# Avoiding Catastrophic Forgetting in Text-Only Adaptation of LLM-based ASR via Multi-View Text Denoising

*Sergio Burdisso* [1,**], *Esaú Villatoro-Tello* [1,**], *Thibault Bañeras-Roux* [1], *Shashi Kumar* [1,2], *Srikanth Madikeri*[3], *Pradeep Rangappa*[1], *Manjunath K E*[4], *Petr Motlicek* [1,5], *Andreas Stolcke*[4]

[1]Idiap Research Institute, Switzerland
[2]EPFL, Switzerland
[3]University of Zurich, Switzerland
[4]Uniphore, USA & India
[5]Brno University of Technology, Czech Republic
{sergio.burdisso, esau.villatoro}@idiap.ch

## Abstract

Adapting LLM-based automatic speech recognition (ASR) systems to new domains using text-only data is challenging. Naively fine-tuning the LLM on target text disrupts the speech–text alignment learned by the projector, causing catastrophic forgetting. We propose a lightweight text-only adaptation strategy that formulates adaptation as a denoising task and introduces a multi-view noise-driven batching scheme to preserve alignment. Each mini-batch mixes paired source audio–text examples, projector-induced noisy transcripts, synthetically corrupted source transcripts, and corrupted target transcripts. This mixing enables domain adaptation while maintaining speech–text alignment. Our method requires no architectural changes or additional parameters and achieves up to 25.4% relative WER improvement over the non-adapted base model, outperforming recent text-only adaptation methods.

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

## 1. Introduction

Recent efforts to integrate speech capabilities into large language models (LLMs) have enabled new voice interaction systems, assistive technologies, and conversational AI applications [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]. LLM-based ASR systems have emerged as a practical and computationally efficient alternative to conventional end-to-end ASR, connecting a pretrained speech encoder to an LLM through a learnable projection layer and using a fixed prompt during training and inference [3, 11, 12, 13, 14, 15, 16, 17, 18]. This modular design leverages advances in speech and text pretraining, enabling scalable transcription without costly instruction tuning. The projection layer maps speech representations into the LLM embedding space, allowing the LLM to interpret them as noisy text and reconstruct transcriptions.

However, training these systems typically requires large audio-text corpora, which are costly to collect and annotate, and performance can degrade on out-of-domain data [12]. Since text is more widely available than paired speech-text data, text-only adaptation is a desirable alternative. Previous text-only adaptation methods for conventional E2E ASR include intermediate-CTC adaptation [19], shallow fusion with external language models [20], text-to-mel generation [21], and decoder adaptation for Whisper [22].

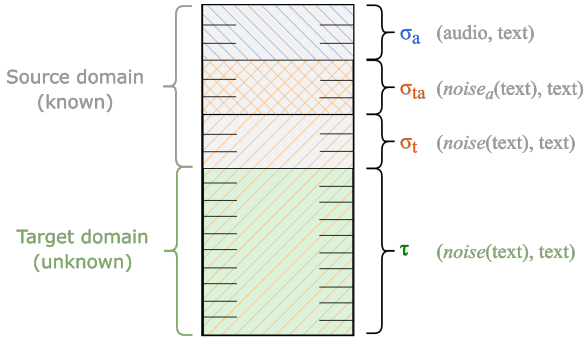


Figure 1: *Batch composition used for fine-tuning the LLM during text-only adaptation to a target domain. Here, $\sigma_a$, $\sigma_{ta}$, and $\sigma_t$ denote the source-domain batch proportions, and $\tau$ denotes the target-domain proportion.*

Nevertheless, text-only adaptation of LLM-based ASR remains largely unexplored, especially regarding preservation of the cross-modal alignment between the speech projector and the LLM. Fine-tuning the LLM only on text can disrupt this alignment, weakening the connection between speech representations and the language model [14, 23]. Fang et al. [14] address this issue by monitoring alignment degradation and stopping fine-tuning before collapse, while Ma et al. [23] introduce soft prompts that replace speech inputs during adaptation.

