

Whispering LLaMA: A Cross-Modal Generative Error Correction Framework for Speech Recognition

Srijith Radhakrishnan^{1,2}, Chao-Han Huck Yang^{1,3}, Sumeer Ahmad Khan¹, Rohit Kumar¹ Narsis A. Kiani⁴, David Gomez-Cabrero¹, Jesper N. Tegner^{1,5}

¹King Abdullah University of Science and Technology, ²Manipal Institute of Technology

³Georgia Institute of Technology, ⁴Karolinska Institute

srijith.radhakrishnan@kaust.edu.sa; huckiyang@gatech.edu

Abstract

We introduce a new cross-modal fusion technique designed for generative error correction in automatic speech recognition (ASR). Our methodology leverages both acoustic information and external linguistic representations to generate accurate speech transcription contexts. This marks a step towards a fresh paradigm in generative error correction within the realm of n-best hypotheses. Unlike the existing ranking-based rescoring methods, our approach adeptly uses distinct initialization techniques and parameter-efficient algorithms to boost ASR performance derived from pre-trained speech and text models. Through evaluation across diverse ASR datasets, we evaluate the stability and reproducibility of our fusion technique, demonstrating its improved word error rate relative (WERR) performance in comparison to n-best hypotheses by relatively 37.66%. To encourage future research, we have made our code and pre-trained models open source at: <https://github.com/Srijith-rkr/Whispering-LLaMA>.

1 Introduction

End-to-end (E2E) trained speech models have demonstrated state-of-the-art performance on Automatic speech recognition (ASR) tasks. Several methods (Xia et al., 2017; Guo et al., 2019; Hu et al., 2021b; Yang et al., 2021a; Xu et al., 2022) have widely adopted a two-pass rescoring paradigm to leverage upon language models to further enhance the capabilities of these models. In the two-pass paradigm, the first pass ASR system “generates” n-best hypotheses using an E2E acoustic model, while the second pass “re-ranks” these hypotheses by incorporating a language model (LM).

This two-pass reranking approach has been several notable advantages over single-pass End-to-End (E2E) ASR systems (Amodei et al., 2016; Chan et al., 2016). Firstly, the subsequent large language model often captures a more comprehen-

sive understanding (Stooke et al., 2023; Tur and De Mori, 2011) of language structures beyond the knowledge of transcribed audio present in the ASR model’s pre-training data, thereby improving performance on unseen words. Furthermore, adapting the two-pass paradigm to accommodate domain shifts (Li et al., 2023; Liu et al., 2021; Yu et al., 2023) is much easier as only the language model needs to be fine-tuned on the new dataset. This alleviates the need for a spoken transcription corpus, which can be particularly beneficial for under-resourced or endangered spoken languages.

The recent emergence of conversational abilities in large language models, such as ChatGPT (OpenAI, 2023a) and GPT-4 (OpenAI, 2023b), has further sparked interest in leveraging representational power of large pre-trained models for more complex tasks involving diverse data modalities (Yang et al., 2021b; Chang et al., 2023). Moreover, this new research direction also introduces a set of unique challenges related to considering information from other input modalities, such as acoustic and visual conditions (Peng et al., 2023; Zhang et al., 2023), in which could enrich using context beyond text-only input.

Recognizing speech signals is one example that requires both acoustic information (Hu et al., 2021a; Hung et al., 2023) (e.g., speaking environments) and linguistic information (Meng et al., 2023; Chen et al., 2023b,c) (e.g., context and domains). Efficiently integrating the information from the acoustic model into the language model to enhance its performance represents a particularly complex research area that necessitates further investigation. In this paper, we introduce a token-level fusion framework that combines two foundational models into a recognition error correction paradigm, aimed at improving the performance of ASR systems.

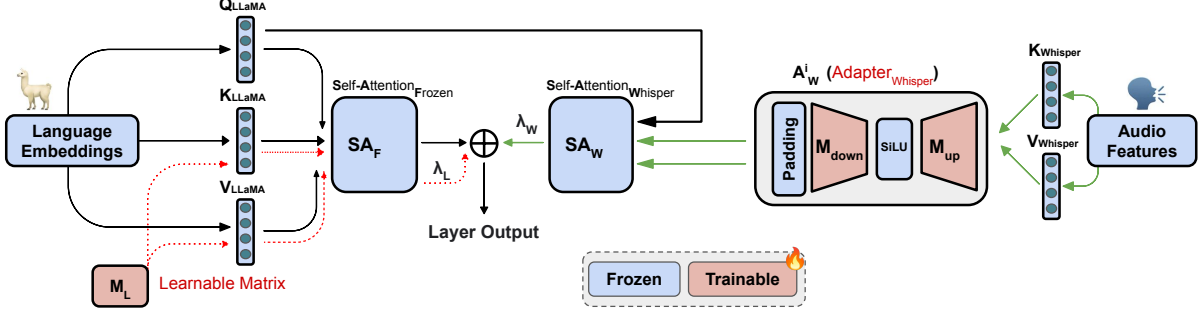


Figure 1: Illustration of proposed generative ASR error correction with a trainable token (M_L) and fusion mechanism inside a self-attention layer described in Section 3.2. A detailed model-wise illustration is discussed in Fig 2.

2 Related Work on ASR Rescoring

Transformer-based language models (Shin et al., 2019; Salazar et al., 2020) approach the two-pass paradigm by utilizing the summation of negative log-likelihoods of individual tokens from the language model to re-score the n-best output. Recent works on deliberation method (Hu et al., 2020; Prabhavalkar et al., 2018) and audio-attention based rescoring (Futami et al., 2021; Gandhe and Rastrow, 2020; Tanaka et al., 2021) have shown evidence with acoustic injection. Recent work on decoder (Yang et al., 2023a) and encoder-decoder based error correction (Chen et al., 2023a; Ma et al., 2023) have demonstrated benefits in using an external language model for reducing the transcription error rate. Meanwhile, how to inject or fuse representations from a large acoustic model into another language model remains under investigation.

