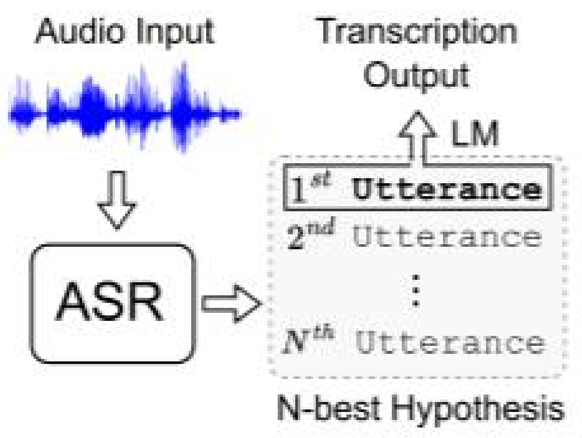
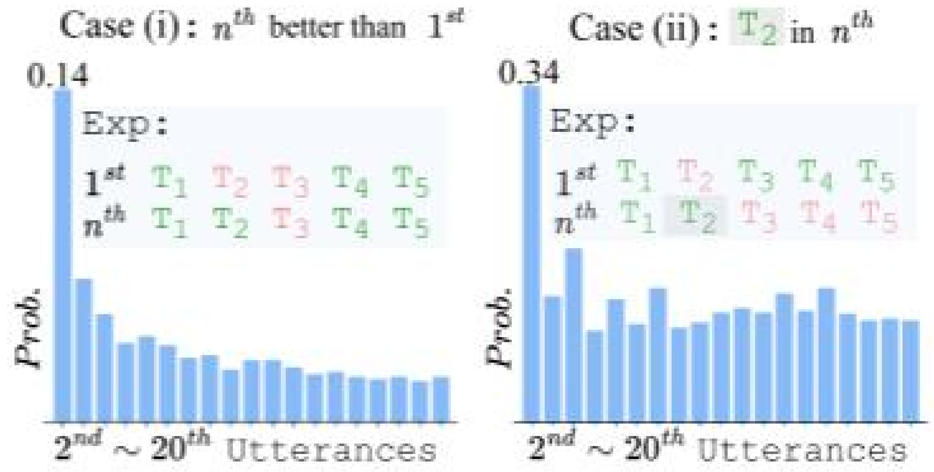
Large Language Models are Efficient Learners of Noise-Robust Speech Recognition

Background

- **Noisy-Robust ASR**
  - Require model take noisy speech as input and still retain its ASR quality, which more consistent with ASR in real scenario.
- **Generative Error Correction (GER) for ASR with LLMs**
  - Different from usually used LM-rescore method just to rerank the N-best hypotheses generated by ASR model and choose the best one as output.



LM-rescore method

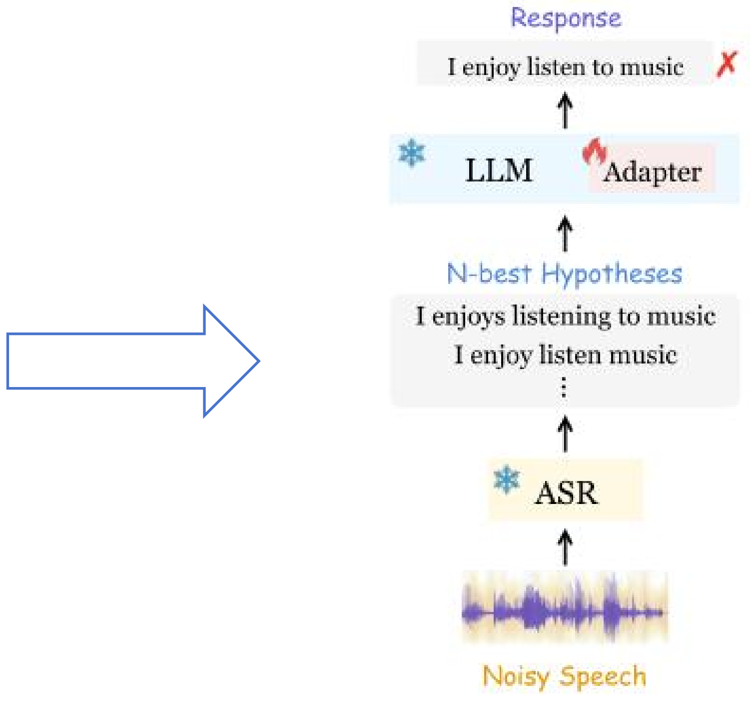


Error Analysis for N-best hypotheses

Background

- **Generative Error Correction (GER) for ASR with LLMs**
  - **LLMs-based GER** use **N-best hypotheses** and corresponding **transcription** for in-context learning (ICL) to **fine-tune** the LLMs, and then use the fine-tuned LLMs to perform Error Correction for ASR model.
  - This method obviously improves the ASR quality, but the performance gain in most noisy conditions is still limited

