TEXT-ONLY ADAPTATION IN LLM-BASED ASR THROUGH TEXT DENOISING

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

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

ABSTRACT

Adapting automatic speech recognition (ASR) systems based on large language models (LLMs) to new domains using text-only data is a significant yet underexplored challenge. Standard fine-tuning of the LLM on target-domain text often disrupts the critical alignment between speech and text modalities learned by the projector, degrading performance. We introduce a novel text-only adaptation method that emulates the audio projection task by treating it as a text denoising task. Our approach thus trains the LLM to recover clean transcripts from noisy inputs. This process effectively adapts the model to a target domain while preserving cross-modal alignment. Our solution is lightweight, requiring no architectural changes or additional parameters. Extensive evaluation on two datasets demonstrates up to 22.1% relative improvement, outperforming recent state-of-the-art text-only adaptation methods.

Index Terms— Text fine-tuning, text denoising, domain adaptation, automatic speech recognition, LLM-based ASR.

1. INTRODUCTION

Recently, there has been growing interest in integrating speech capabilities into large language models (LLMs) to enable seamless voice interaction and advance voice-driven applications, assistive technologies, and conversational AI more broadly [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11].

In this context, LLM-based ASR systems have emerged as a practical and computationally efficient alternative, focusing on transcription by leveraging a fixed, manually defined prompt during both training and inference [3, 4, 12, 13, 14, 15, 16, 17, 18]. This fixed-prompt setup ensures consistency between the training objective and inference behavior, making it well-suited for applications where high-accuracy transcription is the primary goal. A key advantage of LLM-based ASR is the ease of combining strong pretrained speech encoders with powerful LLMs through a learnable projection layer, thereby leveraging advances from pretraining in both speech and text modalities. This modular design enables scalable, high-performance transcription without the need for costly instruction tuning. Intuitively, the projection layer learns to map speech representations into the text embedding space of the LLM (i.e., learns a speech-to-text alignment). Once projected, the resulting representation can be interpreted by the LLM as a noisy text, which the model reconstructs into a clean transcription through its inherent denoising capability.

Despite these advantages, the training of LLM-based ASR typically relies on large amounts of paired audio–text data, which can limit scalability to new domains. In practice, such resources are often scarce or expensive to collect and transcribe. Moreover, existing studies have indicated that performance may degrade when models are applied to domains that differ from the training data [13], highlighting the importance of effective adaptation strategies. Compared to collecting additional audio–text pairs, text-only adaptation offers a more practical alternative given the wide availability of text data.

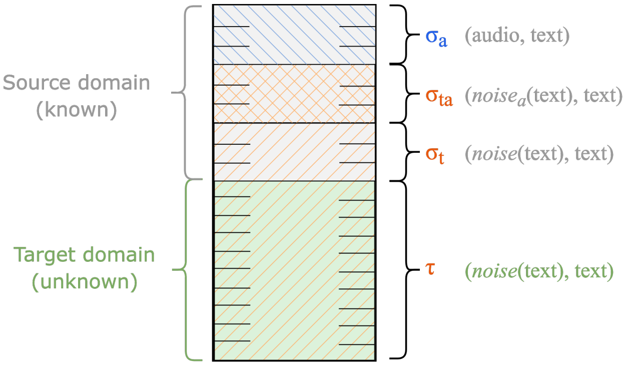
Source domain
(known)
Target domain
(unknown)
σ_a (audio, text)
σ_{ta} (*noise$_a$*(text), text)
σ_t (*noise*(text), text)
τ (*noise*(text), text)

Fig. 1. Batch composition used for fine-tuning the LLM during text-only adaptation to a target domain. Here, σ and $\boldsymbol{\tau}$ denote the proportions of the batch that are drawn from the source domain ($\mathcal{D}_{src}$) and the target domain ($\mathcal{D}_{tgt}$), respectively.