3 Method

We discuss the model architecture and the intuition behind proposed feature combination in Section 3.1. The cross modal fusion mechanism and weight initialization are explained in Section 3.2 and Section 3.3, respectively.

3.1 Generative Error Correction for ASR

Our approach combines two pre-trained models, Whisper (Radford et al., 2022) and LLaMA (Touvron et al., 2023), to facilitate generative error correction (Yang et al., 2023a; Chen et al., 2023a). Firstly, we employ Whisper, a multi-task encoder-decoder-based transformer (Vaswani et al., 2017) speech model trained on 680,000 hours of multi-lingual data, to encode audio representations and generate transcripts of n-best hypothesis. Secondly, we utilize LLaMA, a decoder-based large

language transformer model to generate error-corrected transcripts by utilizing the n-best hypotheses via prompt (illustrated in Appendix, Fig 5) and audio representations via our proposed framework as input. Whisper employs a unique approach of predicting tokens “auto-regressively” during inference. Rather than feeding all previously predicted tokens as input, only the last predicted token, accompanied by positional embeddings and encoded audio representations, is utilized to predict the subsequent token. This suggests that the encoded audio representations provide contextual information through cross-attention, as the native self-attention modules present in the decoder do not have access to the previous tokens. Hence, we choose to fuse the audio representations and the Whisper linear layers that generate the key and value pairs in the decoder cross attention mechanism to the LLaMA model, an overview of the proposed method is represented in Appendix, Fig 2.

3.2 Cross-Modal Fusion Mechanism

We represent our mechanism in Fig 1. To efficient fine-tune large models, we incorporate two residual adapter (Houlsby et al., 2019; Radhakrishnan et al., 2023; Chen et al., 2023d; Yang et al., 2023b) modules (A_L^i and A_W^i) after the self-attention modules (SA_F^i) of the frozen LLaMA model in each layer. Adapter A_L^i represents the adapter in layer i used to fine-tune the LLaMA model using a scaled dot product attention mechanism. Adapter A_W^i represents an adapter in layer i used to fuse Whisper features with the LLaMA model by following an autoencoder mechanism.

In A_L^i , we inject a learnable matrix $M_L^i \in \mathbb{R}^{t \times LE}$ in each layer. Here, t represents the size of the adapter embeddings, and LE represents the size

of the latent LLaMA Embeddings (LE).

We repurpose the frozen LLaMA linear layers K_{llama}^i, V_{llama}^i from LLaMA self-attention (SA_F^i) to transform M_L^i into key and value pairs, thereby reducing the number of trainable parameters. We reuse the query tensor from the frozen LLaMA self-attention module (SA_F^i) to compute A_L^i as shown below. Where S denotes the Softmax function.

$$S \left(\frac{Q_{llama}^i(LE) \cdot K_{llama}^i(M_L^i)^T}{\sqrt{d_k}} \right) V_{llama}^i(M_L^i) \quad (1)$$

To integrate the audio representations and key-value tensors from the Whisper Decoder cross-attention module into the LLaMA model, we introduce two additional linear frozen transformations ($K_{whisper}^i$ and $V_{whisper}^i$) in each layer of the LLaMA model. These transformations are initialized with the respective weights from the cross-attention module of the Whisper decoder model. By applying the audio representations to these additional linear transformations, we generate the key-value pairs that mirror the ones produced by Whisper.

We then utilize the second adapter module A_W^i , to add trainable components to learn cross-modal representation. We apply a learnable projection matrix $M_{down}^i \in \mathbb{R}^{N_W \times \frac{N_W}{r}}$ to down project the obtained key and value pairs. Where N_W denote the shape of the Whisper latent embedding. We then apply the Silu activation function (Elfwing et al., 2018) followed by a learnable up projection $M_{up}^i \in \mathbb{R}^{\frac{N_W}{r} \times N_W}$, to compute trainable adapter:

$$A_W^i(x) \leftarrow \text{Silu}(x \cdot M_{down}^i) M_{up}^i \quad (2)$$

Using this setup, we transform the key-value pair at each layer, merging the hidden representation (H_{audio}) from the output of the Whisper encoder with decoder from LLaMA.

$$\hat{k}_{whisper}^i \leftarrow A_W^i(K_{whisper}^i(H_{audio})) \quad (3)$$

$$\hat{v}_{whisper}^i \leftarrow A_W^i(V_{whisper}^i(H_{audio})) \quad (4)$$

Once we obtain the corresponding Whisper key and value pairs, we apply the padding mechanism described in 3.3 to adjust the shape of $\hat{k}_{whisper}^i$ and $\hat{v}_{whisper}^i$ to match q_{llama}^i to enable computation of Multi Head Attention (MHA) and to preserve the latent structure of the Whisper Key and Value embeddings. We merge it with the LLaMA model

Table 1: Dataset sample statistics are provided with alias names. The Science & Technology category of GigaSpeech (Chen et al., 2021) is divided into two subsets: GS_{SS} (small) and GS_{SM} (medium), to evaluate performance differences with respect to data size.