Input scheme for LLMs-based GER



Introduction

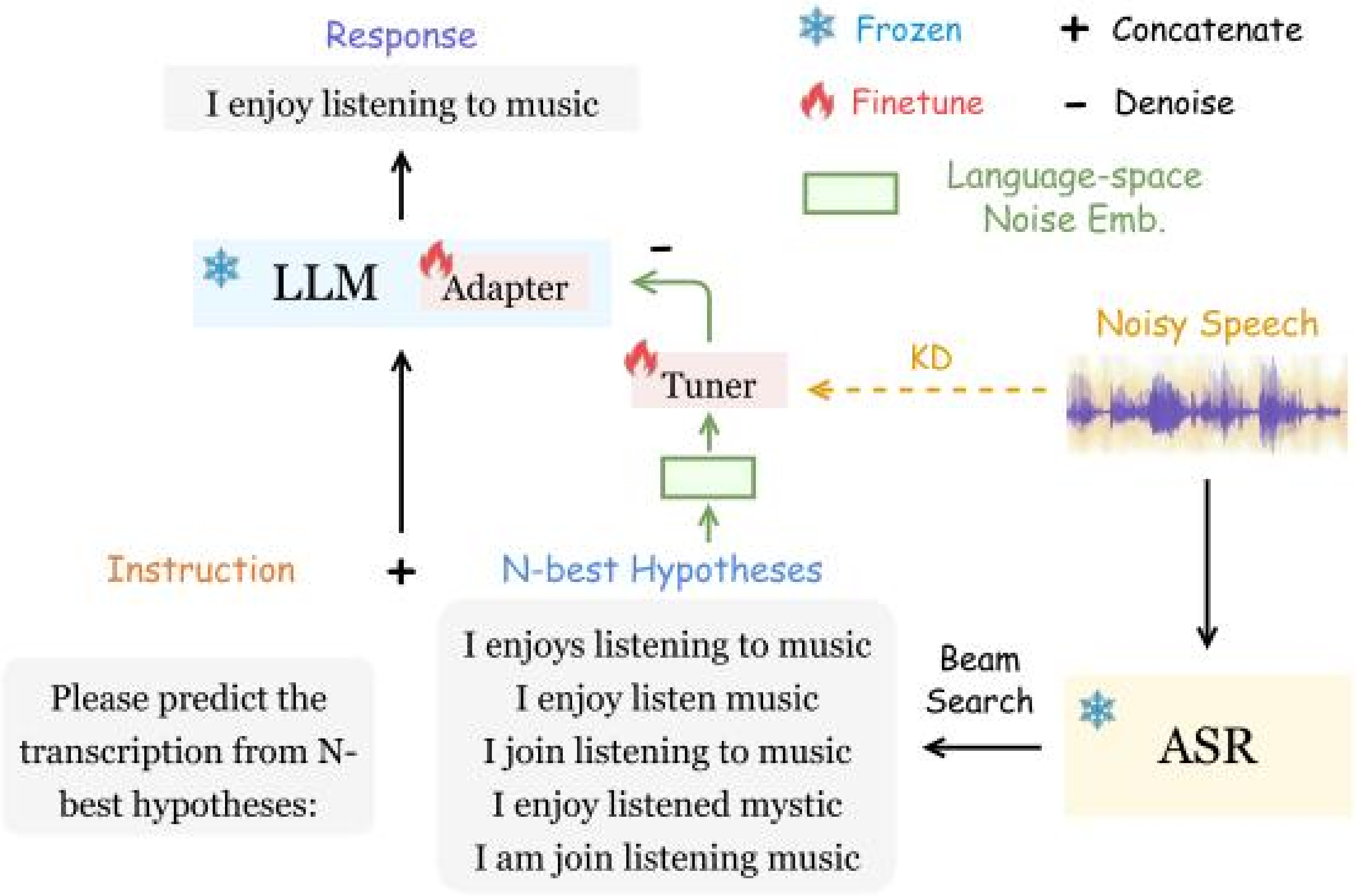
- **Main Contributions**
  - Built a **Robust HyPoradise (RobustHP)** dataset with 113K hypotheses-transcription pairs is collected from various ASR corpus **in common noisy conditions**, extending latest ASR GER benchmark to noise-robust ASR.
  - Proposed **RobustGER**, a noise-aware GER approach based on LLMs to map N-best hypotheses to true transcription, where **an extracted language-space noise embedding with audio distillation** is utilized to teach LLMs to perform denoising.


```
"input":[
    "the sale of the hotels is part of holiday strategy to sell off assets and concentrate on property management",
    "the sale of the hotels is part of holiday strategy to sell off assets and concentrate on property management",
    "the sale of the hotels is part of a holiday strategy to sell off assets and concentrate on property management",
    "the sale of the hotels is part of holiday strategy to sell off assets and concentrate on property management",
    "the sale of the hotels is part of a holiday strategy to sell off assets and concentrate on property management"
],
"output": "the sale of the hotels is part of holiday is strategy to sell off assets and concentrate on property management",
```