To address these challenges, we propose a text-only adaptation strategy that formulates adaptation as a *denoising task*, training the LLM to reconstruct inputs that mimic speech-projector outputs. Our contributions are: *(i)* a denoising formulation for text-only adaptation that preserves speech-LLM alignment; *(ii)* a lightweight multi-view noise-driven batching strategy requiring no additional learnable parameters; *(iii)* extensive evaluation on two datasets, achieving up to 25.4% relative improvement over previous approaches; and *(iv)* release of our source code for reproducibility.[1]

## 2. Method

LLM-based ASR systems consist of three main components: (i) a pretrained speech encoder (e.g., Whisper [24], Hubert [25], or WavLM [26], etc.) that extracts frame-level acoustic rep-

[**]indicates the corresponding author.

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

resentations, (ii) a learnable projector $sp(\cdot)$ that maps these representations into the LLM input embedding space (e.g., pooling-based [2], CNN-based [10, 27], or linear-transform-based [3, 28, 12]),[2] and (iii) a pretrained LLM (e.g., Llama [29], Vicuna [30] ) 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, 12, 13]. 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 [12, 13]. 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 [14]. To address this challenge, 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 $\sigma_a$, $\sigma_{ta}$, $\sigma_t$, and $\tau_t$ denote the proportions (relative shares) of the following components in each batch:

- $\sigma_a$: $(sp(a), t)$ pairs where $(a, t) \in \mathcal{D}_{src}$, i.e., paired audio and transcripts to preserve the original speech–text alignment.
- $\sigma_{ta}$: $(noise_a(t), t)$ pairs where $(a, t) \in \mathcal{D}_{src}$ and $noise_a(t)$ is obtained by projecting $a$ through the model's projector $sp(a)$ 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.
- $\sigma_t$: $(noise(t), t)$ 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 $noise(t)$).
- $\boldsymbol{\tau}$: $(noise(t), t)$ 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.

The proportions are set so that $\sigma_a + \sigma_{ta} + \sigma_t + \boldsymbol{\tau} = 1$. In practice, maintaining a small but nonzero $\sigma_a$ is critical to avoid forgetting the speech-to-text alignment (Table 5), while $\boldsymbol{\tau}$ controls the strength of adaptation. Ideally, $\boldsymbol{\tau}$ 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 heuristic: we set $\tau$ 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., $\sigma_a = \sigma_{ta} = \sigma_t = \frac{1-\boldsymbol{\tau}}{3}$. 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 $\boldsymbol{\tau}$ 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.

Table 1: *Data statistics for the two datasets used in our domain adaptation experiments: DefinedAI and SlideSpeech. For each dataset, source models are trained on selected domains (Source role), and adaptation and evaluation are conducted separately for each target domain (Target role). Domain abbreviations are: B = Banking, I = Insurance, H = Healthcare, L = Life, T = Talent, E = English, Ag = Agriculture, An = Animation, MI = Musical Instruments. We report the number of utterances (**#Utts**) and total duration in hours (**Hrs**) for each split.*

| | Role | Domains | Train | | Val | | Test | |
|---|---|---|---|---|---|---|---|---|
| | | | #Utts | Hrs | #Utts | Hrs | #Utts | Hrs |
| DefinedAI | Source | B/I/H | 17,398 | 38:10 | – | – | – | – |
| | Target | B | 26,704 | 36:20 | 1,625 | 2:08 | 3,111 | 4:16 |
| | Target | I | 32,249 | 46:54 | 1,951 | 2:45 | 3,807 | 5:30 |
| SlideSpeech | Source | L/T/E | 34,682 | 34:59 | – | – | – | – |
| | Target | Ag | 30,498 | 29:20 | 1,679 | 1:43 | 3,470 | 3:26 |
| | Target | An | 56,593 | 49:48 | 3,283 | 2:55 | 7,005 | 5:50 |
| | Target | MI | 9,981 | 8:30 | 852 | 0:44 | 984 | 0:54 |

[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] Exact $\boldsymbol{\tau}$ values appear in the result tables next to each domain.