Dataset	Train	Test
ATIS (Hemphill et al., 1990)	4978	893
GigaSpeech: Entertainment (GS_E)	4709	1000
People & Blogs (GS_P)	6802	1000
Science & Technology (GS_{SS})	6908	1000
Science & Technology (GS_{SM})	10323	1000

by computing MHA with the query (q_{llama}^i) from the frozen LLaMA model as before to obtain its self-attention head (SA_W^i):

$$SA_W^i \leftarrow S \left(\frac{Q_{llama}^i(LE) \cdot (K_{whisper}^i)^T}{\sqrt{d_k}} \right) V_{whisper}^i \quad (5)$$

Then we apply a gated fusion mechanism to fuse all the modules together as shown below, where λ_L and λ_W are learnable scalars.

$$SA_F^i \leftarrow SA_F^i + \lambda_L \cdot A_L^i + \lambda_W \cdot SA_W^i \quad (6)$$

3.3 Weight Initialization

The latent dimensions of the Whisper and LLaMA models are different, making it necessary to reshape the Whisper tensors to match the shape of the LLaMA model while preserving the latent structure and information inherent in the Whisper model. Tensors are shaped in the format of $[B, NH, T, HS]$ which denotes the Batch size, Number of heads, context length and Head Size, respectively. The last two dimensions undergo transformation during the attention mechanism. Hence in order to preserve the Whisper latent structure, We initialize a matrix of zeros of shape $\in \mathbb{R}^{NH_{LLaMA} \times T_{whisper} \times HS_{LLaMA}}$ and fill the principal diagonal of the last two dimensions with ones. We then place K^i and V^i on the top left corner of the padding template. We further initialize the projection matrices M_{down}^i, M_{up}^i on the second adapter module A_W^i as identity matrices. The proposed framework encounters significant losses and fails to converge unless this initialization strategy is followed to preserve Whisper’s latent structure.

Table 2: The experimental results are presented in terms of WER without text normalization. The performance of our proposed framework is reported in rows 2 – 4. Oracle refers to the candidate among the n-best hypothesis with the lowest word error rate compared to the ground truth. Rows 5 – 9 represent different ablation experiments on the best-performing model, \mathcal{WL}_M . The WERR is measured relative to the oracle performance as shown in B.2

#	Method	#Para.	ATIS	GS_E	GS_P	GS_S	GS_M	WER _{Avg} (↓)	WERR (↑)
1	Oracle	-	13.76	28.22	22.84	23.93	19.5	21.64	-
2	\mathcal{WL}_L	26.40M	2.04	21.76	19.21	20.55	11.6	15.03	30.52
3	\mathcal{WL}_M	7.97M	1.77	21.61	16.20	18.02	9.82	13.48	37.66
4	\mathcal{WL}_S	4.89M	1.89	22.24	17.23	19.157	10.185	14.144	34.62
5	\mathcal{WL}_M w/o masking	4.89M	3.94	27.56	18.10	21.71	12.79	20.04	22.25
6	\mathcal{WL}_M w/o $H_{acoustics}$	4.89M	253.20	123.19	203.44	376.81	256.44	242.61	-1020.68
7	\mathcal{WL}_M w/o init.	4.89M	405.83	500.58	414.34	461.63	390.64	434.60	-1907.45
8	\mathcal{WL}_M w/o SA_W	1.22M	1.66	24.99	18.734	20.73	10.86	15.39	28.83
9	Big-scale Adapter	4.91M	1.45	23.65	16.59	19.93	10.62	14.45	33.21

4 Experimental Setup

4.1 Models

For our experiments, we utilize the LLaMA-7B model architecture. As we instruct the language model with the generated hypotheses (as illustrated in Figure 4.3.1) to perform generative error correction, we initialize our model weights with Alpaca (Taori et al., 2023), a model fine-tuned from LLaMA-7B, utilizing 52,000 instruction-following demonstrations to enable instruction following abilities. To extract audio representations from input audio clips, we employ Whisper-Large V2, a model with 1.55B parameters trained on 620,000 hours of audio data. Additionally, we employ Whisper-Tiny, a model with 70M parameters, for generating our transcripts, as described in the subsequent section 4.2. We name our model Whispering LLaMA (\mathcal{WL}) and train three model variants with our proposed framework with $t = 10$ and $r = 8, 16, 32$ named \mathcal{WL}_L (Large), \mathcal{WL}_M (Medium), \mathcal{WL}_S (Small), respectively. We design \mathcal{WL}_L with two separate A_W adapters modules for key and value, respectively. \mathcal{WL}_M and \mathcal{WL}_S use the same A_W adapter module as explained in section 3.2 to reduce trainable parameters.

4.2 Dataset

We curate our own transcripts by leveraging two datasets: the Airline Travel Information System (Hemphill et al., 1990) (ATIS) and GigaSpeech (Chen et al., 2021). ATIS consists of audio recordings of individuals querying flight information. GigaSpeech, contains audio from audio-books, podcasts and YouTube videos on diverse topics. ATIS represents a semantically correct, domain-specific dataset, while GigaSpeech repre-

sents a more noisy, real-world setting in our evaluation. We select domain-specific subsets in GigaSpeech and focus on three specific categories: Entertainment, People and Blogs, and Science and Technology. To explore performance variations with respect to number of data points, we further divide the Science and Technology category into two subsets. Table 1 provides detailed information on the number of training points per dataset. We chose Whisper-Tiny to generate the n-best hypothesis baseline to establish a robust evaluation environment that aligns more closely with real-world settings dealing with sub-optimal hypotheses. By employing Whisper-Tiny, we mimic a weaker acoustic model with lower-quality hypotheses. Feeding the language model with better-quality hypotheses from Whisper-Large would make the generative error correction task less challenging for the language model and does not explore the model’s performance under practical settings where our method is intended to be employed. However, we emphasize that our method remains effective when starting with a Whisper-Large hypothesis in Appendix E.

For each audio clip, we generate 200 hypotheses using a top-k value of 200 and a randomly selected temperature between the range of [0.7, 0.8]. Subsequently, we filter out redundant sentences and select the top 15 hypotheses with the highest log probability.