An example of HPdataset

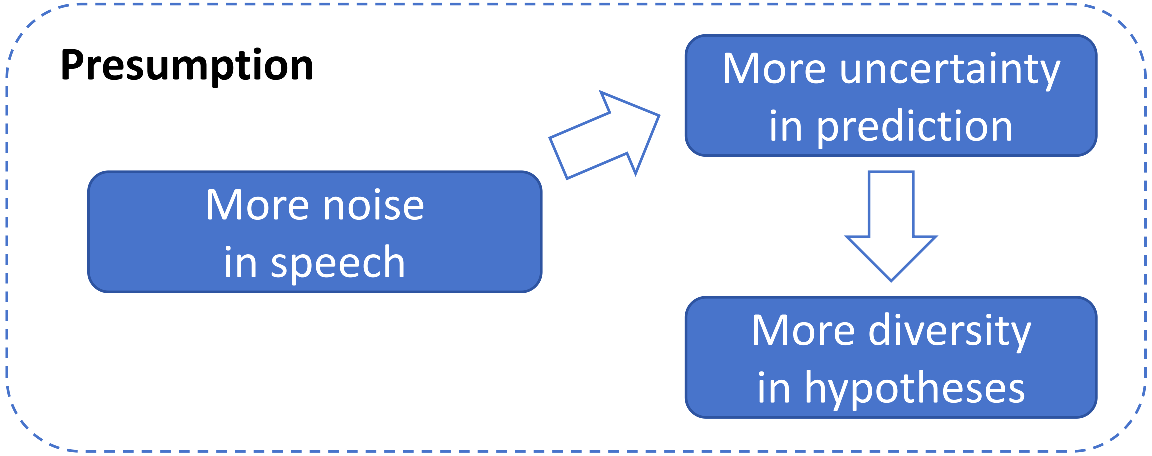
Method

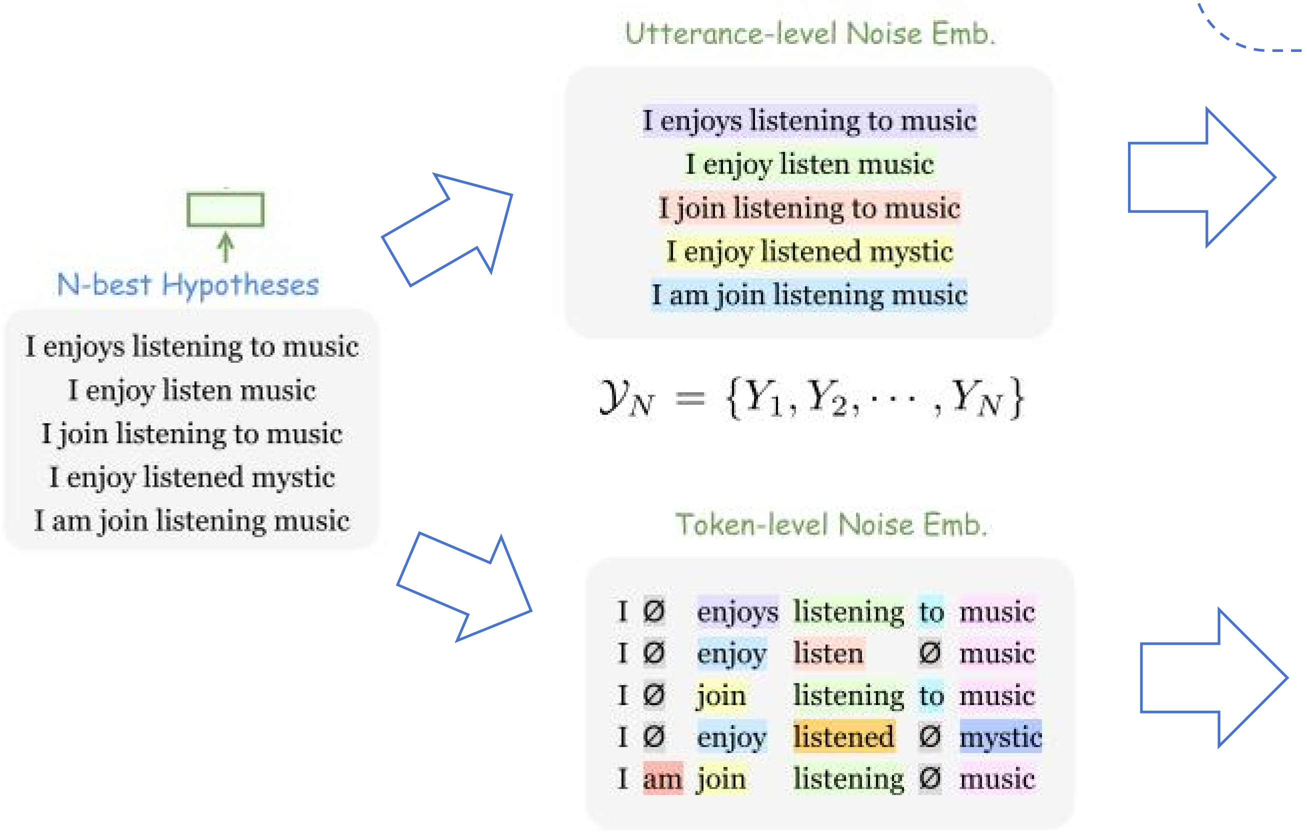
- **Overall Framework**



Method

- **Language-space Noise Embedding**





$$\mathcal{Y}_N = \{Y_1, Y_2, \cdots, Y_N\}$$

$$\mathcal{Y}_N^{ali} = \{Y_1^{ali}, Y_2^{ali}, \cdots, Y_N^{ali}\}$$

$$Y_i^{ali} = [y_{i_1}^{ali}, y_{i_2}^{ali}, \cdots, y_{i_T}^{ali}], \quad y_{i_t}^{ali} \in \mathcal{V} \cup \varnothing,$$

$$E_{\text{LN}}^{utt} = \text{Concat}\{[\mathcal{E}_{\text{sbert}}(Y_i) - \mathcal{E}_{\text{sbert}}(Y_j)]_{i,j=1,i>j}^{N}\} \in \mathbb{R}^{\frac{N\cdot(N-1)}{2} \times D_{\text{sbert}}},$$

$$E_{\text{LN}}^{tok} = \text{Concat}\{E_{\text{edit}}(Y_i^{ali}, Y_j^{ali})_{i,j=1,i>j}^{N}\} \in \mathbb{R}^{\frac{N(N-1)}{2} \times D_{\text{sbert}}},$$

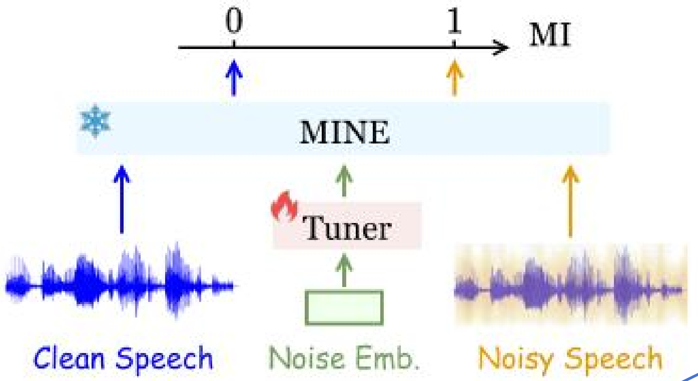
$$E_{\text{edit}}(Y_i^{ali}, Y_j^{ali}) = \sum_{t=1}^{T} [\mathcal{E}_{\text{sbert}}(y_{i_t}^{ali}) - \mathcal{E}_{\text{sbert}}(y_{j_t}^{ali})],$$

Method

- **Audio Noise Distillation**

$$I(X;Z) := H(X) - H(X \mid Z),$$

$$I(X;Z) = D_{KL}(\mathbb{P}_{XZ} \parallel \mathbb{P}_X \mathbb{P}_Z)$$



Presumption

$$(E_{\text{LN}}^{(b)}, \mathcal{E}_{\text{ASR}}(X_n^{(b)})) \sim \mathbb{P}_{XZ}$$

$$\mathcal{E}_{\text{ASR}}(X_c^{(b)}) \sim \mathbb{P}_Z$$

Algorithm 1 Audio noise distillation via mutual information neural estimation (MINE).

Require: LLM $\mathcal{M}_{\text{H2T}}$ with adapter $\mathcal{G}_{\boldsymbol{v}}$, MINE statistics network ψ of parameters $\boldsymbol{\theta}$, language embedding tuner $\mathcal{T}$ of parameters $\boldsymbol{\omega}$. N-best hypotheses $\mathcal{Y}_N$. Parallel noisy speech $\mathcal{X}_n$ and clean speech data $\mathcal{X}_c$. Batch size B and the total number of iterations M. Hyper-parameter weight λ.

for $m = 1$ **to** M **do**

SAMPLE
- Draw B N-best hypotheses samples from RobustHP dataset: $\{\mathcal{Y}_N^{(1)}, \mathcal{Y}_N^{(2)}, \cdots, \mathcal{Y}_N^{(B)}\}$;
- Draw corresponding noisy and clean speech samples: $\{(X_n^{(1)}, X_c^{(1)}), (X_n^{(2)}, X_c^{(2)}), \cdots, (X_n^{(B)}, X_c^{(B)})\}$;

- Extract language-space noise embedding from N-best list using Eq.(4-6): $\{E_{\text{LN}}^{(1)}, E_{\text{LN}}^{(2)}, \cdots, E_{\text{LN}}^{(B)}\}$;

MINE
- Calculate Eq.(8): $\mathcal{I} = \frac{1}{B}\sum_{b=1}^{B} \psi_{\boldsymbol{\theta}}(E_{\text{LN}}^{(b)}, \mathcal{E}_{\text{ASR}}(X_n^{(b)})) - \log(\frac{1}{B}\sum_{b=1}^{B} e^{\psi_{\boldsymbol{\theta}}(E_{\text{LN}}^{(b)}, \mathcal{E}_{\text{ASR}}(X_c^{(b)}))})$;
- Calculate $\boldsymbol{g_\theta} = \nabla_{\boldsymbol{\theta}}(\mathcal{I})$ and update $\boldsymbol{\theta}$ by gradient ascent: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \boldsymbol{g_\theta}$;

Tuner
- Calculate GER cost function $\mathcal{L}_{\text{H2T}}$ using Eq.(2), with $\mathcal{T}_{\boldsymbol{\omega}}(E_{\text{LN}}^{(b)})$ incorporated for denoising;
- Re-calculate the first term of Eq.(8): $\mathcal{I}_1 = \frac{1}{B}\sum_{b=1}^{B} \psi_{\boldsymbol{\theta}}(\mathcal{T}_{\boldsymbol{\omega}}(E_{\text{LN}}^{(b)}), \mathcal{E}_{\text{ASR}}(X_n^{(b)}))$;

- Calculate $\boldsymbol{g}_{\boldsymbol{v},\boldsymbol{\omega}} = \nabla_{\boldsymbol{v},\boldsymbol{\omega}}(\mathcal{L}_{\text{H2T}} - \lambda \mathcal{I}_1)$ and update $\boldsymbol{v}, \boldsymbol{\omega}$ by gradient descent: $\boldsymbol{v} \leftarrow \boldsymbol{v} - \boldsymbol{g}_{\boldsymbol{v}}, \boldsymbol{\omega} \leftarrow \boldsymbol{\omega} - \boldsymbol{g}_{\boldsymbol{\omega}}$;

end for

Algorithm 1 MINE

$\theta \leftarrow$ initialize network parameters

repeat

Draw b minibatch samples from the joint distribution: $(\boldsymbol{x}^{(1)}, \boldsymbol{z}^{(1)}), \ldots, (\boldsymbol{x}^{(b)}, \boldsymbol{z}^{(b)}) \sim \mathbb{P}_{XZ}$