Few studies have explored fine-tuning LLM-based ASR with unpaired text data while preserving cross-modal alignment between the speech projector and the LLM. Fang et al. [15] proposed using a monitoring metric to maintain alignment, but excessive text-only fine-tuning can still degrade recognition, and mitigation strategies only partially address this issue. Ma et al. [19] introduced a two-step approach using trainable soft prompts as pseudo audio embeddings: first optimizing domain-specific soft prompts, then performing text adaptation with the soft prompts fixed. While effective, this method requires tuning additional hyperparameters, such as the number, initialization, and placement of soft tokens.

To address these challenges, we propose a novel text-only adaptation strategy that fine-tunes the LLM within an LLM-based ASR architecture by means of formulating the problem as a *denoising task*. Our contributions are as follows: *(i)* We reformulate text-only adaptation as a denoising task, training the LLM to reconstruct inputs that mimic the outputs of a speech projector in LLM-based ASR architectures. *(ii)* We propose a lightweight training approach that simply consists of a multi-view noise-driven batching strategy, not requiring additional learnable parameters. *(iii)* We present a thorough evaluation across two datasets, achieving up to 22.1% relative improvement, surpassing the state-of-the-art. *(iv)* We release our source code for reproducibility.[1]

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

[1] `https://github.com/idiap/llm-asr-text-adaptation`

2. METHOD

LLM-based ASR systems consist of three main components: (i) a pretrained speech encoder (e.g., Whisper [20], Hubert [21], or WavLM [22], etc.) that extracts frame-level acoustic representations, (ii) a learnable projector $sp(\cdot)$ that maps these representations into the LLM input embedding space (e.g., pooling-based [2],CNN-based [11, 23], or linear-based [3, 24, 13]),[2] and (iii) a pre-trained LLM (e.g., Llama [25], Vicuna [26]) that acts as a decoder, and generates final transcripts.

Prior work has shown that even a simple projector $sp(\cdot)$ (two linear layers with a nonlinearity) combined with a frozen LLM is sufficient to obtain strong transcription performance [3, 13, 14]. This suggests that the projector learns to convert speech into a sequence of *soft tokens* that approximate entries in the LLM vocabulary. For example, the projector may map the audio utterance "yes that would be" into embeddings resembling "mmy Z **YesssS** S SGS **that Will** B **be** S S", as reported in prior work [13, 14]. This illustrates how the projector outputs resemble a noisy transcript rather than raw speech features. As a result, the LLM is forced to interpret these inputs as a corrupted or noisy version of the transcript, which it then recovers.

We interpret this behavior as evidence that LLM-based ASR can be viewed as a *denoising task*: the LLM learns to reconstruct the clean transcript from the noisy, text-like sequence produced by the projector. Building on this insight, we propose a novel approach to adapt LLM-based ASR models using only text data by explicitly teaching the LLM to denoise distorted transcripts from the target domain, even when no target-domain audio is available.

2.1. Task Formulation

Let $\mathcal{D}_{src} = \{(a_i, t_i)\}$ denote the source-domain dataset with paired audio a_i and transcript t_i, and let $\mathcal{D}_{tgt} = \{t_j\}$ denote the target-domain dataset with transcripts only.[3] While standard LLM-based ASR training requires large numbers of (a, t) pairs to achieve high recognition performance, we enable text-only adaptation by introducing a noise function $noise(\cdot)$. This function takes transcripts as input and emulates the audio projection task by adding synthetic noise that approximates the outputs of a trained projector in an LLM-based architecture.

Thus, given only text $t \in \mathcal{D}_{tgt}$, we replace the missing audio with $noise(t)$ and fine-tune the LLM to recover t. Adaptation is therefore reframed as training on $(noise(t), t)$ pairs, i.e., learning to solve a text denoising problem.