4.3 Training Pipeline

The input to our model consists of the encoded audio representations extracted from the Whisper-Large model, accompanied by the 15 best transcripts generated by the Whisper-Tiny model. We employ the prompt template used by the Alpaca model as represented in Appendix Fig 5.

We utilize the Adam optimizer (Kingma and Ba, 2014) and experiment with $1e^{-2}$, $1e^{-3}$, and $5e^{-4}$ as learning rate and choose the optimal one. We train for 25 epochs with early stopping to prevent over fitting using 2 A100 GPUs for efficient parallel processing. We employ an effective batch size of 32 and apply a weight decay of $1e^{-2}$.

4.3.1 LLM Prompt Examples for ASR

We utilize the Alpaca (Taori et al., 2023) prompt template, as depicted in Fig 5 (in the Appendix), to produce the n-best hypotheses. This template features an instructional segment designated by the **Instruction** tag, which offers guidance to the model. Essential contextual data required by the model is housed under the Input tag. The prompt concludes with the **Response** tag, directing the model to enact the specified instruction within the supplied input context. Rather than adopting the recent advancements of task-activating prompting (Yang et al., 2023a) (TAP), we opt to feed the LLM with its task-specific data (e.g., speech recognition in our instance). This approach facilitates second-pass error correction, mitigating the latency issues observed in the extensive context windows of the TAP based generative ASR error correction.

4.4 Performance Studies

Results from our experiments have been reported in Table 2. The $\mathcal{W}\mathcal{L}_M$ model achieves the best performance, a word-error-rate relative (WERR B.2) of 37.66%. Performance between $\mathcal{W}\mathcal{L}_L$ and $\mathcal{W}\mathcal{L}_M$ suggests that having a separate adapter module for key and value pairs does not yield any performance improvements. Further dataset specific analysis is reported in Appendix B. The models perform better on the Gigaspeech dataset as the number of data points increases.

4.5 Ablation Studies

We empirically discover that masking the prompt except for the ground truth in the cross entropy loss function significantly improves the performance. We attribute this improvement to the model’s enhanced capacity to grasp accurate semantics, achieved by refraining from penalizing the model for erroneous sentences found in the n-best hypotheses. Row 5 represents the performance of $\mathcal{W}\mathcal{L}_M$ without masking. We further investigate if the proposed framework is utilizing the audio representations from Whisper by substituting them with random tensors generated from a normal dis-

tribution as the input (Row 6). Additionally, we explore the significance of our weight initialization mechanism by replacing it with random initialization (Row 7). Both of these ablation studies validate our intuition, demonstrating that the method utilizes acoustic features effectively and highlight the importance of the initialization mechanism in preserving the latent structure of the acoustic embeddings. For further insights, please refer to Appendix D. We also remove the Whisper adapter (SA_W) module for an text feature only baseline performance using adapters (Row 9). Since the disparity between the number of trainable parameters is high, we train another model with an increased adapter context of $t' = 4t$ (Row 9).

5 Conclusion

We propose a novel framework to leverage the external knowledge from LLM to improve the transcription accuracy of ASR systems. Our framework presents a parameter-efficient way to integrate large foundational Speech and Language models to achieve competitive WERR improvements. We further conduct extensive ablation experiments to validate our intuitions and open source our code and pretrained-weights to the research community.

6 Limitation

Using large models such as LLaMA is intuitive, as it provides a comprehensive comprehension of language structure owing to its internet-scaled pretraining. However deploying and conducting research with these systems in real-world scenarios is challenging due to the computationally intensive nature of large language models. We further attempt to design our framework in a parameter efficient manner by reusing multiple model components and using adapters to fuse models. However incorporating audio representations into the training pipeline increases the training duration by 394.76%. Furthermore, our proposed solution demonstrates a need for a larger volume of data to achieve optimal performance, despite having a modest parameter count of only 7.97M to integrate foundational models. During our experimentation, we encountered issues related to over-fitting on datasets. To mitigate this problem, we trained with a reduced learning rate and monitored the Word Error Rate (WER) performance throughout the training process and selected the model checkpoint with the best performance to implement early stopping.