Table 2: *In-domain results on DefinedAI: targets B/I; base model trained on the DefinedAI source (B/I/H) split.*

| System | Banking ($\tau = 0.61$) | | Insurance ($\tau = 0.65$) | |
|---|---|---|---|---|
| | **WER** | $\Delta(\uparrow)$ | **WER** | $\Delta(\uparrow)$ |
| Base model | 8.02 | – | 9.36 | – |
| *Adapted model (audio)* | 4.55 | 43.3 | 6.01 | 35.8 |
| *Adapted model (text)* | | | | |
| Fang et al. [14] | 7.89 | 1.6 | 9.10 | 2.8 |
| Ma et al. [23] | 7.75 | 3.4 | 9.24 | 1.3 |
| **Ours** | **6.53** | **18.6** | **7.59** | **18.9** |

Table 3: *Out-of-domain results on SlideSpeech: targets Ag/An/MI; base model trained on SlideSpeech source (L/T/E).*

| System | Ag ($\tau = 0.47$) | | An ($\tau = 0.62$) | | MI ($\tau = 0.22$) | |
|---|---|---|---|---|---|---|
| | **WER** | $\Delta(\uparrow)$ | **WER** | $\Delta(\uparrow)$ | **WER** | $\Delta(\uparrow)$ |
| Base model | 16.25 | – | 16.45 | – | 15.16 | – |
| *Adapted model (audio)* | 13.86 | 14.7 | 13.17 | 19.9 | 13.10 | 13.6 |
| *Adapted model (text)* | | | | | | |
| Fang et al. [14] | 16.19 | 0.4 | 16.27 | 1.1 | 14.58 | 3.8 |
| Ma et al. [23] | 16.13 | 0.7 | 16.34 | 0.7 | 14.75 | 2.7 |
| **Ours** | **15.56** | **4.2** | **15.35** | **6.7** | **14.10** | **7.0** |

## 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 commercially available corpus 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 broaden domain diversity and speaking styles, we use the open-source SlideSpeech [31] 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 computed domain-level perplexity with the same base LLM used in our adaptation pipeline, ranked domains in descending perplexity (higher means less familiar to the base model), and selected domains from the original SlideSpeech training set with 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 are 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 [26] as the speech encoder and Llama 3.2 3B Instruct[6] [29] 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(**input**, **transcript**) = `<|start_header_id|>user<|end_header_id|>\n\n`**`Transcribe speech to text. Speech:{input}`**`<|eot_id|><|start_header_id|>assistant<|end_header_id|>\n\n`**`{transcript}`**

During training, **{transcript}** is the ground truth and **{input}**, depending on the batch component type (Section 2.2), is one of $sp(a)$, $noise_a(t)$, or $noise(t)$. During inference, **{input}** is always $sp(a)$ and **{transcript}** is the model-generated transcription.

• **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 5 epochs with learning rate $10^{-4}$, 1000 warm-up steps, and batch size of 10.

• **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 5 epochs, using LoRA applied to the self-attention *query* and *value* projection layers with rank 16, alpha 32, learning rate $10^{-4}$, 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:

• ***Fang et al.*** [14]: 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 100 steps to identify when catastrophic forgetting occurs. The adapted model is the checkpoint with the minimum perplexity before the sudden increase.