2.2. Batch Construction for Text Denoising Adaptation

Directly fine-tuning the LLM with $(noise(t), t)$ pairs from $\mathcal{D}_{tgt}$ leads to *catastrophic forgetting*, i.e., the alignment between the speech encoder and the LLM degrades, and the projector's mapping becomes ineffective, negatively impacting the model's performance. A similar observation has also been recently reported in [15]. To mitigate this, we propose a more effective batch composition strategy for fine-tuning the LLM. Specifically, each batch is constructed as a mixture of examples from both source and target domains (see Figure 1). Let σ_a, σ_{ta}, σ_t, and τ_t denote the proportions (relative shares) of the following components in each batch:

- σ_a: (a, t) pairs from $\mathcal{D}_{src}$, i.e., paired audio and transcripts, to preserve the original speech–text alignment.
- σ_{ta}: $(noise_a(t), t)$ pairs where $(a, t) \in \mathcal{D}_{src}$ and $noise_a(t)$ is obtained by projecting a through the model's projector (*i.e.* $sp(a)$) and mapping the resulting embeddings to their nearest tokens in the LLM vocabulary. This approximates an optimal noise function, as it corresponds to projector-induced text noise.
- σ_t: $(noise(t), t)$ pairs where t is taken from $\mathcal{D}_{src}$ and $noise(t)$ is generated through random character substitutions and duplications. This serves as a naive approximation of $noise_a(t)$, obtainable without access to audio. By including both projector-induced and naive noise in the same batch for the source domain, the LLM is encouraged to learn to bridge the three views of t (audio, $noise_a(t)$, and $noise(t)$).
- τ_t: $(noise(t), t)$ pairs from $\mathcal{D}_{tgt}$, analogous to the previous item but applied to the target domain, thereby driving adaptation by exposing the LLM to target-domain textual data.

Table 1. Datasets used for training, validation, and testing in our experiments. The source partition comes from either DefinedAI (**B**anking, **I**nsurance, **H**ealthcare) or SlideSpeech (**L**ife, **T**alent, **E**nglish, **An**imation, **Ag**riculture, **M**usical **I**nstruments). For evaluation, each target partition is paired individually with the chosen source partition. For reference, we provide the total number of utterances (**#Utts**) and the duration (**Hrs**) of each partition.

Dataset	**Partition**	**Domains**	**Train**		**Validation**		**Test**	
			#Utts	**Hrs**	**#Utts**	**Hrs**	**#Utts**	**Hrs**
DefinedAI	Source	B/I/H	17,398	38:10	–	–	–	–
SlideSpeech	Source	L/T/E	34,682	34:59	–	–	–	–
DefinedAI	Target	B	26,704	36:20	391	0:48	695	1:24
DefinedAI	Target	I	32,249	46:54	325	0:42	650	1:18
SlideSpeech	Target	Ag	30,498	29:20	1,504	1:40	2,934	3:18
SlideSpeech	Target	An	56,593	49:48	2,896	2:50	5,934	5:32
SlideSpeech	Target	MI	9,981	8:30	719	0:43	863	0:53

The proportions are set so that $\sigma_a + \sigma_{ta} + \sigma_t + \tau = 1$. In practice, maintaining a small but nonzero σ_a is critical to avoid forgetting the speech-to-text alignment between, while τ controls the strength of adaptation. Ideally, τ should be optimized on held-out validation data for each application setting, since it directly governs the balance between source retention and target specialization. In this work, however, we adopt a simple and robust heuristic: we set τ proportional to the relative size of the target domain with respect to the source domain (in terms of training examples), and distribute the remaining source-domain proportions equally, i.e., $\sigma_a = \sigma_{ta} = \sigma_t = \frac{1-\tau}{3}$. This choice ensures that larger target domains naturally receive more adaptation weight, while still preserving source-domain coverage. It also allows us to systematically analyze the effect of varying τ across domains.[4] By carefully mixing audio, projector-based noise, and synthetic textual noise in each batch, we enable the LLM to maintain its original transcription ability (i.e., not forgetting the alignment already learned by the projector) while adapting to the target domain in the absence of target-domain audio.