References

- Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. 2016. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning*, pages 173–182. PMLR.
- William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. 2016. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *Proc. ICASSP*, pages 4960–4964. IEEE.
- Kai-Wei Chang, Yu-Kai Wang, Hua Shen, Iu-thing Kang, Wei-Cheng Tseng, Shang-Wen Li, and Hung-yi Lee. 2023. Speechprompt v2: Prompt tuning for speech classification tasks. *arXiv preprint arXiv:2303.00733*.
- Chen Chen, Yuchen Hu, Chao-Han Huck Yang, Sabato Macro Siniscalchi, Pin-Yu Chen, and Eng Siong Chng. 2023a. Hyporadise: An open baseline for generative speech recognition with large language models. *arXiv preprint arXiv:2309.15701*.
- Guoguo Chen, Shuzhou Chai, Guanbo Wang, Jiayu Du, Wei-Qiang Zhang, Chao Weng, Dan Su, Daniel Povey, Jan Trmal, Junbo Zhang, et al. 2021. GigaSpeech: An evolving, multi-domain asr corpus with 10,000 hours of transcribed audio. In *Proc. Interspeech 2021*.
- Zih-Ching Chen, Chin-Lun Fu, Chih-Ying Liu, Shang-Wen Daniel Li, and Hung-yi Lee. 2023b. Exploring efficient-tuning methods in self-supervised speech models. In *Proc. SLT*, pages 1120–1127. IEEE.
- Zih-Ching Chen, Yu-Shun Sung, and Hung-yi Lee. 2023c. Chapter: Exploiting convolutional neural network adapters for self-supervised speech models. In *Proc ICASSP Workshop*, pages 1–5. IEEE.
- Zih-Ching Chen, Chao-Han Huck Yang, Bo Li, Yu Zhang, Nanxin Chen, Shou-Yiin Chang, Rohit Prabhavalkar, Hung-yi Lee, and Tara N Sainath. 2023d. How to estimate model transferability of pre-trained speech models? In *Proc. Interspeech 2023*.
- Stefan Elfving, Eiji Uchibe, and Kenji Doya. 2018. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107:3–11.
- Hayato Futami, Hirofumi Inaguma, Masato Mimura, Shinsuke Sakai, and Tatsuya Kawahara. 2021. Asr rescoring and confidence estimation with electra. In *Proc. ASRU*, pages 380–387. IEEE.
- Ankur Gandhe and Ariya Rastrow. 2020. Audio-attention discriminative language model for asr rescoring. In *Proc. ICASSP*, pages 7944–7948. IEEE.
- Jinxi Guo, Tara N Sainath, and Ron J Weiss. 2019. A spelling correction model for end-to-end speech recognition. In *Proc. ICASSP*, pages 5651–5655. IEEE.
- Charles T Hemphill, John J Godfrey, and George R Doddington. 1990. The atis spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*.
- Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.
- Hu Hu, Xuesong Yang, Zeynab Raeesy, Jinxi Guo, Gokce Keskin, Harish Arsikere, Ariya Rastrow, Andreas Stolcke, and Roland Maas. 2021a. Redat: Accent-invariant representation for end-to-end asr by domain adversarial training with relabeling. In *Proc. ICASSP*, pages 6408–6412. IEEE.
- Ke Hu, Ruoming Pang, Tara N Sainath, and Trevor Strohman. 2021b. Transformer based deliberation for two-pass speech recognition. In *Proc. SLT*, pages 68–74. IEEE.
- Ke Hu, Tara N Sainath, Ruoming Pang, and Rohit Prabhavalkar. 2020. Deliberation model based two-pass end-to-end speech recognition. In *Proc. ICASSP*, pages 7799–7803. IEEE.
- Yun-Ning Hung, Chao-Han Huck Yang, Pin-Yu Chen, and Alexander Lerch. 2023. Low-resource music genre classification with cross-modal neural model reprogramming. In *Proc. ICASSP*, pages 1–5. IEEE.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Ke Li, Jay Mahadeokar, Jinxi Guo, Yangyang Shi, Gil Keren, Ozlem Kalinli, Michael L Seltzer, and Duc Le. 2023. Improving fast-slow encoder based transducer with streaming deliberation. In *Proc. ICASSP*, pages 1–5. IEEE.
- Linda Liu, Yile Gu, Aditya Gourav, Ankur Gandhe, Shashank Kalmane, Denis Filimonov, Ariya Rastrow, and Ivan Bulyko. 2021. Domain-aware neural language models for speech recognition. In *Proc. ICASSP*, pages 7373–7377. IEEE.
- Rao Ma, Mark JF Gales, Kate Knill, and Mengjie Qian. 2023. N-best t5: Robust asr error correction using multiple input hypotheses and constrained decoding space. *arXiv preprint arXiv:2303.00456*.
- Zhong Meng, Weiran Wang, Rohit Prabhavalkar, Tara N Sainath, Tongzhou Chen, Ehsan Variiani, Yu Zhang, Bo Li, Andrew Rosenberg, and Bhuvana Ramabhadran. 2023. Jeit: Joint end-to-end model and internal language model training for speech recognition. In *Proc. ICASSP*, pages 1–5. IEEE.