Draw n samples from the Z marginal distribution: $\bar{\boldsymbol{z}}^{(1)}, \ldots, \bar{\boldsymbol{z}}^{(b)} \sim \mathbb{P}_Z$

Evaluate the lower-bound:

$\mathcal{V}(\theta) \leftarrow \frac{1}{b}\sum_{i=1}^{b} T_\theta(\boldsymbol{x}^{(i)}, \boldsymbol{z}^{(i)}) - \log(\frac{1}{b}\sum_{i=1}^{b} e^{T_\theta(\boldsymbol{x}^{(i)}, \bar{\boldsymbol{z}}^{(i)})})$

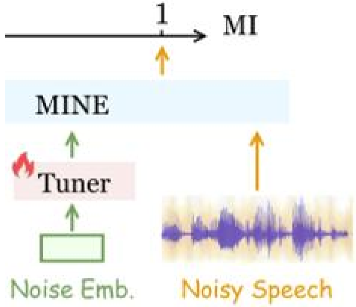
Evaluate bias corrected gradients (e.g., moving average):

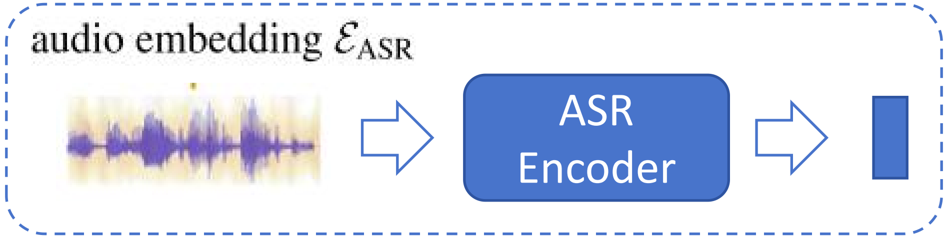
$\widehat{G}(\theta) \leftarrow \widetilde{\nabla}_\theta \mathcal{V}(\theta)$

Update the statistics network parameters:

$\theta \leftarrow \theta + \widehat{G}(\theta)$

until convergence



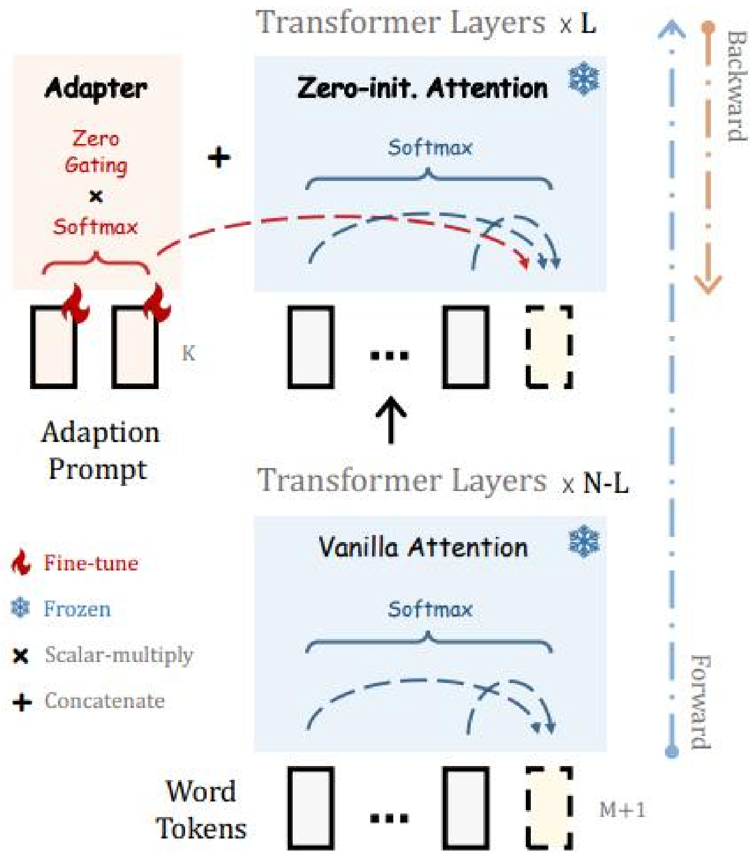


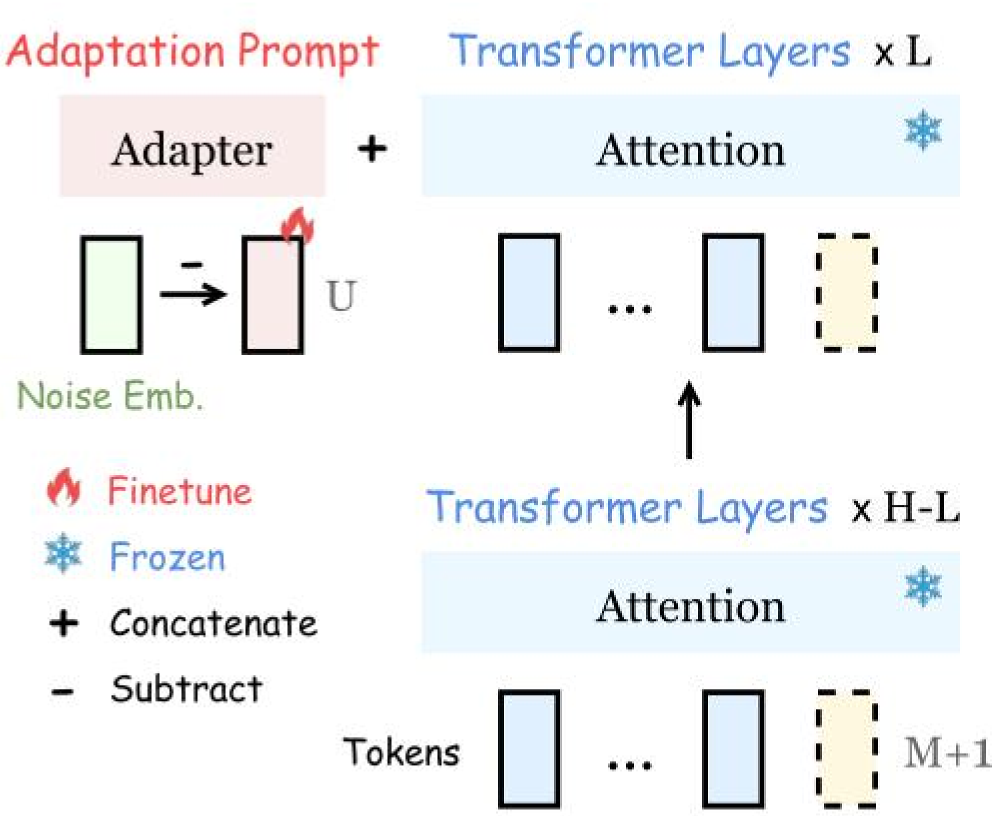
Mutual Information Neural Estimation (Belghazi M I, Baratin A, Rajeshwar S, et al. PMLR 2018)