[2] Also known as "speech projector" in the literature, with some referring to it as a "speech adapter" or "connector".

[3] It is worth mentioning that in all our experiments, the text-only data consists of (ground truth) transcripts from conversational speech, rather than arbitrary text such as that found on web pages.

[4] The exact τ values are provided in the result tables next to each domain.

3. DATASETS PREPARATION

Table 1 presents the composition of the source and target domain datasets, including their training, development, and test splits, as used in our experiments.

Our experiments use two conversational corpora, chosen both for their explicit domain grouping and for their relevance to real-world ASR applications. DefinedAI[5] is a proprietary corpus of manually transcribed customer–agent telephone calls (125 hours used), selected for its production-like conditions. Conversational data like this captures naturalistic speech patterns, e.g., including disfluencies, interruptions, and colloquial expressions, that are critical for evaluating ASR robustness in practical settings. To complement it, we use the open-source SlideSpeech [27] dataset, a large-scale audio-visual dataset generated from online conference videos on YouTube, which contains multi-speaker conversations across 22 domains (1,705 videos; 1,000 hours; 473 hours transcribed). By working with these corpora, we focus on realistic conversational scenarios that are more challenging than scripted speech.

From DefinedAI we prepared three partitions: a source (audio, transcripts pairs) partition covering Banking (B), Insurance (I) and Healthcare (H) domains, and two target (text-only) partitions with Banking (B) and Insurance (I) domains. For the SlideSpeech dataset, although it provides a challenging testbed for evaluating ASR robustness, the limited amount of data in its original development and test partitions (approximately 1 hour per domain) limits its usefulness for our experiments. To address this, we ranked the domains by perplexity and selected only those from the original SlideSpeech training set that contained sufficient examples for partitioning into training and development/test portions. Using this ranking, we constructed the source and target partitions: the source domains comprise Life, Talent, and English, while the target domains include Agriculture, Animation, and Musical Instruments (see Table 1). Note that, in our setup, the training split of the target domain contains text-only data.

4. EXPERIMENTAL SETUP

All experiments were implemented in the SLAM-ASR framework [3] with WavLM-Large [22] as the speech encoder and Llama 3.2 3B Instruct[6] [25] as the decoder LLM. These models are connected via a learnable projector consisting of a single hidden layer followed by a ReLU activation and a regression layer. For all the performed experiments, the prompt template is defined as:

```
PROMPT(i, t) = <|start_header_id|>user
<|end_header_id|>Transcribe speech to text.
 Speech:{i}<|eot_id|><|start_header_id|>
assistant<|end_header_id|>{t}
```

where **{t}** is the ground truth transcript and **{i}** is the input that will change with the type of (input, output) pair in the batch composition, in particular:

- PROMPT($sp(a)$, t) for (a, t)
- PROMPT($noise_a(t)$, t) for $(noise_a(t), t)$
- PROMPT($noise(t)$, t) for $(noise(t), t)$,

• **Base model**: We train a *base model* for each dataset (DefinedAI and SlideSpeech) using the source (audio-text pairs) partition, and evaluated on the test split of the target data. For training the base model, we followed the original setup as described in [3]: the speech encoder and the LLM are frozen, while the projector is trained on the source domain data during four epochs with learning rate 1×10^{-4}, 1000 warm-up steps, and batch size of 4.

Table 2. In-domain WER results. Domain adaptation on DefinedAI (target). Base model trained on DefinedAI (source). Values in %.