[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 4: *Cross-domain results on SlideSpeech targets (Ag/An/MI); base model trained on DefinedAI source (B/I/H).*

| System | Ag ($\tau = 0.64$) | | An ($\tau = 0.77$) | | MI ($\tau = 0.37$) | |
|---|---|---|---|---|---|---|
| | **WER** | $\Delta(\uparrow)$ | **WER** | $\Delta(\uparrow)$ | **WER** | $\Delta(\uparrow)$ |
| Base model | 41.09 | – | 33.92 | – | 31.18 | – |
| *Adapted model (audio)* | 16.03 | 61.0 | 14.12 | 58.4 | 17.30 | 44.5 |
| *Adapted model (text)* | | | | | | |
| Fang et al. [14] | 34.93 | 15.0 | 29.87 | 11.9 | 27.06 | 13.2 |
| Ma et al. [23] | 38.22 | 7.0 | 32.74 | 3.5 | 30.09 | 3.5 |
| **Ours** | **30.78** | **25.1** | **25.32** | **25.4** | **24.76** | **20.6** |

• ***Ma et al.*** [23]: 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 $\textsc{Prompt}(e_1 \dots e_k, t)$; (2) fine-tune the LLM using those $k$ embeddings instead of audio, i.e., batches are composed only of $(e_1 \dots e_k, t)$ pairs. In our experiments, we use $k = 30$ as in the original work. As this setup uses no audio, it is also prone to catastrophic forgetting; we therefore also evaluated checkpoints every 200 steps and selected the best one before the forgetting point.

Finally, all experiments are evaluated in terms of word error rate (WER, %), and we report the relative improvement ($\Delta$, %) over the corresponding *base model*. Unless stated otherwise, all result tables report WER and $\Delta$ in percentages.

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

**(1) In-domain adaptation** - 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 narrows the gap to audio-based adaptation, with 6.53% vs. 4.55% WER in Banking and 7.59% vs. 6.01% WER in Insurance.

**(2) Out-of-domain adaptation** - 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 improves WER across all three target domains, including MI despite its low $\tau$ value.

**(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 consistently outperforms prior baselines, reducing linguistic mismatch, though it remains below the audio-adapted model, which additionally benefits from target-domain audio to address acoustic mismatch.[8]

Table 5: *Ablation of source-side batch components $\sigma_a$, $\sigma_{ta}$, and $\sigma_t$ on DefinedAI targets. Checkmarks indicate included components; $\Delta$ is reported w.r.t. the base model in Table 2.*

| $\sigma_a$ | $\sigma_{ta}$ | $\sigma_t$ | Banking | | Insurance | |
|---|---|---|---|---|---|---|
| | | | **WER** | $\Delta(\uparrow)$ | **WER** | $\Delta(\uparrow)$ |
| ✓ | ✓ | ✓ | 6.53 | 18.6 | **7.59** | **18.9** |
| ✓ | | ✓ | **6.40** | **20.2** | **7.59** | **18.9** |
| ✓ | ✓ | | 6.75 | 15.8 | 7.79 | 16.8 |
| ✓ | | | 7.13 | 11.1 | 8.28 | 11.5 |
| | ✓ | ✓ | 10.55 | -31.5 | 12.95 | -38.4 |
| | | ✓ | 10.63 | -32.5 | 15.24 | -62.8 |
| | ✓ | | 10.33 | -28.8 | 12.72 | -35.9 |

Table 6: *Noise-type ablation on DefinedAI targets. Random samples uniformly from a–z/A–Z/space; length matches the transcript; $\Delta$ is reported w.r.t. the base model in Table 2.*

| Strategy | Batch Item Type | Banking | | Insurance | |
|---|---|---|---|---|---|
| | | **WER** | $\Delta(\uparrow)$ | **WER** | $\Delta(\uparrow)$ |
| Original Noise | $\textsc{Prompt}(noise(t), t)$ | **6.53** | **18.6** | **7.59** | **18.9** |
| Random | $\textsc{Prompt}(\text{random}_{\lvert t \rvert}, t)$ | 6.64 | 17.2 | 8.10 | 13.5 |
| Echoing | $\textsc{Prompt}(t, t)$ | 7.60 | 5.2 | 8.32 | 11.1 |
| Empty | $\textsc{Prompt}(\emptyset, t)$ | 6.81 | 15.1 | 8.03 | 14.2 |

### 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, the full mix yields strong and stable performance; removing projector-induced noise slightly improves Banking, while Insurance remains essentially unchanged. 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. This indicates that framing the task as denoising enables the model to better capture the target domain's lexical and syntactic patterns.