- OpenAI. 2023a. Chatgpt. <https://openai.com/blog/chatgpt/>.
- OpenAI. 2023b. Gpt-4 technical report.
- Puyuan Peng, Brian Yan, Shinji Watanabe, and David Harwath. 2023. Prompting the hidden talent of web-scale speech models for zero-shot task generalization. *arXiv preprint arXiv:2305.11095*.
- Rohit Prabhavalkar, Tara N Sainath, Yonghui Wu, Patrick Nguyen, Zhifeng Chen, Chung-Cheng Chiu, and Anjuli Kannan. 2018. Minimum word error rate training for attention-based sequence-to-sequence models. In *Proc. ICASSP*, pages 4839–4843. IEEE.
- Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision. *arXiv preprint arXiv:2212.04356*.
- Srijith Radhakrishnan, Chao-Han Huck Yang, Sumeer Ahmad Khan, Narsis A. Kiani, David Gomez-Cabrero, and Jesper N. Tegner. 2023. A parameter-efficient learning approach to arabic dialect identification with pre-trained general-purpose speech model. In *Proc. Interspeech 2023*, pages 1958–1962.
- Julian Salazar, Davis Liang, Toan Q Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712.
- Joonbo Shin, Yoonhyung Lee, and Kyomin Jung. 2019. Effective sentence scoring method using bert for speech recognition. In *Asian Conference on Machine Learning*, pages 1081–1093. PMLR.
- Adam Stooke, Khe Chai Sim, Mason Chua, Tsendsuren Munkhdalai, and Trevor Strohman. 2023. Internal language model personalization of e2e automatic speech recognition using random encoder features. In *Proc. SLT*, pages 213–220. IEEE.
- Tomohiro Tanaka, Ryo Masumura, Mana Ihori, Akihiko Takashima, Takafumi Moriya, Takanori Ashihara, Shota Orihashi, and Naoki Makishima. 2021. Cross-modal transformer-based neural correction models for automatic speech recognition. In *Proc. Interspeech 2021*. ISCA.
- Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- Gokhan Tur and Renato De Mori. 2011. *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplín, Jahn Heymann, Matthew Wiesner, Nanxin Chen, et al. 2018. Espnet: End-to-end speech processing toolkit. *Proc. Interspeech 2018*.
- Yingce Xia, Fei Tian, Lijun Wu, Jianxin Lin, Tao Qin, Nenghai Yu, and Tie-Yan Liu. 2017. Deliberation networks: Sequence generation beyond one-pass decoding. *Advances in neural information processing systems*, 30.
- Liyan Xu, Yile Gu, Jari Kolehmainen, Haidar Khan, Ankur Gandhe, Ariya Rastrow, Andreas Stolcke, and Ivan Bulyko. 2022. Rescorebert: Discriminative speech recognition rescoring with bert. In *Proc. ICASSP*, pages 6117–6121. IEEE.
- Chao-Han Huck Yang, Yile Gu, Yi-Chieh Liu, Shalini Ghosh, Ivan Bulyko, and Andreas Stolcke. 2023a. Generative speech recognition error correction with large language models. *arXiv preprint arXiv:2309.15649*.
- Chao-Han Huck Yang, Bo Li, Yu Zhang, Nanxin Chen, Rohit Prabhavalkar, Tara N Sainath, and Trevor Strohman. 2023b. From english to more languages: Parameter-efficient model reprogramming for cross-lingual speech recognition. In *Proc. ICASSP*, pages 1–5. IEEE.
- Chao-Han Huck Yang, Linda Liu, Ankur Gandhe, Yile Gu, Anirudh Raju, Denis Filimonov, and Ivan Bulyko. 2021a. Multi-task language modeling for improving speech recognition of rare words. In *Proc. ASRU*, pages 1087–1093. IEEE.
- Chao-Han Huck Yang, Yun-Yun Tsai, and Pin-Yu Chen. 2021b. Voice2series: Reprogramming acoustic models for time series classification. In *International Conference on Machine Learning*, pages 11808–11819. PMLR.
- Yu Yu, Chao-Han Huck Yang, Jari Kolehmainen, et al. 2023. Low-rank adaptation of large language model rescoring for parameter-efficient speech recognition. *arXiv preprint arXiv:2309.15223*.
- Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. 2023. Multi-modal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*.

A Appendix

In this Appendix, We investigate the performance difference between datasets in Section B, provide illustrations of the model-level architectural design and prompt template in Section C and 4.3.1 respectively. Provide more insight into the results from ablation studies in Section D and Whisper Large Hypothesis baseline in Section E.

B Dataset Analysis

We report the WER from our experiments before and after text normalization on Table 3. We convert the model prediction and the ground-truth to lower-case and remove punctuation during text normalization. The ATIS dataset is not impacted by text normalization because the dataset does not contain any punctuation. It only contains contractions such as “I’d like” instead of “I would like”. ATIS consists of audio recordings of individuals querying automated airline travel inquiry systems for flight information. We believe the lack of punctuation and the consistent structure present within the ATIS dataset enables improved WER performance compared to GigaSpeech. The GigaSpeech dataset contains punctuation and lacks consistency within the dataset because it has diverse categories and sources such as audiobooks, podcasts and YouTube videos.

B.1 More Discussion on Ground Truth Match Rate

During dataset generation, we remove the ground truth if it is present among the Whisper generated n-best hypothesis. This allows us to introduce a new metric called Ground Truth Match Rate (GTMR). GTMR calculates the percentage of predictions generated by the model that exactly match the ground-truth. This metric indicates the model’s to learn the structure of the data source. The GTMR of our experiments before and after text normalization is reported in Table 5. The model is able to learn the structure of the dataset better with more data points as observed from the performance difference between ST-S and ST-M. It can also be observed that the model is able to learn the simpler structure of ATIS much better than other GigaSpeech datasets.

B.2 WERR

Word error rate relative is calculated as

$$WERR(i) \leftarrow \frac{Oracle(i) - WER(i)}{Oracle(i)} \times 100 \quad (7)$$

where $Oracle(i)$ refers to the average oracle performance in terms of WER and $WER(i)$ refers to the average performance of a particular method.

C Proposed Architecture Illustrations

We present a model-level overview of our proposed method described in Section 3.2 in Fig 2. We add two modules into each layer of LLaMA. The Whisper Cross-Attention module and Adapter module which refer to A_W and A_L , respectively. We initialize the Whisper Cross-Attention module with the Whisper decoder model, as illustrated. LLaMA takes the encoded features generated by Whisper encoder and the n-best hypothesis generated by the Whisper in a prompt format as input to generate the error-corrected response.

D Failure Case Studies of Generative ASR with Whispering-LLaMA

Since the WER error rate in row 6 (\mathcal{WL}_M w/o audio representations) and row 7 (\mathcal{WL}_M w/o initialization) of table 2 is high and provides no insight into model performance, we present the training loss graphs of the best performing model (\mathcal{WL}_M) with and without audio representations in Figure 4. The model is not able to converge below a certain threshold without audio representations. Additionally, we include the training loss graphs of \mathcal{WL}_M with and without our initialization mechanism in Figure 3. Without our initialization mechanism, the latent structure of the Whisper encoder embedding is not preserved, leading to an inability to converge.