Method

- **LLM-Adapter tuning with language-space noise embedding**

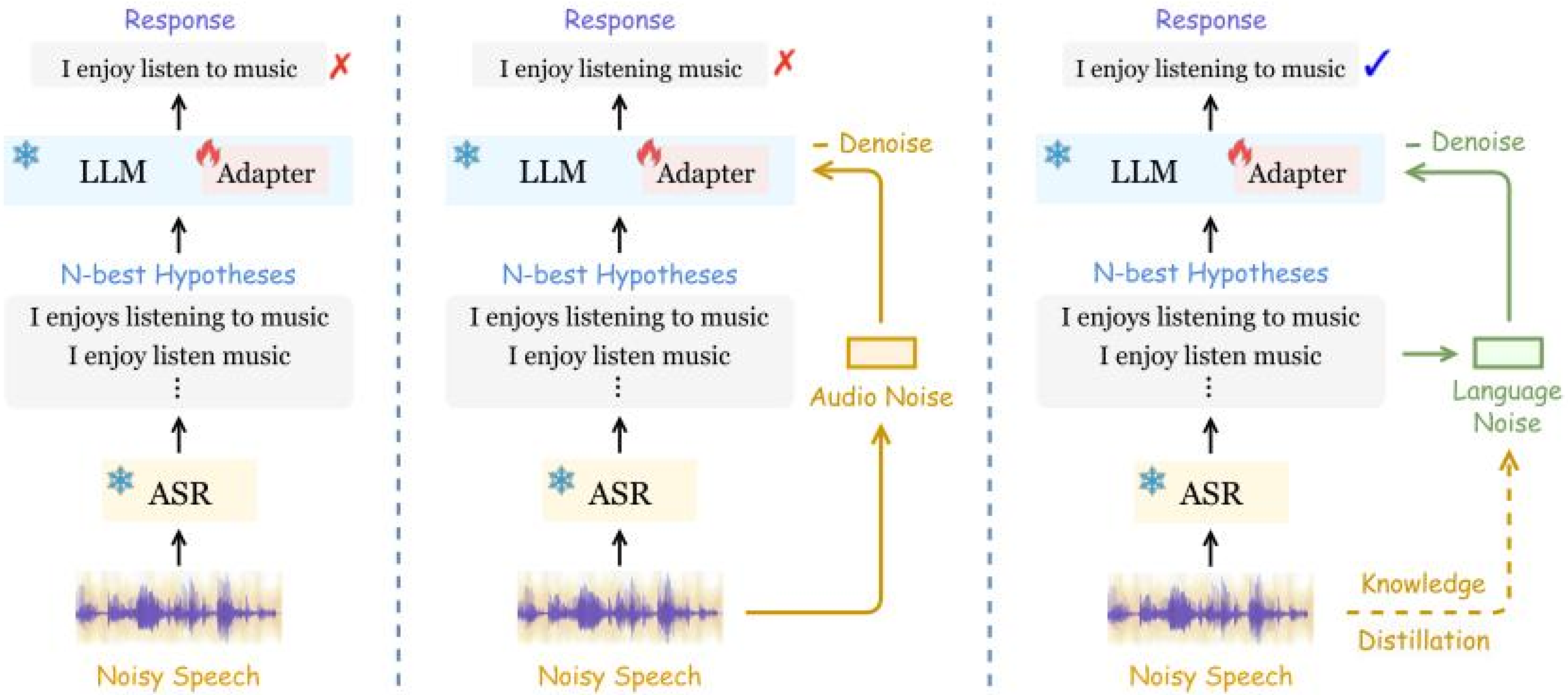
$$\mathcal{G}_l^{\mathrm{dn}} = \mathcal{G}_l - g_l^{\mathrm{dn}} \cdot \mathcal{T}_\omega(E_{\mathrm{LN}}) \in \mathbb{R}^{U \times D}$$





Experiment Results

- **GER vs. GER+Audio Denoising vs. GER+Language Denoising**



(a) GER (b) GER + Audio Denoising (c) GER + Language Denoising (ours)

Experiment Results

- **Main results 1: WER on different noisy speech datasets**

Table 1: WER (%) results of RobustGER with LLaMA-2-7b finetuning. "LM$_{rank}$" denotes LM rescoring. "+ Audio Denoising" denotes introducing audio embedding to denoise GER. o_{nb} and o_{cp} respectively denote the N-best oracle and compositional oracle that are defined in §5.1. The subscript percentage denotes relative WER reduction over ASR baseline, i.e., GER improvement.

Test Set		Baseline	LM$_{rank}$	GER	+ Audio Denoising	RobustGER (ours)	Oracle o_{nb}	Oracle o_{cp}
CHiME-4	*test-real*	12.6	12.2	6.5$_{-48.4\%}$	6.4$_{-49.2\%}$	**5.6**$_{-55.6\%}$	10.5	3.0
	test-simu	15.4	14.5	9.2$_{-40.3\%}$	9.0$_{-41.6\%}$	**8.2**$_{-46.8\%}$	12.9	5.0
	dev-real	10.6	10.3	5.0$_{-52.8\%}$	4.9$_{-53.8\%}$	**4.1**$_{-61.3\%}$	9.1	2.1
	dev-simu	12.4	11.9	6.8$_{-45.2\%}$	6.6$_{-46.8\%}$	**5.8**$_{-53.2\%}$	10.6	3.3
	avg.	12.8	12.2	6.9$_{-46.1\%}$	6.7$_{-47.7\%}$	**5.9**$_{-53.9\%}$	10.8	3.4
VB-DEMAND	*baby-cry*	8.0	7.8	7.0$_{-12.5\%}$	6.9$_{-13.8\%}$	**6.0**$_{-25.0\%}$	4.5	3.0
	helicopter	8.4	8.1	7.4$_{-11.9\%}$	7.3$_{-13.1\%}$	**6.9**$_{-17.9\%}$	4.8	3.2
	crowd-party	22.6	22.3	21.4$_{-5.3\%}$	21.0$_{-7.1\%}$	**19.2**$_{-15.0\%}$	16.5	11.5
	avg.	13.0	12.7	11.9$_{-8.5\%}$	11.7$_{-10.0\%}$	**10.7**$_{-17.7\%}$	8.6	5.9
NOIZEUS	*babble*	16.5	16.7	16.5$_{-0.0\%}$	16.1$_{-2.4\%}$	**14.5**$_{-12.1\%}$	9.5	5.8
	car	17.4	16.8	15.3$_{-12.1\%}$	15.2$_{-12.6\%}$	**14.9**$_{-14.4\%}$	9.9	7.9
	station	12.0	11.6	10.3$_{-14.2\%}$	10.3$_{-14.2\%}$	**9.5**$_{-20.8\%}$	6.6	5.0
	train	15.3	15.2	14.9$_{-2.6\%}$	15.0$_{-2.0\%}$	**14.9**$_{-2.6\%}$	10.3	7.9
	street	17.4	17.2	17.4$_{-0.0\%}$	17.1$_{-1.7\%}$	**16.1**$_{-7.5\%}$	12.4	9.9
	airport	11.2	11.0	10.7$_{-4.5\%}$	10.5$_{-6.3\%}$	**9.5**$_{-15.2\%}$	7.9	4.5
	exhibition	13.2	13.2	12.8$_{-3.0\%}$	12.4$_{-6.1\%}$	**9.5**$_{-28.0\%}$	8.3	5.8
	restaurant	13.2	13.0	12.4$_{-6.1\%}$	12.5$_{-5.3\%}$	**12.0**$_{-9.1\%}$	8.7	6.2
	avg.	14.5	14.3	13.8$_{-4.8\%}$	13.6$_{-6.2\%}$	**12.6**$_{-13.1\%}$	9.2	6.6
LS-FreeSound	*metro*	9.9	9.8	9.5$_{-4.0\%}$	9.4$_{-5.1\%}$	**8.9**$_{-10.1\%}$	7.9	4.9
	car	4.0	4.0	3.7$_{-7.5\%}$	3.5$_{-12.5\%}$	**3.1**$_{-22.5\%}$	3.0	1.8
	traffic	8.3	8.2	8.0$_{-3.6\%}$	7.8$_{-6.0\%}$	**7.5**$_{-9.6\%}$	6.8	4.5
	cafe	9.8	9.5	8.1$_{-17.3\%}$	8.1$_{-17.3\%}$	**7.5**$_{-23.5\%}$	7.1	4.6
	babble	32.0	31.8	31.3$_{-2.2\%}$	31.6$_{-1.3\%}$	**31.1**$_{-2.8\%}$	28.7	19.3
	ac/vacuum	12.4	12.5	12.3$_{-0.8\%}$	12.1$_{-2.4\%}$	**11.4**$_{-8.1\%}$	10.2	6.2
	avg.	12.7	12.6	12.2$_{-3.9\%}$	12.1$_{-4.7\%}$	**11.6**$_{-8.7\%}$	10.6	6.9
RATS	*test*	45.7	45.6	45.2$_{-1.1\%}$	44.8$_{-2.0\%}$	**43.2**$_{-5.5\%}$	38.8	23.6