## 6. Conclusions

We 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 better noise functions and optimal $\tau$ under text-rich conditions.

[8] In a follow-up work [32], we show that adding a small amount of paired speech-text data (approximately 10%) to the training batches is enough to close the modality gap of text-only fine-tuning.

## 7. 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.

## 8. Generative AI Use Disclosure

Generative AI was used to proofread the paper, fix orthographic and grammatical errors, and condense some sections.

## 9. References

[1] A. Goel, S. Ghosh, J. Kim, S. Kumar, Z. Kong, S.-g. Lee, C.-H. H. Yang, R. Duraiswami, D. Manocha, R. Valle *et al.*, "Audio Flamingo 3: Advancing audio intelligence with fully open large audio language models," preprint arXiv:2507.08128, 2025.

[2] Y. Chu, J. Xu, Q. Yang, H. Wei, X. Wei, Z. Guo, Y. Leng, Y. Lv, J. He, J. Lin *et al.*, "Qwen2-audio technical report," preprint arXiv:2407.10759, 2024.

[3] Z. Ma, G. Yang, Y. Yang, Z. Gao, J. Wang, Z. Du, F. Yu, Q. Chen, S. Zheng, S. Zhang *et al.*, "An embarrassingly simple approach for LLM with strong ASR capacity," preprint arXiv:2402.08846, 2024.

[4] W. Ronny Huang, C. Allauzen, T. Chen, K. Gupta, K. Hu, J. Qin, Y. Zhang, Y. Wang, S.-Y. Chang, and T. 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. 13 306–13 310.

[5] Y. Li, Y. Wu, J. Li, and S. 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.

[6] R. Ma, M. Qian, P. Manakul, M. Gales, and K. Knill, "Can Generative Large Language Models Perform ASR Error Correction?" preprint arXiv:2307.04172, 2023.

[7] C.-H. H. Yang, Y. Gu, Y.-C. Liu, S. Ghosh, I. Bulyko, and A. 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.

[8] H. Kheddar, M. Hemis, and Y. Himeur, "Automatic speech recognition using advanced deep learning approaches: A survey," *Information Fusion*, vol. 109, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253524002008

[9] D. Zhang, S. Li, X. Zhang, J. Zhan, P. Wang, Y. Zhou, and X. Qiu, "SpeechGPT: Empowering large language models with intrinsic cross-modal conversational abilities," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023, pp. 15 757–15 773.

[10] N. Das, S. Dingliwal, S. Ronanki, R. Paturi, Z. Huang, P. Mathur, J. Yuan, D. Bekal, X. Niu, S. M. Jayanthi *et al.*, "SpeechVerse: A large-scale generalizable audio language model," preprint arXiv:2405.08295, 2024.

[11] W. Yu, C. Tang, G. Sun, X. Chen, T. Tan, W. Li, L. Lu, Z. Ma, and C. 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. 12 637–12 641.

[12] S. Kumar, I. Thorbecke, S. Burdisso, E. Villatoro-Tello, M. K E, K. Hacioğlu, P. Rangappa, P. Motlicek, A. Ganapathiraju, and A. 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.

[13] M. Yang, S.-J. Chen, J. Xie, and H. L. John Hansen, "Bridging the modality gap: Softly discretizing audio representation for LLM-based automatic speech recognition," in *2025 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2025, pp. 1–7.

[14] Y. Fang, J. Peng, X. Li, Y. Xi, C. Zhang, G. Zhong, and K. Yu, "Low-resource domain adaptation for speech LLMs via text-only fine-tuning," *CoRR*, vol. abs/2506.05671, 2025.

[15] Š. Sedláček, B. Yusuf, J. Švec, P. Hegde, S. Kesiraju, O. Plchot, and J. Černockỳ, "Approaching dialogue state tracking via aligning speech encoders and LLMs," in *Proc. Interspeech*, 2025, pp. 1748–1752.