E Whisper Large Decoding Baseline

We report the results of using the hypothesis generated by Whisper Large to train our best-performing model (\mathcal{WL}_M) on GigaSpeech Entertainment (GS_E) and Science and Technology (GS_E) datasets on Table 4. By leveraging the LLaMA model with the proposed generative error correction mechanism, we are able to match the performance of the Whisper Large model with 1.5 billion parameters by using a Whisper -Tiny model with just 70 million parameters. Using the hypotheses generated by Whisper Large results in a higher

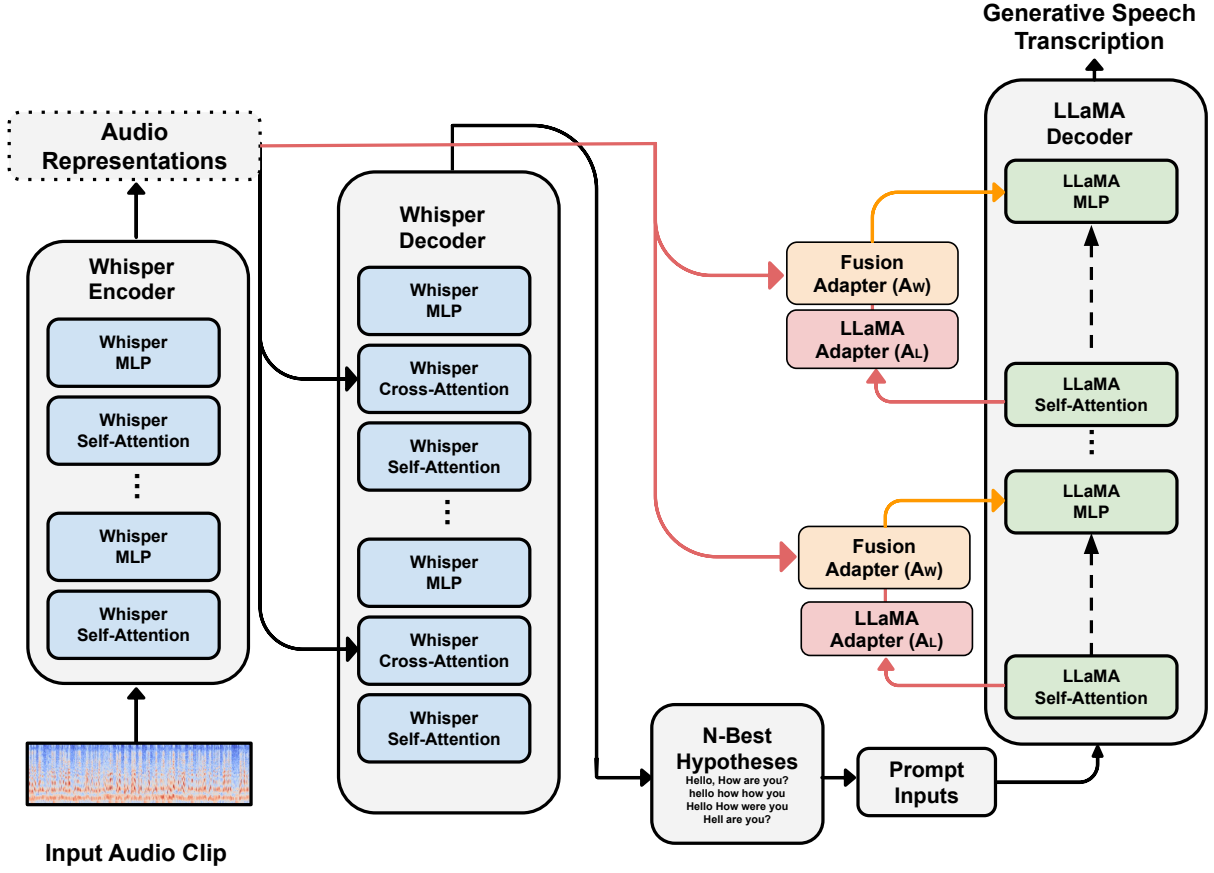


Figure 2: Whispering-LLaMA model-overview of proposed adaptation pipelines described in Section 3.2

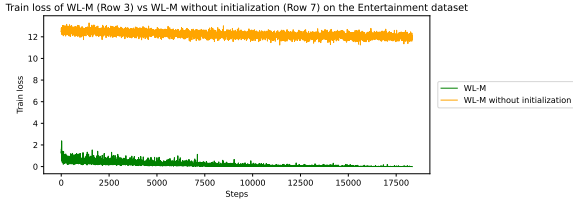


Figure 3: Train loss of \mathcal{WL}_M (Row 3) vs \mathcal{WL}_M without initialization (Row 7) on the Entertainment dataset

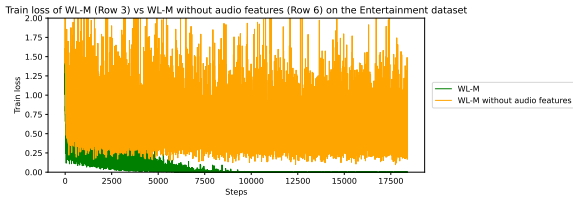


Figure 4: Train loss of \mathcal{WL}_M (Row 3) vs \mathcal{WL}_M without audio representations (Row 6) on the Entertainment dataset

WERR as expected. This finding confirms the effectiveness of the proposed approach, particularly in the context of Whisper Large generated N-best hypotheses.

Table 3: The experimental results in terms of Word Error Rate (WER), Before and after text normalization. We convert all text to lowercase and remove the following punctuation [".", "-", "?", """]. Rows 1-3 represent before text normalization and Rows 4-6 represent after text normalization

#	Method	ATIS	ET	P&B	ST-S	ST-M	Avg
1	\mathcal{WL}_L	2.04	21.76	19.21	20.55	11.6	15.03
2	\mathcal{WL}_M	1.77	21.61	16.20	18.02	9.82	13.48
3	\mathcal{WL}_S	2.11	23.60	17.13	20.16	10.73	14.75
4	\mathcal{WL}_L	2.04	14.71	13.36	14.23	7.44	10.35
5	\mathcal{WL}_M	1.77	17.71	10.83	12.00	6.07	9.67
6	\mathcal{WL}_S	1.89	15.21	11.49	13.03	6.41	9.61

F Reproducibility Resources

We have open-sourced the pre-trained model weights and code, available at <https://github.com/Srijith-rkr/Whispering-LLaMA>. Our future plan includes integrating this baseline into both Espnet (Watanabe et al., 2018) and HyPoradise (Chen et al., 2023a) to accommodate a broader range of use cases.