Experiment Results

- **Main results 2: WER on different noise level**
  - **Also improve ASR quality in clean speech:** Maybe due to its encouragement for the model to mitigate the uncertainty in N-best hypotheses

Table 2: WER (%) results of RobustGER on different SNR-level testing conditions. The test sets are from LS-FreeSound dataset, with five SNR levels on two noise types. More results are in Table 11.

Noise Type	SNR (dB)	Baseline	LM_{rank}	GER	+ Audio Denoising	RobustGER (ours)	Oracle o_{nb}	Oracle o_{cp}
Metro	0	9.9	9.8	9.5$_{-4.0\%}$	9.4$_{-5.1\%}$	**8.9**$_{-10.1\%}$	7.9	4.9
	5	7.2	7.0	6.7$_{-6.9\%}$	6.4$_{-11.1\%}$	**5.5**$_{-23.6\%}$	5.5	3.2
	10	4.8	4.6	4.2$_{-12.5\%}$	4.3$_{-10.4\%}$	**4.0**$_{-16.7\%}$	3.9	2.3
	15	3.9	3.5	3.2$_{-17.9\%}$	3.2$_{-17.9\%}$	**3.0**$_{-23.1\%}$	3.1	1.7
	20	3.3	3.1	2.7$_{-18.2\%}$	2.6$_{-21.2\%}$	**2.3**$_{-30.3\%}$	2.6	1.3
	avg.	5.8	5.6	5.3$_{-8.6\%}$	5.2$_{-10.3\%}$	**4.7**$_{-19.0\%}$	4.6	2.7
AC/Vacuum	0	12.4	12.5	12.3$_{-0.8\%}$	12.1$_{-2.4\%}$	**11.4**$_{-8.1\%}$	10.2	6.2
	5	7.4	7.0	6.5$_{-12.2\%}$	6.3$_{-14.9\%}$	**5.8**$_{-21.6\%}$	5.5	3.1
	10	6.6	6.2	5.5$_{-16.7\%}$	5.6$_{-15.2\%}$	**5.5**$_{-16.7\%}$	4.5	2.6
	15	4.4	4.2	3.7$_{-15.9\%}$	3.7$_{-15.9\%}$	**3.6**$_{-18.2\%}$	3.3	1.8
	20	3.8	3.7	3.3$_{-13.2\%}$	3.2$_{-15.8\%}$	**2.9**$_{-23.7\%}$	2.8	1.4
	avg.	6.9	6.7	6.3$_{-8.7\%}$	6.2$_{-10.1\%}$	**5.8**$_{-15.9\%}$	5.3	3.0
Clean	∞	3.0	2.8	2.5$_{-16.7\%}$	2.4$_{-20.0\%}$	**2.1**$_{-30.0\%}$	2.5	1.4

Ablation Study

- **Effect of two kinds of noise embedding**
  - Token-level noise embedding works better
  - Using both (**concatenate**) has the best effect

Table 3: Ablation study of the language-space noise embedding in terms of utterance and token levels. More studies are presented in Table 13 and Table 14.

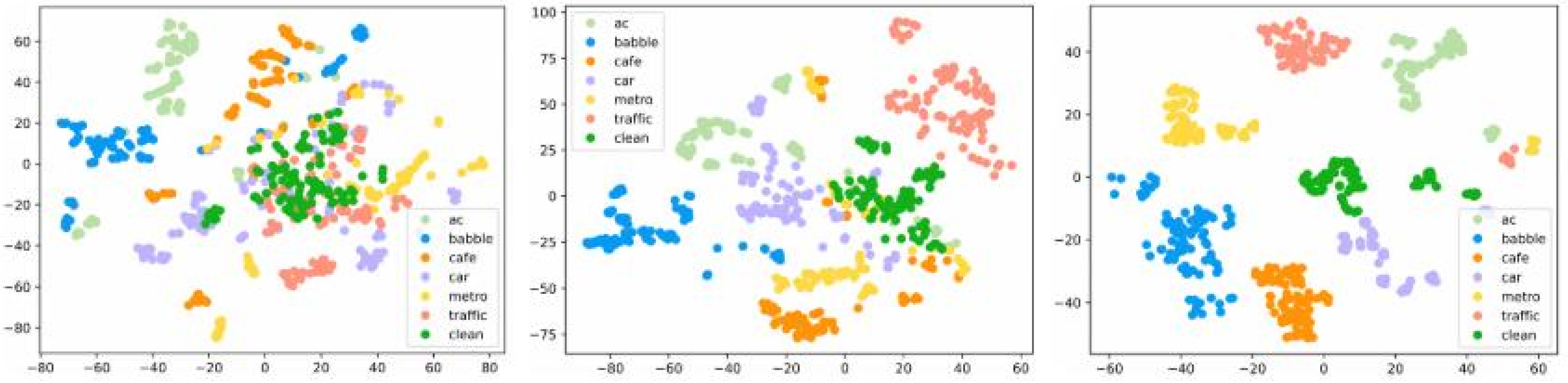
Test Set		Baseline	GER	+ Audio Denoising	+ Language Denoising *Utt.-level*	+ Language Denoising *Tok.-level*	+ Language Denoising *Both*
CHiME-4	*test-real*	12.6	$6.5_{-48.4\%}$	$6.4_{-49.2\%}$	$6.4_{-49.2\%}$	$6.1_{-51.6\%}$	$\mathbf{5.9}_{-53.2\%}$
	test-simu	15.4	$9.2_{-40.3\%}$	$9.0_{-41.6\%}$	$9.1_{-40.9\%}$	$8.9_{-42.2\%}$	$\mathbf{8.6}_{-44.2\%}$
	dev-real	10.6	$5.0_{-52.8\%}$	$4.9_{-53.8\%}$	$4.7_{-55.7\%}$	$4.4_{-58.5\%}$	$\mathbf{4.4}_{-58.5\%}$
	dev-simu	12.4	$6.8_{-45.2\%}$	$6.6_{-46.8\%}$	$6.4_{-48.4\%}$	$6.3_{-49.2\%}$	$\mathbf{6.1}_{-50.8\%}$
	avg.	12.8	$6.9_{-46.1\%}$	$6.7_{-47.7\%}$	$6.7_{-47.7\%}$	$6.4_{-50.0\%}$	$\mathbf{6.3}_{-50.8\%}$
VB-DEMAND	*baby-cry*	8.0	$7.0_{-12.5\%}$	$6.9_{-13.8\%}$	$6.7_{-16.3\%}$	$6.6_{-17.5\%}$	$\mathbf{6.4}_{-20.0\%}$
	helicopter	8.4	$7.4_{-11.9\%}$	$7.3_{-13.1\%}$	$7.3_{-13.1\%}$	$7.1_{-15.5\%}$	$\mathbf{7.1}_{-15.5\%}$
	crowd-party	22.6	$21.4_{-5.3\%}$	$21.0_{-7.1\%}$	$20.8_{-8.0\%}$	$20.3_{-10.2\%}$	$\mathbf{19.9}_{-11.9\%}$
	avg.	13.0	$11.9_{-8.5\%}$	$11.7_{-10.0\%}$	$11.6_{-10.8\%}$	$11.3_{-13.1\%}$	$\mathbf{11.1}_{-14.6\%}$

