture-consistent extension that learns a common projection applied to manual prompt embeddings. Our experiments demonstrate that this simple design improves robustness to prompt choice without modifying the underlying system or introducing special prompt-engineering parameters or tokens.

[8] Unfreezing the underlying model degrades performance, as mentioned in Section 2.2. A detailed analysis is available in the appendix on our GitHub.

[9] CC: $p = 0.00135$; CH: $p = 0.0221$; AMI: $p = 1 \times 10^{-5}$; LS-O: $p = 2.74 \times 10^{-4}$; LS-C: $p = 3 \times 10^{-6}$; all with $p < \alpha = 0.05$.

Table 3: Main results. For reference, we include published results from recent LLM-based ASR systems. **Bold** indicates the best values in each group while **underlines** marks the global best across our *vanilla* and +$pp(\cdot)$ groups.

Method	Prompt	Word Error Rate (WER (in %) ↓)				
		CC	CH	AMI	LS-C	LS-O
SLM[4]		-	-	15.14	2.60	5.00
Q-Former[11]		-	-	-	2.28	5.20
Qwen-Audio[2]		-	-	-	**2.04**	**4.20**
SpeechVerse[18]		-	-	-	2.10	4.40
SALMONN[9]		-	-	-	2.10	4.90
vanilla	*empty*	12.75	27.00	13.88	2.84	5.40
	base	13.00	29.26	13.86	3.09	5.85
	+LoRA	*11.60*	*28.18*	*13.25*	*2.46*	*5.21*
	best	**11.81**	**25.26**	**13.36**	**2.72**	**5.30**
	+LoRA	*11.43*	*24.74*	*12.79*	*2.44*	*4.95*
+ $pp(\cdot)$	*base*	11.23	26.52	13.42	2.34	4.98
	+LoRA	*11.16*	*25.86*	*13.38*	*2.33*	*4.78*
	best	**11.23**	**24.73**	**12.74**	**2.28**	**4.79**
	+LoRA	**11.14**	**24.48**	**12.72**	**2.16**	**4.66**



(a) CC (b) CH (c) AMI

(d) LS-O (e) LS-C

Fig. 3: Boxplots illustrating the impact of applying $pp(\cdot)$ across different datasets. Each subplot compares WER (in %) distributions among the different prompts before (*vanilla*) and after applying $pp(\cdot)$. Improvements are statistically significant across all datasets according to paired statistical tests ($p < 0.05$).[9]

This work is an initial exploration, and several open questions remain. A natural next step is to compare the prompt projector with soft-prompt learning methods [30] that add task-specific learnable tokens to the original prompt, rather than learn a single projection. However, a fair comparison would require extensive experimentation across different hyperparameter settings (e.g., number of learnable tokens, initialization, placement), as well as multiple prompts and datasets, as done in this work. Such a study lies beyond the scope of this paper but would be valuable in a dedicated follow-up. Future work should examine the applicability of the approach to different LLMs and speech encoders, as well as to downstream tasks beyond transcription. Finally, an open question is whether a single projector can generalize across prompts, tasks, or languages.

Acknowledgments This work was supported by an Idiap Research Institute and Uniphore collaboration project. Part of this work was also supported by EU Horizon 2020 project ELOQUENCE.

5. REFERENCES

[1] Arushi Goel, Sreyan Ghosh, Jaehyeon Kim, Sonal Kumar, Zhifeng Kong, Sang-gil Lee, Chao-Han Huck Yang, Ramani Duraiswami, Dinesh Manocha, Rafael Valle, et al., "Audio Flamingo 3: Advancing audio intelligence with fully open large audio language models," preprint arXiv:2507.08128, 2025.

[2] Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, et al., "Qwen2-audio technical report," preprint arXiv:2407.10759, 2024.

[3] Ziyang Ma, Guanrou Yang, Yifan Yang, Zhifu Gao, Jiaming Wang, Zhihao Du, Fan Yu, Qian Chen, Siqi Zheng, Shiliang Zhang, et al., "An embarrassingly simple approach for LLM with strong ASR capacity," preprint arXiv:2402.08846, 2024.

[4] Mingqiu Wang, Wei Han, Izhak Shafran, Zelin Wu, Chung-Cheng Chiu, Yuan Cao, Nanxin Chen, Yu Zhang, Hagen Soltau, Paul K Rubenstein, et al., "SLM: Bridge the thin gap between speech and text foundation models," in *Proc. IEEE ASRU*, 2023.

[5] W Ronny Huang, Cyril Allauzen, Tongzhou Chen, Kilol Gupta, Ke Hu, James Qin, Yu Zhang, Yongqiang Wang, Shuo-Yiin Chang, and Tara N Sainath, "Multilingual and Fully Non-Autoregressive ASR with Large Language Model Fusion: A Comprehensive Study," in *Proc. IEEE ICASSP*, 2024.

[6] Yuang Li, Yu Wu, Jinyu Li, and Shujie Liu, "Prompting Large Language Models for Zero-Shot Domain Adaptation in Speech Recognition," in *Proc. IEEE ASRU*, 2023.

[7] Rao Ma, Mengjie Qian, Potsawee Manakul, Mark Gales, and Kate Knill, "Can Generative Large Language Models Perform ASR Error Correction?," preprint arXiv:2307.04172, 2023.

[8] Chao-Han Huck Yang, Yile Gu, Yi-Chieh Liu, Shalini Ghosh, Ivan Bulyko, and Andreas Stolcke, "Generative Speech Recognition Error Correction With Large Language Models and Task-Activating Prompting," in *Proc. IEEE ASRU*, 2023.