System	**Banking** ($\tau = 0.61$)		**Insurance** ($\tau = 0.65$)	
	WER	Δ	**WER**	Δ
Base model	12.98	–	10.61	–
Adapted model (audio)	9.92	23.6	7.92	25.4
Adapted model (text)				
Fang et al. [15]	10.92	15.9	9.79	7.7
Ma et al. [19]	10.63	18.1	9.68	8.8
Ours	**10.11**	**22.1**	**8.71**	**17.9**

• **Adapted model (audio)**: This experiment reflects the best-case scenario, i.e., when audio-text pairs are available for fine-tuning the LLM for the target domain. Specifically, for adapting the LLM, we fine-tuned on the target partition, using audio-text pairs, for four epochs, using LoRA applied to the self-attention *query* and *value* projection layers with rank 8, alpha 32, learning rate 1×10^{-4}, and 1000 warm-up steps.

• **Text-only adaptation (ours)**: For these experiments, we implement $noise(t)$ as a two-step process: (1) random character substitution followed by (2) random character duplication. In step (1), we use *nlpaug*[7] to select 15% of the words and replace 30% of the characters within those words with random symbols, with a minimum of 1 and a maximum of 10 character edits per utterance. These values match the *nlpaug* defaults, except that we reduce the fraction of edited words from 30% to 15%, which yielded better performance in preliminary experiments. In step (2), each character has a 10% chance of being repeated; when selected, it is duplicated 1 to 3 times with equal probability. This last step is designed to emulate the duplication patterns observed in the optimal noise (*i.e.* $noise_a(t)$).

Additionally, we also reimplemented two recently proposed text-only adaptation approaches for LLM-based ASR:

• ***Ma et al.*** [19]: the text-only adaptation is performed in two stages: (1) first, freeze the model and learn k trainable embeddings e_i inserted where the audio would go in the prompt —i.e. the input of the LLM is PROMPT($e_1 \ldots e_k$, t); (2) fine-tune the LLM using those k embeddings instead of audio, i.e., batches are composed only of $(e_1 \ldots e_k, t)$ pairs. In our experiments, we use $k = 30$ as in the original work.

• ***Fang et al.*** [15]: the adaptation process consists of: (a) fine-tuning the LLM using raw target-domain text (no prompt involved); (b) monitoring the perplexity on the validation set (with audio) every 200 steps to identify when catastrophic forgetting occurs. The adapted model is the checkpoint with the minimum perplexity before the sudden increase.

Finally, all experiments are evaluated in terms of word error rate (WER), and we report the relative improvement (Δ) over the corresponding *base model*.

5. EXPERIMENTAL RESULTS

We evaluate our proposed text-only adaptation method in three scenarios of increasing difficulty: (1) in-domain adaptation, (2) out-of-domain adaptation, and (3) cross-domain adaptation. The main

[5] https://defined.ai

[6] https://huggingface.co/meta-llama/Llama-3.2-1B-Instruct

[7] https://nlpaug.readthedocs.io/en/latest/augmenter/char/random.html

Table 3. Out-of-domain WER results. Domain adaptation on SlideSpeech (target). Base model trained on SlideSpeech (source). Values in %.

System	Ag ($\tau = 0.47$)		An ($\tau = 0.62$)		MI ($\tau = 0.22$)	
	WER	Δ	**WER**	Δ	**WER**	Δ
Base model	14.82	–	15.58	–	14.73	–
Adapted model (audio)	10.80	27.1	10.37	33.4	11.04	25.1
Adapted model (text)						
Fang et al. [15]	14.47	2.4	15.30	1.8	13.70	7.0
Ma et al. [19]	14.23	4.0	15.71	-0.8	**13.35**	**9.4**
Ours	**14.21**	**4.1**	**14.60**	**6.3**	13.43	8.8

results for each of these experiments are reported in Table 2, Table 3 and Table 4, respectively.

(1) In-domain adaptation - In this setting, the target domain corresponds to a domain type already represented in the source-domain data, both in terms of domain exposure and similar speech and acoustic characteristics. The goal of this experiment was to assess the benefit of additional text data for a domain familiar to the base model. Specifically, for this experiment, the source and target partitions are both drawn from DefinedAI.