Analysis

- **Effect of audio noise distillation**

Table 19: Distances between the language noise embeddings from clean and different noisy conditions. The corresponding t-SNE visualizations are presented in Fig. 4.

Clean vs.	ac	babble	cafe	car	metro	traffic	avg.
Language Noise Emb.	**59.7**	54.9	32.4	12.7	19.1	17.4	32.7
+ Audio Distillation	57.6	**87.5**	**53.2**	**37.5**	**32.1**	**51.8**	**53.3**



(a) Language Noise Emb.　(b) Language Noise Emb. + KD　(c) Audio Noise Emb.

Figure 4: t-SNE visualizations of (a) language-space noise embedding, (b) language embedding with audio distillation, (c) audio noise embeddings. Cluster distances are in Table 19. Details are in §C.2.

Analysis

- **t-SNE visualizations of different level of noise**

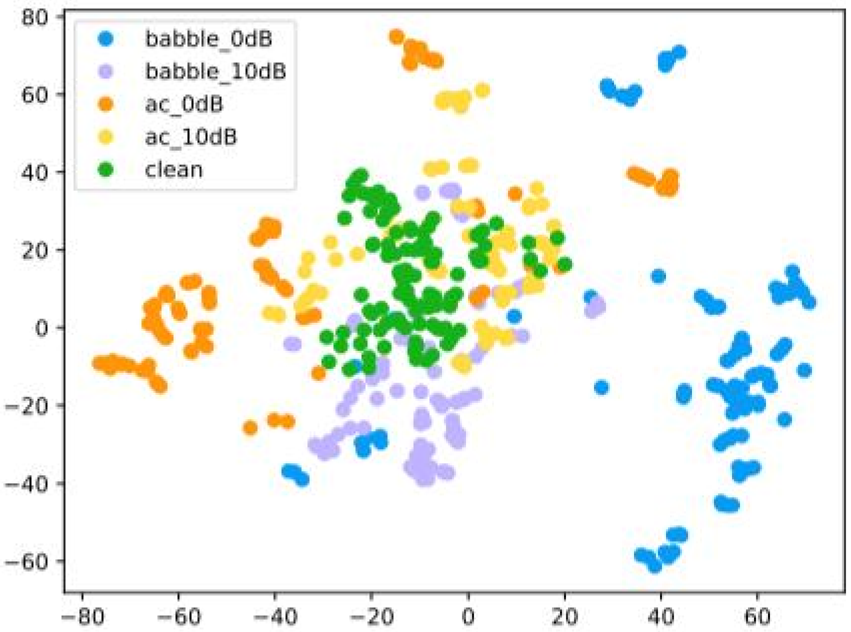


Figure 6: The t-SNE visualizations of language-space noise embeddings from source speech under different noise types and SNR levels. The average distances between embeddings of clean and various noisy conditions are: **58.6** (babble_0dB), **24.5** (babble_10dB), **22.6** (ac_0dB) and **14.3** (ac_10dB).

Other Results

- **Effect of different methods for audio noise distillation**

Table 14: Comparison of different techniques for audio noise distillation. "*T-S Learning*" denotes teacher-student learning with KL regularization, "*Contra. Learning*" denotes contrastive learning.

Test Set		Baseline	GER	+ Lang. Denoising	+ Audio Noise Distillation		
					T-S learning	*Contra. learning*	*MINE*
CHiME-4	*test-real*	12.6	$6.5_{-48.4\%}$	$5.9_{-53.2\%}$	$5.9_{-53.2\%}$	$5.8_{-54.0\%}$	$\mathbf{5.6}_{\mathbf{-55.6\%}}$
	test-simu	15.4	$9.2_{-40.3\%}$	$8.6_{-44.2\%}$	$8.7_{-43.5\%}$	$8.4_{-45.5\%}$	$\mathbf{8.2}_{\mathbf{-46.8\%}}$
	dev-real	10.6	$5.0_{-52.8\%}$	$4.4_{-58.5\%}$	$4.5_{-57.5\%}$	$4.2_{-60.4\%}$	$\mathbf{4.1}_{\mathbf{-61.3\%}}$
	dev-simu	12.4	$6.8_{-45.2\%}$	$6.1_{-50.8\%}$	$6.0_{-51.6\%}$	$6.1_{-50.8\%}$	$\mathbf{5.8}_{\mathbf{-53.2\%}}$
	avg.	12.8	$6.9_{-46.1\%}$	$6.3_{-50.8\%}$	$6.3_{-50.8\%}$	$6.1_{-52.3\%}$	$\mathbf{5.9}_{\mathbf{-53.9\%}}$
VB-DEMAND	*baby-cry*	8.0	$7.0_{-12.5\%}$	$6.4_{-20.0\%}$	$6.4_{-20.0\%}$	$6.2_{-22.5\%}$	$\mathbf{6.0}_{\mathbf{-25.0\%}}$
	helicopter	8.4	$7.4_{-11.9\%}$	$7.1_{-15.5\%}$	$7.2_{-14.3\%}$	$6.9_{-17.9\%}$	$\mathbf{6.9}_{\mathbf{-17.9\%}}$
	crowd-party	22.6	$21.4_{-5.3\%}$	$19.9_{-11.9\%}$	$20.1_{-11.1\%}$	$19.5_{-13.7\%}$	$\mathbf{19.2}_{\mathbf{-15.0\%}}$
	avg.	13.0	$11.9_{-8.5\%}$	$11.1_{-14.6\%}$	$11.2_{-13.8\%}$	$10.8_{-16.9\%}$	$\mathbf{10.7}_{\mathbf{-17.7\%}}$

Other Results

- **Effect of different ASR encoder for audio noise distillation**

Table 17: Comparison between different ASR encoders for audio noise embedding extraction.