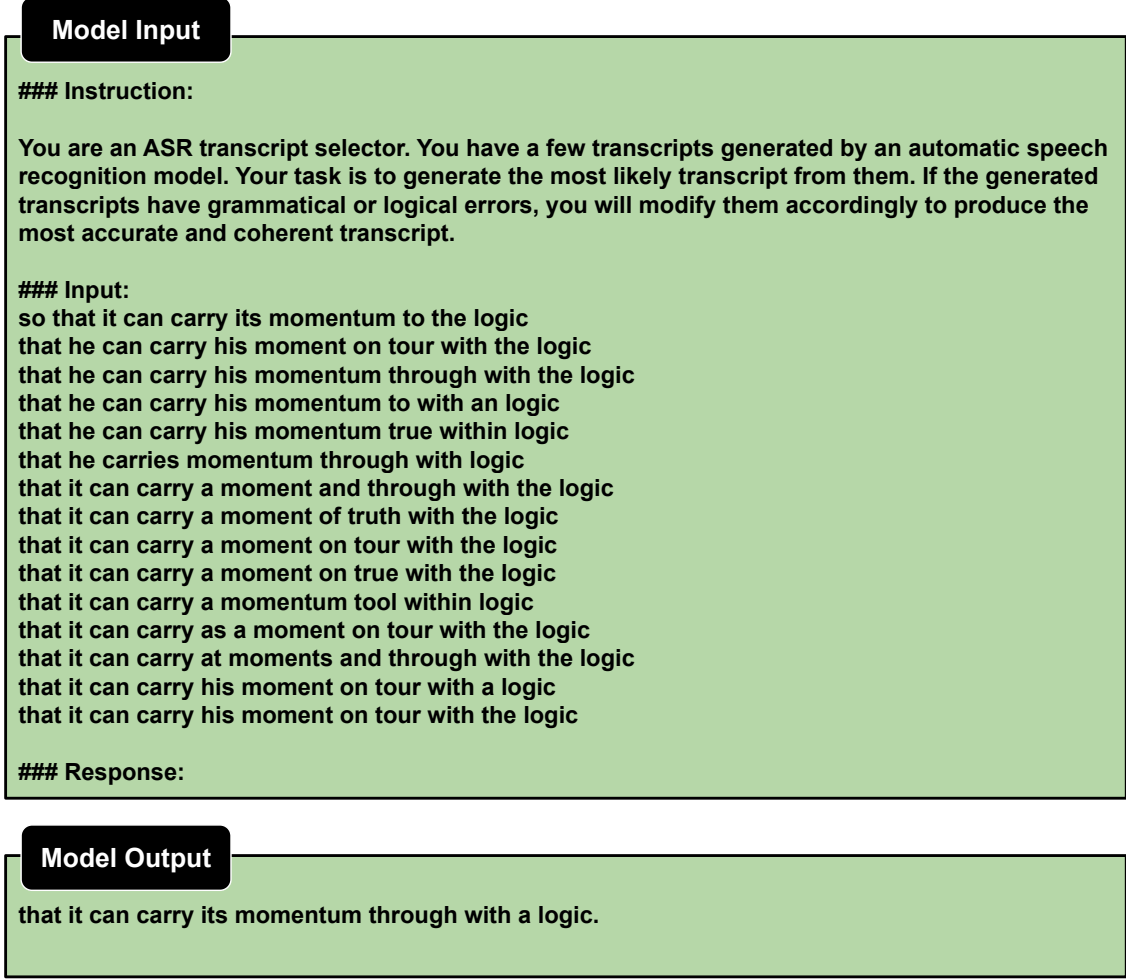


Figure 5: Illustration of the Alpaca prompt template used in our proposed framework

Table 4: Results of employing a Whisper Large generated hypothesis baseline in comparison to the proposed Whisper Tiny hypothesis baseline

Method	GS_E	GS_S	$WER_{Avg} \downarrow$	$WERR \uparrow$
Oracle with Whisper Tiny	28.22	23.93	26.08	-
\mathcal{WL}_M with Whisper Tiny	21.61	18.02	19.82	24.01
Oracle with Whisper Large	21.78	18.09	19.93	-
\mathcal{WL}_M with Whisper Large	15.59	12.77	14.18	28.86

Table 5: The experimental results in terms of Ground Truth Match Rate (GTMR), Before and after text normalization. Rows 1-3 represent before text normalization, and Rows 5-7 represent after text normalization. Row 4 and 6 denote the average GTMR across datasets and columns Avg denotes average GTMR across each method

#	Method	ATIS	ET	P&B	ST-S	ST-M	Avg
1	\mathcal{WL}_L	86.1	26.5	24.1	24.8	36.2	39.5
2	\mathcal{WL}_M	88.3	26.3	31.4	28.8	41.0	43.17
3	\mathcal{WL}_S	87.5	25.2	27.6	26.6	38.6.1	41.10
4	Avg	87.3	26.0	27.7	26.7	38.6	-
5	\mathcal{WL}_L	86.1	46.9	46	45.5	56.4	56.2
6	\mathcal{WL}_M	88.3	47.0	54.3	49.2	62.7	60.3
7	\mathcal{WL}_S	87.5	44.3	49.5	46.3	60.0	57.5
8	Avg	87.3	46.0	49.9	47.0	59.7	-