[9] Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, MA Zejun, and Chao Zhang, "SALMONN: Towards Generic Hearing Abilities for Large Language Models," in *Proc. 12th Intl. Conf. on Learning Representations*, 2024.

[10] Jian Wu, Yashesh Gaur, Zhuo Chen, Long Zhou, Yimeng Zhu, Tianrui Wang, Jinyu Li, Shujie Liu, Bo Ren, Linquan Liu, et al., "On decoder-only architecture for speech-to-text and large language model integration," in *Proc. IEEE ASRU*, 2023.

[11] Wenyi Yu, Changli Tang, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang, "Connecting speech encoder and large language model for ASR," in *Proc. IEEE ICASSP*, 2024.

[12] Shashi Kumar, Iuliia Thorbecke, Sergio Burdisso, Esaú Villatoro-Tello, Manjunath K E, Kadri Hacioğlu, Pradeep Rangappa, Petr Motlicek, Aravind Ganapathiraju, and Andreas Stolcke, "Performance evaluation of SLAM-ASR: The good, the bad, the ugly, and the way forward," in *Proc. IEEE ICASSP Workshop*, 2025.

[13] Mu Yang, Szu-Jui Chen, Jiamin Xie, and John Hansen, "Bridging the modality gap: Softly discretizing audio representation for llm-based automatic speech recognition," preprint arXiv:2506.05706, 2025.

[14] Yangui Fang, Jing Peng, Xu Li, Yu Xi, Chengwei Zhang, Guohui Zhong, and Kai Yu, "Low-resource domain adaptation for speech LLMs via text-only fine-tuning," preprint arXiv:2506.05671, 2025.

[15] Šimon Sedláček, Bolaji Yusuf, Ján Švec, Pradyoth Hegde, Santosh Kesiraju, Oldřich Plchot, and Jan Černocký, "Approaching dialogue state tracking via aligning speech encoders and LLMs," preprint arXiv:2506.08633, 2025.

[16] Guanrou Yang, Ziyang Ma, Fan Yu, Zhifu Gao, Shiliang Zhang, and Xie Chen, "MaLa-ASR: Multimedia-assisted LLM-based ASR," in *Proc. Interspeech*, 2024.

[17] Sergio Burdisso, Esaú Villatoro-Tello, Andrés Carofilis, Shashi Kumar, Kadri Hacioglu, Srikanth Madikeri, Pradeep Rangappa, Manjunath K E, Petr Motlicek, Shankar Venkatesan, and Andreas Stolcke, "Text-only adaptation in LLM-based asr through text denoising," in *Proc. IEEE ICASSP*, 2026.

[18] Nilaksh Das, Saket Dingliwal, Srikanth Ronanki, Rohit Paturi, Zhaocheng Huang, Prashant Mathur, Jie Yuan, Dhanush Bekal, Xing Niu, Sai Muralidhar Jayanthi, et al., "SpeechVerse: A large-scale generalizable audio language model," preprint arXiv:2405.08295, 2024.

[19] Xuelong Geng, Tianyi Xu, Kun Wei, Bingshen Mu, Hongfei Xue, He Wang, Yangze Li, Pengcheng Guo, Yuhang Dai, Longhao Li, Mingchen Shao, and Lei Xie, "Unveiling the potential of LLM-based ASR on chinese open-source datasets," in *Proc. IEEE 14th Intl. Symp. on Chinese Spoken Language Processing (ISCSLP)*, 2024.

[20] Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al., "WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.

[21] Jacob Kahn, Morgane Riviere, Weiyi Zheng, Evgeny Kharitonov, Qiantong Xu, Pierre-Emmanuel Mazaré, Julien Karadayi, Vitaliy Liptchinsky, Ronan Collobert, Christian Fuegen, et al., "Libri-light: A benchmark for ASR with limited or no supervision," in *Proc. IEEE ICASSP*, 2020.

[22] 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)*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, Eds., Online, Aug. 2021, pp. 993–1003, Association for Computational Linguistics.

[23] Guoguo Chen, Shuzhou Chai, Guan-Bo Wang, Jiayu Du, Wei-Qiang Zhang, Chao Weng, Dan Su, Daniel Povey, Jan Trmal, Junbo Zhang, et al., "GigaSpeech: An evolving, multi-domain ASR corpus with 10,000 hours of transcribed audio," *Proc. Interspeech*, 2021.

[24] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al., "Vicuna: An open-source chatbot impressing GPT-4 with 90%* ChatGPT quality," *See https://vicuna. lmsys. org (accessed 14 April 2023)*, vol. 2, no. 3, pp. 6, 2023.

[25] Ilya Loshchilov and Frank Hutter, "Decoupled Weight Decay Regularization," in *Proc. Intl. Conf. on Learning Representations*, 2019.

[26] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. IEEE ICASSP*, 2015.

[27] Linguistic Data Consortium, "Callhome: A telephone speech corpus for language identification and speech recognition," Linguistic Data Consortium, University of Pennsylvania, 2000, LDC2000S98.

[28] Jean Carletta, Simone Ashby, et al., "The AMI meeting corpus: A pre-announcement," in *International Workshop on Machine Learning for Multimodal Interaction*. Springer, 2005, pp. 28–39.

[29] Shangeth Rajaa and Abhinav Tushar, "SpeechLLM: Multi-Modal LLM for Speech Understanding," https://github.com/skit-ai/SpeechLLM, 2024.

[30] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *ACM Comput. Surv.*, vol. 55, no. 9, Jan. 2023.