Table 2 reports the in-domain results. After text-only adaptation, performance approaches that of audio-based adaptation (best case), with 10.11% vs. 9.92% in Banking and 8.71% vs. 7.92% in Insurance, highlighting the benefits of incorporating additional text data for a familiar domain.

(2) Out-of-domain adaptation - In this setting, the target domain is not represented in the source domain partition but shares the same speech and acoustic characteristics. The goal of this experiment was to validate how well the LLM can learn domain-specific lexical and syntactic patterns from text alone, given stable acoustic conditions. This scenario was simulated using the SlideSpeech dataset, where source domain data is represented by L,T,E domains and the target domain is defined by Ag, An, MI domains.

Table 3 presents the out-of-domain results. Our method achieves consistent WER improvements in two of the three target domains, indicating that the LLM learns domain-specific lexicons, albeit modestly due to the low $\boldsymbol{\tau}$ values (< 0.25 in MI). This suggests that higher specialization (larger $\boldsymbol{\tau}$) could yield further improvements.

(3) Cross-domain adaptation - This setting is the most challenging and realistic scenario; the target domain is not represented in the source data and additionally exhibits different speech and acoustic characteristics. The goal of this experiment was to evaluate the impact of the text-only adaptation approach in bridging the linguistic gap in a scenario where there are evident acoustic shifts. For this experiment, we use DefinedAI (B/I/H) as the source domain and SlideSpeech (Ag/Ai/MI) as the target domain.

Table 4 shows that our text-only adaptation approach improves over the base model, achieving performance comparable to Ma et al. [19]. This indicates that the method can reduce the linguistic gap between domains that differ in both lexicon and acoustics. However, performance remains well below that of the audio-adapted model, an expected outcome given that the latter benefits from target-domain audio to address the acoustic mismatch.

5.1. Ablation experiments

We conducted two ablation studies on the DefinedAI target data to isolate the contributions of our method's key components. The first study evaluates the impact of including or removing different source-side components from the training batches. As shown in Table 5, using all three components (σ_a, σ_{ta}, σ_t) yields the best performance. Notably, omitting the audio component ($\sigma_a = 0$) causes a sharp increase in WER, possibly due to catastrophic forgetting of the speech-text alignment. The second study examines the importance of perturbing text-only data from the target domain. Table 6 shows that using syntactic noise as input to the LLM improves performance more effectively than using unperturbed text. This indicates that framing the task as denoising enables the model to better capture the target domain's lexical and syntactic patterns.

Table 4. Cross-domain WER results. Domain adaptation on SlideSpeech (target). Base model trained on DefinedAI (source). Values in %.

System	Ag ($\tau = 0.64$)		An ($\tau = 0.77$)		MI ($\tau = 0.37$)	
	WER	Δ	**WER**	Δ	**WER**	Δ
Base model	32.64	–	29.81	–	28.00	–
Adapted model (audio)	12.25	62.5	11.23	62.3	13.20	52.9
Adapted model (text)						
Fang et al. [15]	31.01	5.0	27.83	6.6	26.72	4.6
Ma et al. [19]	29.22	10.5	25.95	12.9	25.03	10.6
Ours	**29.18**	**10.6**	**25.32**	**15.1**	**23.54**	**15.9**

Table 5. Batch composition ablation study. Checkmarks denote the active components in the batch. Δ is reported over the base model of Table 2. Values in %.

σ_a	σ_{ta}	σ_t	Banking		Insurance	
			WER	Δ	**WER**	Δ
✓			11.29	13.0	9.91	6.6
✓	✓		10.14	21.9	8.94	15.7
✓		✓	11.77	9.3	9.10	14.2
	✓	✓	73.50	-466.3	66.65	-528.2
✓	✓	✓	**10.11**	**22.1**	**8.71**	**17.9**

Table 6. Ablation of different batch item type for σ_t and τ. The original noise is replaced by different strategies. Δ is reported over the base model of Table 2. Values in %.