Test Set		Baseline	LM_{rank}	GER	+ Audio Denoising			Oracle	
					Whisper	WavLM	Wav2vec2	o_{nb}	o_{cp}
CHiME-4	*test-real*	12.6	12.2	6.5	**6.4**	6.6	6.8	10.5	3.0
	test-simu	15.4	14.5	9.2	**9.0**	9.2	9.4	12.9	5.0
	dev-real	10.6	10.3	5.0	**4.9**	5.0	5.4	9.1	2.1
	dev-simu	12.4	11.9	6.8	**6.6**	6.6	7.0	10.6	3.3
	avg.	12.8	12.2	6.9	**6.7**	6.9	7.2	10.8	3.4

Other Results

- **LLM adapter vs. LLM LoRA**

Table 16: Comparison between LLaMA-Adapter and LLaMA-LoRA for efficient LLM finetuning.

Test Set		Baseline	LM_{rank}	GER		Oracle	
				LLaMA-Adapter	LLaMA-LoRA	o_{nb}	o_{cp}
CHiME-4	*test-real*	12.6	12.2	**6.5**$_{-48.4\%}$	6.6$_{-47.6\%}$	10.5	3.0
	test-simu	15.4	14.5	9.2$_{-40.3\%}$	**9.0**$_{-41.6\%}$	12.9	5.0
	dev-real	10.6	10.3	**5.0**$_{-52.8\%}$	5.1$_{-51.9\%}$	9.1	2.1
	dev-simu	12.4	11.9	6.8$_{-45.2\%}$	**6.7**$_{-46.0\%}$	10.6	3.3
	avg.	12.8	12.2	**6.9**$_{-46.1\%}$	**6.9**$_{-46.1\%}$	10.8	3.4
VB-DEMAND	*baby-cry*	8.0	7.8	**7.0**$_{-12.5\%}$	**7.0**$_{-12.5\%}$	4.5	3.0
	helicopter	8.4	8.1	7.4$_{-11.9\%}$	**7.2**$_{-14.3\%}$	4.8	3.2
	crowd-party	22.6	22.3	21.4$_{-5.3\%}$	**21.0**$_{-7.1\%}$	16.5	11.5
	avg.	13.0	12.7	11.9$_{-8.5\%}$	**11.7**$_{-10.0\%}$	8.6	5.9

Case Study

- **Effect on N-best hypotheses under different level of noise**

Table 15: N-best hypotheses from a speech sample under different noise conditions. We use two noise types (i.e., Babble and AC/Vacuum) and two SNR levels (i.e., 0 and 10 dB) from LibriSpeech-FreeSound test set, where the original sample id is “237-134500-0040”. The “Acoustic Score” denotes the decoding score from Whisper Large-V2 model, which is calculated by negative entropy. Red font highlights the wrong tokens compared to ground-truth transcription.

Noise Type	SNR (dB)	N-best Hypotheses	Acoustic Score	WER (%)
Babble	0	i pray for them but that is not the same as i pray for sam	−0.467	33.3
		i pray for them but that is not the same as i pray for science	−0.485	33.3
		i pray for them but that is not the same as if i prayed for sam	−0.516	26.7
		i pray for them but that is not the same as i pray for sons	−0.517	33.3
		i pray for them but that is not the same as if i pray for sam	−0.521	33.3
	10	i pray for you but that is not the same as if you prayed yourself	−0.328	0.0
		i pray for you but that is not the same as if you prayed yourself	−0.328	0.0
		i pray for you but that is not the same as if you pray yourself	−0.340	6.7
		i pray for you but that is not the same as if you pray for yourself	−0.426	13.3
		i pray for you but that is not the same as if you prayed for yourself	−0.449	6.7
AC	0	i pray for you but that is not the same as if you prayed yourself	−0.329	0.0
		i pray for you but that is not the same as if you pray yourself	−0.369	6.7
		i pray for you but that is not the same as if you pray for yourself	−0.388	13.3
		i would pray for you but that is not the same as if you prayed yourself	−0.428	6.7
		i pray for you but that is not the same as if you prayed for yourself	−0.429	6.7
	10	i pray for you but that is not the same as if you prayed yourself	−0.305	0.0
		i pray for you but that is not the same as if you prayed yourself	−0.305	0.0
		i prayed for you but that is not the same as if you prayed yourself	−0.343	6.7
		i prayed for you but that is not the same as if you prayed yourself	−0.343	6.7
		i prayed for you but that is not the same as if you prayed yourself	−0.343	6.7
Clean	∞	i pray for you but that is not the same as if you prayed yourself	−0.280	0.0
		i pray for you but that is not the same as if you prayed yourself	−0.280	0.0
		i pray for you but that is not the same as if you prayed yourself	−0.280	0.0
		i pray for you but that is not the same as if you prayed yourself	−0.280	0.0
		i pray for you but that is not the same as if you prayed yourself	−0.280	0.0
Ground Truth		i pray for you but that is not the same as if you prayed yourself	-	-

Case Study

- **Successful and failed case of RobustGER**

Table 5: Case study of RobustGER. We also implement an in-context learning baseline by ChatGPT for comparison (details are in §C.2). The test sample is selected from the CHiME-4 *dev-real* set.

Method	Utterance	WER (%)
N-best List	the four other utility company owners will also have to take write ups	7.7
	the four other utility company owners will also have to take write ups	7.7
	the four other utility company owners will also have to take write ups	7.7
	the four other utility company owners will also have to take ride outs	15.4
	the four other utility company owners will also have to take ride outs	15.4
In-context Learning	the four other utility company owners will also have to take write-ups	15.4
GER	the four other utility company owners will also have to take write ups	7.7
RobustGER	the four other utility company owners will also have to take write offs	**0.0**
Ground Truth	the four other utility company owners will also have to take write offs	-

Table 18: Failure case of RobustGER. The test sample is from CHiME-4 *dev-real* dataset with ID as "M03_052C010R_BUS".

Method	Utterance	WER (%)
N-best List	miss amsterdam declined to comment	20.0
	miss amsterdam declined to comment	20.0
	ms amsterdam declined to comment	0.0
	miss amsterdam declined to comment	20.0
	miss amsterdam decline to comment	40.0
GER	ms amsterdam declined to comment	**0.0**
RobustGER	miss amsterdam declined to comment	20.0
Ground Truth	ms amsterdam declined to comment	-

Summary

- This paper propose a novel idea **RobustGER**, which **extract a language-space noise embedding from the N-best list** to represent the noise conditions of source speech **to avoid cross-modality gap** and enhance it **with a knowledge distillation utilizing MINE**, gaining good performance improvement on robust-noise ASR.
- However, under certain circumstances this method may lead originally correct hypotheses to faulty prediction due to “bad timing” for denoise. How to address this problem need research in the future.