[16] G. Yang, Z. Ma, F. Yu, Z. Gao, S. Zhang, and X. Chen, "MaLa-ASR: Multimedia-assisted LLM-based ASR," in *Proc. Interspeech*, 2024, pp. 2405–2409.

[17] S. Burdisso, E. Villatoro-Tello, S. Kumar, S. Madikeri, A. Carofilis, P. Rangappa, M. K E, K. Hacioglu, P. Motlicek, and A. Stolcke, "Reducing prompt sensitivity in LLM-based speech recognition through learnable projection," in *Proc. IEEE ICASSP*, 2026.

[18] E. Villatoro-Tello, S. Burdisso, S. Kumar, H. Watawana, S. Madikeri, M. K E, J. Prakash, T. Bañeras-Roux, K. Hacioglu, P. Motlicek, and A. Stolcke, "Context projector: Complementary keyword and dialogue context embeddings for LLM-based ASR," in *Proc. Interspeech*, 2026.

[19] H. Sato, T. Komori, T. Mishima, Y. Kawai, T. Mochizuki, S. Sato, and T. Ogawa, "Text-Only Domain Adaptation Based on Intermediate CTC," in *Proc. Interspeech*, 2022, pp. 2208–2212.

[20] J. Zhu, W. Tong, Y. Xu, C. Song, Z. Wu, Z. You, D. Su, D. Yu, and H. Meng, "Text-Only Domain Adaptation for End-to-End Speech Recognition through Down-Sampling Acoustic Representation," in *Proc. Interspeech*, 2023, pp. 1334–1338.

[21] V. Bataev, R. Korostik, E. Shabalin, V. Lavrukhin, and B. Ginsburg, "Text-only domain adaptation for end-to-end ASR using integrated text-to-mel-spectrogram generator," in *Proc. Interspeech*, 2023, pp. 2928–2932.

[22] B. Kurian, A. Upadhyay, and A. Sengupta, "Domain-specific adaptation for ASR through text-only fine-tuning," in *Proceedings of the 1st Workshop on Multimodal Models for Low-Resource Contexts and Social Impact (MMLoSo 2025)*, A. Shukla, S. Kumar, A. S. Bedi, and T. Chakraborty, Eds. Mumbai, India: Association for Computational Linguistics, Dec. 2025, pp. 78–85. [Online]. Available: https://aclanthology.org/2025.mmloso-1.7/

[23] Y. Ma, Z. Liu, and O. 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.

[24] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *Proc. Intl. Conf. on Machine Learning (PMLR)*, 2023, pp. 28 492–28 518.

[25] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. 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.

[26] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. 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.

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

[28] X. Geng, T. Xu, K. Wei, B. Mu, H. Xue, H. Wang, Y. Li, P. Guo, Y. Dai, L. Li, M. Shao, and L. Xie, "Unveiling the potential of LLM-based ASR on Chinese open-source datasets," in *Proc. Intl. Symp. on Chinese Spoken Language Processing (ISCSLP)*, 2024.

[29] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan *et al.*, "The Llama 3 herd of models," *CoRR*, vol. abs/2407.21783, 2024.

[30] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez, I. Stoica, and E. P. Xing, "Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality," March 2023. [Online]. Available: https://lmsys.org/blog/2023-03-30-vicuna/

[31] H. Wang, F. Yu, X. Shi, Y. Wang, S. Zhang, and M. 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. 11 076–11 080.

[32] T. Bañeras-Roux, S. Burdisso, E. Villatoro-Tello, D. Sánchez-Cortés, S. Liu, S. Baroudi, S. Kumar, H. Watawana, M. K E, K. Hacioglu, P. Motlicek, and A. Stolcke, "Closing the speech-text gap with limited audio for effective domain adaptation in LLM-based ASR," in *Proc. Interspeech*, 2026.