Strategy	Batch Item Type	Banking		Insurance	
		WER	Δ	**WER**	Δ
Noise	$\textsc{Prompt}(noise(t), t)$	**10.11**	**22.1**	**8.71**	**17.9**
Echoing	$\textsc{Prompt}(t, t)$	10.31	20.6	9.09	14.3
Empty	$\textsc{Prompt}(\emptyset, t)$	10.56	18.6	9.01	15.1
No prompt	t	10.53	18.9	9.28	12.5

6. CONCLUSIONS

We have presented a novel text-only adaptation approach for LLM-based ASR architectures. Our method alternates source audio, projector-induced noise, and synthetic noisy text during fine-tuning, enabling the model to retain audio understanding, learn text denoising, and acquire target-domain knowledge. Experiments show consistent gains across domains, outperforming prior text-only adaptation methods. Future work will explore improved noise functions that better approximate projection outputs and evaluate optimal τ values under text-rich, real-world conditions.

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.

7. 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, Lukas Zilka, Dian Yu, Golan Pundak, Nikhil Siddhartha, Johan Schalkwyk, and Yonghui Wu, "SLM: Bridge the thin gap between speech and text foundation models," in *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2023, pp. 1–8.

[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 *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 13306–13310.

[6] Yuang Li, Yu Wu, Jinyu Li, and Shujie Liu, "Prompting large language models for zero-shot domain adaptation in speech recognition," in *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2023, pp. 1–8.

[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 *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2023, pp. 1–8.

[9] Hamza Kheddar, Mustapha Hemis, and Yassine Himeur, "Automatic speech recognition using advanced deep learning approaches: A survey," *Information Fusion*, vol. 109, 2024.

[10] 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," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023, pp. 15757–15773.

[11] 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.

[12] 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 *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 12637–12641.

[13] 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 *2025 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)*, 2025, pp. 1–5.

[14] 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.

[15] 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," *CoRR*, vol. abs/2506.05671, 2025.

[16] Šimon Sedláček, Bolaji Yusuf, Ján Švec, Pradyoth Hegde, Santosh Kesiraju, Oldřich Plchot, and Jan Černockỳ, "Approaching dialogue state tracking via aligning speech encoders and LLMs," in *Proc. Interspeech 2025*, 2025, pp. 1748–1752.

[17] Guanrou Yang, Ziyang Ma, Fan Yu, Zhifu Gao, Shiliang Zhang, and Xie Chen, "MaLa-ASR: Multimedia-assisted LLM-based ASR," in *Proc. Interspeech*, 2024, pp. 2405–2409.

[18] Sergio Burdisso, Esaú Villatoro-Tello, Shashi Kumar, Srikanth Madikeri, Andrés Carofilis, Pradeep Rangappa, Manjunath K E, Kadri Hacioglu, Petr Motlicek, and Andreas Stolcke, "Reducing prompt sensitivity in LLM-based speech recognition through learnable projection," in *Proc. IEEE ICASSP*, 2026.

[19] Yingyi Ma, Zhe Liu, and Ozlem Kalinli, "Effective text adaptation for llm-based ASR through soft prompt fine-tuning," in *Proc. IEEE Spoken Language Technology Workshop (SLT)*, Macau, Dec. 2024, pp. 64–69.

[20] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever, "Robust speech recognition via large-scale weak supervision," in *Proc. Intl. Conf. on Machine Learning (PMLR)*, 2023, pp. 28492–28518.

[21] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[22] 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, 2022.

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

[24] 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. Intl. Symp. on Chinese Spoken Language Processing (ISCSLP)*, 2024.

[25] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al., "The Llama 3 herd of models," *CoRR*, vol. abs/2407.21783, 2024.

[26] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing, "Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality," March 2023.

[27] Haoxu Wang, Fan Yu, Xian Shi, Yuezhang Wang, Shiliang Zhang, and Ming Li, "Slidespeech: A large scale slide-enriched audio-visual corpus," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 11076–11080.